UNIVERSIDAD CARLOS III DE MADRID

TESIS DOCTORAL

Contribuciones a protocolos y mecanismos de análisis y decisión para control de acceso en entornos distribuidos

Autor: **Daniel Díaz Sánchez**
Ingeniero de Telecomunicación

Director: **Andrés Marín López**
Doctor Ingeniero de Telecomunicación

DEPARTAMENTO DE INGENIERÍA TELEMÁTICA
ESCUELA POLITÉCNICA SUPERIOR

Leganés, Enero de 2008

TESIS DOCTORAL

Contribuciones a protocolos y mecanismos de análisis y decisión para control de acceso en entornos distribuidos

Autor: Daniel Díaz Sánchez
Director: Andrés Marín López

Firma del Tribunal Calificador:

Firma

Presidente

Vocal

Vocal

Vocal

Secretario

Calificación:

Leganés, ___ de _____ de _____

A Ana,
a mis padres

Agradecimientos

Durante el tiempo de realización de esta tesis doctoral, un gran número de personas han contribuido de manera directa o indirecta a que todo llegara a buen puerto. Por esta razón quiero agradecérselo en estas líneas.

A Ana, mi compañera y amiga durante todos estos años, gracias por tener tantísima paciencia, por tu apoyo y por tu amor, sin los cuales no lo habría conseguido.

A mis padres y mi hermana, que siempre han creído en mi, dándome apoyo en los buenos y malos momentos. Sin dudarlo un momento, sin vosotros nada de esto habría sido posible. A mis padres, gracias por educarme como lo habéis hecho, por darme vuestro cariño y por ser como sois.

A mi tutor Andrés Marín López por su inestimable ayuda; por estos más de 5 años, desde que comencé mi proyecto de fin de carrera, en los que me ha dado su apoyo y ha confiando en mi, escuchando mis opiniones y propuestas; por todos esos consejos que me han ayudado a crecer como investigador y profesor, así como persona; por el trato tan cercano y humano, sin el que no habría sido posible conseguir lo que hemos logrado en estos años de duro trabajo; por su amistad.

A mis compañeros de grupo de investigación: a Florina, Carlos y Celeste, por estar pendiente de todo, por sus revisiones, por sus consejos, su gran apoyo y por ser unos excelentes compañeros; a Alberto, Jose Carlos y Juan por ser muy buenos compañeros y amigos, dispuestos a prestar su ayuda en todo momento.

A mis compañeros de departamento: Carlos Delgado Kloss por creer en mi, permitirme ser parte del departamento y por hacer que todo funcione; a todos los compañeros del “C03”, a todos los que están ahora y todos los que se fueron, todos fueron grandes compañeros: Juan Jesús, Jose Alberto, Jose María, Jorge, Jopez, Luis, Pedro. . . ; a todos los demás miembros del departamento, por hacer de éste un gran departamento.

A mis antiguos compañeros de carrera, Pedro Peris y Fernando Carrasco, con los que emprendí proyectos que me sirvieron para aprender mucho de muchas cosas.

A mis todos mis amigos, por su apoyo; a Javi y María, por ser más que amigos y apoyarme durante tantos años.

A todos aquellos cuya ayuda no es perceptible, pero que sin los cuales nada funcionaría, a todos los que me dejo en tintero...

¡GRACIAS!

Resumen

Mark Weiser estableció que las tecnologías más profundas son aquellas que no vemos, que no requieren intervención del usuario, que desaparecen.... Esta visión de Weiser recibe el nombre de **Computación Ubicua**: dispositivos que se integran de forma amigable con los humanos de modo que la interacción se realiza de forma inconsciente. Estos dispositivos, desde dispositivos personales, hasta dispositivos implantados en el cuerpo, pasando por ropa inteligente, elementos de visualización, etc. han evolucionado no sólo en capacidad de cómputo, sino en capacidad de comunicación. Las tecnologías radio proporcionan mayor cobertura, velocidades de enlace superiores y calidad de servicio mejorada lo que les proporciona capacidad para conectarse a distintas redes y proveedores, así como constituir nuevas redes entre pares sin necesidad de infraestructura. El ambiente que nos rodea puede considerarse, por tanto, un entorno dinámico, formado por una población de dispositivos y servicios con alta movilidad, ricos en información de contexto y con un mayor número posibilidades de constituir redes.

Este hecho nos lleva a la necesidad de procesar la información de contexto para operar por debajo de la consciencia de usuario; presentar al usuario esa información de contexto, de forma intuitiva, e inclusive tratar de imitar su forma de pensar o resolver problemas; no requerir intervención del usuario en cuestiones cotidianas, como seleccionar puntos de acceso u otros dispositivos con los que interactuar. Además, la seguridad es una pieza fundamental en estos entornos, dado que la movilidad aumenta el número de interacciones y los riesgos. Dado el gran número de dispositivos existentes en estos entornos y la gran distribución y replicación de los servicios, los dispositivos personales deben asistir al usuario en los procesos de control de acceso distribuido.

Esta tesis define unos objetivos y un marco de trabajo orientado a mejorar el con-

trol de acceso, respetando la autonomía de todos los actores como dispositivos, elementos de red, servicios..., permitiendo que, mediante mecanismos de decisión y selección, se pueda negociar de forma flexible el acceso a los servicios y que, utilizando extensiones a protocolos de seguridad, se facilite el acceso a los mismos.

Por otro lado, la tesis se enfocará también en hacer la seguridad más usable y eficaz; capaces de comunicar eficazmente a los humanos la información de seguridad, analizando sistemas de comunicación gráfica con el usuario, que permitan a personas sin conocimientos técnicos entender los riesgos de determinadas acciones.

Abstract

Mark Weiser stated that the most profound technologies are those that disappear, those that do not require user intervention, those that we forget... **Ubiquitous computing** describes devices integrating friendly with humans in such a way that interactions become unconscious. Ubiquitous devices, that range from personal devices to body implanted devices, including smart clothes, visualization devices... have evolved not only increasing their computing performance, but also their communication capacity. Radio technologies provide now more coverage, increased link speeds and improved quality of service; so they can connect to many different network and even constitute peer to peer networks when there is no infrastructure.

Context information processing is needed to work under users' consciousness threshold; to intuitively show context information to the user; to mimick users' way of thinking; to avoid disturbing the user requiring his intervention for regular tasks, as selecting the best access point or peer to interact with. Moreover, security is the key topic for this environments since the increasing mobility raise interaction rate and, thus, the risk of attack. So, the great number of devices present in these environments, together with the ubiquitous service availability, makes necessary to develop smart mechanisms for assisting the user to face distributed access control tasks.

This thesis aims at improving access control, respecting the autonomy and independence of every actor: devices, network elements and services... I define a framework which provides selection and decision engines that allow accessing to services with a flexible yet fair negotiation. I have also defined security protocol extensions, to communicate requirements and credentials that guarantee the fairness of the process.

Furthermore, this thesis focuses on making user-device interaction more effective in terms of risks communication: analyzing data processing techniques that produce results that can be visually interpreted. So users with no technical knowledge could

understand the risks of their actions.

Índice general

Índice de figuras

Índice de tablas

CAPÍTULO

1 Introducción

Mark Weiser estableció en [1] lo siguiente: "*the most profound technologies are those that disappear*". Con esta frase, Weiser, quiso afirmar que las tecnologías más profundas son aquellas que olvidamos, que no requieren intervención del usuario, que desaparecen... Esta frase sugiere un entorno saturado de dispositivos con capacidad de cómputo que facilita el acceso permanente a la información; esta visión de Weiser recibe el nombre de **Computación Ubicua**. Estos dispositivos se integran de forma amigable con los humanos, de modo que la interacción se realiza de forma inconsciente. La escala de estos dispositivos varía, desde dispositivos personales, hasta dispositivos implantados en el cuerpo, pasando por ropa inteligente, elementos de visualización...

Las tecnologías de acceso radio están evolucionando y actualmente proporcionan mayor cobertura, velocidades de enlace superiores y calidad de servicio mejorada. La bajada de costes de este tipo de tecnología ha facilitado su despliegue, llegando al usuario medio. Los dispositivos personales, como los móviles, comienzan a incorporar cada día un mayor número de interfaces de red. Esto les proporciona capacidad para conectarse a distintas redes y proveedores, así como constituir nuevas redes entre pares sin necesidad de infraestructura. El ambiente que nos rodea puede considerarse, por tanto, un entorno dinámico, formado por una población de dispositivos y servicios con alta movilidad, ricos en información de contexto y con un mayor número posibilidades de constituir redes.

Este hecho nos lleva a advertir la necesidad de procesar la información de contexto para operar por debajo de la consciencia de usuario. Otra característica deseable es la de presentar al usuario esa información de contexto, de forma intuitiva, e inclusive tratar de imitar su forma de pensar o resolver problemas, para no requerir intervención del usuario en cuestiones cotidianas, como seleccionar puntos de acceso u otros dispositivos con los que interactuar.

Cuando los dispositivos móviles se mueven de un lugar a otro, se hace necesario seleccionar apropiadamente la red o el par con el que conectar de forma automática, para satisfacer las necesidades de conexión del usuario o, si ya estamos conectados a una red, simplemente para reducir el tiempo de handover o incrementar la calidad y el número de servicios. Decidir a que red o con que entidad conectar, dependerá de muchos factores, pero es necesario observar, que cuando el usuario interviene, típicamente éste tiende a simplificar el problema. Esa es la razón de preguntarnos: ¿Por qué no implementar un motor de decisión que simplifique esas decisiones?

Además, la seguridad es una pieza fundamental en estos entornos porque la movilidad aumenta el número de interacciones y, por tanto, los riesgos. Dado el gran número de dispositivos existentes en estos entornos y la gran distribución y replicación de los servicios, los dispositivos personales deben asistir al usuario en los procesos de control de acceso. La mecánica del control de acceso se basa en encontrar una respuesta adecuada a la pregunta: ¿puede el usuario o entidad acceder a un recurso en concreto, dadas unas ciertas restricciones? Sobre esta pregunta genérica se pueden hacer un sinfín de apreciaciones que afectan a cualesquiera de las partes involucradas, por lo que los dispositivos deben actuar de forma autónoma, gestionando su propia evolución y cambios de configuración evitando la intervención explícita del usuario (“autonomic computing"[2]).

Es por ello que los dispositivos, sea cual sea su rol, deben ser capaces de protegerse contra el desvelamiento, modificación o pérdida de la información; el uso inadecuado de los recursos y servicios; y los abusos. Además, para no limitar la movilidad, deben disponer de mecanismos de negociación, que permitan a usuarios desconocidos interactuar o componer servicios. También es importante disponer mecanismos para informar al usuario de forma comprensible sobre los riesgos de seguridad y del proceso de toma de decisiones, evitando así que los usuarios con menor formación técnica se encuentren en situación de indefensión ante eventualidades.

Esta de tesis define unos objetivos, soportados por un marco de trabajo, orientado a mejorar el control de acceso, respetando la autonomía de todos los actores como dispositivos, elementos de red, servicios...; permitiendo que, mediante el uso de mecanismos de decisión y selección, se pueda negociar de forma flexible el acceso a los servicios; y que, utilizando extensiones a protocolos de seguridad, se facilite el intercambio de credenciales. Por otro lado, la tesis se enfocará en hacer más comunicativos y humanos a los dispositivos en cuanto a comunicación de decisiones sobre seguridad, analizando sistemas de comunicación gráfica con el usuario, que ayuden a personas con menos conocimientos técnicos a entender los riesgos de determinadas acciones.

1.1. Motivación de la tesis

Los entornos dinámicos, comentados en la introducción, sugieren entornos formados por grandes poblaciones de dispositivos (y servicios) con alta movilidad, ricos en información de contexto y con mayor posibilidad de constituir redes.

La seguridad es necesaria para garantizar la protección de los recursos, la privacidad y para minimizar los riesgos de perdida de información. Por otro lado, el control de acceso, puede ayudar a un mejor aprovechamiento de los recursos si no solo controla el acceso, sino que toma decisiones proactivas. Dentro de estas decisiones proactivas, estarían la selección de redes, entidades y servicios con los que o a través de los que interactuar.

Las tecnologías avanzan y su complejidad también a la vez que comienzan a ser utilizadas por los usuarios en un espacio de tiempo cada vez menor. Esta reducción del *time to market* ha motivado que los usuarios medios no comprendan realmente el funcionamiento, ni los riesgos de seguridad del uso de estas tecnologías, pero que se vean obligados a utilizarlas para sus tareas cotidianas. Por esa razón, es necesario utilizar mecanismos de comunicación con el usuario, fáciles de entender, que le ayuden a: tomar decisiones de seguridad, entender los riesgos que conllevan las decisiones tomadas y entender por qué se seleccionan determinadas redes y servicios.

Como se pondrá de manifiesto en las siguientes secciones, el soporte que dan los sistemas de control de acceso actuales, así como los protocolos de autenticación y autorización, no es suficiente para dar soporte a estos entornos.

El trabajo realizado hasta el momento en varios proyectos de investigación y la difusión que se ha realizado en congresos y revistas, justifican el interés de esta investigación y su relevancia técnica. Existen varios proyectos en los que se encuentra actualmente inmerso el grupo investigador y otros finalizados, en los que se ha participado activamente, donde se han propuesto y se trabaja en conseguir soluciones para estos retos. Entre dichos proyectos cabe destacar los siguientes proyectos competitivos y del plan nacional de I+D:

- TRUST-eS: Technology Responses To Ubiquitous Security Threats For e-Security (EUREKA / MEDEA).
- UBISEC: Ubiquitous Networks with a Secure Provision of Services, Access, and Content Delivery (Sixth Framework Programme / IST-2002-506926).
- EasyWireless (EUREKA / ITEA).
- Everyware (CICYT)

Motivación personal

Desde el año 2002 hasta diciembre de 2003 realicé el proyecto de fin de carrera, con Andrés Marín López, desarrollando un CSP de Microsoft CryptoAPI (Cryptographic Service Provider) que pudiera utilizar módulos PKCS#11 [3] para realizar operaciones. La dificultad del desarrollo, que requería la utilización de un depurador de kernel unido a otra máquina por puerto serie para depurar, alargó sensiblemente el proyecto.

A continuación, en enero de 2004 me integré en el grupo de investigación Pervasive, del Departamento de Ingeniería Telemática de la Universidad Carlos III de Madrid, donde continué con mi trabajo, mejorando el "puente" entre PKCS#11 y CryptoAPI. La idea consistía en poder acceder, con seguridad, a todas las credenciales del sistema, desde CryptoAPI o desde PKCS#11, con independencia de que estuvieran almacenadas en uno u otro. Así se dotaría a los sistemas de mayor flexibilidad y movilidad.

Una vez finalizamos esta línea de trabajo, tras desarrollar satisfactoriamente el "puente", comenzamos a proporcionar soporte de autorización a través de ambos APIs criptográficos, portando además la solución a dispositivos Windows Mobile. El objetivo que se planteó, y que finalizaría con la redacción de esta tesis, fue agilizar las interacciones proporcionando capacidades de negociación, que incluían autorización y autenticación; así como contribuir a la mejora del control de acceso interno del dispositivo sin requerir excesiva intervención del usuario.

Para ello, se desarrollo un middleware que permitiera verificar certificados almacenados en cualquiera de los sistemas criptográficos y emitir credenciales tanto de autenticación como de autorización. Este middleware daba soporte además a TrustAC [4], que permitía interactuar de forma segura en entornos desconectados.

El soporte de autorización estaba basado en certificados de atributos y más adelante en SAML; sin embargo, debíamos proporcionar una manera eficiente de utilizar dichas credenciales de autorización, que era uno de los objetivos. Se analizaron, por tanto, varios protocolos prestando especial atención a TLS, que era un protocolo ampliamente usado y soportado; que proporcionaba confidencialidad y autenticación; pero, que carecía de soporte de autorización. Durante los siguientes meses se definió la extensión de TLS que permitiría el uso de credenciales de autorización.

Una vez dispusimos de soporte de autorización en TLS, un middleware en dispositivo móvil que permitía el procesado de varias credenciales y un puente entre interfaces criptográficos, podíamos afirmar que habíamos incrementado mucho las posibilidades de interacción entre dispositivos. Así que comenzamos a abordar el soporte avanzado de autenticación y autorización para ayudar en el roaming entre redes, ya que al poder comunicar privilegios explícitamente, podíamos simplificar y agilizar el acceso a la red. De esta manera, evitábamos que los servidores de autenticación tu-

vieran que contactar a la red origen durante el proceso de roaming, lo que reducía sensiblemente el tiempo empleado en el acceso. Para ampliar la capacidad de negociación, comenzamos a estudiar el trabajo, que estaban realizando algunos grupos de investigación, que tomaba el nombre de "negociación de confianza".

Durante esta etapa de la investigación, se propusieron nuevas versiones del a extensión de TLS para autorización, cada vez más completas, y una extensión de TLS que, acompañada del formato de solicitud de certificados de atributos, completaba las anteriores propuestas. En paralelo a estas propuestas sobre TLS, se analizaron los requisitos que tendría la negociación de confianza sobre un protocolo como TLS.

Si bien es cierto que, con las propuestas anteriores, soportábamos el uso de trust tickets y emisión de credenciales bajo demanda, también era cierto que lo limitamos a certificados de atributos y SAML. Para dotar a los dispositivos de mayor capacidad de interacción, diseñamos motores agnósticos de decisión, que pudieran hacer uso del software diseñado con anterioridad (gestores de credenciales y motores de control de acceso) y de otros ajenos al nuestro. Por esta razón, diseñamos, simulamos y validamos ciertas mejoras en el control de acceso de los terminales, que incluían monitores y puntos de aplicación de la política empotrados en drivers de red NDIS de Windows CE; un mecanismo de selección de red, que obtenía la información de NDIS y de la capa Radio Interface Layer (RIL) de Windows CE (United States Patent 6826762); y un gestor de políticas ya utilizado con anterioridad en el proyecto UBISEC.

En la etapa final de la investigación, estudiamos los protocolos de autenticación usados para el acceso a la red, tratando de encontrar una solución que pudiera ser utilizada por la mayoría de tecnologías de acceso. Así fue como nos centramos en protocolos de niveles superiores capaces de transportar EAP [5] (que era a su vez extensible y permitía el uso de TLS). Eso nos llevó a un protocolo en definición llamado PANA. PANA acompañado de EAP-TLS permitía autenticar ambas partes y generar claves para proteger los mensajes extremo a extremo, lo que era muy necesario, dado el punto de acceso podría redirijir los mensajes EAP a otro servidor.

Una vez mejorado el control de acceso y la decisión en el terminal, estudiamos los requisitos para un protocolo que permitiera el intercambio gradual de políticas y credenciales. Para lograrlo, se diseñó una tercera extensión a TLS que lo permitiera; sin embargo, aun quedaban cosas por hacer, ¿Cómo orquestar todo el intercambio de credenciales, cuando exista más de un motor de control de acceso?. El problema era de visibilidad: cada motor de control de acceso puede ver el exterior, utilizar las variables de contexto y decidir en consecuencia; pero no se coordina con el resto de los motores de control de acceso, lo que nos puede llevar a un problema de eficiencia bastante importante. Por esta razón, se diseño un sistema que permitiera dotar de visión global a todos los motores de control de acceso, mediando entre ellos y el exterior, orquestando toda la negociación de confianza.

Finalmente, pese a que es interesante trabajar bajo el nivel de consciencia del usuario, en aquellas ocasiones en las que fuese necesario comunicar decisiones, resultados de análisis y riesgos, debíamos hacerlo mediante comparación, dado que es una manera efectiva. Por esta razón se propuso mostrar al usuario el espacio de decisión de manera que, visualmente, pudiera entender mejor los riesgos y el espacio de decisión por complicado que éste fuera.

Publicaciones

Respecto a las contribuciones a congresos y revistas, las publicaciones que avalan el interés de esta investigación son las siguientes:

- Contribuciones a congresos nacionales:
  1. Título: A framework for authorization and delegation in ubiquitous computing. Fecha: 09-2005. Autores: D.Díaz, A.Marín, F.Almenárez. Congreso: Ubiquitous Computing and Ambient Intelligence (UCAmI 2005). Libro o actas: CEDI Editor: Thomsom Lugar: Granada (Spain). [6].
  2. Título: Securing interactions in emerging environments. Fecha: 06-2006. Autores: D.Díaz, A.Marín, F.Almenárez, C.Campo, C.Garcia-Rubio. Congreso: 2nd International Workshop on Ubiquitous Computing and Ambient Intelligence - 2006 Libro o actas: Conference proceedings. Lugar: Puertollano (Spain). [7].
  3. Título: Mecanismo de selección de red sensible al contexto para entornos dinámicos. Fecha: 09-2007. Autores: D.Díaz, A.Marín, F.Almenárez. Congreso: Jornadas de Ingeniería Telemática (JITEL) Libro o actas: Conference proceedings Editor: Jitel. Lugar: Málaga, Spain. [8].
  4. Título: Mejorando el control de acceso para dispositivos móviles con un motor de decisión agnóstico para negociación de confianza. Fecha: 09-2007. Autores: D.Díaz, A.Marín, F.Almenárez, C. Campo, C. García. Congreso: Libro o actas: Actas de congreso Editor: Thompson. Lugar: Zaragoza, Spain. [9].
- Contribuciones a congresos internacionales:
  1. Título: Secure Ad-hoc mBusiness: Enhancing WindowsCE Security Fecha: 09-2004. Autores: F.Almenárez, D.Díaz, A.Marín, Congreso: Trust and Privacy in Digital Business. First Conference on Trust Digital Business (TrustBus). Libro o actas: Lecture Notes In Computer Science. Editor: Springer-Verlag GMBH. Lugar: Zaragoza (Spain). [10]

2. Título: Developing a Model for Trust Management in Pervasive Devices Fecha: 05-2006. Autores: F.Almenárez, A.Marín, D.Díaz, J.Sánchez, Congreso: Third IEEE International Workshop on Pervasive Computing and Communication Security held in conjunction with IEEE PerCom 2006. Libro o actas: IEEE PerCom. Editor: IEEE. Lugar: Pisa (Italy). [11].

3. Título: Cards and Residential Gateways: Improving OSGi gateways services with Java Cards. Fecha: 04-2006. Autores: J.Sánchez, D.Díaz, J.Vigo, N.Martinez, R.Seepold. Congreso: Seventh Smart Card Research and Advanced Application IFIP Conference (CARDIS 2006). Libro o actas: Lecture Notes In Computer Science Editor: Springer-Verlag GMBH. Lugar: Tarragona (Spain). [12].

4. Título: A Smart card solution for Access Control and Trust Management for Nomadic Users. Fecha: 04-2006. Autores: D.Díaz, A.Marín, F.Almenárez, Congreso: Seventh Smart Card Research and Advanced Application IFIP Conference (CARDIS 2006) Libro o actas: Lecture Notes In Computer Science Editor: Springer-Verlag GMBH. Lugar: Tarragona (Spain). [13]

5. Título: Interaction Distance determination with PervsIM. Fecha: 06-2006. Autores: D.Díaz, A.Marín, F.Almenárez, C.Garcia-Rubio, C.Campo. Congreso: 15th IST Mobile and Wireless Communication Summit. Libro o actas: Conference proceedings Editor: IST Lugar: Myconos (Greece). [14]

6. Título: Context awareness in network selection for dynamic environments. Fecha: 06-2006. Autores: D.Díaz, A.Marín, F.Almenárez, C.Garcia-Rubio, C.Campo. Congreso: 11th IFIP International Conference on Personal Wireless Communications "PWC06". Libro o actas: Lecture Notes In Computer Science Editor: Springer-Verlag GMBH. Lugar: Albacete (Spain). [15]

7. Título: Middleware for Secure Home Access and Control. Fecha: 03-2007. Autores: A.Marín, W.Mueller, R.Schaefer, F.Almenárez, D.Díaz, M.Ziegler. Congreso: IEEE Pervasive Communications (PERCOM) 2007. Libro o actas: Conference proceedings/IEEE Library. Lugar: White Plains, New York (USA). [16]

8. Título: Access Control Agnostic Trust Negotiation Decision Engine. Fecha: 09-2007. Autores: D.Díaz, A.Marín, F.Almenárez. Congreso: 18th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. Libro o actas: IEEE. Lugar: Athens, Greece. [17]

9. Enhancing access control for mobile devices with an agnostic trust negotiation decision engine. Fecha: 2007. Autores: D.Díaz, A.Marín, F.Almenárez. Título Revista: Personal Wireless Communications. Revista: Springer series in Computer Science. ISSN: 1571-5736. [18]

10. DVB-H key management system for UMTS capable devices. Fecha: Enero de 2008. Autores: D. Díaz, A. Marín, F. Almenarez, A. Cortés. Congreso: IEEE International Conference on Consumer Electronics. Libro o actas: IEEE Editor. Lugar: Las Vegas, Nevada (USA). [19]

- Contribuciones a revistas:

1. Título: Secure Service Discovery based on Trust Management for adhoc Networks. Fecha: 03-2006. Autores: C.Campo, F.Almenárez, D.Díaz, C.Garcia-Rubio, A.Marín, Título Revista: vol. 12. Revista: Journal of Universal Computer Science. [20]
2. Título: Smart card-based agents for fair non-repudiation. Fecha: 06-2007. Autores: A.Marín, D.Díaz, F.Almenárez, C.Garcia-Rubio, C.Campo. Título Revista: Special Issue: Advances in Smart Cards. vol. 51, Issue 9, ISSN 1389-1286. Revista: Computer Networks. [21]
3. Título: Context awareness in network selection for dynamic environments. Fecha: 2007. Autores: D.Díaz, A.Marín, F.Almenárez, C.Campo, C.Garcia-Rubio. Título Revista: Personal Wireless Communications Special Issue. Revista: Telecommunication Systems Journal. DOI: 10.1007/s11235-007-9058-9. [22]

1.2. Objetivos de la tesis

En el capítulo 2, que versa sobre estado del arte, así como en el resto de capítulos, en los que se describen las propuestas, identificaremos las carencias de los diferentes elementos involucrados en la seguridad de los entornos dinámicos. En esta sección, se ponen de manifiesto los objetivos que ha perseguido esta tesis doctoral para mitigar dichas carencias y deficiencias. En esta tesis, se proporcionan una serie de contribuciones que permiten satisfacer las demandas de los diferentes actores como el equipo de usuario, los equipos de la red y los servicios. Para ello se plantean los siguientes objetivos específicos:

- Mejora de los mecanismos relacionados con el control de acceso en los equipos de usuario:
    - Estudiar los mecanismos de procesado de información multivariable, de forma que se puedan analizar diferencias entre entidades descritas con distinto número de atributos o de diferente naturaleza (los atributos son la información utilizada para caracterizar una entidad), sin necesidad de exigir

un modelo común. Básicamente se estudiarán algoritmos de análisis de datos en la sección 2.4. Estos algoritmos no son, evidentemente, mecanismos de selección de red como tal, pero no ha sido posible encontrar en la literatura, trabajos de selección de red con resultados comparables a lo que en este objetivo se pretende. Las características de los algoritmos de análisis de datos, deben ser tales que permitan seleccionar entre distintas entidades como si se tratara de iguales, pese a que las tecnologías, y por tanto, los datos disponibles para caracterizar a cada una de las entidades, sean distintas.

- Diseñar mecanismos orientados a mejorar el control de acceso en los terminales que permitan:
  - Caracterizar el entorno: proporcionar la suficiente información para reconocerlo como conocido o desconocido, caracterizar los servicios disponibles y redes disponibles.
  - Tomar decisiones respecto a que red conectar. El algoritmo utilizado, deberá ser capaz de encontrar la mejor red, en base a unas preferencias establecidas, con independencia de la tecnología de acceso a la red.
  - Aplicar políticas específicas para cada entorno o adecuadas al nivel de confianza otorgado a dicho entorno.

  Dichos mecanismos deberán acercarse, en la medida de lo posible, a las siguientes características ideales:
  - Autónomo: no debe necesitar apoyo de la red u otras entidades. Las decisiones, en estas circunstancias, se tomarán en base a los datos que pueden ser recopilados por el dispositivo sin desvelar su ubicación, manteniendo su privacidad y sin utilizar servicios externos.
  - Independiente de la tecnología: deberá se capaz de proporcionar una medida de similitud o disimilitud entre entidades sin que sea necesario un modelo de datos común. De esta forma se permitirá representar entidades de diferentes tecnologías y, por tanto, descritas con diferente número de atributos.
  - Sin estado o estado mínimo: debe ser capaz de tomar decisiones, apoyándose en datos actuales, reduciendo a máximo la información a almacenar en el dispositivo. Se deberán poder utilizar datos ya almacenados, como datos de confianza, credenciales o reputación, pero se deberá minimizar esa cantidad de datos.
  - Sensible al contexto: Como parte de los datos a procesar se deben incluir los relativos al contexto, como localización absoluta o relativa, hora, fecha, batería restante... Se debe poder determinar qué restric-

ciones impone el contexto a las diferentes entidades sobre las que se elige.

- Colaborativo bajo demanda: debe ser capaz de funcionar en entornos colaborativos, cuando el contexto lo favorezca, y de forma autónoma en el resto de la ocasiones.
- Sensible a preferencias de usuario: Debe ser capaz de modular la selección en base a las preferencias del usuario.

- Diseñar un sistema de evaluación de riesgo que permita, mediante comparación, agrupar y mostrar al usuario los recursos afectados por ciertas decisiones, usando un modelo heterogéneo. Cualquier elemento, con independencia de su naturaleza, debe poder considerarse un recurso a la hora de calcular el riesgo.

  Deberá ser capaz de orquestar un intercambio de credenciales-requisitos protegiendo al terminal de ataques o de abusos.

  Para lograr este intercambio justo, este sistema deberá ser capaz de proporcionar una visión global de la situación, mediante análisis de riesgo, a los diferentes motores de control de acceso que se encuentren funcionando en el sistema, en lugar de que cada motor de acceso lo haga por separado. De esta manera, se podría decir, que el grado de flexibilidad total, es mayor que la suma de los grados de flexibilidad proporcionados por cada motor de control de acceso.

- Contribuir, con el diseño de nuevas extensiones a protocolos estándares, al soporte de negociación de confianza y por ello, al acceso flexible a los servicios de red y de valor añadido:

  - Diseñar una extensión, sobre un protocolo estándar, que permita la utilización de una capa para la gestión de autorización, de forma que el cliente pueda proporcionar una credencial de autorización directamente o proporcionar información de dónde obtenerla.

    Esta extensión deberá negociar una capa que asuma el procesamiento de dicha información de autorización, sin lastrar a otras capas que no fueron diseñadas para ello.

  - Diseñar una extensión sobre un protocolo estándar que permita la emisión de credenciales mediante pruebas de posesión de una determinada clave o cualquier otro objeto.

  - Diseñar una extensión sobre un protocolo estándar, agnóstico en cuanto a sintaxis, que permita comunicar requisitos y credenciales durante una negociación extremo a extremo. La extensión debe permitir la negociación de

confianza permitiendo el intercambio de requisitos (o políticas) y credenciales entre iguales. La extensión deberá permitir que se puedan realizar procesos de autenticación con diferentes calidades y enviar credenciales de autorización, así como cualquier otra información, que permita establecer un estado de seguridad gradual entre las partes.

Debe ejecutarse en paralelo a la capa de aplicación o por debajo de la misma, mediando entre el protocolo original y la aplicación. La motivación de esta capa es independizar a las aplicaciones de ciertas tareas como, por ejemplo, la negociación de confianza, y evitar que otras capas asuman tareas para las que no fueron diseñadas.

Por otro lado, debe hacer lo posible por mitigar los ataques de man-in-the-middle descritos en [23].

- Analizar la viabilidad de su uso con EAP u otro protocolo que permita, no solo usar información de autorización o negociación de confianza entre pares, sino la utilización de estas extensiones para el acceso a la red.
- Estos protocolos no deben interferir con otros protocolos de seguridad o de soporte de movilidad.

Objetivos funcionales

Las modificaciones de protocolos propuestas deberán ser totalmente integrable con los protocolos existentes. La función de las modificaciones no será la de sustituir, sino la de complementar, los protocolos existentes para dar soporte al control de acceso en entornos distribuidos y dinámicos.

No son objeto de la tesis el diseñar protocolos ni mecanismos de accounting, charging y billing. Tampoco son objeto de la tesis proporcionar soluciones a tareas que involucren gestión del conocimiento, como pueden ser las ontologías.

Todas las soluciones planteadas deben considerar las siguientes restricciones:

- Respetar la autonomía del dispositivo de forma que la dependencia con el resto de los dispositivos sea baja.
- Permitir la colaboración con otros dispositivos para complementar la información.
- Utilizar protocolos existentes, realizando modificaciones a través de sus mecanismos de extensión, sin realizar cambios estructurales que impidan compatibilidad hacia atrás.

- Diseñar arquitecturas extensibles y reutilizables. Considerando la restricción del punto anterior, todo lo que se plantee debe poder ser reutilizado.
- Las soluciones deben ser sencillas y fáciles de extender.
- Independencia de la tecnología y lenguajes: Los diseños deberán poder utilizar cualquier tipo de credenciales y políticas presentes y futuras y no deberán asumir modelos de datos fijos.

1.3. Estructura del documento

En las siguientes secciones se presentará la tesis doctoral.

En el capítulo 1 se describen las motivaciones (1.1) y la relevancia de la investigación a través de las publicaciones admitidas en congresos nacionales, internacionales y revistas (sección 1.1).

En dicho capítulo se detallan los objetivos de la tesis (sección 1.2).

Más adelante se detalla el estado del arte, repasando las soluciones más relevantes y de mayor importancia en la investigación. Para ello describiremos sus puntos fuertes, así como las carencias que motivan la realización de la tesis. El estado del arte se encuentra en el capítulo 2 y se divide en varias secciones. La sección 2.1 describe la mecánica del control de acceso desde las diferentes perspectivas que pueden tener los equipos de usuario, red y los servicios.

En el estado del arte se describen, en la sección 2.2, las credenciales para autenticación y autorización más representativas (secciones 2.2 y 2.2), así como las técnicas de negociación de confianza (sección 2.2).

Los protocolos que dan soporte a la autenticación y autorización para el acceso a la red y los servicios, son comentados en la sección 2.3. Finalmente, en la sección 2.4 del estado del arte, se describen algunas técnicas de procesado de información, que serán útiles durante la realización de la investigación, para el análisis de entidades mediante modelos heterogéneos. Más adelante, en la sección 2.5, se repasarán las necesidades identificadas en la sección de estado del arte, que la tesis tratará de satisfacer.

Finalizada la exposición del estado del arte que afecta a la investigación, comenzamos con las contribuciones. En el capítulo 3 se propone una arquitectura y unos algoritmos para la mejora del control de acceso en los equipos de usuario. Dicho capítulo se divide en varias secciones comenzando con la descripción exhaustiva del problema, en la sección 3.1; más adelante se propone la arquitectura del sistema, sección 3.2; un mecanismo para caracterizar entornos (sección 3.3) y eventualmente utilizar protocolos colaborativos para valorar adecuadamente los mismos (véase la sección 3.4);

terminando con la propuesta de un mecanismo de selección de red, independiente de la tecnología de acceso, que el lector encontrará en la sección 3.5.

En el capítulo 4, se procede a la validación del mecanismo de selección de red, propuesto en el capítulo anterior, mediante dos simulaciones: una cuyo objetivo es demostrar el concepto, en la sección 4.1 y otra, orientada a un caso práctico, que permite realizar precisiones sobre el procesamiento adecuado de los datos, ajuste del modelo... que podrá encontrar en la sección 4.2.

El capítulo 5 presenta, tras una descripción del problema (sección 5.2), la arquitectura (sección 5.3), funcionamiento del algoritmo (secciones 5.4, 5.4 y 5.4), así como consideraciones sobre la medida del riesgo de exponer determinados recursos (sección 5.4), de un sistema de negociación de confianza.

Posteriormente, en el capítulo 6, se presenta una simulación representativa que permite determinar la adecuidad de la solución para los objetivos previstos.

Para terminar las contribuciones, en el capítulo 7, se presenta el diseño de varias extensiones a protocolos estándares para el soporte de autorización, en la sección 7.2; emisión de credenciales de autorización, en la sección 7.3; y negociación de confianza en la sección 7.4. Además, en dicho capítulo, se presentan posibles escenarios de aplicación para las soluciones propuestas 7.5.

Finalmente, el capítulo 8, repasa las principales contribuciones (sección 8.1), conclusiones (sección 8.2) y futuras líneas de trabajo (8.3).

CAPÍTULO

2 Estado del arte

Esta sección describe los trabajos más relevantes para el desarrollo de la tesis. Se divide en varias secciones: la sección 2.1 describe la mecánica del control de acceso desde las diferentes perspectivas. En la sección 2.2, las credenciales para autenticación y autorización (secciones 2.2 y 2.2) y las técnicas de negociación de confianza (sección 2.2). Los protocolos que dan soporte a la autenticación y autorización para el acceso a la red y los servicios, se detallan en la sección 2.3. Finalmente, en la sección 2.4 se cubren las principales familias de técnicas de procesado de información.

2.1. Control de acceso distribuido

La mecánica del control de acceso se basa en encontrar una respuesta adecuada a la pregunta: ¿puede el usuario o entidad acceder a un recurso en concreto dadas unas ciertas restricciones? Sobre esta pregunta genérica se pueden hacer un sinfín de apreciaciones que afectan a cualquiera de las partes involucradas.

Un recurso **R** tiene una sensibilidad **S**, que depende en general de su valor. La medida de este valor puede medirse de forma directa, por el valor económico que expone, o de forma indirecta mediante el valor, perdida o coste de oportunidad que representa su exposición a un tercero. La medida de ese valor puede ser por tanto abstracta o quizá, mejor expresado, subjetiva, influyendo en gran medida en su valor la persona en concreto que lo determina.

La determinación de la sensibilidad del recurso depende también del ámbito de exposición o dominio de utilización, pudiéndose distinguir entre recursos locales, recursos pertenecientes a un dominio o recursos pertenecientes a un conjunto de dominios.

En un proceso de decisión relacionado con el control de acceso, intervienen de forma directa o indirecta las siguientes entidades:

- La entidad que trata de acceder al recurso que podría ser visto como el "cliente" en arquitecturas cliente-servidor.
- La entidad que controla el acceso al recurso o aquella donde se aplican las políticas, es decir, la que determina si es posible el acceso o no.
- La entidad que dispone de ese recurso y de los mecanismos para ofrecerlo de forma compartida. En ocasiones la entidad que controla el acceso al recurso es la misma que dispone de él. Podría llamarse "servidor" en arquitecturas cliente-servidor.

A la hora de determinar si un recurso es accesible o no, se consideran varios aspectos:

- La sensibilidad del recurso, que se recoge en forma de política. El editor de la política es aquel que determina la sensibilidad del recurso. Es posible encontrar políticas de dominio o políticas de usuario; políticas para un recurso en concreto o para un conjunto de recursos; aplicables a un conjunto de dispositivos o servidores o a solo uno de ellos. Si estuvieran presentes más de una política habría que fusionarlas.
- La condiciones del entorno, como pueden ser la fecha, hora, localización u otras variables sean físicas o subjetivas.
- La información disponible sobre la entidad que trata de acceder al recurso que podrían ser, identidad, claves, propiedades o atributos.

En esta sección haremos un repaso de las diferentes perspectivas de control de acceso según su ámbito y por tanto sensibilidad, analizando qué recursos se pueden identificar como susceptibles de proteger, mediante control de acceso, en esos diferentes ámbitos.

Perspectiva del equipo de usuario

Los equipos de usuario, que incluyen desde el ordenador personal a la PDA o teléfono móvil, disponen de una serie de recursos que deben ser protegidos desde fuera hacia dentro, que sería el caso de un equipo que ofrece servicios al exterior y que es protegido por un cortafuegos. Del mismo modo es necesario proteger el equipo de

dentro hacia fuera, como en el caso de un gestor de conexiones basado en preferencias de usuario. El control de acceso en los equipos de usuario se realiza por tanto (ver Figura 2.1):

- De fuera a dentro: protegiendo el acceso a los recursos compartidos localizados en el terminal. Este tipo de control de acceso es muy conocido y trata de evitar accesos no autorizados requiriendo autenticación/autorización.
- De dentro a fuera: protegiendo los recursos radio del acceso de las aplicaciones. Este control de acceso debe ser capaz de:
  - Lograr que las aplicaciones del terminal se conecten solo a través de las redes adecuadas a sus demandas de tráfico.
  - Conectar a redes conocidas y probadas reduciendo el tiempo de conexión.
  - Utilizar el contexto para determinar las redes más atractivas permitiendo, por ejemplo, que ciertas aplicaciones sólo envíen paquetes dentro de una red segura, como la del trabajo.
  - Garantizar un uso eficiente de los recursos seleccionado redes con el menor coste posible.
  - Garantizar un cierto grado de protección de forma autónoma, es decir, sin intervención de la red, servicios o terceras partes que ayuden en el proceso.

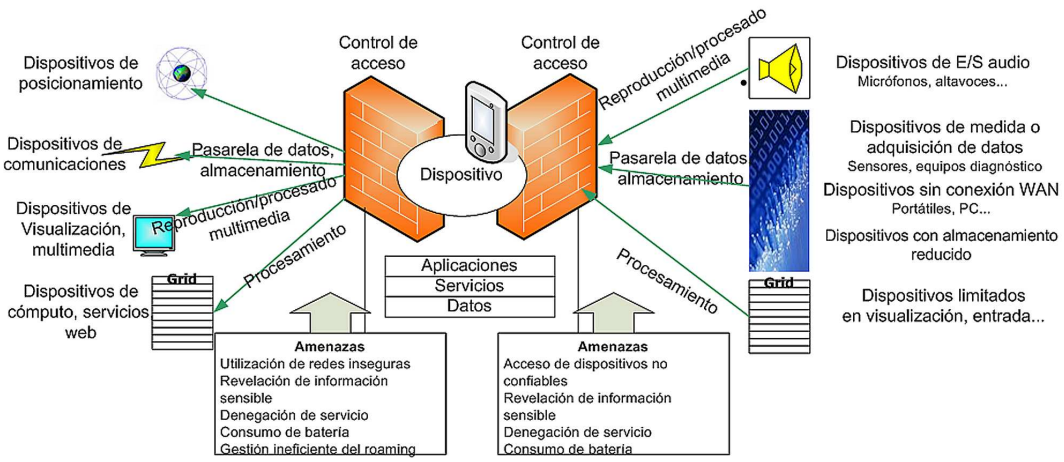


Figura 2.1: Control de acceso de entrada y control de acceso de salida y su relación con dispositivos que típicamente rodean a un dispositivo personal como una PDA o móvil

Dado que hoy en día, las arquitecturas cliente-servidor están más difuminadas, debido a la aparición de paradigmas de computación ubicua y redes de pares, los servicios ofrecidos por terminales de usuario se caracterizarán en la sección 2.1, tratándolos como servicios en general.

Respecto al acceso a la red, la proliferación de dispositivos personales junto con los avances en las tecnologías de comunicación, han fomentado la aparición de dispositivos que soportan varios sistemas de acceso radio como UMTS [24], 802.11x [25], WiMax [26] bluetooth [27]. Por otro lado, avances en el procesado de señales y el hardware reconfigurable han dado lugar a lo que se conoce como *Software Defined Radio* (SDR). Introducido por primera vez en el año 1995 en [28], el SDR, tenía como principales características satisfacer las demandas de las comunicaciones militares [29]: aparatos de radio capaces de sintonizar una gran cantidad de bandas de frecuencias e implementar cualquier tipo de modulación, permitiendo cambios en la funcionalidad radio mediante cambios en el software que controla el hardware programable. Hoy en día el SDR se está haciendo un hueco en el mercado de consumo ya que una mejora de ciertas rutinas del firmware de radio de los dispositivos móviles, proporciona mejor cobertura, calidad de voz, etc. Aún no han aparecido terminales capaces de "mutar" a nuevas modulaciones radio con un simple cambio en el software, o con un hardware programable tan genérico que lo permita, pero a medida que las economías de escala abaraten los dispositivos programables y de procesado de señal, este tipo de dispositivos llegarán a ser cotidianos.

El roaming entre proveedores se define como la capacidad de obtener conectividad/servicio utilizando las redes de otro proveedor distinto de aquel con el que realizó el registro inicial el equipo de usuario. Dado el creciente número de tecnologías radio, es de suponer que los proveedores se diversificarán en tecnología de acceso y, a su vez, aparecerán nuevos proveedores que utilicen tecnologías nuevas, de amplia cobertura, alternativas a UMTS, como Wimax [26]. Este fenómeno dará lugar a acuerdos entre proveedores que se materializarán en acuerdos de roaming efectivos y soluciones que permitan roaming espontáneo en base a las preferencias del usuario. Es decir, el roaming dejará de estar motivado por la falta de cobertura de la red original, para estar motivado por necesidades puntuales de conectividad o de calidad de servicio de las aplicaciones de usuario.

La movilidad del usuario da lugar a otros conceptos, como el *handover*, ya conocidos desde los tiempos de la telefonía celular, pero que se amplían sensiblemente. El handover en GSM se conoce como el cambio de una celda a otra. En GSM, teniendo en cuenta unas restricciones de velocidad (por efecto doppler) y de atenuación de la señal, el handover se puede realizar sin interrumpir los servicios de voz y datos. En el caso que nos ocupa, aparece el *handover vertical*, en el cual se varía la tecnología de conexión manteniendo la sesión; y el *handover horizontal*, que se define como el cambio

de hotspot o punto de acceso sin interrumpir la conexión manteniendo la tecnología. La figura 2.2 muestra multiples acciones de roaming entre proveedores y handover.

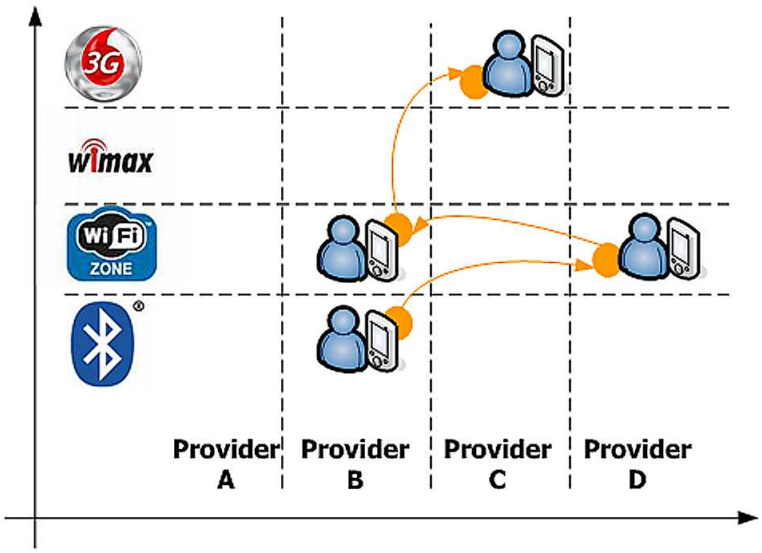


Figura 2.2: Handover vertical y horizontal.

Hasta ahora, teniendo en cuenta los escenarios de un solo proveedor o en los que las conexiones, por ejemplo, de tipo WiFi, se configuraban a mano, el control de acceso se relegaba al operador o dueño de las redes de acceso. En este sentido, algunos trabajos intentan resolver el problema de roaming entre redes trasladando la gestión en mayor medida a la red [30], realizando asunciones en cuanto al tipo de credenciales o protocolos a utilizar [31, 32], estando los protocolos tan acoplados al nivel de enlace [5], que restan generalidad al problema e impiden que escale. En presencia de multiples proveedores y tecnologías de acceso, dado el gran número de interfaces radio y proveedores, sería interesante disponer de un control de acceso en los dispositivos de usuario que, considerando el contexto en el que las distintas aplicaciones solicitan acceso a la red y los requisitos de calidad que demandan, permita o no realizar las conexiones.

Algunas de las entidades con las que interactuará el equipo de usuario en los procesos de seguridad (autenticación, autorización y gestión de claves) son las siguientes:

- Proveedor de acceso a la red o Network Access Provider (NAP): entidad responsable de los equipos que permiten el acceso a la red utilizando una tecnología de acceso dada.

- Proveedor de servicios de Internet (ISP): utilizan varios NAPs para lograr acceso

capilar al usuario dado que éste podrá utilizar varias tecnologías de acceso. Los ISPs podrán proporcionar acceso básico o por servicios.

- Servicios de terceros: Aquellos servicios de datos, multimedia u otros, que utilizan las redes de los ISPs, e indirectamente las de los NAPs, para llegar al usuario.

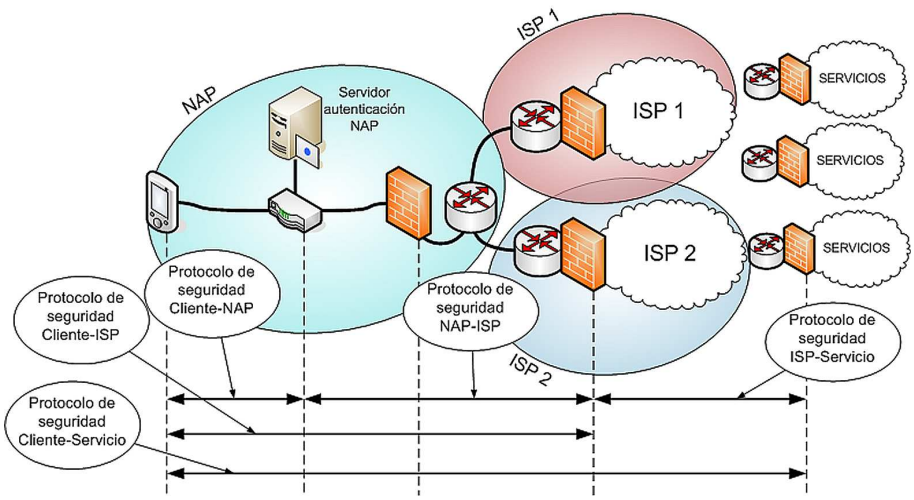


Figura 2.3: Entidades relacionadas en el acceso a la red y a los servicios: proveedores de acceso a la red, proveedores de servicios de internet y proveedores de servicio.

Como se observa en la figura 2.3, el cliente puede necesitar iniciar y mantener varios intercambios de credenciales con las diversas entidades en el camino de acceso a los servicios. Es posible incluso que el equipo de usuario deba, para acceder a los servicios, proporcionar credenciales bajo demanda a las entidades entre el cliente y el servicio, con la intención habilitar el transporte de esos servicios por la red.

Existen tareas asociadas al control de acceso, a las que debe dar cobertura el equipo de usuario, como la valoración y comunicación de riesgos al usuario. El equipo de usuario debe velar también por la seguridad de aquel que lo usa y proporcionarle información sobre las decisiones tomadas. En ocasiones, **la verdadera vulnerabilidad del sistema es el propio usuario** que, por falta de conocimiento o negligencia, lleva a cabo acciones que contravienen la seguridad de sus propios datos. La velocidad a la que aparecen nuevos productos y servicios de comunicación, unidos a la popularidad creciente de Internet y a su difusión entre el público en general, hacen que los usuarios relajen su percepción del riesgo, pensando que todo es mucho más sencillo de lo que parece. Albert O. Hirschman recoge en [33] una teoría similar a ésta para inversiones económicas, que permite determinar las causas que impiden la percepción del riesgo asociado a una decisión económica.

La mayoría de estos riesgos se mitigan con la formación básica del usuario en temas de seguridad, pero existen otras maneras de conseguir el mismo resultado, por ejemplo, hacer al usuario consciente de los riesgos que conlleva una acción usando un modelo mental alternativo conocido para él.

La forma típica de comunicar el riesgo al usuario es a través de diálogos (ventanas) de warning como las que aparecen, por ejemplo, al aceptar como confiable un certificado o una clave. En estas advertencias se pregunta al usuario si desea aceptar el certificado o clave como válido, a lo cual puede responder sí permanentemente, sólo durante la sesión o responder que no. Los usuarios medios que desean acceder a servicios no son conscientes de la posibilidad de un ataque de man-in-the-middle o phishing y suelen aceptar el certificado.

Este comportamiento nos lleva a la necesidad de utilizar mecanismos de autenticación y autorización combinados que permitan alcanzar un estado de seguridad más alto a medida que se requiera. Estas técnicas llamadas de negociación de confianza (Trust Negotiation) serán analizadas más adelante en la sección 2.2.

De momento vamos a intentar analizar el comportamiento y la visión del riesgo de los usuarios. Existen varios estudios que tratan de analizar el problema de la percepción del riesgo por parte de los usuarios. En [34] se estudian varios modelos mentales cotidianos sobre un grupo de expertos y otro grupo de inexpertos para los más cercanos al modelo de seguridad informática. En este artículo se relacionan otros estudios, como [35], donde se muestra cómo las personas asumen el riesgo de forma poco realista en comunidades online. Este estudio basa sus conclusiones en la decisión de las personas en cuanto qué cantidad de datos personales deben aparecer en su perfil dentro de una comunidad.

Por otro lado, en [36], se analizan varios modelos mentales, poniéndose de manifiesto que las personas interactúan con los sistemas en base a sus creencias y a aquello que asumen como cierto respecto al mismo: el estudio razona cómo la capacidad del usuario de aprender a usar el sistema, la usabilidad y la funcionalidad del mismo, dependen de la correlación entre el modelo mental del diseñador y el del usuario final. De esto se concluye que la efectividad de la comunicación al usuario, del riesgo asociado a un comportamiento o acción, dependerá de a que grupo pertenece el usuario. De ahí que en [37] se analicen los modelos mentales para prevenir riesgos domésticos, de forma que se minimice el exceso de confianza que el usuario pueda tener en su seguridad personal.

En el estudio [34], donde se analizan los modelos mentales cotidianos, se analizaron los siguientes: seguridad física, infecciones médicas, comportamiento criminal, guerra y fallos del mercado. El estudio realiza dos experimentos en los que se diferencian expertos de inexpertos. En cada uno de los experimentos se clasifica de forma

diferente los dos grupos dando lugar a dos subgrupos dentro de cada grupo. Aquellos con el apelativo de "fuerte" son individuos que se pueden clasificar como de un grupo cuando cumplen los requisitos más restrictivos de pertenencia o "débil" cuando los requisitos son laxos. Así se obtienen cuatro grupos diferenciados: un grupo de inexpertos fuertes, inexpertos débiles, expertos fuertes y expertos débiles.

Los participantes en el experimento asocian palabras con categorías mediante un juego de ordenación de cartas. Los modelos considerados eran: económico o fallos de mercado, guerrilla, físico, criminal y médico. Tanto los expertos débiles como los inexpertos débiles rechazan el modelo mental médico; los expertos débiles excluyen el modelo de fallos de mercado y el de guerra. Los inexpertos y expertos fuertes rechazan el modelo médico. Para los inexpertos fuertes los modelos criminal, físico y económico están más cercanos al modelo de seguridad que los demás. En cambio para los expertos fuertes ocurre que los más cercanos son el criminal, físico y de guerra. Por tanto parece ser mucho más adecuado comunicar los riesgos a los usuarios, expertos y no expertos, mediante metáforas provenientes de modelos mentales criminales y de seguridad física como podrían ser *robo, secuestro, puerta, llave, guardaespaldas...*

La razón de buscar similitudes entre modelos mentales es poder comunicar el riesgo mediante comparaciones, para que sea sencillo de entender por el usuario. La comparación de riesgos es útil ya que permite comparar un riesgo, desconocido para el usuario, con algo que conoce o que puede manejar. De esta forma puede hacerse una idea de las consecuencias de una acción. A parte de la comparación con otros modelos mentales, la comparación entre riesgos del mismo modelo es muy útil para la comprensión, sobre todo porque la comparación de riesgos es más exitosa cuando se comparan riesgos similares. Por ejemplo, es más sencillo que una persona entienda el riesgo que conlleva para la salud consumir un alimento desconocido, si se compara con otro conocido, cuyo consumo acaree las mismas consecuencias; en cambio, puede ser complicado de entender si se compara, por ejemplo, con riesgos financieros.

En [38] se describen las formas de comparar riegos que mayor impacto tiene en las personas, siendo las comparaciones con riesgos similares, las comparaciones con alternativas o la comparación de riesgos con beneficios, las que producen mayor impacto.

Es por esa razón que el dispositivo de usuario debe proporcionar a éste, de la forma más comprensible, información del proceso de toma de decisiones. Para ello el dispositivo puede necesitar recurrir a técnicas de análisis y representación de datos como las que describiremos en la sección 2.4.

Perspectiva de la red

Las redes de acceso disponen de varios mecanismos de protección relacionados, directa o indirectamente, con el control de acceso: la autenticación, la autorización, el cifrado y los filtros de paquetes.

La autenticación de dispositivo o de usuario consiste en que el usuario o equipo proporcione una prueba de identidad durante el proceso de conexión a la red. La autenticación en las redes wireless depende de la tecnología, ya que típicamente, se realiza a nivel de enlace. En muchas ocasiones se ha hablado de los pros y contras de realizar la autenticación a niveles superiores como en [39].

Entre los factores determinantes a la hora de elegir el nivel al que hacer la autenticación se encuentra el número de paquetes intercambiados y el tiempo empleado en comunicarse con la entidad que realiza la autenticación, ya sea el elemento situado en el primer salto o un servidor de autenticación situado en otro punto. Por otro lado, destaca la sencillez ya que cuanto más sencillo es, por parte del usuario instalar y usar la red, más atractiva se hace la tecnología comercialmente. Realizar la autenticación a nivel de enlace, afecta a la implementación, y por tanto, al coste de la tecnología de manera que, la sencillez de la misma, puede afectar a la implantación por motivos de coste, haciéndose más atractiva aquella menos costosa de implementar. En la sección 2.3 se hace un repaso a los distintos protocolos de seguridad entre los que se encuentran, en la sección 2.3, los protocolos de autenticación para el acceso a la red.

Como se comentó en la sección anterior, es necesario distinguir entre el proveedor de acceso a la red, NAP, y el proveedor de servicios de Internet (ISP). En un ambiente de diversificación de negocio, ayudado por el desarrollo de nuevas tecnologías de acceso, los ISPs tenderían a proporcionar paquetes de servicios de transporte a través de varias tecnologías de acceso gestionadas por los NAPs. La situación actual ha llevado a un proceso de sinergias y fusiones de empresas para proporcionar acceso a servicios de transporte y de valor añadido a través de varias redes, tanto móviles como fijas, asumiendo la misma marca comercial el papel de NAP e ISP. En cambio, con la aparición de nuevas tecnologías será posible disponer de un mayor abanico de tecnologías de acceso y por ello, será muy probable la aparición de nuevos proveedores que actúen como NAPs. Lo que nos lleva a la necesidad de diseñar nuevos mecanismos para establecer relaciones de confianza entre las partes involucradas en el acceso a los servicios.

Respecto al acceso a los servicios es posible considerar dos escenarios. En el primero, el cliente gestiona todas las credenciales de acceso a los servicios, disponiendo de credenciales de acceso a varios NAPs, ISPs y servicios. Se entiende que existen relaciones de confianza y acuerdos de colaboración preestablecidos entre algunas partes. En este escenario se podrían llegar a situaciones en las que no se pudiera utilizar un ISP a través de un determinado NAP, determinados servicios a través de ciertos NAPs o

ISPs. En el segundo, el cliente dispone de credenciales de varios NAPs e ISPs, así como las credenciales de acceso a los servicios. Los ISPs y NAPs no disponen de acuerdos preestablecidos, sino de **reglas y estrategias de negociación** con el cliente y entre entidades. Esta flexibilidad permite determinar si un cliente puede acceder a la red o no, e informar a todas las partes involucradas de los detalles de las relaciones establecidas bajo demanda como, por ejemplo, la facturación aplicada.

Estos dos escenarios son antagónicos. El primero carece de flexibilidad, requiriendo acuerdos preestablecidos entre todas las partes para conseguir acceso al servicio de forma universal. El segundo escenario es abierto y flexible, y permite cualquier tipo de interacción entre las partes, aunque puede requerir negociaciones de seguridad complejas, que aumentarían la latencia en situaciones de gran tasa de movilidad. Por tanto, un escenario **balanceado** puede ser más atractivo. En este escenario balanceado, existirían cierto número de acuerdos entre partes y se permitiría la negociación dinámica de nuevos servicios y calidades. La negociación podría ser necesaria en casos de falta de conectividad a través de los operadores habituales o, cuando fuese necesario incrementar, de forma dinámica, la calidad asociada a la conexión o los servicios soportados por la misma. Esta negociación puede involucrar a todas las redes de transporte intermedias.

La conclusión a esta sección es que con independencia del mecanismo de control de acceso utilizado, el operador debe proteger sus recursos de una forma eficiente, sin ser un lastre para la movilidad y permitir el despliegue de nuevas tecnologías de red, así como servicios de valor añadido, sin grandes esfuerzos. El operador debe también asumir como tarea el facilitar a los clientes el acceso a los recursos, proporcionando **mecanismos sencillos de acceso**. Estos mecanismos de acceso deben poder ser utilizados por varias tecnologías de acceso, de forma que no suponga una barrera de entrada al usuario no experto. En la sección 2.3 discutiremos los distintos protocolos de autenticación y autorización, dando una mayor importancia a aquellos menos dependientes del hardware, que operan en o por encima de la capa de red, dado que proporcionan mayor flexibilidad que otros.

Perspectiva del los servicios

Los servicios se proporcionan al usuario a través de las redes de acceso o en redes de pares. El control de acceso a los servicios se realiza típicamente extremo a extremo. Las credenciales involucradas en el proceso de autenticación y autorización suelen ser distintas a las de acceso a la red, haciéndose más patente esa separación, en esquemas en los que las redes de acceso están separadas administrativamente de los proveedores de servicios.

En situaciones en las que los ISPs proporcionen además servicios de valor añadido, las credenciales para el acceso al ISP pueden ser las mismas a usar con el servicio, por ejemplo, los casos de distribución de contenidos multimedia, en los que terceros proporcionan servicios de pago por visión utilizando las redes, mecanismos de autenticación, autorización y facturación del ISP. El control de acceso a servicios necesita garantías acerca de la seguridad de las redes de acceso para evitar ataques de tipo man-in-the-middle, como los recogidos en [40]. Este tipo de ataques consisten en que un atacante finge ser un punto de acceso válido, enrutando paquetes de autenticación y autorización para el acceso a la red y servicios, permaneciendo el tiempo suficiente para luego robar el servicio al cliente original. El atacante se aprovecha de la apertura de filtros, por parte del proveedor, tras una autenticación/autorización exitosa. Esto ocurre cuando el cliente utiliza mecanismos de autenticación que no generan claves, sobre túneles ya establecidos sin verificar adecuadamente los extremos del túnel.

Pese a que son necesarias unas ciertas garantías sobre la seguridad de las redes de acceso, en última instancia, los servicios deben disponer de mecanismos de **autenticación extremo a extremo** que, pudiendo hacer uso de información de capas inferiores, permitan autenticar al cliente con independencia de las capas inferiores. El protocolo SSL (Secure Socket Layer) [41, 42] desarrollado inicialmente por Netscape, proporciona seguridad extremo a extremo para autenticar entidades por encima del nivel de red. SSL se ejecuta sobre TCP/IP y proporciona autenticación de servidor y autenticación opcional de cliente. La evolución de SSL, conocida como TLS (Transport Layer Security) [43] y estandarizada por la IETF, proporciona, al igual que SSL, autenticación de servidor y opcionalmente mutua, junto con un mecanismo de extensión para soporte de nuevas funcionalidades.

En ocasiones se ha sugerido que la vida de TLS estaría supeditada a la aparición de un protocolo que protegiera la red nodo a nodo, como es caso de IPSEC [44]. Es lógico suponer, que no sería necesaria esa seguridad extremo a extremo si existiese una seguridad proporcionada globalmente a nivel de red. Pero la situación es, en realidad, diferente dado que, ni el cliente ni el servicio, pueden obtener garantías fehacientes de que se está cifrando nodo a nodo en todo el camino desde el cliente al servicio. Esta imposibilidad de verificar la confianza de todo el camino hacen que sea complicado prescindir de seguridad extremo a extremo.

TLS, que se explicará con detalle en la sección 2.3, no define nuevos mecanismos de cifrado ni de intercambio de claves, sino que reutiliza los existentes. TLS permite la utilización de mecanismos de intercambio de claves anónimos y otros basados en PKI (certificados de clave pública) [45], que permiten además, autenticar al servidor/servicio.

Como se explicará más adelante en la sección 2.2, determinados esquemas de clave pública cuya validez es larga en el tiempo, como PKI, requieren que su validez sea com-

probada en el momento, para verificar que no ha sido revocado por perdida o compromiso de la clave. En interacciones sobre redes de pares, en las que no existen posibilidad de obtener acceso a una red con conectividad a Internet, es necesario recurrir a mecanismos colaborativos, como Pervasive Trust Manager (PTM) [10], para determinar la confianza de otras entidades que ofrecen servicios o que tratan de acceder a los ofrecidos.

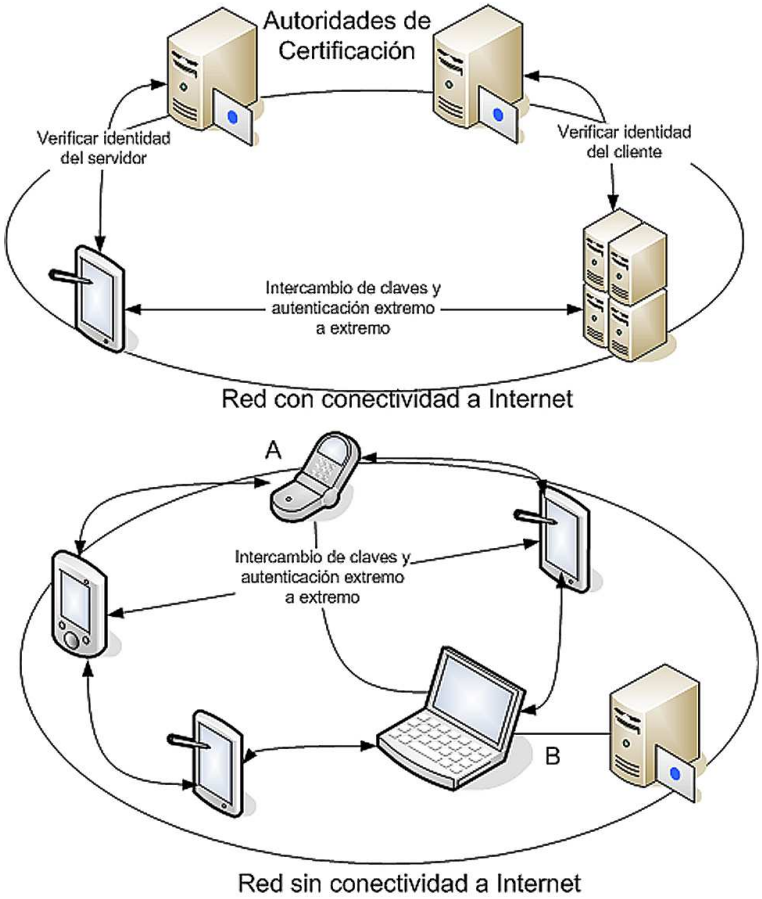


Figura 2.4: Interacción entre entidades para acceso a servicios en redes con conectividad donde es posible verificar el estado de revocación de las credenciales y redes donde no es posible realizar esa comprobación.

En la figura 2.4 se muestra dos posibles escenarios. En el primero, al haber conexión a Internet, es posible para ambas partes, verificar las credenciales mediante consulta a las autoridades que las emitieron. Las decisiones en ese escenario se toman en una situación de **información perfecta**, por lo que la incertidumbre puede considerarse nula. En cambio en el escenario de redes sin conexión a Internet no es posible. Por

tanto, es necesario utilizar la **información contenida en el entorno**, mediante el uso de protocolos colaborativos, para verificar una credencial o reducir la incertidumbre. Podría darse el caso de que un dispositivo, como **A** en la figura, que se acabe de incorporar a la red, disponga de información de revocación fresca que aportar al resto de las entidades. O que alguno de los dispositivos disponga de otro interfaz conectado a otra red (**B** en la figura) con conexión a Internet.

Los protocolos colaborativos, aprovechan la información contenida en el entorno para reducir la incertidumbre en las decisiones, pese a que no se llegue a disponer de información perfecta. Existen varios trabajos en ese sentido entre los que cabe destacar PTM [46, 47] por su adaptabilidad a terminales con capacidad de cómputo limitada y buenos resultados.

Otro de los procesos involucrados en el control de acceso que realiza el servicio es el de autorización. Para determinar si un individuo está autorizado a utilizar un servicio, dicho servicio debe comprobar los privilegios del individuo, una vez éste se ha autenticado y no cabe duda de que es quien dice ser. Para determinar los privilegios se han propuesto muchos sistemas, desde la estructura de permisos de UNIX tipo "r-x-w" a esquemas de roles [48] o certificados de atributos [49], pasando por listas de control de acceso discrecionales (DACL). En la sección 2.2 se analizan las credenciales más representativas para el proyecto.

2.2. Credenciales y políticas de soporte

Autenticación

La autenticación es un proceso que permite determinar la procedencia en caso de objetos o la identidad en caso de personas. Se trata de confirmar que algo sobre el objeto o persona es auténtico. La autenticación siempre es el paso previo a la autorización dentro de un esquema de control de acceso. La autenticación se basa en algo conocido, como contraseñas o claves; en algo poseído como tarjetas inteligentes; en propiedades del sujeto (biometría), como huellas o iris; o en propiedades de los dispositivos, como los parches de seguridad que tiene instalados... En esta sección describiremos aquellos que tienen relevancia para el trabajo a desarrollar.

Históricamente se ha recurrido a diversas técnicas para verificar la autenticación, pero ninguna a resultado tan atractiva y duradera en el tiempo como la palabra de paso. Una entidad o usuario se identifica mediante un nombre y una palabra de paso, siendo el nombre único y público, dentro de un entorno, y la palabra de paso, el secreto compartido, entre el que autentica y el autenticado.

La autenticación mediante el uso de certificados X.509, de clave pública o certificados PKI, descrita en [50][51] la conocida infraestructura de clave pública o PKI. PKI sustituye el par nombre de usuario y contraseña por un certificado de clave pública y su correspondiente clave privada, habilitando además, un conjunto de servicios añadidos, como firmas o cifrados asimétricos. Aquello que identifica unívocamente al usuario en un dominio es el certificado, y la prueba de autenticación se realiza mediante una prueba de posesión de la clave privada, asociada a la clave pública del certificado. En la infraestructura de clave pública aparecen las Autoridades de Certificación o CAs. Estas autoridades son las encargadas de garantizar que la clave contenida en el certificado pertenece al usuario cuyo nombre aparece en el certificado. El proceso de verificación de una identidad en PKI tiene dos fases, en la primera se verifica una prueba de posesión de la clave privada y en la segunda se verifica la validez del certificado contra la autoridad de certificación. De lo expresado con anterioridad, se deduce que la autoridad de certificación deberá mantener una lista actualizada de aquellas identidades que han sido revocadas, así como proporcionar esa información a todo autenticador que la necesite.

Un usuario en PKI se identifica por un nombre de directorio contenido en el certificado. Este nombre es único en un espacio de nombres global. PKI utiliza nombres para identificar a las entidades y permite periodos de validez altos por lo que, para lograr que escale mejor, se organiza de forma jerárquica. Una identidad basada en un nombre presenta ciertos problemas de escalabilidad en entornos dinámicos y distribuidos, dado que es necesario tener acceso a las listas de revocación en todo momento. Por otro lado, la utilidad de estas credenciales en aplicaciones de e-government o para identidad en pasaportes o tarjetas de identidad electrónicas, esta fuera de toda duda. Otra de las ventajas de PKI es la automatización del proceso de autenticación sumada a la posibilidad de intercambio de claves mediante el protocolo TLS [52][53].

Otros sistemas de autenticación asocian identidades a claves en lugar de a nombres. De esta manera, se elimina la dependencia con un espacio de nombres global (se sustituye por el espacio de claves). Este tipo de credenciales se conocen en inglés como “key-centric”, entre los que destacan KeyNote [54], SPKI [55] o [56]. Estas credenciales proporcionan datos de autenticación, e incluso de autorización, asociando atributos a claves. El problema es que la autenticación y la autorización pasan a estar muy fuertemente acopladas.

Esfuerzos actuales basados en XML, como Security Assertion Markup Language (SAML) [57], se pueden considerar importantes en materia de autenticación y autorización combinada. SAML en si mismo no es un mecanismo de autenticación ni autorización, sino una forma de comunicar decisiones de autenticación y autorización para ayudar en dichas tareas. SAML es un lenguaje basado en XML, que define un lenguaje para codificar atributos y asertos de otros mecanismos de autenticación, y una serie

de definiciones para su utilización conjunta con otros protocolos. Es decir, es simplemente un vehículo para comunicar decisiones de autenticación y autorización. Dado que SAML utiliza mecanismos de autenticación existentes sufre de las mismas limitaciones relativas a los espacios de nombres de PKI y de las de los sistemas centrados en claves. Por otro lado, SAML es más adecuado para autenticación en web que para autenticación de acceso a la red o a servicios tradicionales via TLS, por lo que no es una solución adecuada para todos los escenarios.

Autorización

Tras la autenticación, se procede a encontrar los privilegios asociados a la entidad que acaba de ser autenticada. El procedimiento más simple es buscar el nombre en una lista que asocie privilegios sobre un recurso a identidades. Estas listas se conocen como listas de control de acceso o DACLs. Un caso aun más sencillo es el esquema de permisos de Unix.

Las técnicas modernas de autorización utilizan Roles para asignar privilegios a identidades. Se asignan roles a identidades y privilegios a roles, así el sistema escala mejor, ya que pueden asignarse nuevos roles a identidades o cambiar los privilegios de un rol unilateralmente, sin necesidad de cambiar los privilegios identidad a identidad. Los roles pueden ser expresados con certificados de clave pública [58] y con certificados de atributos [50], pero tiene problemas derivados de asignar identidades a nombres. SAML [57] y otros esquemas como el descrito en [59], pueden ser utilizados para expresar decisiones de autorización o privilegios, pero tienen las limitaciones de los sistemas basados en claves. Estos últimos no disponen de expresividad suficiente, para definir separación de tareas u otras características deseables, como se pone de manifiesto en [60], donde se analizan los sistemas de control de acceso y lenguajes que permiten expresar privilegios.

La aproximación natural de PKI a la expresión de privilegios se consigue mediante Certificados de Atributos. Los certificados de atributos son el mecanismo que permite enlazar los nombres de las entidades, extraídos de los certificados de PKI, con los privilegios, y compararlos con los requisitos del recurso. La recomendación ITU X.509 [50] define un modelo jerárquico de autorización basado en Certificados de Atributos llamado PMI (Privilege Management Infrastructure). Los certificados de atributos son emitidos por Autoridades de Atributos (AAs). Cada AA tiene potestad para asignar privilegios sobre recursos pertenecientes al mismo dominio administrativo de la AA, pudiendo definir roles o delegar privilegios.

La autorización es muy dinámica en PKI, mientras que la autenticación es prácticamente estática. La identidad como tal no varía en gran medida. En la recomendación

de la ITU se define una extensión para incluir atributos, que proporcionen información de autorización, en los certificados de identidad; pero no tiene mucho sentido mantener un único certificado para la identidad y la información de autorización. Esto se debe a que la fuente de información de autenticación puede ser externa, mientras que la fuente de autorización debe ser parte del dominio donde se encuentran los recursos. Por lo tanto, carece de sentido que una autoridad externa al mismo sea la encargada de gestionar esos privilegios.

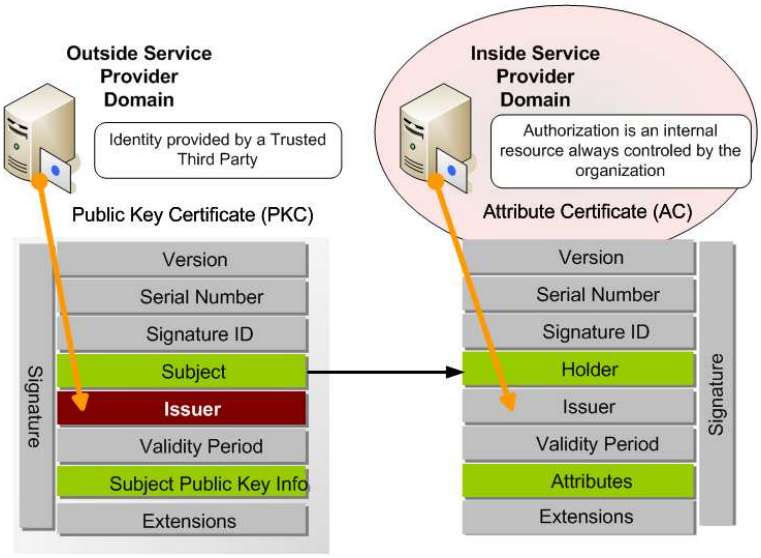


Figura 2.5: Estructura de los certificados de atributos y las similitudes con los certificados de clave pública.

PMI complementa a PKI proporcionando cobertura a la autorización. Los certificados de atributos (ACs) asignan privilegios a identidades como se muestra en la figura 2.5. Los certificados de atributos pueden almacenarse en repositorios. En este caso, las aplicaciones se encargan de buscar en ellos para poder determinar los privilegios. Estos mecanismos son conocidos como “pull”, dado que es la aplicación o servicio quien realiza la búsqueda en el repositorio (observe la figura 2.6). Este tipo de mecanismos permiten incorporar PMI sin modificar las aplicaciones de los clientes. Por otro lado, existe la posibilidad de que los certificados sean transportados por el cliente y se envíen al servicio tras la autenticación. Estos mecanismos de “push”, requieren cambios en las aplicaciones de cliente, pero aumentan el rendimiento.

En escenarios en los que no se dispone de conexión a Internet, las entidades debe llevar consigo los certificados para poder interactuar o delegar privilegios a terceros. En ese caso es más adecuado el modelo “push”. Los problemas de autorización y delegación en este tipo de entornos se tratan en [6].

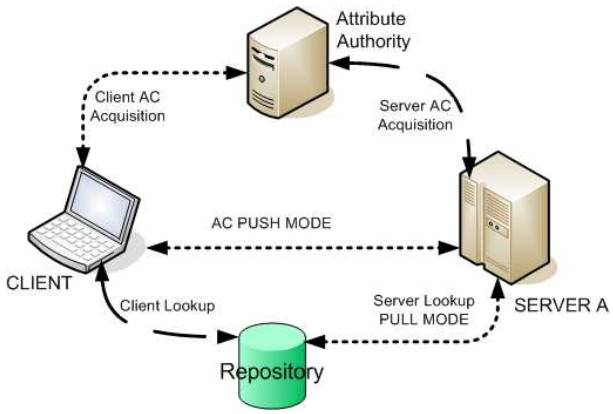


Figura 2.6: Intercambio de certificados de atributos entre entidades. Los certificados puede enviarlos en cliente directamente al servidor o autenticarse y que el servidor consulte un repositorio.

Otras credenciales que permiten incorporar información de autorización son, como ya hemos comentado, KeyNote [54], SPKI [55] o [56]. Los esquemas de push y pull son aplicables también a estas credenciales. SAML de hecho, permite aún mayor flexibilidad permitiendo la participación de varias partes en la distribución de hojas SAML.

Las políticas y la negociación de la confianza

Los mecanismos de autenticación y autorización comentados en las secciones anteriores asumen que las entidades ya se conocían previamente o, al menos, confían en las autoridades que emitieron las credenciales. Por ejemplo, la autoridad de certificación en PKI debe ser confiable para el sistema o, en el caso de roles, el rol o grupo debe ser conocido.

En entornos dinámicos, en los que la movilidad es muy alta, el número de interacciones con dispositivos no conocidos aumenta. Para permitir las interacciones en estos entornos, es necesario recurrir a técnicas que permitan alcanzar un estado de seguridad o confianza entre entidades que no tienen conocimiento previo la una de la otra, o que no necesariamente conocen las autoridades de certificación que han emitido las credenciales. Estos mecanismos se conocen como **negociación de confianza** aunque, en realidad, sea un proceso de autenticación y autorización en varias fases.

Existen varios esfuerzos en esta temática orientados a entornos web, como los que se pueden encontrar en [61][62], que permiten negociar confianza con extraños, proporcionando autenticación y autorización. La negociación se lleva a cabo liberando credenciales y políticas. Dado que no existe confianza previa, es necesario liberar va-

rias credenciales y, en algunos casos, utilizar credenciales basadas en propiedades de la entidad, en lugar de algo que tiene o conoce. Las credenciales basadas en propiedades proporcionan detalles acerca de las características de la entidad y no de su identidad y sus capacidades.

De esta manera un extraño puede ser autenticado y autorizado aunque, la calidad de servicio proporcionada dependerá de la política aplicable al servicio. La negociación se basa en el hecho de que existe una política o grupo de políticas que protegen un recurso y que, por tanto, existe una manera de averiguar las credenciales que es necesario aportar para obtener acceso a dicho servicio. Un enfoque un poco ingenuo, podría proponer que en la negociación, el cliente mostrara todas las credenciales de las que dispone, decidiendo el servicio si le concede o no acceso y con qué calidad. Un ejemplo cotidiano similar, es acudir a una tienda y dejar todo el dinero sobre la mesa para que el tendero decida que productos nos vende. En este caso el cliente desvela claramente más información de la que debe y la negociación en sí no puede considerarse justa para el cliente.

Otra forma de atacar el problema sería que el tendero proporcionase al cliente una lista con todos los productos y precios finales a los que está dispuesto a llegar, y que el cliente decidiese cuáles compra. En este caso el servicio mostraría la política al usuario y este decidiría qué credenciales aporta. La negociación, en este caso, no es justa para el servicio, ya que desvela más información de la necesaria.

Detrás de toda negociación debe haber una estrategia que vele por los intereses de ambas partes para así, proteger los intereses de ambas partes y su privacidad. Imaginemos una negociación, en un entorno P2P, en la que la política de una de las partes, requiere una credencial que acredite la pertenencia a un area de hospitalización para revelar los informes médicos almacenados en el dispositivo (recurso). Suponiendo, como es lógico, que no deseamos revelar a cualquiera que disponemos de informes médicos en nuestro dispositivo, no deberíamos desvelar jamás la política completa, dado que aparecen detalles sobre los requisitos necesarios para acceder a los informes. Sólo se deberá revelar la parte de la política que solicita pertenencia al hospital, a aquellas personas en las que confíe o que tengan que ver con el tratamiento. Para lograr esa grado de confianza, en el que el riesgo de revelar información sensible haya descendido lo suficiente, ha debido existir un intercambio de credenciales previo, utilizando una política o partes de otras políticas, que no contenían alusión alguna al hospital. Del mismo modo querremos mantener protegido quién gestiona nuestras cuentas, qué límites tenemos en las tarjetas, qué aseguradora usamos, dónde compramos...

En [63] se describen algunos requisitos que debe cumplir un sistema de negociación, como son: semántica bien definida, monótonos, sometidos a restricción, protección de políticas sensibles y formalismo unificado. Las políticas juegan un rol muy importante en este tipo de negociaciones. En [61] se estudian estos sistemas y se reconoce

que, la única manera de proteger la privacidad de las partes, es liberar las credenciales y políticas que contienen información sensible de forma gradual, acorde al nivel de confianza alcanzado hasta el momento.

Pese a que, tanto las credenciales como las políticas se liberaran de forma gradual seguiría siendo posible un ataque: que entidades maliciosas construyeran políticas abusivas, buscando obtener información sensible de otros. En este caso, liberar políticas y credenciales de forma gradual no protege la privacidad del usuario, dado que gradualmente o no se termina por revelar información sensible de más a la otra parte. Una forma de evitar esto es permitir que se invierta el rol, de forma que el cliente pueda pedir credenciales al servidor para evitar abusos. De esta manera, antes de que el cliente proporcione credenciales que puedan comprometer su privacidad, éste puede solicitar credenciales al servidor para asegurarse de que se las proporciona a quien debe. Por ejemplo, un cliente que trata de acceder a un sistema de banca online primero se autentica. Posteriormente el servidor le solicita una credencial que demuestre la titularidad de ciertos valores bursátiles, así como los límites que tiene definidos para sus operaciones. El cliente para defenderse de un posible abuso, solicita al servidor credenciales que garanticen su pertenencia al sistema bancario y otras que le identifiquen como capaz de operar en un determinado parqué. La conclusión final es que **toda negociación debe estar gobernada por un motor de decisión**, que vele por los intereses del usuario o servicio, evitando abusos.

Respecto al rendimiento, el lector puede considerar que repetir el proceso de negociación para cada interacción es costoso. La idea es que la negociación ayude en aquellos casos en los que no exista confianza previa, pero ¿qué ocurre cuando ya se ha negociado una vez?, ¿es necesario reproducir la negociación siempre que se interactúe con esa entidad? Una característica interesante, que puede ayudar a mejorar el rendimiento de estas negociaciones, se describe en [61] y en [60]. Se trata del denominado en inglés **"trust ticket"**. La intención de este ticket es la actuar como credencial que indica el nivel alcanzado en una negociación de forma que se pueda reutilizar un estado de negociación alcanzado previamente.

Este ticket puede ser utilizado, por ejemplo, para habilitar el roaming en redes desconocidas. La primera vez que se usa la red desconocida, el cliente llega a un acuerdo con el proveedor en cuanto a coste y calidad de servicio; se le permite utilizar la red; se habilitan ciertos servicios. Posteriormente, si el cliente vuelve a interactuar con el mismo proveedor, puede utilizar el ticket para agilizar el proceso, ya que tras la primera negociación, dejaron de ser desconocidos el uno para el otro. Además, otras entidades pueden valorar el hecho de que un desconocido haya llegado con anterioridad a un acuerdo con otra entidad que si conocen, de forma que se acelere el acceso.

En [61], el ticket se define como una hoja XML firmada por la parte que dispone del recurso. El ticket en sí, no garantiza el acceso al recurso, pero indica que dos partes

ya han negociado satisfactoriamente con anterioridad, simplificando las futuras negociaciones. En [60], el ticket es una hoja SAML firmada por una autoridad federada con facultades de emitir credenciales de autenticación y autorización, por lo que este tipo de negociaciones puede utilizarse para flexibilizar el acceso en esquemas de Single Sign On (**SSO**): se pueden acceder a recursos que requieran un estado de seguridad similar en todo un dominio, sin negociar repetidamente el estado de seguridad. Estos tickets no dejan de ser similares a otros, como SAML, ACs o credenciales KeyNote, con la salvedad de que no son emitidos sólo a entidades dentro del dominio administrativo, sino que pueden ser emitidas a extraños tras una negociación exitosa.

Los sistemas basados en negociación tienen otra característica interesante: la posibilidad de establecer calidad de servicio en autenticación y autorización. En [64] se explica cómo varias credenciales provenientes de diferentes autoridades puede ser requeridas, por un sistema de control de acceso, para garantizar la seguridad. Así se minimizan las probabilidades de exponer información de alto valor al perder o comprometer una credencial. Es decir, si una credencial por si sola no permite más que acceder a un conjunto acotado de información y en cambio, para acceder al resto se necesitan más credenciales, se minimiza el desastre ocasionado por la perdida de una credencial.

Dejando a un lado el riesgo, el hecho de utilizar varias credenciales para negociar un estado de seguridad de forma monótona creciente, permite establecer calidad de servicio asociada a la autenticación. Si un cliente necesita más calidad o más cantidad de servicios debe negociar para habilitarlos.

2.3. Protocolos de seguridad

En esta sección vamos a analizar los protocolos de autenticación y autorización más destacados. Distinguiremos aquellos que se utilizan para el acceso a la red de aquellos que se utilizan para el acceso a los servicios.

Protocolos de autenticación y autorización de red

La autenticación de usuario se realizada en multiples niveles dependiendo del escenario y los protocolos usados. 802.11b/g [65], realiza la autenticación a nivel físico (PHY). Como ventaja tiene que no es necesario realizar cambios en el nivel MAC o en protocolos superiores como TCP/IP. Todo está soportado en el firmware de los dispositivos involucrados, tanto tarjetas como los puntos de acceso gestionan todo. Como contrapartida, requiere que el firmware sea actualizado para soportar nuevos mecanismos de autenticación o corregir errores, dado que la implementación depende de cada fabricante. Las actualizaciones y la distribución de parches de seguridad se hace

tediosa. Son soluciones complejas de integrar con sistemas de autenticación, autorización y accounting (AAA).

En 802.11 el cliente busca un punto de acceso activo enviando tramas de gestión *probe request* en cada canal y recibe un *probe response* por cada uno de los puntos de acceso que se adecúa a la descripción de la trama enviada por el cliente. Una vez el cliente ha encontrado un punto de acceso adecuado debe autenticarse. En 802.11 se soportan dos mecanismos de autenticación, uno de autenticación abierta protegido o no por WEP [66] y otro de autenticación mediante secreto compartido protegido por WEP. En el primero, la autenticación es un algoritmo nulo, por lo que el punto de acceso aceptará a cualquier cliente. Esto puede parecer ridículo, pero en el momento de desarrollo de 802.11 se vio necesario dar soporte a dispositivos limitados como, por ejemplo, lectores de códigos de barras, que no disponían de capacidades de cómputo necesarias para cifrados complejos. Este mecanismo de autenticación sin cifrado WEP, permite que cualquier dispositivo acceda a la red. Si por el contrario se utiliza una clave WEP, esta actúa como control de acceso ya que sin ella el dispositivo no sería capaz de transmitir datos ni recibirlos. El otro mecanismo de autenticación soportado utiliza una clave para cifrar un reto transmitido por el punto de acceso, típicamente se usa una clave WEP local.

Otro mecanismo que, sin autenticar, actúa como control de acceso en 802.11, es el filtrado por dirección MAC. Este mecanismo es eficaz en la práctica, pero escala muy mal en redes grandes y además no garantiza la seguridad, dado que existe hardware que permite el cambio de la MAC.

WPA (WiFi Protected Access) [67] se diseñó para cubrir las deficiencias de WEP. En realidad es el estándar 801.11i [68] el que las corrige, pero WPA se implementó como transición a la espera de la finalización de 802.11i. WPA soporta varios mecanismos de seguridad, y aunque fue diseñado para utilizar un servidor de autenticación RADIUS, que se encargase de distribuir claves diferentes a cada usuario utilizando el protocolo 802.1x [69], también es posible utilizarlo, de forma menos segura, con claves compartidas (PSK Pre Shared Key).

Típicamente en WiFi se ha prescindido de una autenticación de usuario o equipo utilizando una autenticación abierta. La autenticación de usuario en estas redes se logra a través del cifrado de datos en el nivel enlace, de forma que aquellos usuarios que no disponen de la clave secreta, no pueden tener acceso a la red. Obviamente, en este caso no es posible distinguir usuarios salvo por MAC u otros mecanismos que pueden ser falseados. En otros casos, la autenticación en Wifi a nivel de enlace es abierta y el punto de control de acceso hace de la pasarela, de forma que, cuando un usuario trata de acceder a la web, tiene que autenticarse previamente utilizando un navegador HTTP. Este último caso asume que la conexión se utilizará para navegar, por lo que las aplicaciones, que de forma automática, descargan el correo o en general utilizan

cualquier otro protocolo diferente a HTTP, no tienen posibilidades de autenticarse.

Para soporte de autenticación sobre nivel de enlace (MAC), tenemos estándares como PPP [70] y 802.1x [69], WIMAX (802.16) [26]. Como ventaja, estos protocolos son menos dependientes del hardware, aunque siguen siendo algo dependientes. Se relaja la dependencia con el firmware haciendo más sencilla la actualización y corrección de los fallos. Permiten integrar el acceso con servidores AAA, estando el acceso a la red sujeto a una autenticación/autorización satisfactoria por parte de los servidores. Aquellos que utilizan EAP [5] permiten su extensión a otros mecanismos de autenticación/autorización. Como hemos dicho, la dependencia con el firmware se relaja, pero sigue existiendo una dependencia, que se hace mayor en los casos en los que estos protocolos se implementan en el hardware, en lugar de en el driver.

EAP (Extensible Authentication Protocol) es un protocolo de autenticación extensible que se utiliza frecuentemente en redes inalámbricas y conexiones punto a punto. Su característica principal es definir un framework que permite la negociación y utilización de múltiples mecanismos de autenticación llamados métodos EAP. EAP se utiliza sobre nivel de enlace con PPP o 802.11 y, por tanto, no requiere protocolo IP. Esta característica era muy deseable hace años, cuando otros protocolos diferentes de IP eran utilizados en las redes locales. En la actualidad IP se utilizada en todos los entornos y, por esa razón, no se puede considerar una ventaja específica el soporte de otros protocolos. Sí se puede considerar una ventaja específica el hecho de poder redirigir la autenticación a un servidor o infraestructura de AAA (Fig. 2.7), por ejemplo, RADIUS o DIAMETER [71]. EAP sólo permite transmitir un paquete al mismo tiempo y soporta retransmisiones, pero no soportar recepción fuera de orden. La fragmentación y la posterior unión de los fragmentos la debe proporcionar el mecanismo de EAP, ya que EAP por si mismo no proporciona esa funcionalidad.

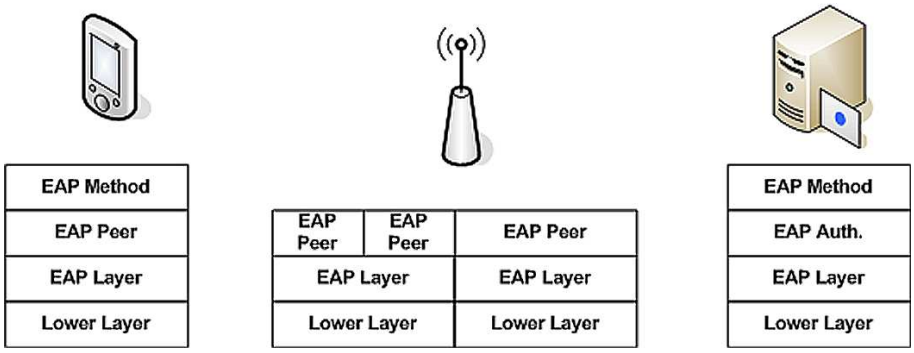


Figura 2.7: EAP en modo Pass-Through. Redirige los paquetes EAP a un servidor AAA.

802.1x es un protocolo que utiliza EAP al igual que PPP. PPP es un protocolo de tunneling de nivel 2 que evolucionó, permitiendo el uso de EAP, para poder definir nuevos

métodos de autenticación además del par usuario-contraseña. PPP actúa como portador de paquetes EAP, permitiendo redirigir la autenticación, via EAP, a un servidor externo. 802.1x actúa al igual que PPP, es decir, como vehículo para EAP, sobre una red LAN convencional o inalámbrica, proporcionando únicamente autenticación y no configuración de la conexión. 802.1x es un protocolo de encapsulación de EAP sobre LAN (EAPOL). Por esa razón, cuando no se negocian protocolos diferentes a TCP/IP, PPP no es necesario. En 802.1x se consideran tres entidades como muestra la figura 2.8: la que accede a la red o *supplicant* (cliente), el servidor que hace la autenticación y el dispositivo entre el servidor y el cliente, que es el autenticador (punto de acceso). Es posible que el punto de acceso y el servidor de autenticación sean la misma entidad. La característica fundamental de 802.1x es que simplifica mucho el protocolo de cara al autenticador (punto de acceso), ya que éste simplemente debe redirijir los paquetes de EAP al servidor. De esta forma se simplifica el hardware y se reduce el coste.

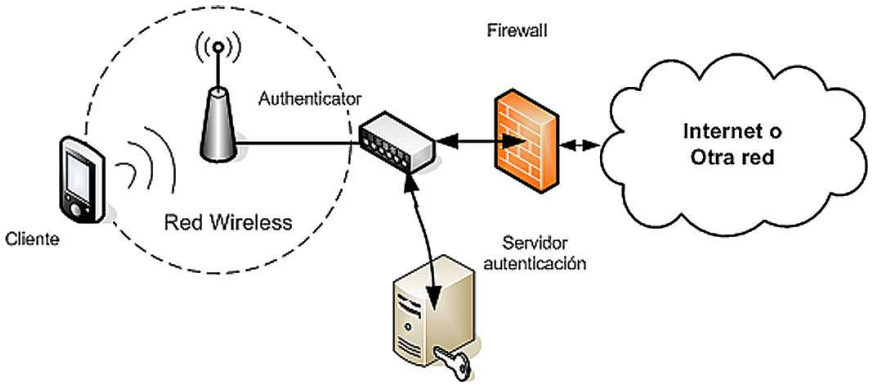


Figura 2.8: Componentes en un sistema de autenticación basado en 802.1x para una red LAN inalámbrica.

Existen mecanismos de autenticación sobre IP. Típicamente se implementan mecanismos de redirección de protocolos como ICMP. Así se redirigen las peticiones de protocolos superiores a, por ejemplo, un servidor de autenticación que presenta un interfaz web. Este tipo de autenticación está muy presente en los hoteles, aeropuertos... Pese a que no presenta ninguna mejora tecnológica, merece mención dado su despliegue. La desventaja de este sistema es clara, está orientado a la navegación web, por lo que no funciona para todas las aplicaciones.

Otros mecanismos de autenticación se utilizan sobre IP, como IP móvil, tanto en su versión para IP versión 4 [72] como IP versión 6 [73]. En este caso no se trata de un mecanismo de autenticación frente al proveedor, sino un mecanismo de autenticación que permite salvaguardar la seguridad, en el caso de IPv4 para evitar spoofing; y para evitar routing triangular, optimizando rutas, en IPv6.

PPP se hizo popular en su día por que permitía utilizar otros protocolos además de IP, por eso, además de permitir la autenticación, se involucraba en la configuración de la conexión. En la actualidad, la mayoría de esos protocolos se han extinguido e IP se puede considerar el factor común de todos las comunicaciones. En esta dirección, existe un esfuerzo importante de la IETF que actúa como protocolo de tunneling de EAP sobre capas superiores a IP, de forma que es posible utilizarlo con todas las tecnologías de conexión. Este protocolo es **PANA** (Protocol for carrying Authentication for Network Access) [74], que describiremos en detalle a continuación dado que se encuentra en desarrollo y es probablemente menos conocido para el lector.

El protocolo PANA

PANA surgió por la necesidad de proporciona un mecanismo para que los dispositivos IP se autentiquen antes de ser autorizados a utilizar la red. PANA permite utilizar varios mecanismos de autenticación, selección dinámica de proveedor de servicio y roaming. PANA reorganiza la pila de protocolos para evitar que se introduzcan capas adicional entre, por ejemplo, la capa de enlace y la capa de red; que se cargue a otros protocolos con tareas para las que no fueron diseñados, como los flags *Registration Required* de Mobile IPv4; que existan redirecciones para login basado en web. PANA es un protocolo que corre sobre UDP y permite también redirigir la autenticación a servidores de AAA utilizando protocolos estándares que abstraigan la estructura de AAA subyacente. Así no es necesario recurrir a modificaciones en los niveles de enlace, siendo válido PANA, tanto para conexiones punto a punto como para medios compartidos.

PANA no define nuevos protocolos para autenticación ni distribución de claves, sino que se apoya en los ya conocidos y de probada solvencia. PANA usa directamente EAP, que permite la ejecución de varios métodos de autenticación, salvo por una serie de extensiones que debe soportar EAP para que pueda ser usado en todas las situaciones.

PANA distingue varias entidades participantes en el proceso de autenticación y autorización y propone algunos términos y definiciones que facilitan la descripción del protocolo:

- PANA Client (PaC): Es la parte cliente del protocolo residente en el dispositivo que accede a la red. Este dispositivo puede ser una PDA, teléfono móvil, ordenador portatil... Respecto a EAP, el PaC está localizado conjuntamente con el EAP Peer.

- Agente de autenticación de PANA (PAA): Es la entidad cuya responsabilidad es la de verificar las credenciales proporcionadas por el PaC y autorizar el acceso a

la red. Respecto a EAP, esta entidad es el EAP authenticator y puede instanciar también el rol de servidor EAP cuando las credenciales se verifiquen finalmente en el PAA. En otros casos, la autenticación es redirigida a un servidor de la red que no es el PAA. El PAA debe estar siempre a un salto de distancia del PaC.

- Sesión PANA: una sesión de PANA se establece entre el PaC y el PAA y tiene una duración establecida. La terminación de esta sesión, durante la cual el PaC tiene acceso a la red, se produce por la no renovación de la sesión o por una terminación explícita de la misma, iniciada por el PaC o el PAA. Esta sesión no puede ser compartida por varios interfaces de red. Tiene varios parámetros, a destacar:
  - Tiempo de vida: es la duración asociada a la sesión de PANA. Está relacionada con el tiempo de vida de la autorización proporcionada para el acceso a la red.
  - Identificador: Este identificador de sesión identifica unívocamente una sesión entre el PaC y el PAA.
- Asociación de seguridad de PANA: se forma una asociación de seguridad entre el PaC y el PAA intercambiando claves y un contexto asociado. Se utiliza para proteger la señalización entre el PaC y el PAA.
- Punto de aplicación (Enforcement Point o EP): es el nodo de la red cuya tarea consiste en aplicar las políticas y filtros por paquete necesarios para evitar que clientes no autorizados penetren en la red, o que salga tráfico hacia clientes no autorizados.
- Clave maestra de sesión (MSK): es la clave obtenida por el método EAP utilizado entre ambas partes.

PANA es un protocolo que consta de 3 fases bien diferenciadas:

- **Autenticación y autorización**: En esta etapa del protocolo, se inicia una nueva sesión entre el PaC y el PAA y se ejecuta un intercambio EAP entre las dos partes. El resultado de la autenticación y autorización se transmite al PaC una vez finalizado el método EAP.
- **Fase de acceso a la red**: Esta fase tiene lugar tras una autenticación y autorización exitosa. Durante esta fase, el dispositivo obtiene acceso a la red, de forma que puede enviar y recibir tráfico a través del EP. Es posible que durante esta fase tanto el PaC como el PAA envíen mensajes para comprobar que la sesión sigue activa.

- **Reautenticación**: Durante la fase de acceso es posible que el PAA solicite al PaC una reautenticación de modo que se pueda reactivar la sesión antes de que expire su tiempo de vida. Durante esta fase se vuelve a realizar un intercambio de paquetes EAP sobre PANA.

- **Terminación**: Tanto el PaC como el PAA pueden terminar el servicio en cualquier momento enviando un mensaje de desconexión explícito. Si el envío del mensaje no se produce debido a una interrupción en la conexión, o por algún tipo de fallo en el sistema (baterías, reinicio...) la sesión termina cuando se cumple su tiempo de vida.

Tras el intercambio de mensajes EAP se genera una clave compartida entre el PaC y PAA que se utiliza para crear una asociación de seguridad de PANA (PANA SA). Esa asociación de PANA permite generar códigos de autenticación de mensaje (MAC) de forma que se protege la integridad y la autenticación de la señalización de PANA.

La figura 2.9 muestra un ejemplo de ejecución del protocolo PANA. El protocolo puede ser iniciado tanto por el PAA como por el PaC. En la figura, la ejecución la comienza el PaC. Si el PaC no tiene preconfigurada la dirección del PAA se puede utilizar el protocolo DHCP con PANA [75], u otros métodos alternativos que no están cubiertos por PANA. Si el PAA conoce la dirección IP del PaC puede iniciarlo también, pero los métodos de descubrimiento de PaC no están cubiertos por PANA.

En el primer mensaje de tipo PANA-Auth-Request enviado por el PAA se informa del identificador de sesión, que deberá ser usado en los siguientes intercambios de mensajes entre el PaC y el PAA. Una vez el PaC recibe el primer mensaje del PAA con el identificador, éste responde con un mensaje PANA-Auth-Answer si desea continuar con la negociación. Esto nos lleva a un posible ataque de DoS contra el PAA, dado que es posible solicitar el inicio de varias sesiones por parte de un conjunto de PaCs maliciosos, de forma que, obligando al PAA a almacenar el estado de la sesiones, se consuman los recursos del PAA. Es por ello que el protocolo limita la tasa a la que el PAA procesa los mensajes de inicio de los PaCs. Para mitigar aún más posibles ataques de DoS, la primera respuesta por parte del PAA no debe contener ninguna EAP-Payload, forzando al PaC a volver a enviar el mensaje de inicio hasta que el PAA pueda responderle (el PAA procesa mensajes con una tasa fija). Esta respuesta sin estado ayuda a prevenir ataques. Por otro lado no hay detalles en la especificación sobre la tasa con la que el PAA debe atender a los mensajes del PaC. En este primer mensaje PANA-Auth-Request, se envía también un Nonce, que no deberá volver a ser usado salvo en reautenticaciones durante la fase de acceso.

Como contenido de los mensajes PANA contempla varias AVPs entre las que tenemos:

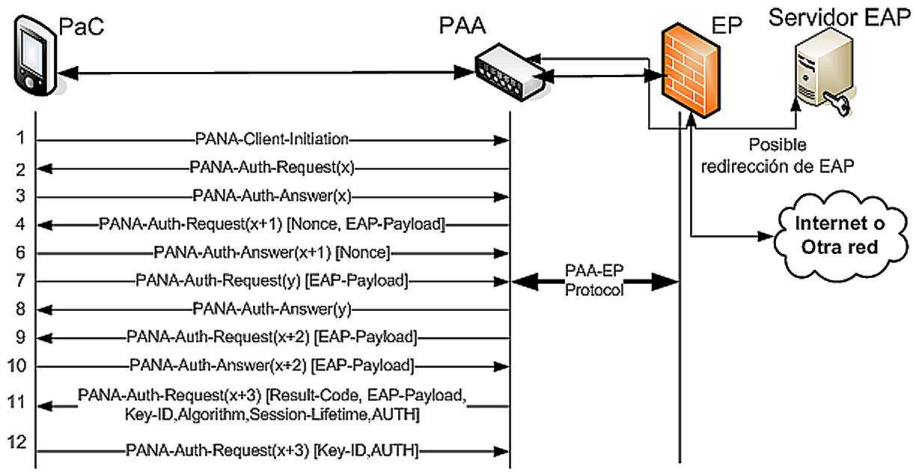


Figura 2.9: Intercambio de mensajes en PANA (ejemplo de ejecución del protocolo).

- **Algorithm**: Se utiliza para comunicar la función pseudo aleatoria que se utilizará para generar la clave PANA_AUTH_KEY, así como el algoritmo a utilizar para generar las APVs de tipo AUTH, que protegen algunos mensajes, proporcionando integridad y autenticación de mensaje. Contiene un campo cuyos primeros 16 bits indican el Transform ID, de tipo Transform Type 2 de IKEv2 [76], que corresponde a una función para obtener una clave. Los subsiguientes 16 bits contienen el algoritmo de integridad. PANA debe soportar, al menos, PRF_HMAC_SHA1 para la clave así como AUTH_HMAC_SHA1_160 para la integridad de mensaje.

- **AUTH**: Contiene el código de autenticación de mensaje para proteger la integridad del mismo.

- **EAP-Payload**: Encapsula el mensaje EAP intercambiado entre el EAP Peer (PaC) y el EAP Authenticator (PAA).

- **Key-Id**: Contiene un identificador para la clave master (Master Session Key, MSK) generada por el método EAP.

- **Nonce**: Contiene un número aleatorio utilizado para el calculo de la clave. Su longitud se determina en base al algoritmo de generación de clave, aunque se recomienda usar el máximo posible, que corresponde a 20 octetos.

- **Result-Code**: Código que indica el resultado de la operación indicando éxito, fallo en la autenticación o fallo en la autorización.

- **Session-Lifetime**: Segundos que restan hasta la finalización de la sesión.

- **Termination-Cause**: Causa de la terminación, que se indica en el mensaje de terminación.

Como se puede ver en el ejemplo de la figura 2.9, la AVP de tipo Algorithm no se envía hasta el mensaje número 11. Esto se debe a que el canal es no confiable: si la capa inferior dispone de cifrado por IPSec o por paquete de nivel 2, la AVP Algorithm puede enviarse sin problemas; en otro caso, para prevenir ataques de man-in-the-middle, la AVP Algorithm se envía una vez se ha ejecutado correctamente el método EAP y se dispone de una MSK para generar una AVP AUTH que proteja el mensaje. Una vez se ha generado una clave, PANA permite su uso para proteger la señalización con cifrado de nivel de enlace o IPsec [77].

Como hemos comentado con anterioridad, un mecanismo de autenticación y autorización mide parte de su éxito en función del número de round-trips necesarios para el establecimiento de la sesión, sobre todo en tecnologías de acceso a la red. PANA dispone de dos modos de funcionamiento, en uno de ellos, los mensajes PANA-Auth-Answer se utilizan simplemente para verificar la correcta recepción del mensaje anterior. En otro caso, este mensaje puede incluir también un paquete EAP. La elección de un modo u otro, dependerá del tiempo que lleve generar la respuesta EAP: si requiere intervención del usuario no se debe esperar para enviar el mensaje de PANA-Auth-Answer evitando retransmisiones innecesarias.

Existen una serie de requisitos generales y de seguridad, así como recomendaciones, que PANA debe resolver todavía antes de su despliegue. Estos requisitos y recomendaciones, que se pueden consultar en [78] y [79], se resumen a continuación:

- Requisitos de autenticación.
    - La autenticación debe realizarse por dispositivo/interfaz y no por usuario, aunque varios usuarios utilicen el mismo interfaz.
    - La filosofía de PANA, en lo que se refiere a autenticación, es utilizar mecanismos EAP existentes en lugar de crear nuevos.
    - La asociación de seguridad de PANA debe utilizar mecanismos que permitan generar un canal seguro IPsec en entornos en los que el canal es compartido y no existe seguridad en la capa de nivel 2.
    - La autenticación requerida puede ser doble, entendiendo que el cliente debe autenticarse primero con el Proveedor de acceso a la red (Network Access Provider, NAP) y posteriormente con el proveedor de servicios de Internet (ISP).
    - Es necesario que se produzca una autenticación mutua PaC-PAA para garantizar la seguridad del proceso: no solo el PAA debe autenticar al PaC, sino

también el PaC al PAA, ya en caso contrario un atacante podría fingir ser un PAA.

- El mecanismo EAP debe soportar reautenticaciones.
- Si se utilizan métodos de autenticación compuestos es necesario tener en cuenta las recomendaciones contenidas en [23] para evitar ataques de man-in-the-middle.

- Requisitos de los métodos EAP: De [70] pueden extraerse una serie de requisitos que debe cumplir el método que se ejecuta sobre EAP. EAP no asume transporte fiable por eso PANA utiliza UDP. Dado que EAP no proporciona corrección de errores, requiere que la capa inferior reordene los paquetes; sin embargo, es capaz de detectar duplicados. Por otro lado, pese a que no es necesario seguridad en capas inferiores es recomendable por motivos de privacidad.

- Protocolo entre PAA y EP: Es obvio que de alguna manera el PAA debe ser capaz de modificar las reglas del EP para que no filtre los paquetes del PaC, una vez éste ha conseguido autenticarse correctamente y ha sido autorizado a utilizar la red. El grupo de trabajo de PANA está planteándose utilizar COPS, SNMP o DIAMETER para ello. Por otro lado, como la seguridad de la conexión ha de protegerse con cifrado de nivel de enlace o IPsec, es necesario que el PAA le proporcione al EP las claves de sesión necesarias y que las actualice siempre que sea necesario. Por esta razón, la comunicación entre el PAA y el EP debe protegerse.

- Requisitos del nivel de red: PANA no asume un solo interfaz de red por PaC. PANA puede funcionar sobre enlaces orientados a conexión o no, por lo que los PaCs pueden o no enviar el mensaje final de disconnect, razón por la cual PANA limita en tiempo las sesiones y las protege con cifrado, evitando el robo de servicio. PANA asume que el PAA está localizado a un salto del PaC pero no define mecanismos de descubrimiento, esto se resuelve mediante DHCP u otros mecanismos.

- Interacción con otros protocolos: PANA no gestiona la movilidad como lo hace IP móvil, por tanto, debe poder convivir con el resto de protocolos incluidos los de movilidad.

- Filtrado: El proveedor a través del EP debe proporcionar mecanismos de filtrado Ingress Filtering y Egress Filtering que eviten ataques de DoS y de suplantación. El Ingress Filtering se encarga de filtrar paquetes de cliente que tengan direcciones IP diferentes a las que deberían tener [80]. El Egress Filtering es un firewall de salida que evita el uso de ciertos servicios impidiendo que las redes sean usadas para realizar ataques hacia el exterior (más propio de redes corporativas).

Tras esta descripción de los protocolos es lógico suponer que cuanta mayor independencia presente un protocolo frente al hardware, más útil será para entornos dinámicos donde se hace uso de diferentes tecnologías de red. El hecho de utilizar diferentes tecnologías de acceso a la red, y por tanto, diferentes niveles físicos así como capas de enlace, require una solución a niveles superiores que permita la reutilización de código, credenciales... facilitando el despliegue.

Protocolos de autenticación para acceso a servicios

Existen varios protocolos involucrados en la autenticación de usuario durante el acceso a los servicios. IPSec [44] interviene como garantizador de la confidencialidad entre los nodos de una red proporcionando cifrado de la información entre los nodos. IPSec puede utilizarse intercambiando claves entre los nodos, tras una autenticación exitosa, o utilizando claves previamente distribuidas. IPSec utiliza IKE [76] para intercambio de claves y otros mecanismos, que no protocolos, de autenticación para comprobar la identidad de un nodo, por ejemplo, para el acceso a través de VPN. Pero IPSec no es un protocolo de autenticación per sé, aparte de estar relacionado con la red y no con la aplicación o servicio, por lo que no puede proporcionar seguridad extremo a extremo. Otro protocolo muy asociado a la idea de autenticación para el acceso a servicios es Kerberos pero no proporciona autenticación por si mismo, sino que comunica autorizaciones generadas por otros servicios como SAML. Además, Kerberos necesita que las aplicaciones que lo usan hayan sido diseñadas teniendo en consideración el protocolo.

Existen otros protocolos que permiten la autenticación en entornos distribuidos pero hablando de acceso a servicios extremo a extremo, el único protocolo universalmente aceptado es TLS (Transport Layer Security) [52, 53].

El protocolo TLS proporciona confidencialidad extremo a extremo con autenticación para el acceso a servicio. TLS permite la negociación segura de mecanismos de generación e intercambio de claves así como de autenticación. TLS fue desarrollado por la IETF como una nueva versión de SSL, que fue originalmente desarrollado por Netscape Communications. Los cambios introducidos en TLS respecto a SSL son mínimos. SSL se hizo tan importante para el despliegue del comercio electrónico y el acceso a servicios en Internet, que la IETF desarrolló TLS para evitar la dependencia con una empresa privada como era Netscape.

TLS funciona sobre una pila de protocolos TCP/IP y es, por tanto, orientado a conexión, pero dispone de una versión no orientada a conexión que funciona sobre UDP [81] y otra que permite su uso en protocolos de transporte con múltiples flujos como SCTP. Servicios bien conocidos como HTTP y FTP utilizan TLS. TLS dispone de un me-

canismo de extensión para el soporte de nuevas funcionalidades que se describe en los estándares [82, 52, 83].

Las extensiones son el mecanismo adecuado para añadir funcionalidad a TLS. El mecanismo de extensión se basa en añadir información extra a los mensajes de ClientHello y ServerHello y pueden ser utilizados por los servidores y los clientes para aumentar las capacidades de negociación del protocolo. Como hemos comentado, el mecanismo de extensión permite añadir funcionalidad extra al protocolo pero es compatible hacia atrás con versiones anteriores de TLS. Para ello, los servidores deben ignorar aquellas extensiones que no comprendan.

TLS define el protocolo *TLS Record Protocol*. Los mensajes de protocolos, provenientes de clientes del protocolo Record, incluyen campos para la longitud, descripción y contenido. Los mensajes en esa capa son fragmentados en bloques manejables, opcionalmente comprimidos, cifrados con un cifrador de clave simétrica y autenticados con una función MAC (Message Authentic Code). Los mensajes se cifran, autentican y comprimen en base a lo expresado en el estado de la conexión (*Connection State*) negociado durante la etapa de HandShake del protocolo. El estado de la conexión contiene, entre otros, el algoritmo usado para la función de MAC, el algoritmo de cifrado por bloques, el algoritmo de compresión y la clave maestra de sesión generada durante el handshake.

Hay cuatro clientes estándar del protocolo Record: el protocolo de handshake, que negocia todos los parámetros de seguridad; el protocolo de alerta (alert protocol), que produce mensajes de alerta y una descripción de la severidad; el protocolo *change cipher spec*, que notifica a ambas partes los cambios en el estado de la conexión; y el protocolo de aplicación, como HTTP, FTP o SMTP, que envía los datos de aplicación a través del canal seguro.

Los parámetros de seguridad de una conexión TLS se negocian durante la etapa de handshake. El intercambio de mensajes del protocolo de handshake permite intercambiar material de forma segura para generar una clave que proteja el canal. El material para generar la clave se intercambia utilizando un algoritmo asimétrico como RSA o Diffie Hellman. Una vez se genera la clave, el estado de la conexión se cambia utilizando la clave para proteger el canal. Los mensajes intercambiados durante el handshake son los siguientes (el * indica opcional):

```
Client                                                Server

ClientHello
<Extensions*>                 -------->
                                                  ServerHello
                                                 <Extensions*>
```

```
                                                    Certificate*
                                                ServerKeyExchange*
                                               CertificateRequest*
                                <--------          ServerHelloDone
    Certificate*
    ClientKeyExchange
    CertificateVerify*
    [ChangeCipherSpec]
    Finished                    -------->
                                                [ChangeCipherSpec]
                                <--------                Finished
    Application Data            <------->        Application Data
```

Handshake: El cliente inicia el protocolo de handshake enviando un mensaje *Client Hello* seguido de un conjunto opcional de extensiones. En ese mensaje, el cliente proporciona datos aleatorios, una marca de tiempo y un conjunto de mecanismos de cifrado y comprensión. El servidor entonces responde con un mensaje *Server Hello*, proporcionando también datos aleatorios y seleccionando entre los mecanismos de cifrado y compresión de datos proporcionados por el cliente. El mensaje *Server Hello* puede ir seguido de una serie de extensiones siempre y cuando coincidan con las enviadas por el cliente. De esta manera se puede extender la negociación de TLS sin afectar a los servidores antiguos.

El servidor puede enviar un certificado si el intercambio de claves se va a realizar utilizando certificados PKI o un mensaje de tipo *ServerKeyExchange* para casos en los que se utilicen claves Diffie-Hellman. El intercambio finaliza con el mensaje de *ServerHelloDone*.

El servidor, tras autenticarse mediante el envío del certificado, puede requerir que el cliente envíe también un certificado (autenticación mutua). Más adelante, el cliente genera una clave y la cifra con la clave privada del servidor. Esta clave, denominada pre-master, se envía en un mensaje *ClientKeyExchange* al servidor. La clave pre-master será utilizada para generar la clave simétrica final (master secret key o MSK) mediante una función seudoaleatoria (PFR) que utiliza una combinación de hashes SHA-1 y MD5. La MSK se utilizará para proteger el canal. TLS utiliza el protocolo de único mensaje *ChangeCipherSpec* para indicar el cambio en el estado de seguridad, de forma que se deja de utilizar un cifrado nulo, para utilizar la clave derivada. Para verificar que el handshake no ha sido manipulado se envía el mensaje *Finished*, que proporciona una combinación de hashes de todos los mensajes intercambiados durante el handshake, de forma que es posible comprobar si algún mensaje intercambiado durante la ne-

gociación ha sido manipulado. El mensaje *Finished* trata de minimizar el impacto de ataques de denegación de servicio (DoS).

Protocolos de autorización

No existen protocolos de autorización como tales, es decir, que manejen la autorización en capas separadas de la autenticación. Sí existen trabajos relevantes en autorización, como GreenPass [30], que permiten poca flexibilidad. En general, la información de autenticación y autorización suele coexistir, como en SAML o KeyNote, y las tareas de verificación se dejan a la red o al servicio, que comprueba los privilegios del usuario mediante una consulta a una base de datos o repositorio de donde extrae los privilegios.

Como protocolos o, más bien, extensiones a protocolos para soportar autorización, caben destacar dos esfuerzos. El primero, que data del año 1998. Se trata de un draft de la IETF escrito por S. Farrel [84], que define una modificación a TLS para incluir nuevos mensajes de protocolo que permitan enviar credenciales de autorización, en concreto, certificados de atributos. El incluir certificados de atributos, tenía su motivación dado que TLS está muy vinculado a PKI. El problema fundamental de esta propuesta, que no llego a considerarse como RFC, es que estos nuevos mensajes no son negociados previamente mediante extensiones, lo que rompe la compatibilidad hacia atrás con los servidores estándar.

Por otro lado, en [85], M. Brown and R. Housely, definen una modificación que persigue la misma idea pero con un enfoque distinto. El trabajo propone utilizar extensiones para negociar los mecanismos de autorización disponibles, sin enviar las credenciales en si, dado que el canal durante el proceso de handshake no está protegido y puede revelar información sensible. Más adelante, una vez los mecanismos de autorización han sido negociados y ha finalizado el handshake, generándose una clave en ambos extremos, se realiza un nuevo handshake donde las extensiones de los mensajes de Hello sí contienen credenciales. Esta modificación requiere un doble intercambio de mensajes de handshake, por lo que se aumenta la latencia del acceso a los servicios, sin garantía de autorización satisfactoria, por lo que es muy sensible a ataques de DoS. Por otro lado solo soporta certificados de atributos y SAML.

Por otro lado, dejan las tareas de verificación y gestión de la información de autorización al protocolo de handshake, que no fue diseñando con ese fin. Además, y esto es lo más importante, la interacción con el servicio es muy limitada:

- Solo se pueden enviar credenciales que estén vinculadas al certificado de usuario, por lo que el servidor deberá solicitar siempre autenticación mutua y el clien-

te solo podrá hacer uso de las credenciales de autorización vinculadas a dicho certificado.

- La negociación se limita a una ronda siendo siempre iniciada por el cliente y sobre la cual el servidor solo puede decidir si es válida o no. Si fuera necesario proporcionar credenciales vinculadas a otro certificado sería necesario volver a hacer un handshake. Por otro lado, el cliente no puede solicitar al servidor credenciales, simplemente dispone del certificado enviado por el servidor y del cual solo puede extraer la identidad (no información de autorización).

- Este enfoque sería ingenuo respecto a la negociación de confianza, dado que, admitida la validez del certificado del servidor, el cliente proporciona todas las credenciales que tiene o que está dispuesto a usar, para que el servidor seleccione entre ellas.

- No queda claro en que posición de riesgo quedaría un usuario si el servidor utiliza credenciales tipo Diffie-Hellman para el intercambio de claves, dado que no puede comprobar la identidad del servidor.

- La expresividad de la extensión es muy limitada para SAML dado que no permite especificar el *protocol binding* ni el *assertion consumer endpoint*. El *protocol binding* es necesario para conocer la estructura de los mensajes de protocolo, ya que existen multitud de bindings. La propuesta sólo permite proporcionar una URL de donde el servidor debe descargar la hoja SAML. Por otro lado, SAML se diseño para que una vez el cliente se haya autenticado con el proveedor de identidad (IDP), sea el IDP el encargado de enviar el resultado al servicio. Por lo que es necesario especificar dónde (*assertion consumer*) debe el proveedor de identidad de SAML enviar la información solicitada.

2.4. Procesado de la información y representación al usuario

Como parte de la protección que todo equipo de usuario debe proporcionar a su dueño o que cada servicio debe proporcionar al sistema donde se aloja, debemos incluir, como ya hemos discutido, mecanismos para la evaluación del riesgo. Para suavizar el riesgo debe procesarse un conjunto amplio de información de las entidades involucradas y del contexto. De esta manera se posibilita la selección de servicios, redes y entidades con las que interactuar, considerando toda la información disponible, tanto de las entidades por separado, como del contexto. Así se elegirá siempre lo más

adecuado en cada momento como forma proactiva de defenderse ante ciertos ataques y para maximizar el rendimiento de los recursos radio.

Un sistema así permite valorar el riesgo de las decisiones de seguridad de forma comparativa respecto a los recursos involucrados. Entre las decisiones cuyo riesgo debe ser valorado, se encontrarían aquellas que afectan a la transmisión o aceptación de credenciales, la transmisión de políticas de seguridad que expresen requisitos y la oferta de recursos al exterior.

Para obtener los resultados deseados necesitaremos técnicas de procesado de datos que nos permitan establecer similitudes y diferencias entre elementos para poder elegir adecuadamente. La idea es agrupar los recursos, credenciales y políticas de manera que, mediante comparación, se pueda valorar el riesgo de diversas acciones.

Dentro de los múltiples algoritmos pertenecientes a la estadística multivariable encontramos Multidimensional Scaling (MDS); análisis factorial, dentro del que se enmarcaría el análisis de componentes principales (PCA); y análisis de clusters. De entre todos, el más atractivo es MDS dado que, como veremos más adelante, no solo permite analizar los datos manteniendo las diferencias conceptuales, sino que, al estar orientado a la visualización de problemas complejos, cumple el propósito de representar el espacio de decisión y los riesgos al usuario.

Multidimensional Scaling [86] (MDS), es un conjunto de técnicas ampliamente utilizado en ciencias de comportamiento, sicología así como en econometría y en otras disciplinas para analizar similitudes entre entidades.

A partir de una matriz de (di)similitudes entre pares, típicamente distancias euclídeas m-dimensionales [87], MDS puede ser utilizado para representar fielmente relaciones entre datos proporcionando una representación geométrica de dichas relaciones. MDS se utiliza para reducir las dimensionalidad de un problema a un valor más reducido o manejable. Puede considerar cualquier tipo de evaluación de disimilitudes además de distancias euclídeas: las disimilitudes pueden ser calculadas para datos cualitativos o cuantitativos, según la naturaleza de los atributos utilizados para su cálculo. Por otra parte, se pueden aplicar pesos a esos atributos, así, asignado distintos pesos a los atributos (MDS ponderado), pueden obtenerse resultados particularizados para diferentes problemas. De esta forma, un problema complejo m-dimensional puede ser simplificado preservando la información esencial y la percepción.

Existen multitud de variantes de MDS con diferentes funciones de coste y algoritmos de optimización. El primer MDS, que data del año 1930, se utilizaba para análisis de datos métricos. Más tarde fue generalizado para analizar datos no métricos [88].

En el algoritmo clásico, las proximidades (así se denominaba a las similitudes) se trataban como distancias, sin embargo, cualquier medida de disimilitud podía derivarse de los atributos para obtener una métrica, siempre y cuando se mantuviesen, la

no degeneración (las diagonales de la matriz a cero $d_{i,i} = 0$) y la desigualdad triangular para todos los elementos ($d_{i,j} + d_{i,k} \geq d_{j,k}$ para todo i, j, k). Dadas esas restricciones, la distancia entre dos puntos i y j en un espacio euclídeo m-dimensional se define de la siguiente manera:

$$d_{i,j} = [\sum_{a=1}^{m} (x_{i,a} - x_{j,a})^2]^{\frac{1}{2}} \tag{2.1}$$

Para distancias euclídeas, las distancias $d_{i,j}$ se relacionan con las proximidades observadas $p_{i,j}$ mediante una transformación apropiada $d_{i,j} = f(p_{i,j})$, que dependerá de las características de la medida. Una transformación lineal, $d_{i,j} = a + bp_{i,j}$, con $b < 0$ para similitudes y $b > 0$ para disimilitudes.

Si la solución se obtiene utilizando mínimos cuadrados, una transformación lineal de las proximidades $I(P)$, se puede definir como $I(P) = D + E$, con D la matrix de distancias, que es función de las coordenadas, y E el error residual. La solución obtenida es la X, tal que la suma de los cuadrados de E sea mínima. La matriz de productos escalares, B, se puede definir como $B = XX^T$ donde X es la matriz de coordenadas. El valor de B es:

$$B = -\frac{1}{2}[I - \frac{1}{n} i i^T] D^2 [I - \frac{1}{n} i i^T] \tag{2.2}$$

donde n es el número de entidades, I es una matriz identidad $n \times n$ e i un vector unidad de longitud n. Descomponiendo la matriz B en sus valores singulares, $B = VAV^T$, la matriz de coordenadas X se puede calcular como $X = VA^{\frac{1}{2}}$.

Para reducir la complejidad de un problema m-dimensional, podemos elegir $l < m$ autovalores y autovectores. Eligiendo solo los l autovalores y autovectores más grandes el problema queda simplificado a un problema l-dimensional.

Sin embargo, cuando se trata con datos ordinales, otro procedimiento ha de seguirse en lugar de la descomposición en valores singulares, ya que el objetivo es recuperar el orden de las proximidades y no las proximidades en si. Shepard en [89], dio una solución a este problema que fue más tarde refinada por Kruskal [90]. Esta solución minimiza iterativamente una medida llamada *Stress*, que permitía abordar el cálculo informático eficiente (dado que la descomposición en valores singulares es costosa). Para la implementación en dispositivos más limitados, se puede considerar un algoritmo llamado ALSCAL [91], que utiliza mínimos cuadrados combinado con (di)similitudes ponderadas y que es adecuado para un análisis métrico y no métrico. Además, el algoritmo ALSCAL funciona en ausencia de datos.

El análisis de factores (Factor Analysis) permite un análisis similar al que realiza MDS. En concreto, el análisis de factores, trata de analizar relaciones entre variables de dos formas. La primera, conocida como análisis de componentes principales, calcula la varianza total entre todas las variables y crea tantos factores como variables, por lo que no reduce el problema. La segunda trata de analizar la varianza común entre variables, de forma tal que puede reducir la dimensionalidad del problema a un conjunto reducido de factores. La diferencia radica en el tipo de datos a analizar. Mientras MDS permite cualquier tipo de datos, incluso permite realizar comparaciones entre entidades con diferente número de atributos, el análisis por factores requiere disponer de la información en forma de matrices de covarianza, cosa que es complicada a menos que se disponga de un modelo común de datos.

Por otro lado, el análisis de factores se considera una técnica R, es decir, está indicado para análisis de las correlaciones de las variables observadas, midiendo aspectos del comportamiento o propiedades de un conjunto de entidades; en cambio, MDS es una técnica Q orientada, por tanto, a analizar relaciones entre las entidades y no entre las variables observadas.

Los análisis de clusters o técnicas de agrupamiento son útiles en este caso dado que permiten agrupar recursos o entidades sobre las que seleccionar después. En MDS la similitud entre entidades se mide dependiendo de la distancia a la que se encuentran unos de otros, por lo que habría que discretizar esa medida de distancia para definir los distintos grupos. Agrupar puede ayudar a comunicar riesgos al usuario, pero la solución no es estática, es decir, algo puede ser arriesgado en un momento dado, teniendo en cuenta condicionantes del contexto y en otro momento puede ser seguro. MDS permite caracterizar esa variación en el contexto y ponderarla para enfatizar ciertos atributos en un momento dado, en cambio las técnicas de clustering atienden a un concepto de clasificación más estático.

2.5. Análisis de las soluciones existentes

En la sección 2.1 se ha descrito la perspectiva ideal de los diferentes actores en un sistema de control de acceso con alta movilidad de terminales y con una disponibilidad de recursos, tanto radio como de valor añadido, altamente distribuida. Durante este repaso a las perspectivas, se han puesto de manifiesto ciertas características deseables de estos sistemas de control de acceso que no están cubiertas, en general, por los sistemas de control de acceso actuales.

El equipo de usuario debe tener un sistema de control de acceso que medie en las interacciones con clientes que tratan de acceder a servicios localizados en el terminal. Ésta es la visión tradicional del control de acceso: un sistema que protege la utilización

de un recurso propio. Pero, en la sección 2.1, discutimos acerca de la necesidad de un control de acceso que limite el uso que las aplicaciones hacen de los recursos radio, teniendo en cuenta las demandas de calidad de servicio, su ámbito de utilización y el contexto que define la situación actual.

Otra funcionalidad deseable de los sistemas de control de acceso del equipo de usuario es que dispongan de un mecanismo de selección de red, servicio o entidad con el que interactuar, que considere el contexto y las preferencias del usuario. Este sistema de selección es una defensa proactiva. Además el control de acceso debe proporcionar una medida del riesgo asociado a las decisiones de control de acceso, y debe representar al usuario la información de manera comprensible.

En la sección 2.1 se analizan las consecuencias del incremento del número de proveedores de servicio, realizando una distinción conceptual entre proveedores de acceso a la red (NAPs), proveedores de servicios de Internet (ISPs) y proveedores de servicio de valor añadido. En la sección se comentan las limitaciones de los actuales sistemas de control de acceso localizados en la red respecto a este incremento en los proveedores, fruto de combinar distintos NAPs, ISPs y proveedores de servios de valor añadido. Los sistemas de control de acceso actuales, carecen de la flexibilidad necesaria para permitir composición de servicios mediante el uso de varias redes, y carecen de expresividad suficiente para permitir relaciones de roaming bajo demanda, que serían útiles en este entorno. Los sistemas de control de acceso de la red deberían tener reglas para gestionar negociaciones, además de reglas para autorizar entidades.

Respecto a los servicios, en la sección 2.1, se discute acerca de las limitaciones de los servicios en función de su conectividad, identificando dos grandes grupos, aquellos con capacidad de conexión a Internet y aquellos que se prestan en redes sin infraestructura. Se insiste en las limitaciones de protocolos de confidencialidad salto a salto y en la necesidad de mantener autenticación extremo a extremo como garantía de seguridad. Se analizan las carencias que tienen los protocolos de autenticación para comprobar las credenciales en entornos sin conexión, haciéndose necesario proporcionar soporte para mecanismos colaborativos. Estos mecanismos colaborativos, no solo pueden ayudar en redes sin infraestructura, sino que pueden ayudar a reducir el tiempo de acceso y la incertidumbre del control de acceso aportando más información. Estos mecanismos deben permitir, que dispositivos con información relevante acerca del proceso de control de acceso, puedan comunicarla a las partes involucradas.

En la sección 2.2, se repasan las credenciales representativas en autenticación y autorización atendiendo a cómo identifican a las entidades (espacio de nombres) o a cómo incluyen la información de autorización. No se analizan todas las credenciales existentes, sino las más representativas, para dejar patente que desarrollar un nuevo sistema, adaptado por completo a las necesidades de los entornos dinámicos, sería

complicado. Además, dependiendo de la aplicación, las credenciales tradicionales son útiles, pese a defectos como pueden ser el asociar identidades únicamente a nombres o claves. La conclusión es que los sistemas deben soportar varias credenciales conjuntamente con sistemas de negociación. Los modelos jerárquicos de PKI-PMI son útiles para aplicaciones basadas en la identidad de personas físicas y jurídicas, como aquellas que dan soporte a trámites con el estado, dado que un nombre único, dentro de un espacio de nombres global, es adecuado en dichas situaciones. Por otro lado, el tamaño de estas credenciales es adecuado para su uso en tokens o tarjetas inteligentes, como las utilizadas por los miembros de la unión para la identidad de los ciudadanos. También permite su uso en tarjetas de telefonía móvil. Otros credenciales o vehículos de comunicación de resultados de autenticación y autorización, como SAML, son atractivos en situaciones en las que se requiere comunicar decisiones de control de acceso a varios sistemas diferentes.

La sección 2.2 describe los sistemas de negociación de confianza. Estos sistemas permiten la negociación de un estado de seguridad en base a credenciales de varios tipos. Las limitaciones de estos sistemas, en la actualidad, es que carecen de motores de decisión que controlen la negociación que puedan ser adaptados a dispositivos con menor capacidad de cómputo como móviles. Si se dispusiese de estos sistemas en dispositivos móviles o personales, las interacciones con otros dispositivos y servicios serían posibles a través de multitud de redes, sin necesidad de que el usuario tuviese conocimientos técnicos para configurar todas las redes y servicios. La mayoría de los enfoques descritos en esa sección proponen típicamente las reglas, describen las estrategias necesarias para su funcionamiento y describen en ocasiones nuevos lenguajes para la emisión de credenciales; pero no proponen una estructura que permita su uso con credenciales y protocolos ampliamente desplegados. Además están centrados en interacción web y asumen alta capacidad de cómputo, dado que aluden a estrategias basadas en árboles y ontologías, difícilmente adaptables a dispositivos limitados. Es cierto que para realizar parte de las tareas de procesado de la información, estos dispositivos podrían contar con ayuda de máquinas más potentes utilizando la red, pero por otro lado, es igual de cierto, que la autonomía del dispositivo es necesaria.

Otra de las cuestiones relativa a al control de acceso distribuido y que se discute en la sección 2.3, es encontrar transportes adecuados que permitan la negociación tanto para el acceso a la red como para el acceso a los servicios. Como hemos visto, los protocolos de autenticación y autorización para el acceso a la red que tienen una gran dependencia con el hardware no son adecuados para un acceso flexible a la red. Cuanto más independientes son del hardware, es decir, cuando más alto es el nivel de la pila de protocolos, donde se realiza la autenticación y más flexible es un protocolo, permitiendo extensiones, más útil resulta para los entornos de alta movilidad descritos. Es por ello que protocolos de autenticación extensibles de alto nivel, como PANA, parecen encajar en los requisitos de estos entornos más adecuadamente que otros mas

cercanos al hardware.

Por otro lado la cobertura que se proporciona a la autorización en protocolos como TLS es muy limitada, siendo un proceso clave en el sistema de control de acceso. Los trabajos descritos están centrados en un conjunto cerrado de credenciales y no presentan una estructura escalable, sino que lastran a capas estándares del protocolo con tareas para las que no fueron diseñadas. Esto nos puede llevar a problemas similares a los que tienen los protocolos de acceso a la red, que a nivel de enlace o físico, añaden artificialmente una capa de autenticación. En el capítulo 7, al final de cada propuesta relacionada con modificación de protocolos, se realiza una comparación más profunda con los trabajos similares que afectan a cada una de las propuestas.

CAPÍTULO

3 Propuesta para la mejora de mecanismos de control de acceso en terminales

Existen multitud de soluciones para el control de acceso, entre las que encontramos familias de soluciones que combinan soporte a la autenticación y autorización junto a la definición de políticas. De entre las existentes caben destacar aquellas que disponen de mecanismos de extensión [92][57], de forma que pueden soportarse tanto las credenciales actuales como las futuras.

Durante la investigación se han utilizado credenciales y políticas ya existentes. Es decir, no se han diseñado nuevos lenguajes para la descripción de políticas, ni nuevas credenciales dado que actualmente se puede considerar que existen sistemas de probada solvencia en control de acceso para distintos entornos, desde tecnologías web a redes ad-hoc. Todas estas tecnologías se muestran útiles en entornos donde el control de acceso es distribuido dado que los participantes son heterogéneos y las credenciales se fortalecen cuando se combinan con otras.

En esta sección vamos a describir las contribuciones realizadas en materia de control de acceso cubriendo las deficiencias encontradas en los terminales. Comenzaremos analizando el problema.

3.1. Descripción del problema

En la sección 2.1 se mostró la perspectiva que un equipo de usuario tiene del control de acceso. Los equipos de usuario deben protegerse desde fuera hacia dentro, que sería el caso de un cortafuegos, que protege a un equipo que ofrece servicios al exterior. Del mismo modo es necesario proteger el equipo de dentro hacia fuera, como en el caso de un gestor de conexiones basado en preferencias de usuario. El control de acceso en los equipos de usuario se debería realizar por tanto:

- De fuera a dentro: protegiendo el acceso a los recursos compartidos localizados en el terminal. Este tipo de control de acceso es muy conocido y trata de evitar accesos no autorizados requiriendo autenticación/autorización.
- De dentro a fuera: protegiendo a los recursos radio del acceso uso indebido de algunas aplicaciones. Este control de acceso debe ser capaz de:
  - lograr que las aplicaciones del terminal se conecten solo a través de las redes adecuadas a sus demandas de tráfico conectando a redes conocidas y probadas, reduciendo el tiempo de conexión.
  - utilizar el contexto para determinar las redes más atractivas permitiendo, por ejemplo que ciertas aplicaciones, solo envíen paquetes dentro de una red segura como la del trabajo.
  - garantizar un uso eficientemente económico de los recursos seleccionado redes de menor coste posible.
  - garantizar un cierto grado de protección de forma autónoma, es decir, sin intervención de la red, servicios o terceras partes que ayuden en el proceso.

Para lograr los objetivos previstos deberemos proporcionar soluciones que permitan realizar algunas tareas necesarias que se discuten en lo que resta de esta sección.

Necesidad de caracterizar entornos

Uno de nuestro objetivos es reconocer el entorno que nos rodea para poder particularizar el comportamiento del terminal. En el control de acceso, tal y como se ha definido, es necesario controlar el acceso de “dentro a fuera” tanto como de “fuera a dentro”. Como es lógico, existen entornos confiables en los que el terminal no tiene por qué restringir sus comunicaciones ni sus servicios ofrecidos, estos entornos pueden ser su casa o su oficina. Esto no quiere decir que se pueda descuidar la seguridad y la confidencialidad de las comunicaciones, sino que en dicho entorno se puede interactuar con el resto de los dispositivos conocidos y asociados a ese entorno sin restricción en cuanto al consumo y a la oferta de servicios.

Para determinar las políticas aplicables a cada entorno podemos proceder preguntando al usuario, de forma que éste califique cada entorno y decida cuales son los dispositivos confiables en cada entorno; o recurrir a mecanismos automáticos como se discutirá más adelante en esta sección. Determinar el entorno en el que nos encontramos, y caracterizarlo desde la perspectiva de la confianza así como de los servicios ofrecidos, es clave para proteger proactivamente al terminal, dado que esto permitirá establecer filtros, reglas en el firewalls etc. con un doble propósito:

- protegernos de accesos externos, aumentando la protección en los entornos menos favorables; y restringiendo el acceso a los servicios localizados en el terminal, aplicando requisitos más fuertes.
- establecer una política o políticas de selección de red o par con el que interactuar, minimizando así los riesgos de usar redes no confiables. De esta manera evitamos que las aplicaciones accedan a los recursos del terminal limitando la conectividad de ciertas aplicaciones en determinados entornos.

Necesidad de seleccionar adecuadamente las redes

Queremos permitir que los terminales puedan cambiar de red de forma espontanea, teniendo en cuenta las demandas de conectividad de las aplicaciones y no sólo por perdida de conectividad o degradación de cobertura. Además, como queremos aplicar las restricciones que nos impone el entorno y, por supuesto, utilizar cualquier tecnología de conexión actual y futura, sin que importe el modelo de datos con que describe cada tecnología. Por otro lado, debemos ser capaces de trabajar sin que sea necesario disponer de información perfecta dado que no siempre podremos recopilar el mismo número de datos.

Es necesario un mecanismo de selección universal que utilice los mismos mecanismos y criterios para encontrar la red adecuada con independencia de la tecnología de conexión. Actualmente disponemos de mecanismos adaptados a cada una de las tecnologías, lo que nos obliga a seleccionar por separado, proporcionando sólo resultados de la tecnología para la que está pensado el mecanismo. Por ejemplo, si una persona no consigue conexión WiFi, deberá cambiar sus preferencias permitiendo que sus aplicaciones utilicen GPRS o EDGE. El caso contrario, sería el de una persona que está utilizando GPRS para leer el correo, llega a su casa y tiene que desconectar GPRS manualmente y realizar una búsqueda con las herramientas de WiFi, para que, a partir de ese momento, las aplicaciones utilicen WiFi.

Si se utiliza un único mecanismo de selección, sólo hace falta considerar las demandas de tráfico actuales así como las políticas en información de contexto, para encontrar un “máximo global” y no uno local, es decir, la mejor red ya sea GPRS, Wifi, UWB, Wimax...

Necesidad de comunicar efectivamente al usuario el espacio de decisión

Existen tareas asociadas al control de acceso, a las que debe dar cobertura el equipo de usuario, como la valoración y comunicación de riesgos o del espacio de decisión

al usuario. El usuario tiene asociada **la idea de distancia o potencia de señal recibida** como el criterio más adecuado para la selección de una red, debido a que lógicamente, la potencia de señal aumenta al disminuir la distancia.

En un escenario en el que coexisten gran cantidad de tecnologías de red a disposición del usuario, será complicado que el usuario medio conozca suficientemente bien los detalles técnicos de cada una de las tecnologías como para poder elegir adecuadamente. Por esa razón, deben proporcionarse herramientas que permitan al usuario comprender el problema y para ello se debe recurrir a modelos mentales similares al del usuario medio, que en el caso de la selección de red, corresponde a la distancia (o potencia de señal recibida).

Para lograr una comprensión del problema será necesario simplificar un problema de multiples dimensiones, a uno visualizable gráficamente, que encaje con el modelo mental del usuario. Esta forma de proceder es adecuada con el principio de la navaja de Occam o principio de economía o de parsimonia, que hace referencia a un tipo de razonamiento basado en una premisa muy simple: “en igualdad de condiciones la solución más sencilla es probablemente la correcta” y a su postulado “No ha de presumirse la existencia de más cosas que las absolutamente necesarias”.

Por esta razón, trataremos de ofrecer al usuario una visión simple del problema para que pueda realizar una decisión adecuada y además simplificaremos el problema para que el terminal decida siguiendo la misma filosofía.

3.2. Arquitectura

Como se explicó en la sección anterior, las políticas son un elemento clave para el sistema. Típicamente un sistema podrá estar gobernado por varias políticas que deberán fundirse en una para su aplicación efectiva. Las políticas permitirán aplicar las restricciones del dominio estableciendo las **preferencias de selección**. Dichas preferencias de selección serán consumidas por el sistema en forma de **vector de pesos** como se explicará a lo largo del resto del capítulo. Las políticas contribuirán a dichos vectores de pesos y la demanda de tráfico contribuirá a caracterizar los **elementos ideales**. Así, se podrán tener en consideración todas las restricciones descritas en las políticas en conjunción con las limitaciones impuestas por el entorno.

La figura 3.1 muestra las relaciones entre los diferentes módulos del sistema. Las relaciones entre las diferentes entidades se describen a continuación:

- A través de los sensores o los interfaces de red, se obtiene toda la información necesaria acerca de lo que nos rodea. Ésta información define el entorno que

nos rodea en base a los dispositivos cercanos. De estos dispositivos, se marcarán aquellos que no sean móviles para reconocer el entorno en futuras ocasiones.

- Con esta información se valora el entorno, de forma autónoma o mediante protocolos colaborativos y se informa del resultado al *punto de decisión*.
- El *punto de decisión* selecciona las políticas adecuadas y las comunica al *motor de decisión*.
- El *motor de decisión* utiliza estos datos junto con los datos de tráfico demandado para seleccionar las redes adecuadas. La decisión se comunica al punto de decisión y éste la comunica a los interfaces de red para conectarse a la red elegida.
- Por otro lado, el punto de decisión informa de los filtros y restricciones que deben aplicarse al tráfico que pase por el punto de *aplicación de políticas*.

Los dispositivos limitados almacenan recursos susceptible de ser protegidos, por esta razón utilizamos un gestor de políticas, para tomar decisiones de control de acceso en base a esas políticas. Las políticas de control de acceso junto con los sensores, permiten definir un mecanismo dinámico y semiautomático de protección, de forma que las aplicaciones se adaptan al contexto y minimizan la intervención del usuario.

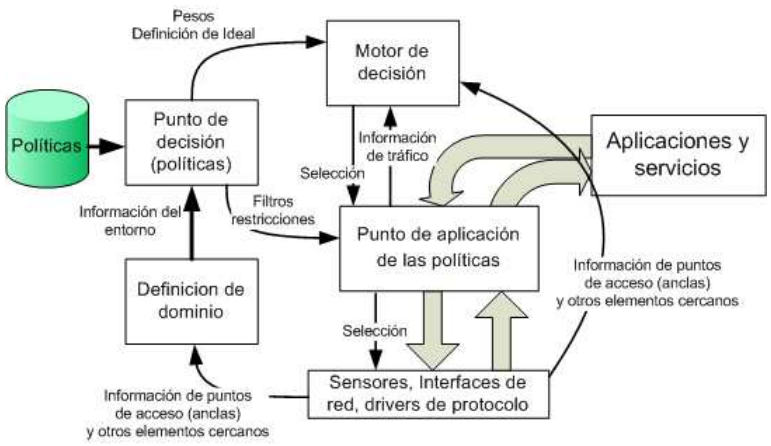


Figura 3.1: Relación entre las políticas, los puntos de aplicación de política y el motor de decisión en el motor de control de acceso propuesto.

En [93] se define un sistema genérico de control de acceso basado en confianza. En este trabajo, incluimos un sistema específico para controlar el acceso a los interfaces de red. Dicho sistema se basa en el estándar XACML [92] para definir las políticas y el intercambio de información.

XACML define una arquitectura para control de acceso en sistemas típicamente web en los que los dispositivos involucrados son PCs y servidores. XACML es un enfoque flexible que permite especificar diferentes políticas y reglas, que serán evaluadas en el punto de decisión, *Policy Decision Point* (PDP), para permitir o denegar el acceso los recursos. Las peticiones de acceso a recursos deben pasar por el punto donde se controla el acceso, *Policy Enforcement Point* (PEP). La colaboración entre el PEP y el PDP garantizan que el control de acceso a los recursos se realiza para todas y cada una de las peticiones.

Respecto al PEP, hay dos enfoques: el PEP puede estar incluido en la aplicación o las aplicaciones acceden al PEP a través de un API. No obstante, las aplicaciones no diseñadas para cooperar con un sistema de control de acceso o incluso aplicaciones maliciosas como virus, troyanos... pueden esquivar el PEP y acceder a los recursos directamente. Una posible solución, la adoptada por nuestra solución, es embeber el PEP a nivel de sistema operativo (Kernel), complicando así el acceso a los recursos a este tipo de aplicaciones. Por otro lado, nos aseguramos que las aplicaciones del fabricante, presentes en el dispositivo móvil, cumplen también con el control de acceso definido por el usuario.

Nos beneficiamos de la flexibilidad de XACML, extendiendo los atributos para incluir valores de confianza y datos de contexto externos. De esta forma, las decisiones se toman en base a la confianza asignada a las otras entidades y en función de la información de contexto disponible, como localización relativa, preferencias y coste.

3.3. Descripción y caracterización del entorno

Vamos a dar una breve definición de algunos de los términos que utilizaremos a lo largo de esta sección:

- Los **dispositivos** se agrupan en **dominios**. El conjunto de dispositivos mas cercano, que nos rodea, se considera el **dominio** actual.
- Los dispositivos dentro de un dominio se dividen en estáticos, que llamaremos **anclas** y móviles que llamaremos **pares** o iguales.

Caracterizar el entorno actual pasa por poder identificar unívocamente el entorno actual y agrupar los dispositivos en dominios para más tarde descubrir y calificar los servicios disponibles. De esta forma se dispone de la máxima información para tomar una decisión adecuada.

Satyanarayanan dijo que las interacciones en entornos de computación ubicua decaen con el cuadrado de la distancia [94]. Ésta afirmación es aplicable, en general, a

cualquier interacción, ya que la energía de las señales radioeléctricas decae de la misma manera. El logro de Satyanarayanan es establecer una medida de lo que podría llamarse *distancia de interacción*.

Pero, ¿Qué ocurre con otros atributos métricos o no métricos como confianza, coste económico, tipo de servicio o cualquier otro definido por el usuario? ¿Deben ser tenidos en cuenta cuando se busca otro par con el que interactuar? ¿Cómo podemos asistir al usuario a la hora de seleccionar la red con la menor distancia de interacción y hacerlo a su vez de forma *invisible* al usuario?

La mayor de las restricciones, como ya hemos comentado, es la distancia física. Esa es la razón de que el conjunto más cercano de dispositivos definan el dominio actual. Se utiliza esta forma de localización relativa combinada con otra información de los dispositivos cercanos para definir un dominio.

Dado un dominio, los dispositivos inalámbricos estáticos dentro de él, es decir, aquellos que por su naturaleza no son móviles, por ejemplo, puntos de acceso, impresoras y pantallas, son identificados por su dirección MAC o por otros medios, por ejemplo, criptográficos. Estos dispositivos no móviles son marcados como **anclas** o puntos de referencia. Las anclas de un dominio ayudan al dispositivo móvil a reconocer un dominio.

Para cada elemento del dominio, el módulo averigua todos los atributos que serán utilizados para calcular la distancia de interacción. Los atributos representan información que describe el elemento como tal, en el caso de un punto de acceso Wifi podemos encontrar atributos como la MAC, el BSSID o el SSID, así como su seguridad (WEP o WPA). Por otro lado, hay que modelar los elementos respecto al contexto, de forma que se pueda calibrar el efecto que tiene el contexto sobre ese elemento. Como atributos se puede incorporar información de cualquier tipo ya sea cuantitativa, cualitativa o de pertenencia a categoría.

La información a averiguar dependerá de en que atributos basa el usuario su decisión. El tipo y número de atributos son definidos por el usuario pero, como mínimo, dos deberían ser considerados: la distancia física y la confianza. Otros atributos interesantes son la información de servicios, obtenida a través de protocolos de descubrimiento [95]; credenciales requeridas; coste económico... La figura 3.2 muestra el modelo lógico de descripción de elementos.

Como el lector puede inferir, obviamente la descripción de cada entidad participante en la decisión puede variar en número y tipo de atributos, dependiendo de la tecnología de red que utilice o de las preferencias del usuario en cuanto al modelado de contexto. En un corto espacio de tiempo, varios estándares estarán disponibles para su uso comercial, entre ellos cabe destacar UWB, WiMax y LTE. Por esta razón, no se podría considerar extraño encontrarnos en un futuro con un amplio espectro de tec-

nologías y de operadores coexistiendo en un mercado de alta movilidad. Esta serie de condicionantes nos llevan a dos conclusiones importantes:

- No es posible ni lógico modelar todas las tecnologías de la misma manera, es decir, con el mismo conjunto de datos, dado que son tecnologías distintas. Por otro cada usuarios puede tener distintas preferencias de modelado.
- El modelado de datos debe ser capaz de describir un elemento de forma que se pueda soportar un roaming espontáneo, motivado más por las demandas de tráfico o calidad de las aplicaciones que por la falta de cobertura.

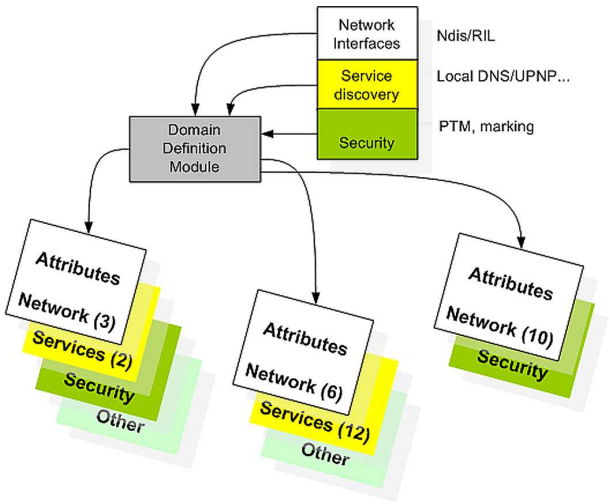


Figura 3.2: Descripción lógica de la información que modela un elemento dentro de un dominio

Respecto a la distancia física, dado que las modulaciones, frecuencias y características radio son muy variables de una tecnología a otra, esta distancia física se determina a través de las medidas de señal recibida. Se toman muestras de potencia de señal recibida para todos los puntos de acceso o anclas y posteriormente se escalan por un factor que depende de la tecnología de conexión, de esta forma se pueden obtener valores normalizados entre 0 y 1.

Las técnicas de localización basadas en la potencia de la señal recibida proporcionan un buen grado de privacidad, además de ser baratas: el mismo hardware se utiliza no solo para establecer conexiones sino para determinar la posición relativa. La exactitud de estos mecanismos es limitada y empeora en entornos cerrados como edificios. Datos precisos sobre localización por potencia de señal pueden leerse en los artículos

[96] [97]. En cualquier caso, este tipo de localización es adecuada para nuestros propósitos: reconocer dominios conocidos y determinar si nos acercamos o alejamos de los mismos.

Obviamente, los bordes del dominio, obtenidos mediante el uso de la potencia de señal recibida, son inexactos, pero en combinación con el resto de atributos, pueden proporcionar una medida útil de *distancia de interacción* suficiente para tomar decisiones.

Otro de los atributos que debería estar presente siempre a la hora de modelar un elemento es el valor de confianza. El valor de confianza es un indicador de cómo de seguro es, para el usuario o terminal (según se establezca), el uso de un elemento. Este valor puede ser establecido en función de la experiencia, por el resultado de usos anteriores; por el propio usuario; o por protocolos colaborativos. Durante la investigación se utilizó PTM (Pervasive Trust Manager) [98] para obtener el valor de confianza de pares y las anclas.

Todos los atributos se almacenan en elementos XML, que contienen al menos información para identificar un dominio (sus anclas) y un tiempo de vida.

3.4. Valoración del entorno

Como introdujimos en la sección anterior, uno de los parámetros importantes que deben incluirse en el modelado de los elementos del entorno es el valor de confianza. La determinación de este valor puede realizarse de diferentes formas, desde la valoración subjetiva del usuario, hasta mecanismos de marcado automático. Cada uno de ellos tiene sus ventajas y sus inconvenientes. Respecto a las ventajas, el marcado de los elementos permite realizar una valoración subjetiva que no conseguiría con un mecanismo de marcado automático pero, esta forma de proceder se convierte en inviable si el número de elementos a valorar es demasiado grande o variable en el tiempo como para que el usuario pueda personalmente dar un valor a todos.

El objetivo es dar marcas automáticamente a las anclas del dominio en lugar de preguntar al usuario si debe confiar en un punto de acceso o no. Para ello se consulta a otros pares, dispositivos cercanos. Cada dispositivo puede definir un dominio de forma dado que importa la distancia, pudiendo ser nuestra visión de un dominio distinta a la visión del más cercano de los dispositivos que nos rodean.

Por esa razón, cuando los dispositivos cercanos intercambian información acerca de las anclas, el que la recibe solo procesa atributos relativos a anclas que tiene en común con el que ha enviado la información. El modelo está diseñado para funcionar con cualquier tipo de información, pero en este momento solo consideraremos la

confianza.

El procesado de los valores de confianza es simple, los valores de confianza son intercambiados de forma segura entre los pares y escalados por un valor que depende de la confianza asignada por PTM al recomendador. El par i usa la información recibida por el par k para obtener un valor, $\beta_{i,j}$, que es el valor de confianza que el par i tiene en el ancla j. El par i cuantifica su confianza en otro par k con un valor entre 0 and 1, $\alpha_{i,k}$, y sólo acepta recomendaciones de pares con un valor de confianza mayor que α_{min}. El incremento del valor de confianza $\beta_{i,j}$ para la recomendación $n-$ésima se calcula mediante la siguiente expresión:

$$\Delta\beta_{i,j} = \frac{\alpha_{min}}{n\log n}(\beta_{k,j} - \beta_{i,j})\alpha_{i,k} \quad \forall \quad (\alpha_{min} < \alpha_{i,k}) \tag{3.1}$$

$$\Delta\beta_{i,j} = 0 \quad \forall \quad (\alpha_{min} > \alpha_{i,k}) \tag{3.2}$$

El la expresión anterior utiliza un factor ($\frac{\alpha_{min}}{n\log n}$) que permite un arranque rápido, es decir, permite un incremento rápido del valor de confianza para un ancla y evita ataques colaborativos ya que el factor decrece con el número de recomendaciones. Este factor puede ser modificado por el usuario para conseguir otros resultados. La figura 3.3 muestra la evolución del valor de confianza para un ancla usando un factor de $\frac{\alpha_{min}}{n\log n}$.

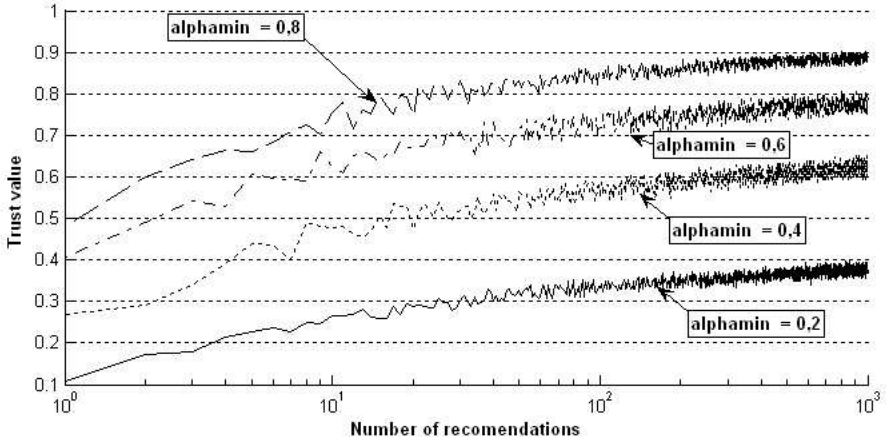


Figura 3.3: Evolución del valor de confianza para un ancla desde 0 en función del número de recomendaciones. El valor recomendado es siempre 1.0 y la expresión utilizada es de tipo conservador.

Como puede verse en la figura 3.3, los resultados están condicionados por el valor de α_{min}. Éste es un enfoque muy conservativo que suele ser empleado en sistemas de

reputación, que tienden a proteger al dispositivo frente a recomendaciones maliciosas. Cuanto más alto es α_{min}, mayor valor de confianza puede ser alcanzado, pero menor número de recomendaciones serán tenidas en cuenta ($\alpha_{i,k}$ debe ser mayor que α_{min}). Otros modelos han sido considerados: un enfoque menos conservativo puede obtenerse utilizando como factor $\frac{\alpha_{i,k}}{n \log n}$, de forma que las recomendaciones provenientes de pares muy confiables influencien mucho más el valor de confianza final.

Este mecanismo ayuda a calcular un valor de confianza de forma colaborativa permitiendo al dispositivo móvil identificar entornos como confiables o peligrosos y reaccionar en consecuencia.

3.5. Mecanismo de selección de red o par

Elección del algoritmo

Las técnicas de análisis de datos basadas en MDS, cuyas ventajas se describen en la sección 2.4, se utilizarán para encontrar una secuencia ordenada de pares o puntos de acceso a los que conectar dependiendo de las políticas aplicables, preferencias del usuario y la información de contexto.

La razón de utilizar MDS es su capacidad de simplificar problemas. Como ya se ha razonado en la sección 3.1, simplificar problemas que ayuden a acercarnos a modelos mentales o formas de razonar habituales son muy deseables.

El problema de decidir cuál es la mejor red a la que conectar o con qué par interactuar en entornos complejos, se resuelve utilizando técnicas que permiten *"simplificar los problemas como hacen los humanos"*. Dado que cada entidad involucrada en la decisión puede estar descrito con un número diferente de atributos, es necesario simplificar dicho problema, de multiples dimensiones, a algo más manejable. Al realizar una simplificación, hay que valorar qué información se esta perdiendo y cómo se adapta el resultado a la conceptualización del problema inicial. MDS ALSCAL proporciona parámetros que permiten medir la bondad del ajuste y, siempre que respeten cierto límites para esos parámetros, se puede decir que se mantiene la conceptualización del problema.

Al simplificar el problema y reducir éste a un número de variables bajo, una o dos variables, podemos calcular lo que denominaremos *distancia de interacción* para cada uno de las entidades. De esta forma podemos aproximarnos al modelo mental de distancias, logrando que dichas distancias recojan tanto las restricciones de la política, como las preferencias del usuario además de la información de contexto.

Procesado de los atributos

Consideremos un entorno con varios pares y anclas (elementos). Las (di)similitudes entre elementos se pueden calcular, dependiendo del tipo de atributos, de la siguiente manera:

- **Atributos cuantitativos**: aquellos que tienen un valor numérico contenido en un intervalo definido. El procesamiento un atributo cuantitativo α, para calcular la diferencia entre dos entidades i y j, consiste en calcular la diferencia del valor de sus atributos en relación al intervalo en el que se encuentran los posibles valores para dicho atributo:

$$\delta_{i,j,\alpha} = \frac{|u_{i,\alpha} - u_{j,\alpha}|}{max(u_\alpha) - min(u_\alpha)} \tag{3.3}$$

  donde $u_{i,\alpha}$ es el valor del atributo $\alpha-$ésimo del elemento i. Los datos cuantitativos que pueden utilizan para describir relaciones de confianza, distancias físicas [87]...

- **Atributos ordinales**: aquellos que tienen un valor que corresponde al orden que ocupan en una escala o clasificación. El procesamiento un atributo ordinal α, para calcular la diferencia entre dos entidades i y j es el siguiente:

$$\delta_{i,j,\alpha} = \frac{|rank(u_{i,\alpha}) - rank(u_{j,\alpha})|}{max(rank(u_\alpha)) - 1} \tag{3.4}$$

  donde $rank(u_{i,\alpha})$ es lugar o rango del atributo $\alpha-$ésimo del elemento i. Los datos ordinales permiten, por ejemplo, distinguir clases de QoS por orden: calidad oro, plata, bronce.

- **Atributos de categoría o clase**: aquellos que permiten clasificar una entidad como perteneciente a una clase o categoría. El procesamiento un atributo de categoría α, para calcular la diferencia entre dos entidades i y j es el siguiente:

$$\delta_{i,j,\alpha} = \begin{cases} 0 : u_{i,\alpha} = u_{j,\alpha} \\ 1 : otherwise \end{cases} \tag{3.5}$$

  donde $u_{i,\alpha}$ es el valor del atributo $\alpha-$ésimo del elemento i. Permiten, por ejemplo, clasificar elementos, de forma que distinguimos entre pares ad-hoc o puntos de acceso pertenecientes a redes con infraestructura.

- **Atributos con umbral**: Algunos atributos necesitan compararse con un umbral antes de compararse con el mismo atributo de otra entidad, por ejemplo, si se establece un coste máximo, o un ancho de banda mínimo. Dado un atributo α y dado un umbral $Umbral$, para calcula la diferencia entre dos entidades i y j, se procesa, dependiendo del tipo de umbral, de la siguiente manera:

  - Umbral mínimo:

$$\delta_{i,j,\alpha} = \begin{cases} \frac{|u_{i,\alpha}-u_{j,\alpha}|}{Umbral-min(u_\alpha)} : & si\ u_{i,\alpha} \leq Umbral\ y\ u_{j,\alpha} \leq Umbral \\ \frac{|Umbral-u_{j,\alpha}|}{Umbral-min(u_\alpha)} : & si\ u_{i,\alpha} > Umbral \\ \frac{|u_{i,\alpha}-Umbral|}{Umbral-min(u_\alpha)} : & si\ u_{j,\alpha} > Umbral \\ 0 : & en\ otro\ caso \end{cases} \tag{3.6}$$

    Si se establece como condición que un atributo α sea mayor que un umbral U, la diferencia entre dos entidades, respecto al atributo α, será 0 si el valor de sus atributos α son mayores que U. En otro caso se calcula la diferencia como se recoge en la expresión anterior.

  - Umbral máximo:

$$\delta_{i,j,\alpha} = \begin{cases} \frac{|u_{i,\alpha}-u_{j,\alpha}|}{max(u_\alpha)-min(u_\alpha)} : & si\ u_{i,\alpha} \geq Umbral\ y\ u_{j,\alpha} \geq Umbral \\ \frac{|u_{i,\alpha}-Umbral|}{max(u_\alpha)-Umbral} : & si\ u_{j,\alpha} < Umbral \\ \frac{|Umbral-u_{j,\alpha}|}{max(u_\alpha)-Umbral} : & si\ u_{i,\alpha} < Umbral \\ 0 : & en\ otro\ caso \end{cases} \tag{3.7}$$

    Si se establece como condición que un atributo α sea menor que un umbral U.

  - Umbral máximo y mínimo combinados, Umbral menor o igual, Umbral mayor o igual: En el caso de los combinados se aplican las dos expresiones anteriores. En el caso de mayor o igual, se modifican las inecuaciones.

El elemento Ideal y los pesos

Para poder elegir el punto de acceso o par más adecuado necesitamos establecer en primer lugar cuáles son las demandas de tráfico, o qué es lo que necesitan las aplicaciones. De esta manera se pueden comparar las alternativas disponibles con la demanda. Para ello se crea un **elemento ideal**, un punto de acceso ficticio que recoja dicha información.

Por otro lado, es necesario establecer las preferencias. Es decir, cómo de importante es una diferencia entre dos entidades respecto a un atributo en concreto. Así se puede

expresar, por ejemplo, preferencias del tipo: encontrar una red que sea muy segura aunque disponga de poco ancho de banda.

Por lo tanto, el resultado de la decisión estará condicionado a dos parámetros:

- El elemento **Ideal**: Este elemento recoge las demandas de las aplicaciones corriendo en el sistema. Dependiendo de los servicios a los que se requiera acceso así como de la calidad de la información de contexto, algunos atributos quedarán indefinidos por falta de información o por ser irrelevantes para el caso. El elemento Ideal, recoge además preferencias generales como costes y confianza.

  Este elemento se incluirá en el cómputo de similitudes de manera que se puedan establecer las diferencias entre lo que hay disponible y lo que se requiere.

- El **vector de pesos**: Este vector recoge la información de las políticas y las preferencias del usuario, de forma que se potencian determinados atributos frente a otros.

Procesado de los datos en el tiempo

Hasta este punto se han descrito las tareas del algoritmo correspondientes a la recopilación de datos. En la figura 3.4 pueden apreciarse todas las tareas involucradas en el algoritmo y que resumimos a continuación:

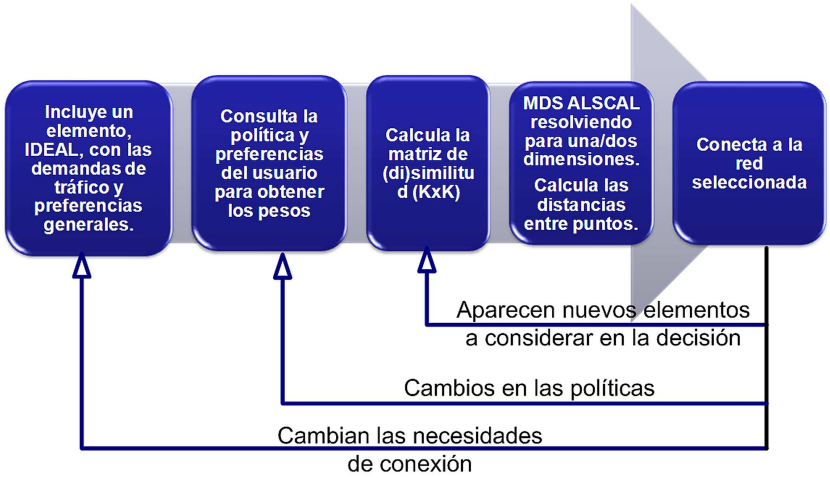


Figura 3.4: Orden en el que se ejecutan las tareas del algoritmo de selección de red.

- Establecimiento del Ideal: Se analizan las demandas de tráfico de las aplicaciones así como otras necesidades, como calidad de servicio necesaria, tipo de mo-

vilidad soportada... El elemento ideal recoge además preferencias generales, como pueden ser costes máximos, operadores favoritos,etc.

- Obtención de los pesos: Se consultan las políticas y las preferencias de usuario y se extraen los pesos.

- Calculo de la matriz de disimilitudes: Una vez las similitudes entre elementos se han calculado, se ponderan con los pesos. De esta forma se obtiene una matriz cuadrada de similitudes entre entidades particularizada para el espacio de decisión actual. La particularización se logra incluyendo la información de contexto y ponderando con los pesos asociados a dicho contexto. Estas similitudes promediadas se definen, para un conjunto de n entidades descritas con q atributos, de la siguiente manera:

$$\delta_{i,j} = \left(\frac{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha \delta_{i,j,\alpha}^{\lambda}}{\sum_{\alpha=1}^{q} w_{i,j,\alpha} w_\alpha}\right)^{\frac{1}{\lambda}} \tag{3.8}$$

  donde $w_{i,j,\alpha}$ valdrá 0 si las entidades i y j no pueden compararse en el atributo $\alpha-$ésimo y 1 en caso contrario. De esta manera se consigue comparar entidades definidas con distinto numero de atributos.

  w_α es el peso dado a dicho atributo α y $\delta_{i,j,\alpha}$ es finalmente la similitud entre las entidades i y j para el atributo $\alpha-$ésimo.

- Conectar a la red seleccionada: Una vez se resuelve el algoritmo, se puede proceder a conectar a la red seleccionada. En aquellos casos en los que se requiera redundancia, se puede conectar a varios o ejecutar varias veces el algoritmo con diferentes requisitos, para determinar a qué puntos conectar para conseguir diferentes flujos que puedan ser controlados, por ejemplo, con SCTP.

  Una vez en este estado, el mecanismo volverá a activarse si se producen determinados cambios que se expone a continuación:

  - Si aparecen nuevas entidades a considerar: Si aparecen nuevos puntos de acceso o pares a los que conectar, se vuelve a recalcular la matriz de disimilitudes, utilizando los mismos pesos y el mismo elemento ideal, por si el nuevo elemento es una mejor elección.

    Este comportamiento debe dotarse de una cierta histéresis de forma que se mitiguen posibles ataques de DoS que fuercen al dispositivo a recalcular constante la matriz.

- Si cambian las políticas debido a un cambio de dominio (el usuario se mueve), porque cambian las preferencias o, simplemente, la política se actualiza, es necesario volver a obtener los pesos y por tanto volver a ejecutar el algoritmo a partir de la tarea de obtención de pesos.
- Si cambian las necesidades de conexión los cambios deben repercutir en el elemento Ideal. Esto afecta sin duda al resultado de la decisión y por tanto debe ejecutarse de nuevo el algoritmo completo. En este caso, es también conveniente considerar cierta histéresis.

CAPÍTULO

4 Validación mediante simulación del algoritmo de selección de red

En este capítulo mostraremos los resultados de las simulaciones realizadas para validar el algoritmo. El objetivo es demostrar que el algoritmo es útil para los propósitos para los que ha sido diseñado. Plantearemos dos simulaciones, en la primera haremos una demostración de concepto utilizando un ejemplo muy simple, pero instructivo. Los vectores de pesos para la primera simulación se han exagerado de modo que el ejemplo sea más comprensible para el lector. En la segunda simulación mostraremos un escenario de decisión muy cercano a un caso real utilizando otros criterios, selección de pesos, más razonables.

4.1. Prueba de concepto

Proponemos un ejemplo cuyos datos se pueden encontrar en la tabla 4.1. Ésta tabla muestra un posible escenario de decisión para un sistema que mide la *distancia de interacción* en términos de confianza, que modelaremos con un valor continuo entre 0 y 1; distancia física, obtenida a partir de la potencia de señal recibida; y coste económico.

El primer elemento de la tabla representa el **elemento ideal** que se utilizará como referencia para medir la *distancia de interacción*, como se comentó en el capítulo anterior. Este elemento es Ideal dado que presenta un valor de confianza de 1, está muy cerca del dispositivo (distancia 0) y es gratis interactuar con él.

Para resolver el problema usaremos el algoritmo MDS ALSCAL, simplificando a una dimensión y utilizando $\lambda = 2$ para tratar los atributos como distancias. De esta forma, una vez tengamos la solución, podremos calcular la *distancia de interacción* así como clasificar los elementos.

En el ejemplo consideramos tres posibles situaciones antagónicas en las que uno

	Ideal(1)	2	3	4	5	6
Conf.	1.0000	0.9429	0.8430	0.9573	0.8344	0.0206
Dist.	0	0.5259	0.5048	0.4633	0.5270	0.4757
Coste	0	0.2054	0.2738	0.8636	0.8931	0.8461
	7	8	9	10	11	
Conf.	0.0464	0.0075	0.0597	0.0191	0.0935	
Dist.	0.5635	0.2540	0.2587	0.2509	0.2670	
Coste	0.8513	0.8424	0.8416	0.0	0.0	

Tabla 4.1: Valores de los atributos en un escenario de selección de red. Este escenario se utilizará para demostrar la validez conceptual del algoritmo.

de los atributos se ponderará siempre con un peso mucho mayor que los demás, de forma que se aprecien las diferencias.

Elección de un elemento seguro

En esta situación, la política establece que el vector de pesos debe ser {Confianza,Distancia,Coste}= {0.8,0.1,0.1}. El motor de decisión proporciona una lista ordenada de entidades en función de su distancia al ideal además una tabla con su distancia al dicho elemento ideal.

En la figura 4.1 hay dos representaciones de este espacio de decisión, una de ellas corresponde a una simplificación del problema a una dimensión y la otra a dos dimensiones. Los ejes de las figuras no se corresponden con ninguna atributo en concreto, es decir, la figura simplemente representa cómo de cerca están unas entidades de otras.

El resultado de la decisión, en forma de vector ordenado de entidades, según su cercanía al elemento ideal, es {1,4,2,5,3,11,9,7,6,10,8}. Examinando los resultados se puede ver que las entidades pueden ser divididas en dos grupos, el primer grupo de entidades ({4,2,5,3}), al estar cerca del elemento ideal 1, se pueden considerar elegibles. Las otras, están agrupados lejos del elemento ideal, razón por la cual no deberían ser elegidas.

Elección de un elemento cercano

En la segunda situación, la política establece el vector de pesos {Confianza,Distancia,Coste}= {0.1,0.8,0.1}. El resultado, que puede apreciarse gráficamente en la figura 4.2, es el siguiente: {1,10,11,8,9,6,4,3,2,5,7}. El vector de resultados,

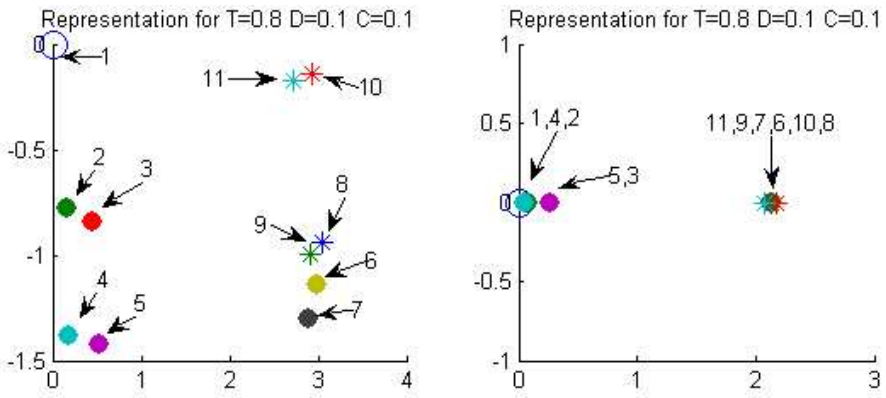


Figura 4.1: Selección de punto de acceso (anchor) favoreciendo la confianza con los siguientes pesos : Confianza 0.8, Distancia 0.1, Coste 0.1.

clasifica las entidades por su distancia de interacción y muestra que la distancia entre el elemento ideal 1 y el grupo más cercano al ideal, {10,11,8,9}, es muy alta, por lo que el dispositivo móvil podría decidir no interactuar.

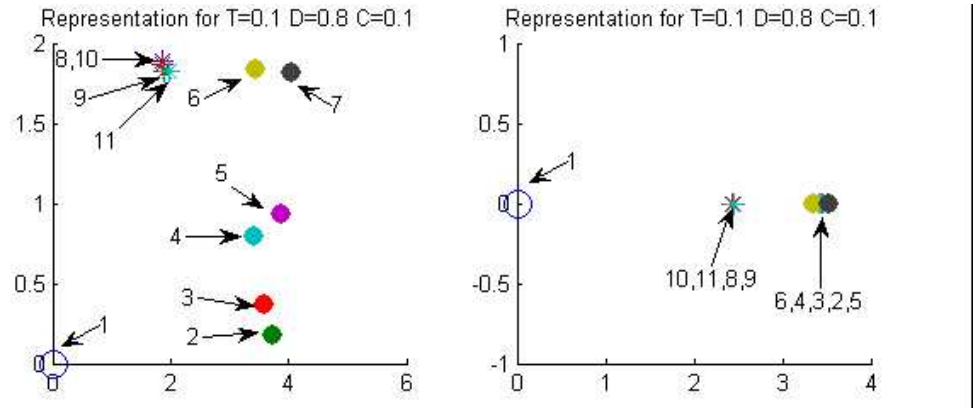


Figura 4.2: Selección de punto de acceso (anchor) favoreciendo la distancia con los siguientes pesos : Confianza 0.1, Distancia 0.8, Coste 0.1.

Elección de un elemento de bajo coste

En la tercera situación, el vector de pesos es $\{Conf, Dist, Coste\} = \{0,1,0,1,0,8\}$. En la figura 4.3 podemos ver el espacio de decisión. Como se puede apreciar, el conjunto de entidades que más se acerca al criterio de selección es $\{10,11\}$ que puede ser, por ejemplo, un conjunto de puntos de acceso cuyo servicio es gratuito. El segundo grupo

de entidades más cercano es el grupo {2,3}, que están lo suficientemente cerca para ser considerados una opción. El resto de las entidades están demasiado lejos para ser consideradas.

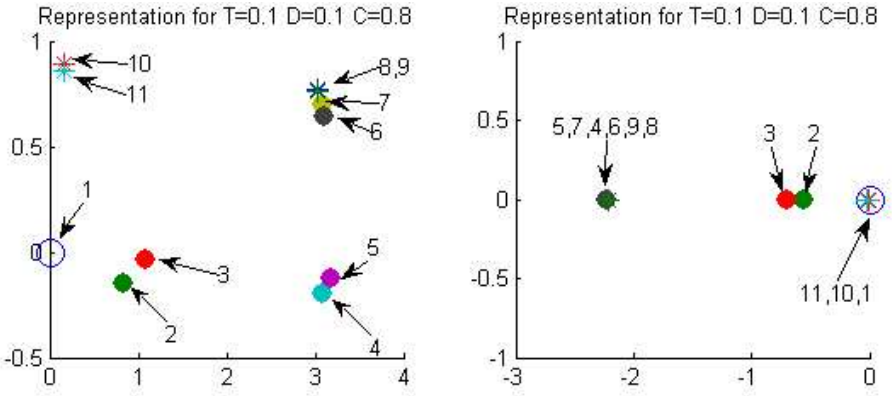


Figura 4.3: Selección de punto de acceso favoreciendo el coste económico con los siguientes pesos : Confianza 0.1, Distancia 0.1, Coste 0.8.

Datos de ajuste al modelo

Las simulaciones realizadas muestran que el modelo se ajusta a los datos si analizamos los parámetros de ajuste. El algoritmo utilizado, ALSCAL, minimiza un parámetro llamado S-STRESS que se utiliza para detener las iteraciones cuando su valor cae por debajo de un valor. Cuanto mayor sea el número de variables a las que se reduce en problema, mejores resultados se obtienen, ya que un número de variables demasiado pequeño puede ocasionar una pérdida demasiado grande de información.

El valor medio obtenido para S-STRESS durante las simulaciones, en las que se han variado los elementos de 2 a 60 y se ha reducido a 1 dimensión, es de 0.2728. Este valor es adecuado dado que S-STRESS varía entre 0 para el mejor ajuste y 1 para el peor. El resultado es satisfactorio sobre todo si tenemos en cuenta que se ha medido en el peor de los casos, es decir, reduciendo a una dimensión.

Cuando MDS se aplica a otros problemas podría ser necesario obtener un mejor ajuste, pero en esta serie de simulaciones sencilla hemos podido determinar que el valor obtenido de S-STRESS es suficiente para nuestros propósitos de selección de red. Además, se ha tenido en cuenta la correlación cuadrática entre las (di)similitudes y las distancias, llamada RSQ, que da una idea de cuán adecuado es el ajuste al modelo. RSQ varía entre 1 para un ajuste perfecto y 0 para un desajuste. Los valores de RSQ obtenidos en las simulaciones estaban en un entorno de 0,8 a 1.

	Ideal(1)	2	3	4	5	6
Confianza	1	0,026	0,589	1	1	1
Distancia	0	0,5259	0,5048	0,4633	0,5270	0,8461
Coste (kByte)	0	NaN	NaN	0,004878	NaN	0,004878
Coste (s)	0	0	0,01197	NaN	0	NaN
Autenticación abierta	x	1	1	0	0	0
Velocidad enlace (kbps)	x	11000	6000	40	11000	130
Movilidad soportada	x	1	1	3	1	3
Soporte llamada SOS	x	0	0	1	0	1
Permite roaming	1	0	1	1	1	1
Personal/Trabajo	x	0	0	2	1	0
Tipo de red	x	802.11b	802.11b	GPRS	802.11b	EDGE

	7	8	9	10	11	
Confianza	1	0,874	1	0,191	0,843	
Distancia	0,5635	0,2540	0,2587	0,2509	0,2670	
Coste (kByte)	0,001953	NaN	0,004878	NaN	NaN	
Coste (s)	NaN	0	NaN	0	0,01197	
Autenticación abierta	0	0	0	1	0	
Velocidad enlace (kbps)	320	54000	700	54000	54000	
Movilidad soportada	3	0	3	0	0	
Soporte llamada SOS	1	0	1	0	0	
Permite roaming	1	0	1	0	0	
Personal/Trabajo	1	0	2	0	0	
Tipo de red	UMTS	802.11g	HSDPA	802.11g	802.11g	

Tabla 4.2: Espacio de decisión incluyendo datos de categoría y ausencia de datos para distintas entidades.

4.2. Simulación de un caso real

En esta simulación podemos encontrar un escenario más cercano a la realidad. La tabla 4.2 muestra otro espacio de decisión con valores más razonables para las diferentes entidades y utiliza datos de pertenencia a categoría.

Para demostrar el correcto funcionamiento del algoritmo incluimos datos de pertenencia a categoría además de atributos cuantitativos, que son:

- *Autenticación abierta*: Este atributo toma el valor 0 cuando la red no requiere autenticación y 1 para autenticación de nivel de enlace.
- *Movilidad soportada*: 0 para transmisiones en las que el dispositivo móvil perma-

nece estático o realiza pequeños movimientos, 1 para tecnologías que soportan movilidad a la velocidad de un peatón, 2 para aquellas tecnologías que permiten movimientos a la velocidad de un coche y 3 para movilidad completa (limitada por doppler o por atenuación de señal)

- *Soporte de llamada SOS*: permite saber si es posible realizar una llamada de emergencia usando esa red o no, tendrá el valor 1 si se permite.
- *Permite roaming*: La red soporta roaming. Las redes celulares siempre soportan roaming y algunas redes WiFi también (1 si lo permite).
- *Personal/Trabajo*: En el ejemplo, entre las redes descubiertas, hay dos que deben usarse por motivos de trabajo, dado que la compañía paga por esos servicios. Como se pude ver, se dispone de una posible conexión usando GPRS, que puede usarse para leer el correo o en general para aplicaciones sin altas demandas de tráfico. Por otro lado, se dispone de una conexión HSDPA para aplicaciones con una mayor demanda de ancho de banda. Para distinguirlos, esta variable puede tener tres diferentes valores, 2 para las redes pagadas por la compañía, 1 para las redes pagadas por el usuario y 0 para otras redes.

La tabla 4.2 muestra el espacio de decisión para esta segunda simulación. En dicha tabla se puede apreciar que algunos de los atributos tienen el valor *NaN* o "Not a Number", que es la representación aritmética de la IEEE para un valor que no es un número. Por otro lado la x, presente en algunos valores de los atributos del elemento Ideal, indica que es variable en el tiempo, dado que se adaptará a las demandas de tráfico del momento. Para este espacio de decisión vamos a suponer 4 escenarios con parámetros de selección diferentes para analizar así el comportamiento del algoritmo.

Selección de red con múltiples atributos

Este escenario corresponde a una selección compleja donde se utilizan muchos atributos dado que se requiere una red o par muy concreto.

Definamos el escenario: el usuario está conduciendo y a la vez su móvil está sincronizando ciertos ficheros importantes con el servidor de la empresa. Asumamos que el contenido es importante para la compañía y que debe ser tratado como confidencial. El motor de decisión modela el punto de acceso ideal como aquel capaz de proporcionar un tráfico de 20 kBps según la demanda de tráfico actual.

A través del sistema de definición de dominios, el sistema detecta que el dominio actual es el coche, por lo que el punto de acceso o red ideal, debe soportar movilidad completa o aquella que permita la transferencia de datos a la velocidad del coche. Por

otro lado se le da el valor de 2 al parámetro “Personal/Trabajo” indicando así trabajo. Es entonces cuando el motor de decisión, a través de la política, obtiene valores para el vector de pesos que potenciarán unos atributos por encima de otros. El vector de pesos será cero para todos los atributos exceptuando: “confianza” al que se le da el valor 0.2 para que trate de elegir un punto de acceso o red confiable; “coste por kByte” al que se dará el valor de 0.1, movilidad cuyo valor será de 0.2, “roaming” 0.2 y “Personal/Trabajo” 0.3.

El resultado de ejecutar el algoritmo es el mostrado en la tabla 4.3. Dicha tabla muestra en la columna “escenario 1” las distancias en valor absoluto entre el elemento ideal, aquel que cumpliría todos los requisitos de conexión, y el resto de entidades, que representan las redes disponibles. La entidad que aparece como más atractivo es la número 4 como puede verse en la figura 4.4.

Por otro lado, tanto el 7 como el 9 están cerca del ideal. La red número 4 proporciona la tasa de transferencia adecuada, pertenece a las redes de trabajo y soporta movilidad, en cambio la red número 7 no pertenece a las redes de trabajo y está seleccionada como preferible a la número 9. Además, la red número 9 es más atractiva dado que permite una mayor transferencia de datos al mismo coste que la red número 4. Esta deficiencia se debe a que no se ha dado importancia al atributo velocidad de enlace.

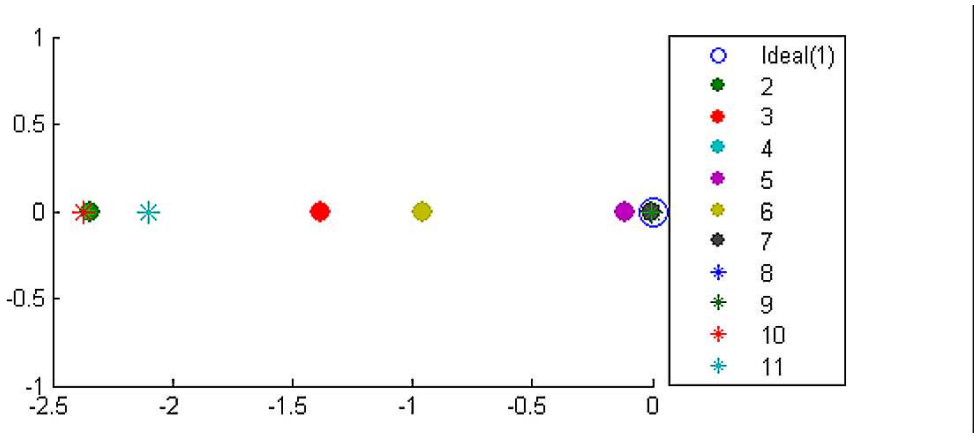


Figura 4.4: Selección de punto de acceso en el escenario 1 que consiste en una selección compleja en la que influyen varios atributos.

Reflexiones sobre el número de dimensiones

En la tabla 4.3 se puede apreciar que los puntos 4, 7 y 9 están muy próximos al ideal. Pese a ello sorprende que la red 7, pagada por el usuario, esté en realidad tan cerca del punto ideal. Pese a que el conjunto de redes seleccionadas, cubre de sobra

	escenario 1	escenario 1 (2D)	escenario 1b	escenario 1b (2D)
2	2,3415	3.1477	2.1547	3.0297
3	1,3818	2.0320	1.5565	2.1193
4	**0,0021**	**0.0047**	0.0001	0.0197
5	0,1107	1.9479	0.5221	1.9029
6	0,9542	1.1281	1.0410	1.2294
7	0,0043	0.7555	0.0056	0.6751
8	2,1010	2.7978	2.1544	2.8438
9	0,0058	0.0129	0.0006	0.0330
10	2,3709	3.0949	2.6148	3.1963
11	2,0986	2.7974	2.1535	2.8514

Tabla 4.3: Distancias desde el elemento ideal para varios casos en el escenario 1, que consiste en una selección compleja en la que influyen varios atributos. Muestra por otro lado los resultados en dos dimensiones.

autovalor	7.06	2.66	1.24	0.89	0.70	0.51	0.42	0.33	0.30	0.014	0
autovalor(%)	50.0	18.8	8.7	6.3	4.9	3.6	2.9	2.3	2.1	0.1	0

Tabla 4.4: Autovalores de la matriz de (di)similitudes en el escenario 1.

las necesidades de la aplicación, demostrando que el algoritmo es válido, existen una serie de apreciaciones respecto al ajuste que, si son tenidas en cuenta, mejoran los resultados.

Tenemos como objetivo permitir que el usuario pueda consultar siempre el espacio de decisión, por lo que tendremos siempre que recurrir a una representación en una o dos dimensiones. Los valores de los parámetros usados para medir la bondad del ajuste son S-STRESS=0.191 y RSQ=0.901 para la reducción a una dimensión, de dónde se extrae que el ajuste es relativamente bueno: S-STRESS varía entre 0 y 1, siendo 0 perfecto; RSQ varía entre 0 y 1, siendo 1 perfecto.

Por otro lado, consultar el valor de los autovalores de la matriz de diferencias puede ayudar mucho a calcular las dimensiones adecuadas a las que reducir el problema en MDS tradicional, donde se usa descomposición en valores propios. El mismo procedimiento nos puede ayudar a dimensionar el problema en ALSCAL. Los autovalores se pueden observar en la tabla 4.4, donde se aprecia que el primer autovalor concentra el 50% de la información, por lo que una reducción a una dimensión puede perder demasiada información.

Si reducimos a dos dimensiones en lugar de a una, usando los dos primeros autovalores, cubriremos prácticamente el 70 % de la información. El resto de la información está bien distribuida entre los demás atributos, por lo que no se obviaría ningún atributo importante. Una gráfica de Pareto sobre los autovalores de la matriz de (di) similitudes, como la que se muestra en la figura 4.5, permite ver la evolución de la cobertura de la información.

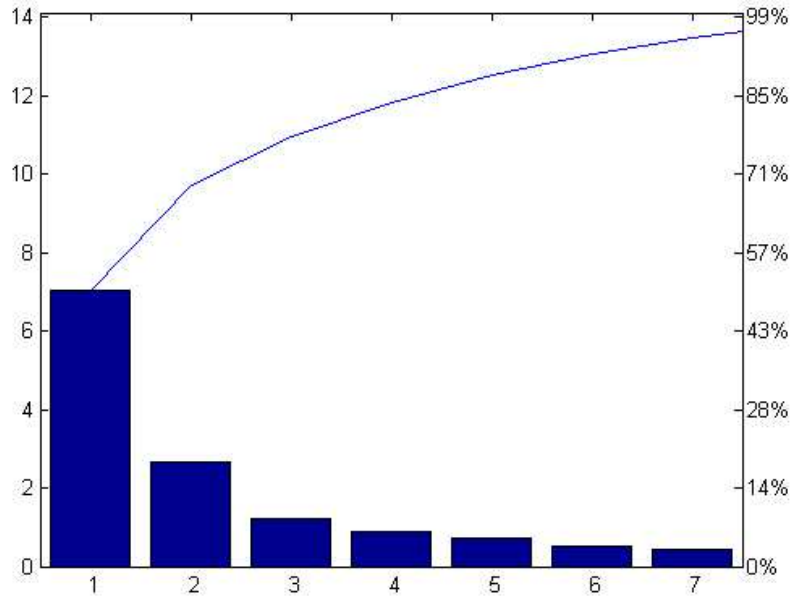


Figura 4.5: Gráfico de Pareto para los autovalores que muestra que una reducción a dos dimensiones equivalen al 70 % de la información.

Para ratificar la validez de este mecanismo de cálculo vamos a cambiar momentáneamente los pesos reduciendo a 0 el peso asignado al "coste por kByte" y repartiendo la suma total de pesos entre los demás atributos equitativamente. Hecho esto, obtenemos un gráfico de Pareto prácticamente idéntico al anterior y obtenemos un 75 % al usar dos autovalores. De esta manera, eliminamos la dependencia con el coste por kByte que no está definido para todos los elementos (NaN). Al eliminar estos atributos indefinidos podemos realizar un **Análisis de Componentes Principales** sobre todos los elementos menos el ideal. Como el hecho de usar un peso 0 para los costes hace que estos no computen en la matriz de disimilitudes de MDS, estaríamos analizando un problema prácticamente idéntico mediante Análisis de Componentes Principales.

Si obtenemos un gráfico de Pareto (Fig. 4.6) con los resultados del las varianzas explicadas del Análisis de Componentes Principales obtenemos resultados similares,

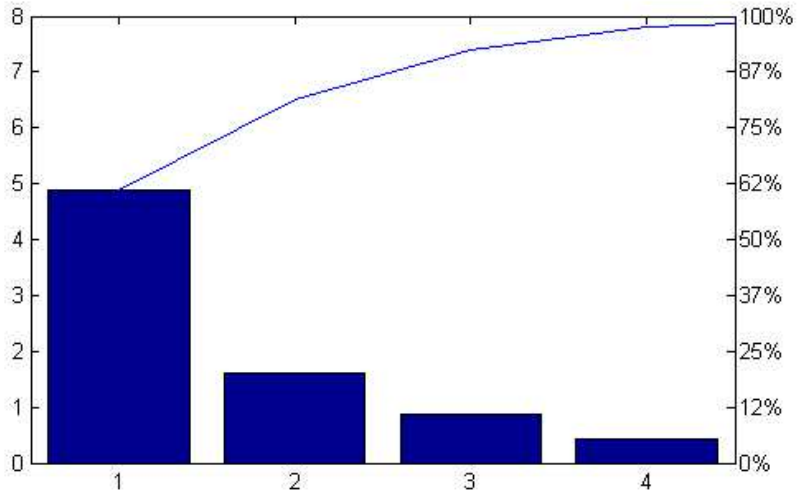


Figura 4.6: Gráfico de Pareto para las varianzas explicadas obtenidas del Análisis de Componentes Principales para determinar el número de variables a las que se reduce el problema.

donde se ve perfectamente que con dos variables se puede llegar a explicar más del 70 % del problema. El tratamiento de los datos con Análisis de Componentes Principales no permite el tratamiento cualitativo de los datos, de hecho este análisis presuponen datos métricos, pero ayuda a estimar el número de dimensiones del problema general.

Para finalizar, en la figura 4.7, se puede apreciar la evolución de los parámetros que miden la bondad del ajuste en MDS respecto al incremento de dimensiones para el caso original, sin la modificación realizada a los pesos para el análisis con Análisis de Componentes Principales. Este modo de proceder, representar S-STRESS y RSQ para varias dimensiones, es apropiado para determinar las dimensiones como se recoge en [90]. Si analizamos la figura, vemos que de una a dos variables hay una ligera mejora que sólo es útil para precisar un poco la decisión como ocurre en este escenario, dado que el ajuste en todos los casos es bastante bueno para una dimensión.

En la figura 4.8 así como en la columna “escenario 1 2D” de la tabla 4.3, se muestra información sobre la decisión cuando se reduce el problema a dos dimensiones. Los puntos más afines al punto ideal son, en orden, los puntos 4, 9 y 7. Se puede comprobar como el resultado es más adecuado al problema por lo que se debe incluir mayor información en la decisión. Los datos del ajuste en este caso son S-STRESS=0.0540 y RSQ=0.9870, que mejoran respecto a la reducción a una dimensión. Esto nos lleva a la conclusión de que es necesario que el motor de decisión no solo procese los datos,

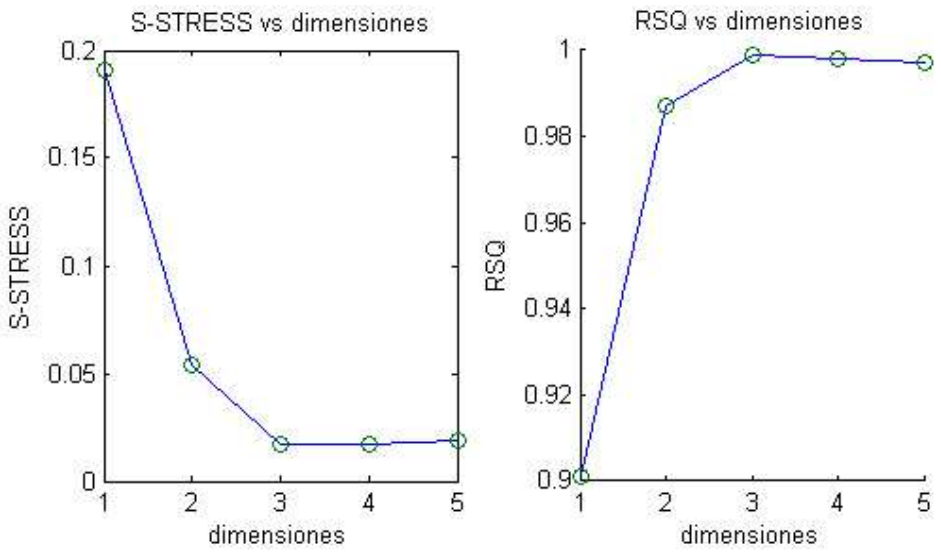


Figura 4.7: Variación de los parámetros de ajuste, RSQ y S-STRESS, con el aumento del número de variables.

sino que decida qué reducción es más adecuada tratando, en la medida de lo posible, que sean representables sobre una pantalla.

Reflexiones sobre la elección de los pesos y la naturaleza de los atributos

La asignación de pesos depende de varios factores como se ha visto en la sección 3.2. Este proceso de asignación de pesos es muy importante ya que permite una mayor libertad en la selección de la red: se puede dar más importancia subjetiva a determinados atributos. En el vector de pesos utilizado con anterioridad se ha ponderado con 0 la velocidad de enlace, por lo que el motor de decisión no la ha tenido en cuenta. Veamos ahora en qué afecta esta asignación de pesos cambiando el valor de "roaming" a 0.1 y asignando a "Velocidad de enlace" el valor de 0.1 de modo que se tenga en cuenta la velocidad del enlace y se mantenga la suma de pesos constante. Los resultados de la reducción a dimensión se pueden ver en la figura 4.9.

La figura 4.9 muestra una distribución de puntos un tanto peculiar, dado que se encuentran dispuestos en grupos equidistantes, no pareciendo muy lógico el agrupamiento si contrastamos con los datos. Veamos lo que ocurre si simplificamos a dos dimensiones como se muestra en la figura 4.10 y en la tabla 4.3 (columnas "escenario 1b" y "escenario 1b (2D)"). En dos dimensiones se puede apreciar una distribución de puntos más coherente: los elementos 4 y 9 son los más atractivos y están localizados sumamente cerca del punto Ideal. Por otro lado el elemento 7 se encuentra separa-

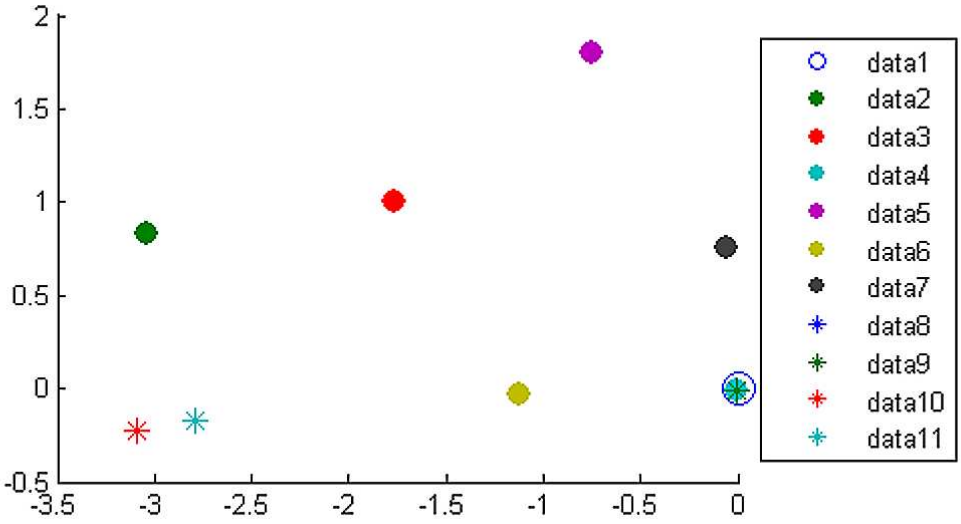


Figura 4.8: Selección de punto de acceso en el escenario 1. Gráfico para dos dimensiones.

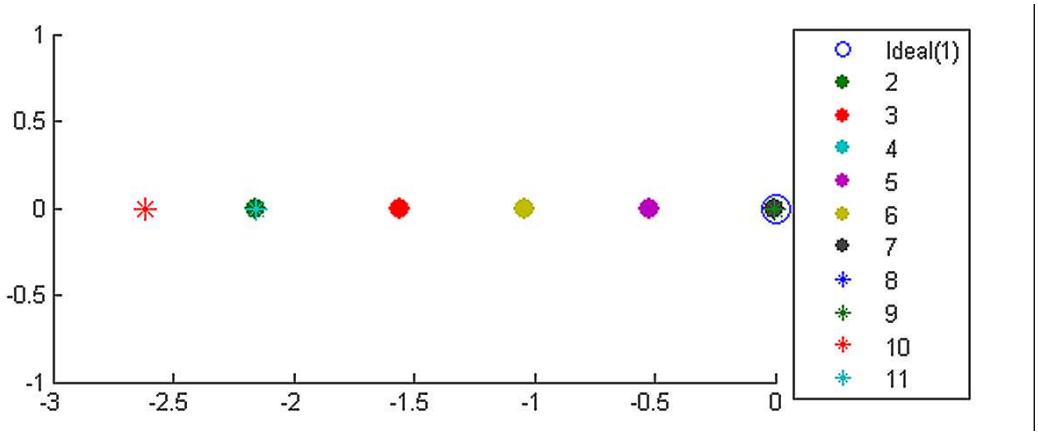


Figura 4.9: Selección de punto de acceso en el escenario 1 considerando además la velocidad de enlace.

do del anterior grupo dado que la red no es de las suministradas por el trabajo. Los elementos 5 y 6 están a una distancia similar del Ideal pero en regiones del espacio distintas dado que, pese a ser redes de confianza, no tienen similitudes en velocidad de enlace; ni soporte a la movilidad; y no pertenecen al grupo de redes de trabajo. El resto de elementos están alejados del Ideal pudiéndose observar un agrupamiento de los elementos 10, 8 y 11, que obedece a la gran similitud de estos puntos entre si, siendo

el más separado el punto 10 por motivos de autenticación. Puede consultar la tabla 4.3 para contrastar que estas diferencias no eran visibles ni medibles en una reducción a una dimensión.

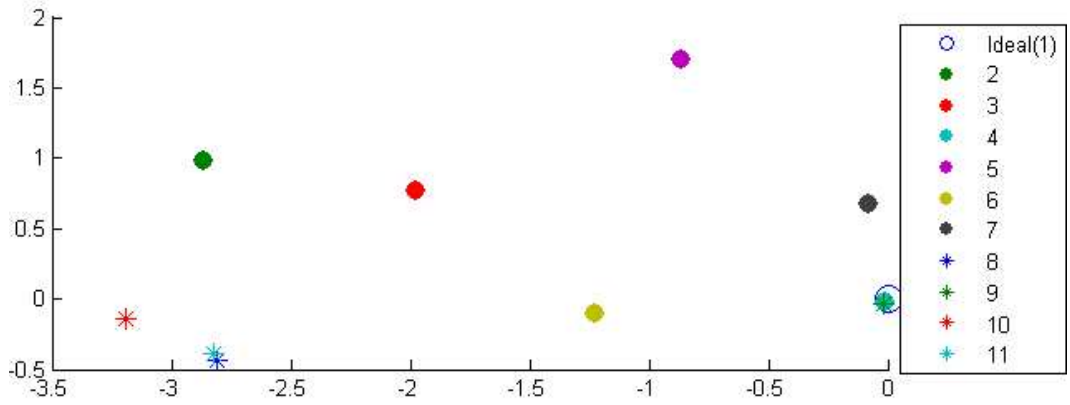


Figura 4.10: Selección de punto de acceso en el escenario 1 considerando la velocidad de enlace.

El procedimiento utilizado para el cálculo de la velocidad de enlace utiliza procedimientos métricos, es decir, establece diferencias siempre entre los elementos, con independencia de que puedan cursar el tráfico solicitado. Por esta razón, se elige antes una conexión GPRS (elemento 4) que una HSDPA (elemento 9) pese a que la segunda tiene el mismo coste así como ancho de banda suficiente para cursar el tráfico. Un pequeño cambio en el procesado del atributo, que pasaría a ser un **atributo cuantitativo con umbral** a partir del cual no se calcula la diferencia, permitiría el calculo de diferencias entre elementos en base a ese atributo, siempre y cuando su velocidad de enlace no fuera suficiente para cursar el tráfico requerido. De esta forma no se incrementaría la diferencia en los casos en los que la velocidad de enlace fuera superior a la marcada por el umbral: el algoritmo no encontraría diferencia alguna respecto al atributo “Velocidad de enlace” entre los elementos 4 y 9. El atributo “Velocidad de enlace” se procesará como métrico con umbral en un escenario diferente más adelante.

A continuación vamos a considerar, además de la velocidad del enlace, la distancia siendo el vector de pesos tal que asigne 0.2 a la confianza; 0.3 al atributo “Personal/Trabajo”; y 0.1 a los atributos de distancia, coste por kByte, velocidad de enlace, movilidad y roaming; permaneciendo los demás a 0. El espacio de decisión sería, representado en dos dimensiones, el mostrado en la figura 4.11. En este cálculo se obtienen los valores de S-STRESS=0.221 y RSQ=0.865 resolviendo para una dimensión y de S-STRESS=0.0390 y RSQ=0.9930 resolviendo para 2 dimensiones (un ajuste mucho mejor). Las distancias pueden verse en la tabla 4.5 pudiéndose apreciar cómo, en este

	escenario 1c (2D)
2	3.4833
3	2.2251
4	**0.2798**
5	1.9146
6	1.8686
7	1.7404
8	2.7088
9	**0.2427**
10	3.2132
11	2.7365

Tabla 4.5: Distancias desde el elemento ideal en el escenario 1 considerando la distancia física.

caso, el elemento más adecuado pasa a ser el 9, debido al efecto de la distancia física.

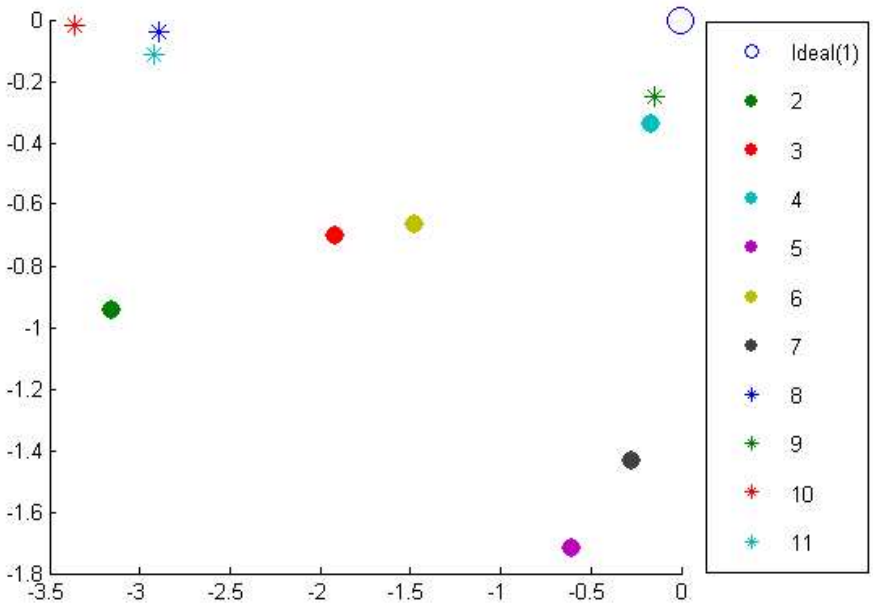


Figura 4.11: Selección de punto de acceso en el escenario 1 considerando además la distancia (Dos dimensiones).

En este escenario se podría considerar que si todas las redes de trabajo son con-

fiables, bastaría un aumento del peso asignado al atributo “Personal/Trabajo”, previa reducción del atributo “confianza”, para asegurar la selección de una red costeada por la compañía. Esto obligaría a analizar cada caso en concreto antes de hacer una elección de pesos y requeriría intervención humana para determinar que atributos están conceptualmente correlados, cosa que queremos evitar a toda costa.

Selección de red para ocio

En este escenario veremos una selección de red un poco más sencilla, que permitirá apreciar mejor el agrupamiento que realiza el algoritmo.

El usuario está esperando para embarcar en un vuelo. Éste dispone de un tiempo para el esparcimiento, por lo que decide navegar por Internet. El elemento ideal se modela con un valor de “movilidad” 0, es decir estático, dado que se detecta que no hay movimiento. La velocidad del enlace se establece en 1000 kBps y se da el valor de 1 al atributo “autenticación abierta”. El resto de los atributos se dejan sin especificar con el valor NaN. El vector de pesos que se utilizará es cero para todos los atributos salvo para: los costes (0.5), “autenticación abierta” (0.2) y “velocidad de enlace” (0.3).

Los pesos se establecen de esta manera porque la política o el registro de preferencias del usuario o una combinación de ambas establecen que, para tráfico de ocio, en el cual no se comprometa ninguna información, aquello que debe tenerse en cuenta por encima de lo demás es el coste. Además, debe intentarse que la red no requiera autenticación, un host spot gratuito o la red de algún incauto que sea capaz de cursar el tráfico demandado por la aplicación.

El punto seleccionado es el 2 como puede comprobarse gráficamente en la figura 4.12 así como en la tabla de distancias 4.6, en la columna “escenario 2”. Como puede apreciarse, existen cuatro grupos bien diferenciados. El grupo 0 son aquellos elementos más afines a la búsqueda realizada y corresponden a dos puntos de acceso que proporcionan la velocidad requerida sin coste para el usuario. Además, disponen de autenticación abierta.

Los elementos 8 y 10, que forman el grupo 1, están cercanos al Ideal pero se diferencian en la velocidad de enlace y el 8 en la autenticación. El algoritmo agrupa las tecnologías de alta movilidad y bajo ancho de banda en el grupo 2. El grupo 3 está separado por motivos de ancho de banda y el 4 por motivos de coste.

Procesando un atributo con umbral

Volviendo al tema del procesado del atributo velocidad, si procesamos el atributo velocidad como se expresó en el anterior escenario, es decir, utilizando un umbral

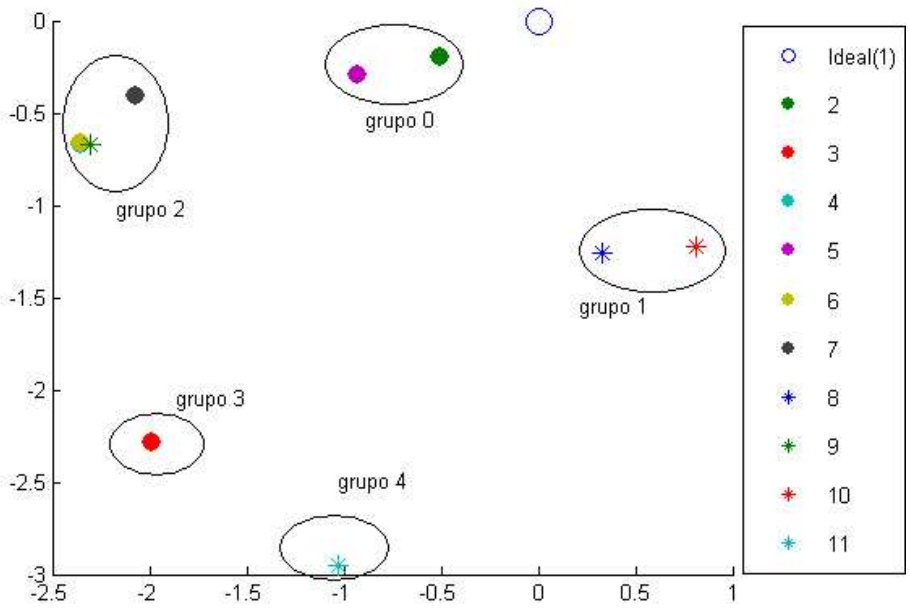


Figura 4.12: Selección de punto de acceso en el escenario 2: selección de red para ocio.

mínimo ancho de banda a partir del cual las redes son equivalentes a efectos de velocidad, vemos algunas variaciones que pueden apreciarse en la figura 4.13.

Se puede ver como tecnologías similares en criterios de coste, con un ancho de banda mayor, se acercan mucho al ideal (véase en la figura 4.13 el grupo 0). En este grupo vemos como tenemos cerca del ideal tanto a los puntos 2 como 8 y un poco más lejos, pero aún cerca del ideal, el 10, hecho que obedece a una indefinición en el coste. Las tecnologías con un ancho de banda menor y mayor movilidad se agrupan en el grupo 3. Los grupos 1 y 2 se alejan del ideal debido a indefiniciones en algunos atributos o diferencias de coste.

Vamos ahora a darle cierta importancia, como es lógico a la confianza, en detrimento del coste. Además no requeriremos autenticación abierta. Esto se debe a que rara vez, en un dominio identificado como desconocido (el usuario no frecuenta la sala de embarque tanto como su casa o trabajo) como sería el aeropuerto, un hotspot con autenticación abierta no requiere pago. También es posible considerarlo una amenaza. Por ello el punto ideal tendrá el atributo de “Autenticación abierta” a 0 y se moverá el peso atribuido a “Autenticación abierta” al parámetro “Confianza” junto con parte del coste. El vector de pesos resultante es el siguiente: {Confianza=0.3,Coste(s)=0.3, Velocidaddeenlace=0.3,movilidad=0.1}.

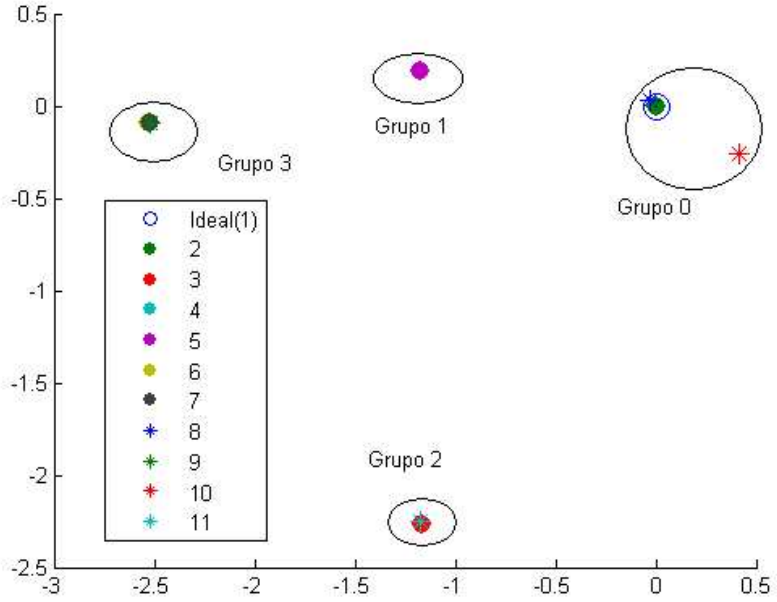


Figura 4.13: Selección de punto de acceso en el escenario 2, red para ocio, procesando la velocidad de enlace con umbral mínimo.

El resultado se puede observar en la figura 4.14. En esta figura puede apreciarse como la red más adecuada es la 5, ya que como es lógico, además de satisfacer correctamente las necesidades de conexión, es una red altamente confiable. Cerca del Ideal, tendremos el puntos 8, con una confianza también alta. Si observamos la figura, podemos ver como el punto 2 se encuentra relativamente cerca del Ideal pese a ser poco confiable: la razón es que se pondera de igual manera el coste por segundos que la confianza y, de alguna manera, se podría decir que las redes libres sin coste (no pagadas por el usuario ni la empresa) son poco confiables.

Vamos a establecer un umbral para el coste, es decir, un máximo que estamos dispuestos a gastar, al igual que hicimos para la velocidad de enlace, pero en este caso se tratará de un umbral máximo (coste máximo). Los costes seleccionados como máximo son 0.0048 para coste por kByte y 0.02 para coste por segundos. El resultado cambia, como se puede ver en la figura 4.15 y la tabla 4.6 (columna “escenario 2c (2D)”). En dicha figura podemos observar que los puntos más cercanos al Ideal son ahora confiables y cumplen los requisitos de tráfico (puntos del interior de la elipse).

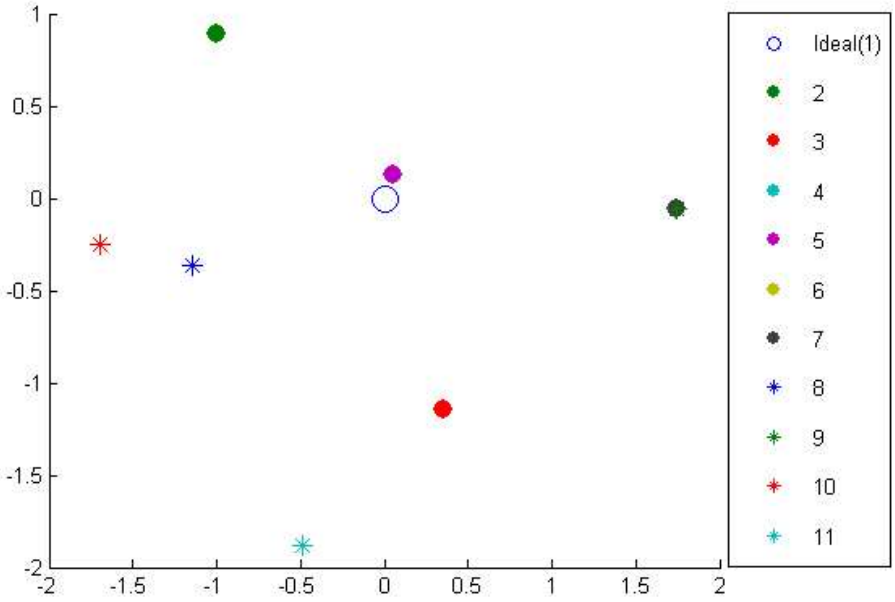


Figura 4.14: Selección de punto de acceso en el escenario 2, procesando la velocidad de enlace con umbral mínimo y teniendo en cuenta la confianza.

Selección de red para difusión de video

En este escenario el usuario pretende ver un video en su casa. El punto de acceso ideal se establece en aquel capaz de proporcionar al menos una velocidad de enlace de 11000 kBps en situación de movilidad baja (0). Se intenta minimizar el coste (0) y no es necesario el uso de redes de trabajo: Personal/Trabajo a 1. El resto de los atributos se dejan indefinidos dándoles el valor de NaN, salvo la confianza, que siempre se establece en el Ideal a 1. El vector de pesos será {confianza,coste,velocidad de enlace,personal/trabajo}= {0.3,0.3,0.2,0.2}. Utilizamos umbrales tanto para la velocidad de conexión como para el coste, con los mismos valores para el coste que en el escenario anterior y 11000 kBps para la velocidad de enlace.

El resultado del algoritmo muestra que los puntos más adecuados serían, en orden de proximidad al ideal, el 5, 8 y 11 (mirar figura 4.16 y tabla 4.7).

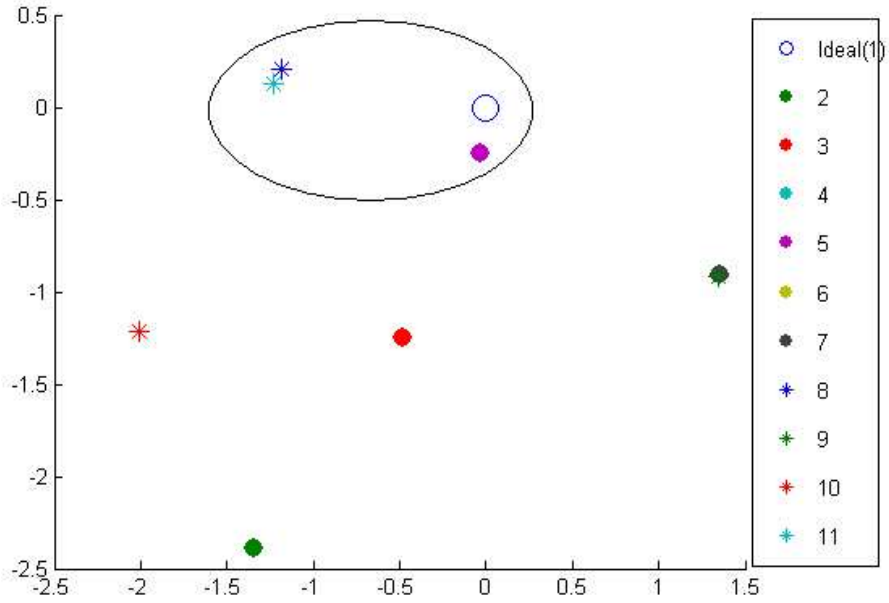


Figura 4.15: Selección de punto de acceso en el escenario 2 procesando la velocidad de enlace con umbral, la confianza y procesando el coste con umbral máximo.

Selección de red para llamada de emergencia

En una situación de emergencia, una vez el usuario pulsa un botón de pánico o realiza una llamada al 112 (o 911), el par Ideal es aquel más cercano, por motivos de calidad de señal y para minimizar los posibles fallos derivados de una cobertura insuficiente; capaz de cursar una llamada de emergencia. El resto de los atributos no son importantes y por ello estarán sin especificar. El vector de pesos potenciará ambos atributos dando el valor de 0.1 a la distancia, para seleccionar un punto de acceso cercano y 0.9 a la capacidad de llamada de SOS. Como puede verse en la tabla 4.8, los cuatro primeros puntos de acceso a los que conectarse son: 9, 4, 7 y 6.

Conclusiones

En esta segunda serie de simulaciones se demuestra que, en ausencia de datos (atributos con valor NaN), el algoritmo funciona proporcionando una selección adecuada. La tabla de distancias permite realizar una selección apropiada al dispositivo móvil.

	escenario 2 (2D)	escenario 2b (2D)	escenario 2c (2D)
2	**0.5449**	**0.0006**	2.7459
3	3.0313	2.5478	1.3342
4	2.4469	2.5268	1.6333
5	0.9739	1.1858	0.2500
6	2.4432	2.5261	1.6308
7	2.1102	2.5237	1.6300
8	1.3041	0.0424	1.2024
9	2.4029	2.5186	1.6295
10	1.4636	0.4912	2.3462
11	3.1280	2.5401	1.2323

Tabla 4.6: Distancias desde el elemento ideal para las diferentes simulaciones dentro del escenario 2 (selección para ocio).

	escenario 3 (2D)
2	2.4134
3	1.5336
4	1.6066
5	**0.2595**
6	1.6067
7	1.6065
8	1.1713
9	1.6020
10	2.1893
11	1.1713

Tabla 4.7: Distancias desde el elemento ideal para el escenario 3 (selección para difusión de video).

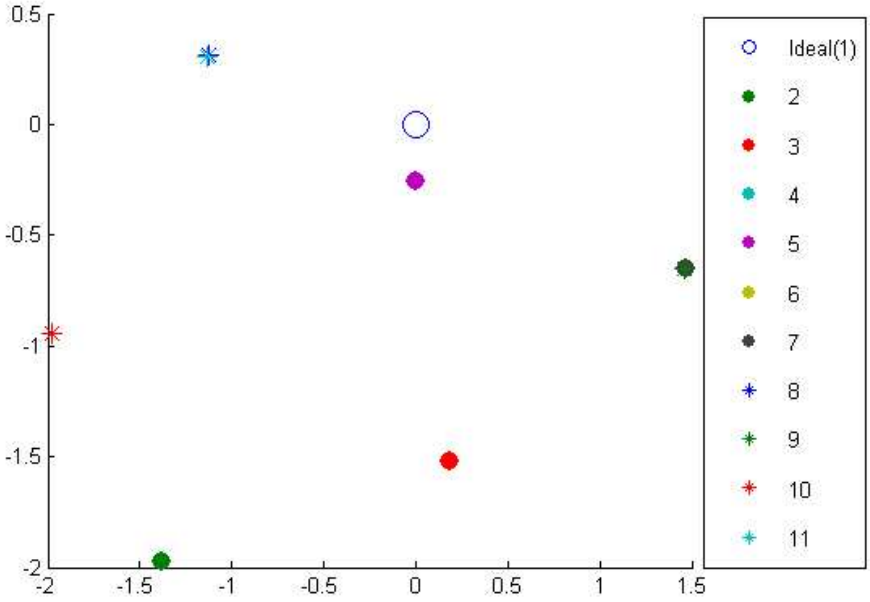


Figura 4.16: Selección de punto de acceso en el escenario 3 (selección para difusión de video).

Además de la selección automática, uno de los objetivos que perseguimos es la aumentar la comprensión del usuario y su percepción del riesgo, por eso se muestra al usuario el espacio de decisión y se recurre a métodos gráficos. Éstos permiten que el usuario valore la conveniencia de determinadas decisiones mediante comparación y agrupamiento.

4.3. Rendimiento

La complejidad medida del algoritmo es de $O(n^{2,65})$ siendo n el número de elementos. La figura 4.17 muestra los resultados. Para realizar la prueba de rendimiento se ha utilizado Matlab, para preparar los datos del problema, y una versión compilada en C del algoritmo ALSCAL, corriendo en un solo núcleo del procesador (mediante una máscara). Previamente a su llamada, se han utilizado contadores de alta precisión que permiten medir el tiempo de proceso con ALSCAL. La resolución del contador es la siguiente: durante un segundo se cuentan 3.579.545,00 unidades.

El rendimiento, medido en tiempo, es suficiente para resolver problemas de selec-

	escenario 4
2	2.8592
3	2.8536
4	0.5132
5	2.8592
6	0.9349
7	0.6235
8	2.8129
9	**0.2878**
10	2.8126
11	2.8150

Tabla 4.8: Distancias desde el elemento ideal, escenario 4 (selección para emergencia).

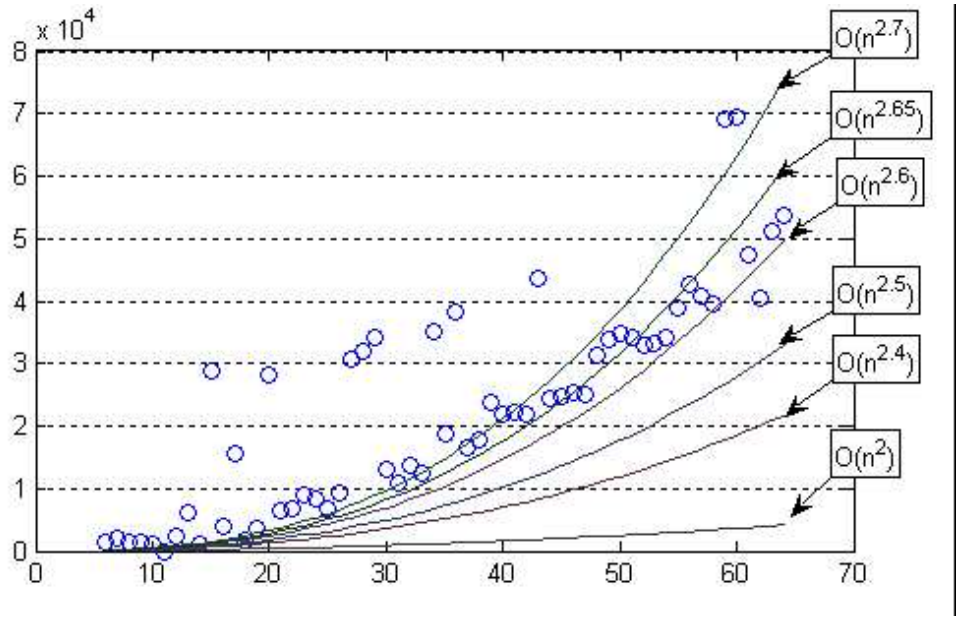


Figura 4.17: Valor del contador de precisión vs. número de elementos de la simulación, para 3 atributos y simplificación a una dimensión

ción de red en un tiempo no percibido por el usuario como molesto. El lector debe tener en cuenta la importancia de este requisito de tiempo, dado que se trata de actuar en todo momento por debajo del umbral de consciencia del usuario.

CAPÍTULO

5 Propuesta de un mecanismo agnóstico de asistencia a la decisión en negociación de confianza

Como se introdujo en el estado del arte (capítulo 2), realizar control de acceso requiere determinar si un dispositivo que solicita acceso está autorizado a usar ese servicio o no. Aparte del acceso, en determinadas circunstancias, se puede requerir control sobre otros parámetros como la calidad de servicio que se debe proporcionar a ese dispositivo. El control de acceso es un proceso que, en general, comprende autenticación, autorización y la utilización de políticas, que cubren el contexto y la sensibilidad de los recursos y credenciales.

En este capítulo presentamos una solución para ayudar al dispositivo en la toma de decisiones en negociaciones P2P. El "servidor" podrá comunicar los requisitos al "cliente" en base a una política. El "cliente" podrá determinar si los requisitos solicitados son justos, es decir, se ajustan al estado actual de la negociación, lo que le permitirá detectar posibles abusos antes de proporcionar credenciales al "servidor". Nuestro motor de decisión proporciona además una representación gráfica del espacio de decisión que permitirá a los usuarios sin conocimientos técnicos entender el problema.

Como medida de la dimensión del problema que apunta a la necesidad de motores genéricos, en el capítulo 2 se hace un recorrido por las credenciales, políticas, mecanismos de autenticación y protocolos. A lo largo de este capítulo se mostrará el algoritmo utilizado para combinar datos puede ser utilizado para asistir al usuario durante una negociación de confianza. El algoritmo es capaz de utilizar cualquier credencial, política o protocolo, dado que simplemente orquesta la negociación tratando de que sea justa para ambas partes, sin procesar cada credencial o política. El motor simplemente utiliza los metadatos que acompañan a cada elemento involucrado en la decisión. Más adelante se mostrará un ejemplo para que sea más sencillo de entender.

5.1. Definiciones

Para ayudar al lector a entender el resto del capítulo, se proporcionan las siguientes definiciones:

Las **políticas** son un conjunto de reglas que se utilizan para guiar decisiones y acciones. Son emitidas por los gestores de recursos, el propio usuario, administradores de dominio o un proveedor de servicio. Para proteger un **recurso** puede usarse más de una política cuya combinación tiene como resultado un conjunto de requisitos que deberán ser satisfechos para acceder al recurso.

Un **requisito** representa, por tanto, la información que debe ser revelada para satisfacer parte de una política o conjunto de políticas. De una política o conjunto de políticas se deriva un requisito o conjunto de requisitos.

Un **policy item** es la definición formal de un requisito que puede ser usado por otras entidades para determinar qué credenciales enviar durante un proceso de negociación de forma que se satisfaga un requisito.

Una **credencial** es cualquier información que tenga que ser desvelada para satisfacer un requisito. Finalmente, como **recurso** entendemos cualquier información, servicio o mecanismo cuyo acceso entraña un riesgo de cualquier tipo. Por esta razón las credenciales deben considerarse recursos y deben ser protegidas por políticas, limitando el ámbito y las circunstancias en las que se utilizan. Del mismo modo, determinadas partes de la política deben considerarse recursos y protegerse dado que pueden desvelar información sensible.

Un **punto de entrada** es un policy item, una expresión de requisitos, que siempre está accesible para todos aquellos que interactúen con el sistema y que permite comenzar la negociación. Las diferentes combinaciones de requisitos, desde un punto de entrada hasta poder acceder a un recurso, se pueden definir como **rutas**.

Los **atributos** son cualquier información ya sea numérica, de pertenencia a categoría o grupo, ordinal, lógica... utilizada para caracterizar una entidad.

Notación

En este capítulo y en el siguiente se usará esta notación para distinguir los diferentes elementos en las expresiones:

- Los requisitos se expresan con Rq_X, donde la X será un número que lo distinga de los demás, por ejemplo, Rq_1, Rq_8...

- Los policy items se representan con P_X donde X será un número, es decir, P_1, P_2. Se establece una relación unívoca entre los requisitos y los policy items que los especifican, de manera que, por ejemplo, el requisito Rq_3 lo especifica formalmente el policy item P_3, el Rq_8 es especificado por P_8...

- Los recursos que no son credenciales, se representan con R_X donde X será una o varias letras de la A a la Z. Por ejemplo, R_A, R_F...

- Las credenciales se representan con C_X donde X será una o varias letras de la A a la Z: C_A, C_E, C_C.

5.2. Descripción del problema

En la sección 2.1 describíamos las perspectivas que cada una de las entidades involucradas en un proceso de comunicación tiene del control de acceso. En dicha sección se podía observar cómo cada una de las partes, equipo de usuario, proveedor de acceso a la red, proveedor de servicios... tiene unos intereses que no tienen por qué estar en sintonía con los intereses del resto de los participantes. Ahora veremos qué grandes problemas se pretenden resolver con el sistema que se propone:

Aumento del número de tipos de credenciales y políticas

Para gestionar correctamente los recursos y encontrar afinidades entre los participantes, se recurre a mecanismos de control de acceso que, una vez ejecutados, permiten determinar si las partes involucradas tienen derecho a interactuar. Este "derecho" como tal, no se establece de forma universal para todas los participantes, sino que se establece a través de diversos agentes que son los encargados de determinar: qué restricciones de acceso se aplican a cada uno de los recursos y qué requisitos deben satisfacerse para su utilización.

Para sintonizar los intereses de todos los elementos es necesario recurrir a dos mecanismos, uno de ellos es la autenticación y el otro la autorización. El primero permite, a través de relaciones de confianza, identificar unívocamente a la entidad como tal o como perteneciente a un grupo de entidades; el segundo, la autorización, permite identificar los derechos asociados a esa entidad o grupo de entidades.

Si extrapolamos la situación a escenarios donde el número de entidades de cualquiera de los tipos, ya sean clientes o proveedores, aumenta significativamente, parece complicado establecer relaciones de confianza eficientes entre todos ellos sin recurrir a un único proveedor de confianza o a un modelo jerárquico. Típicamente se recurre

a esquemas jerárquicos como PKI/PMI [45][99], que gestionan dichas relaciones de confianza, pero con el problema de que no siempre escalan bien con el número de entidades. Uno de los motivos son los costes de revocación existentes.

Considerar una jerarquía con una única raíz, algo así como una autoridad de certificación a nivel global, que tuviera delegados en un modelo similar al de DNS, es impensable por la imposibilidad de manejar tantas credenciales y consultas. Si la jerarquía estuviera muy ramificada podría funcionar, pero solo podría soportar un tipo de credenciales con sus ventajas e inconvenientes. Por otro lado, pese a que el proveedor de confianza global existiera para autenticación, cada dominio en concreto debería disponer de una infraestructura para la gestión de derechos o privilegios. Esto es necesario dado que es privativo de cada organización o dominio administrativo establecer los permisos de sus usuarios, ya no solo por cuestiones administrativas o de separación de responsabilidades, sino por el alto coste de revocación asociado al hecho de unir autenticación y autorización. La razón es que la autenticación es información relativamente estática y la autorización muy dinámica.

Un escenario con una única fuente de confianza parece imposible de abordar desde el punto de vista práctico y, hasta la fecha, no se ha logrado conseguir. En cambio sí es posible establecer relaciones entre jerarquías, lo que permite que cada dominio disponga de sus propios sistemas permitiendo estableciendo relaciones de confianza entre sus sistemas y los de otro dominio. En cualquier caso, pese a ser posible, o bien exige el uso del mismo sistema (o formato de credenciales) por ambas partes, o se hace necesario disponer de entidades intermedias capaces de traducir de un formato a otro.

La utilización de único sistema o de un conjunto limitado de sistemas requiere un proceso de estandarización y, por descontado, de revisión continua, de modo que se vaya adaptando a las necesidades futuras. Si extrapolamos el caso al de la autenticación en la red, como se discute en el 2.3, recurrir a soluciones globales a niveles inferiores garantiza una bajada del coste; pero aumenta los problemas de actualización y corrección de errores; a la vez que aumentan los riesgos, debido a que se expone una mayor cantidad de recursos cuando se producen fallos (hasta que se encuentran y distribuyen las soluciones).

Si se utilizan varios sistemas y reglas de equivalencias entre sistemas, orquestadas a través de mecanismos de decisión, pese a que en principio parezca que aumentan los costes directos, se aumenta la capacidad de interacción y se disminuyen los fallos dado que existe una cierta redundancia. Al aumentar la capacidad de interacción aumentan los clientes potenciales; y al aumentar la redundancia se evitan caídas continuadas ya que se puede recurrir a otros sistemas de autenticación/autorización.

Además, es necesario considerar la separación de identidades y de roles. Mientras que una persona o dispositivo tiene una identidad en un dominio, por ejemplo cor-

porativo, donde está identificado como trabajador o como perteneciente al mismo; la misma persona o dispositivo no se identifica o autentica de la misma manera en otro tipo de dominios, por ejemplo, en una comunidad, foro o web de entretenimiento. Por otro lado, la misma persona o entidad no tiene por qué utilizar las mismas redes para diferentes tareas, incluso es posible que disponga de varios proveedores de servicio a los que acceda con diferentes identidades. Si además tenemos en cuenta la privacidad del usuario, nada justifica el uso de una identidad global para autenticación.

En los sistemas donde pueden convivir varios mecanismos de autenticación y autorización junto con diversos tipos de credenciales, se pueden considerar como elemento clave las políticas. Las políticas determinan qué hace falta para acceder a un recurso y además, si éstas políticas están escritas en lenguajes extensibles, como es el caso, por ejemplo, de [92] o [100], las políticas hacen las veces de "pegamento" que une todo lo demás.

Gracias a las políticas y la extensibilidad de algunos de los lenguajes de políticas, que permiten el uso de distintos sistemas de autenticación y autorización, queda fuera de toda lógica el recurrir a un único conjunto de credenciales y mecanismos para el control de acceso. Por otro lado, tampoco es lógico suponer que con un único lenguaje de políticas pueden describirse todas las sensibilidades de los recursos. Sobre este hecho se discute en el capítulo 2 y más en concreto en la sección 2.2, donde se ponen de manifiesto las deficiencias de ciertos lenguajes de políticas que no permiten, por ejemplo, expresar separación de tareas. Por todas estas razones hemos de centrarnos en diseñar mecanismos que ayuden a la toma de decisión cuando hay involucradas muchas y diferentes credenciales, así como políticas, con nuevos mecanismos que no dependan de los lenguajes con los que se han escrito ni de la naturaleza de dichas credenciales y políticas.

Necesidad de negociación

Una vez analizado el escenario desde la perspectiva de la existencia de múltiples credenciales y lenguajes de políticas es necesario razonar sobre cómo se aplican las políticas. En algunas ocasiones, como en [101], se han propuesto protocolos en los que se hace pública la política, de forma que el cliente puede determinar si podrá acceder o no al recurso.

Estos sistemas, se puede considerar como de información perfecta. En ellos, los clientes que traten de acceder a un recurso pueden examinar la política para obtener lo que necesitan antes de enfrentarse al mecanismo de control de acceso. Enfoques similares desde la perspectiva del cliente se pueden encontrar por ejemplo en [50], donde se proponen repositorios públicos para el almacenamiento de las credenciales,

que puedan ser consultados por las aplicaciones para comprobar la identidad o los permisos de los clientes.

Si analizamos las relaciones de confianza entre dominios podemos encontrar dos tipos de relaciones: directas, en las que las credenciales de un dominio son reconocidas como válidas en otros dominios, y otras indirectas. Respecto a las indirectas podemos decir que no existe una relación de confianza previa entre dominios pero que se puede llegar a ella tras la presentación de varias credenciales. Es decir, si para acceder a un recurso es necesario la presentación de la credencial A, es posible que aquellos que no disponen de la credencial A como tal, puedan presentar digamos las credenciales B, C y D o las credenciales E y F, dado que en algún lugar de la política en aplicación se considera lo siguiente: $A \approx (B\&C\&D)|(E\&F)$. Por ello una entidad que en primera instancia se considera no confiable, puede llegar a un estado de seguridad o confianza, que le permite acceder a un determinado recurso mediante el intercambio de diferentes credenciales.

Para que esta negociación de confianza sea posible, el cliente debe presentar al servidor todas aquellas credenciales que éste le pida y el servidor debe informar al cliente acerca de qué credenciales debe proporcionar para acceder. Para lograr esto, el servidor podría enviar la política completa al cliente, de esta manera el cliente podría determinar qué credenciales presentar; sin embargo, esta forma de proceder se puede considerar ingenua. Un caso comparable en la vida real, sería acudir a una tienda a comprar un determinado objeto y pretender que el tendero nos proporcione su lista de clientes y cómo y cuánto pagan para conseguir determinados objetos. Es decir, al revelar la política o políticas que gobiernan el sistema, el servidor puede estar exponiendo información sensible de clientes o información que comprometa el sistema a atacantes.

Tampoco sería lógico que el cliente enviara todas las credenciales que el servidor le pidiera sin analizar el riesgo que conlleva liberar ciertas credenciales. Sería como acudir a una tienda, poner sobre el mostrador todo el dinero del que se dispone y decir al tendero "¿que puedo comprar con esto?". El cliente debe tratar de protegerse contra abusos analizando el riesgo de liberar determinadas credenciales y solicitando al servidor alguna credencial en caso de duda. La posibilidad de invertir el rol, pasar de ser cliente a servidor, convierte una negociación de este tipo en P2P, por lo que el modelo cliente/servidor se difumina.

Por tanto, está claro que el servidor debe liberar la política de forma monótona creciente atendiendo al estado de la negociación. En la sección 5.1, se pueden leer algunas definiciones entre las que se encuentran los **policy items**, que son "trozos" de políticas que hacen referencia a un requisito en concreto. La razón de "trocear" las políticas es la de poder liberarlas gradualmente.

Para poder negociar en todas las circunstancias posibles, siempre debe existir uno o varios requisitos que puedan liberarse siempre, que no comprometan el sistema y que denominamos **punto de entrada**. Un punto de entrada puede ser, por ejemplo, una autenticación básica, un reto, un puzzle, en definitiva algo que nos lleve al siguiente paso, porque hemos determinado la identidad o la pertenencia a un grupo de la otra parte; porque dispone de una clave dada; o debido a que tiene interés en continuar y no es un ataque de DoS, dado que ha resuelto un puzzle costoso computacionalmente....

El cliente deberá disponer de un sistema que evalúe el riesgo de comprometer determinadas credenciales para evitar abusos. Por tanto deberá disponer de una serie de **policy items**, asociados a las diferentes credenciales, de forma que, si la situación lo exige, pueda enviarse ese **policy item** a la otra parte para que ésta proporcione una credencial que lo identifique como consumidor de dicha credencial. Son por lo tanto necesarios los protocolos y motores de decisión para negociación de confianza.

Inclusión de información de contexto

Muchos lenguajes de políticas admiten información de contexto. Entre ellos destaca XACML dado que al ser extensible, permite la inclusión de cualquier tipo de dato no previamente considerado. En [46][10] y en [4] se propone utilizar valores de confianza, entre 0 y 1, para realizar el control de acceso. Este sistema de control de acceso basado en confianza llamado TrustAC, añade con éxito estos niveles de confianza a políticas XACML.

Existen otros lenguajes de políticas más complejos y capaces de describir diferentes y complejas sensibilidades que también incluyen información de contexto. La pregunta es ¿Cómo se procesa la información de contexto? En [102] se discute acerca de como procesar variables de entorno estables para mejorar la eficiencia de un sistema basado en XACML, pero en el caso que nos ocupa, no nos limitamos a un solo sistema, por lo que es necesario realizar ciertas consideraciones. Bajo estas líneas se muestra una política de ejemplo, escrita en XACML, en la cual se establece como condición de acceso a un recurso, que la variable “variableContexto1” tenga un valor menor de 0.6:

```
<Policy PolicyId="ExamplePolicy"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:
        rule-combining-algorithm:permit-overrides">
  <Target>
    <Subjects>
      <AnySubject/>
    </Subjects>
    <Resources>
      <Resource>
        <ResourceMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
```

```
          http://mirecurso.com
          </AttributeValue>
          <ResourceAttributeDesignator DataType="XMLSchema#anyURI"
          AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <AnyAction/>
    </Actions>
  </Target>
  <Rule RuleId="ReadRule" Effect="Permit">
    <Target>
      <Subjects>
        <AnySubject/>
      </Subjects>
      <Resources>
        <AnyResource/>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
            read
            </AttributeValue>
            <ActionAttributeDesignator DataType="XMLSchema#string"
              AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition FunctionId="urn:oasis:names:tc:xacml:1.0:function:double-less-than">">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:double-one-and-only">
        <SubjectAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#double"
                                    AttributeId="variableContexto1"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#double">0.6</AttributeValue>
    </Condition>
  </Rule>
</Policy>
```

En este caso no hay duda alguna al procesar la regla, dependiendo de dicha variable de contexto se establece la posibilidad de acceder al recurso o no. XACML permite expresar cómo afecta la información de contexto a cada recurso, pero la evaluación de dicha información de contexto arroja un resultado de decisor duro, es decir, por encima de un umbral se permite el paso, por debajo no. Consideremos la existencia de múltiples variables de contexto; contempladas por más de una regla contenida en diferentes políticas, aplicadas sobre el mismo recurso; y la posibilidad de que no todas afecten por igual a los diferentes recursos: ¿Cómo se podría manejar este tipo de situaciones?

Puede ser interesante establecer diferentes calidades de servicio dependiendo de qué valores tienen determinadas variables de entorno. Para ello, se podría complicar

la política del ejemplo y otras que afectaran al recurso, creando nuevos recursos con distintas calidades de servicio (http://mirecurso.com/index.html?calidad=1) o se podría incluir la calidad de servicio directamente en aquellas políticas que lo soportaran. El problema es que esto es inmanejable cuando existen varias políticas, dado habría que obligar a todas esas políticas a considerar los distintos niveles de calidad, aumentando el número de reglas: si se consideran **n** nuevas calidades de servicio para un recurso, sobre el que se establece una restricción respecto a una variable de entorno en concreto, habría que incluir en cada una de las políticas que gobiernan el recurso y que consideran esa variable de entorno, **n** nuevas reglas.

Vamos a considerar un ejemplo. Supongamos tres variables de entorno que afectan a una decisión y que tienen valores comprendidos entre 0 y 1. Supongamos por otro lado que se establece en 0.8 el valor mínimo de cada una de las variables para conceder acceso a un recurso. Si se presenta dos casos en los que las variables tienen los valores {0.1,0.1,0.1} y {0.8,0.9,0.7}, ambos serían tratados igual por el sistema, es decir, se denegaría el acceso al recurso. Si se quisiera dotar de cierta tolerancia al sistema, de forma que, a entidades muy cerca de reunir las condiciones necesarias se les permitiera acceder al servicio mientras que no se comprometiera el recurso, considerando, por ejemplo, que una pequeña diferencia no tiene porqué entrañar un riesgo elevado; sería necesario incrementar el número de reglas de la política y sería necesario hacerlo cada vez que se considerara una nueva variable de entorno.

Las razones analizadas en esta sección nos llevan a la conclusión de que la información de contexto debe ser considerada en global, atendiendo a las restricciones de las políticas, pero permitiendo al sistema que lo gestiona establecer o medir el riesgo en conjunto y tomar decisiones en consecuencia.

Comprensión de riesgos por parte del usuario

Típicamente los detalles de seguridad, y aquí se incluyen credenciales, mecanismos de autenticación y autorización, políticas, etc... son desconocidos para el usuario medio, lo cual le incapacita para evaluar el riesgo de sus acciones.

Pongamos un ejemplo sencillo. Supongamos que al acceder una página web el usuario recibe una alerta sobre la confiabilidad del certificado dado que éste está mal construido o no identifica correctamente a la máquina a la que quiere acceder. Al acceder repetidas veces a dicha página web, la alerta se torna incómoda para el usuario, por lo que decide añadir el certificado al almacén de entidades confiables. El almacén de entidades confiables (Trusted Root Certificate Authorities, en adelante TRCA) se comparte con todas las aplicaciones del sistema que utilizan PKI. El usuario está, por lo tanto, confiando en todas las páginas web cuyos certificados han sido emitidos

por una autoridad de certificación que no es necesariamente confiable, dado que el certificado se encuentra en el TRCA. Si de forma gráfica, o más comprensible, se pudiera mostrar al usuario todos los servicios que se ven afectados por su decisión, éste podría comprender el riesgo de su acción al entender que dicha decisión **no es aislada**, sino que afecta a otros recursos que no quiere comprometer.

En la sección 2.1 se discute razonadamente sobre cuáles son los medios más efectivos de comunicación de riesgos. En dicha sección se llega a la conclusión que, dependiendo del modelo mental de dicho usuario y grado de su experiencia, unos mecanismos de comunicación de riesgos son más efectivos que otros, pero que en cualquier caso es el modelo físico (objetos presentes en la vida cotidiana) es el más adecuado si es necesario generalizar. Por otro lado también es muy eficiente expresar riesgos mediante comparaciones con elementos similares.

Por esta razón, se tratará en lo posible de comunicar el riesgo al usuario de forma comprensible gráficamente, logrando así una mejor comprensión de los resultados indeseables de determinadas acciones. Para ello se deberá presentar al usuario, el espacio de decisión de forma comprensible para él.

5.3. Arquitectura

Para tomar decisiones de control de acceso es necesario tener en consideración toda la información disponible incluyendo el contexto en un instante de tiempo. Este razonamiento se empleó en el capítulo anterior, demostrando los beneficios de incluir el contexto en las decisiones para la selección de red; sobre la aplicación concreta al caso que nos ocupa, ya se ha discutido en 5.2.

Consideremos un dispositivo móvil cuyos recursos se gobiernan por un conjunto de políticas escritas por el usuario, el administrador de dominio de la compañía para la que trabaja el usuario y el proveedor de servicios UMTS. Estas políticas pueden gestionarse a través de diferentes **motores de control de acceso (ACEs)** debido a que, como se expuso en la sección 5.2, pueden estar escritas en diferentes lenguajes. Cada ACE procesa la política o conjunto de políticas, que es capaz de procesar, extrayendo los **requisitos**; la restricciones que impone el contexto... generando posteriormente los **policy items**. Los **requisitos** que se registran en el motor de decisión, permiten que éste tenga en cuenta dicho requisito; sepa cómo encontrar el policy item que lo describe, así como qué ACE lo debe procesar; y le permite saber a qué recursos afecta dicho **requisito**. Los **policy items**, que son la expresión formal de un requisito, permiten que otros sepan cómo proceder para satisfacer un requisito.

El número de requisitos dependerá de cada política y todos ellos, deben registrar-

se en el motor de decisión. Un requisito, tal y como se entiende en el sistema, puede requerir la liberación de una única credencial o varias. Cómo discernir cuándo una parte de una política da lugar a un requisito o a varios dependerá del procesamiento que se haga de la misma; sin embargo, no es uno de los objetivos de la investigación. En cualquier caso, se deben aplicar ciertos principios de eficiencia tratando de encontrar puntos en común, de forma que, requisitos que se repitan varias veces afectando a varios recursos, se aíslen en un único **requisito** para evitar redundancia.

En la figura 5.1 se presenta un esquema de lo que podría ser una extracción de datos (recursos, requisitos, restricciones de contexto) de una política XACML. En la figura se aprecia que, de la información contenida en el elemento <Resources>, se extraen los recursos R_A, R_B y R_C. Más abajo en la política tenemos la expresión del requisito como tal, en forma de regla y, aún más abajo, las restricciones en cuanto al contexto que se aplican a los recursos. En el ejemplo, se registra un requisito en el motor de decisión , Rq_1, que afecta a tres recursos (R_A, R_B y R_C) y se complementa con información de contexto, Ctx1, extraída de forma separada (no se incluye como parte del requisito), de forma que pueda procesarse globalmente.

En dicha figura, se puede ver además, cómo contribuyen con más información otras políticas, añadiendo requisitos (Rq_2 y Rq_8) y condiciones de contorno (Ctx2 y Ctx3). Por otro lado, se aprecia cómo ciertos recursos se ven afectados por algunas variables de entorno y otros no, por lo que de nuevo volvemos a encontrarnos con un problema en el cual los elementos están definidos por **distinto número de parámetros**.

Al recopilar de esta manera la información, el motor de decisión no necesita entender los distintos formatos de políticas o credenciales, simplemente maneja requisitos y recursos. Es más, sólo manejará elementos descritos con diferentes atributos y los procesará con diferentes pesos. Los requisitos, extraídos de diferentes políticas escritas en diferentes lenguajes, podrán ser combinados utilizando un único motor de decisión. Las políticas pueden estar escritas en cualquier lenguaje y puede que sean muy generales, por ejemplo, para cubrir todos los posibles casos y usuarios dentro de un dominio.

Los ACEs procesan las políticas y extraen los requisitos. Se requiere, por tanto, que los motores de control de acceso subyacentes sean capaces de extraer la parte de cada política que se aplica a su dispositivo durante la instalación y en las ocasiones en las que se cambie la política. El usuario puede participar estableciendo sus preferencias durante la instalación, pero éstas preferencias de usuario afectan únicamente a aquellos parámetros que son susceptibles de modificación por parte del usuario, es decir, no puede alterar las políticas de otros proveedores.

Por otro lado, los policy items son un recurso más a proteger, ya que revelarlos pue-

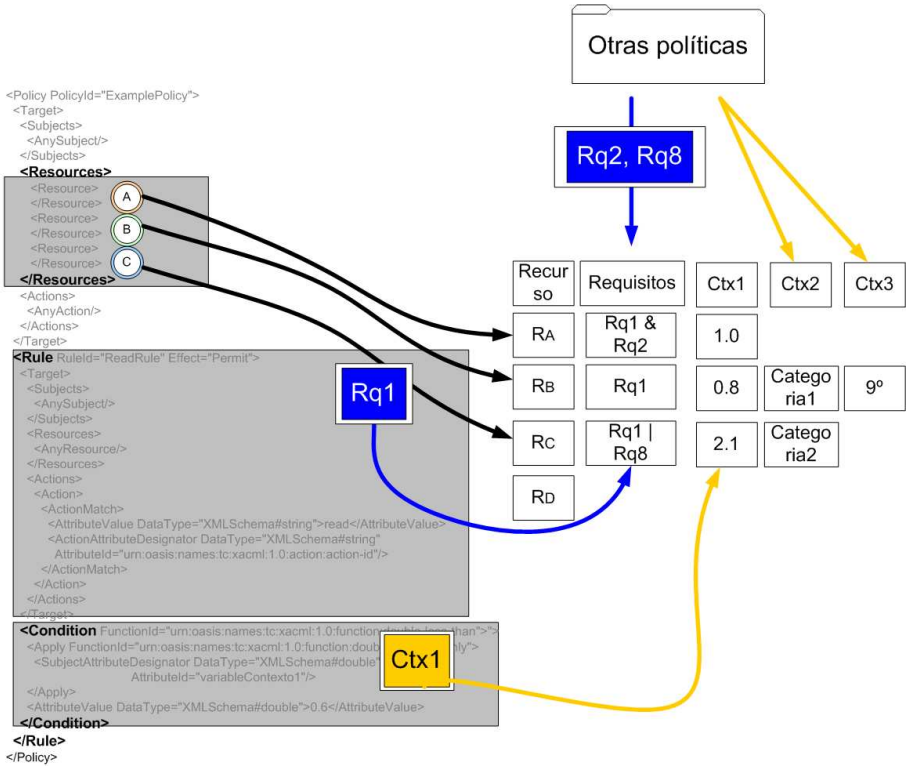


Figura 5.1: Ejemplo de la extracción de información correspondiente a recursos, requisitos e información de contexto, de una política XACML, para su procesado en el motor de decisión de negociación de confianza

de suponer un riesgo. Los recursos a proteger en un dispositivo serán por tanto:

- Servicios: como pueden ser servidores web, de intercambio de ficheros, ftp...
- Policy Items: que son expresiones formales de un requisito, extraídos de una política y que, al ser susceptibles de revelar información comprometedora del sistema o de sus clientes, requieren ser protegidos por requisitos de forma que su liberación sea escalonada.
- Credenciales: por cuestiones de privacidad.

Esto nos lleva a que las políticas tendrán que disponer de información acerca de en qué orden se aplican los requisitos, de forma que los policy items se puedan liberar de forma gradual. Pese a que no es uno de los objetivos de la investigación, bastaría con especificar una sensibilidad para cada uno de los requisitos con el mayor grado

de detalle posible. Esta sensibilidad podría ser expresada con un número entre 0 y 1, siendo 0 sensibilidad mínima y 1 máxima. Así sería posible determinar el orden de aplicación de los mismos.

La figura 5.2 muestra la arquitectura, poniendo de manifiesto las relaciones entre los ACEs y el motor de decisión, y cómo éste último media en todas las interacciones.

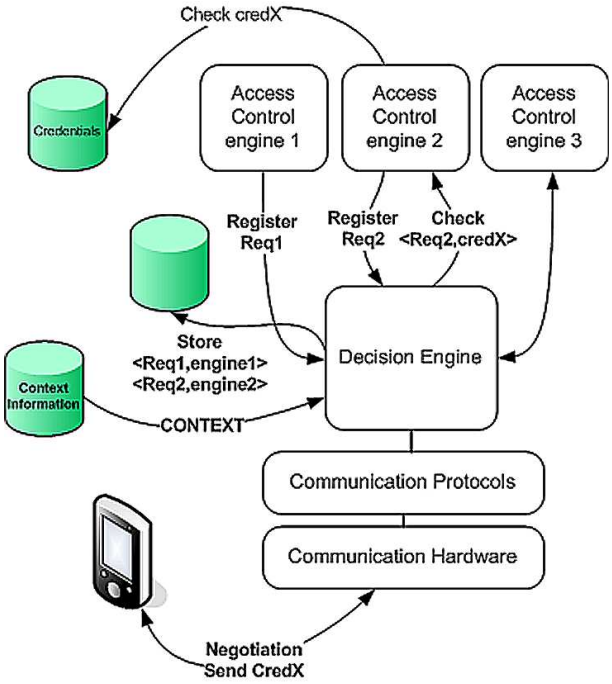


Figura 5.2: Arquitectura del motor de decisión para negociación de confianza.

5.4. Descripción del algoritmo

En esta sección describiremos en qué consiste el algoritmo utilizado para tomar decisiones durante un proceso de negociación de confianza.

Elección del algoritmo

Como se razonó en la sección anterior (sección 5.3), el problema final consiste en calcular las similitudes entre entidades en base a un número de parámetros variable. Si bien es cierto que en el capítulo 3 se resolvió un problema similar, el problema de la

negociación de confianza requiere un tratamiento distinto de algunos de sus parámetros como se razonará más adelante.

El algoritmo seleccionado para realizar un análisis de datos, que nos permita resolver un problema cuyas particularidades se han razonado en la sección 5.2, es MDS. Este algoritmo se elige sobre otros por dos grandes motivos. El primero de ellos obedece a los mismos razonamientos realizados en la sección 2.4, donde se analizaron las ventajas e inconvenientes de diferentes técnicas de análisis estadístico, concluyendo que MDS era el más acertado para determinados objetivos.

Son muchas las ventajas de MDS frente a otros algoritmos dado que es el único capaz de reducir la dimensionalidad de un problema manteniendo su conceptualización, que es el objetivo perseguido; calculando similitudes entre entidades y no entre las variables que los describen, como hace el Análisis de Componentes Principales (PCA). MDS puede procesar cualquier tipo de dato, entre los que tendremos métricos, datos de categoría o pertenencia a grupos y ordinales. En cambio, Análisis de Factores (FA) requiere matrices de covarianza y el PCA requiere que todos los elementos sean descritos con el mismo número de variables. Además, la versión iterativa de MDS seleccionada, ALSCAL, tiene como ventaja la capacidad de funcionar en ausencia de datos.

El segundo motivo es el de reutilización de recursos existentes. En el capítulo 3 se ha propuesto la utilización del mismo algoritmo para resolver el problema de la selección de red, por lo que la misma implementación, así como los mecanismos de representación gráfica, pueden reutilizarse con otra finalidad.

Pese a que MDS se posiciona como un algoritmo muy adecuado para este problema, vamos a discutir sobre la necesidad o no de recurrir al análisis estadístico para la gestión de las negociaciones de confianza. Habitualmente las políticas se recorren linealmente hasta encontrar los requisitos y condiciones aplicables a un recurso. En el caso de políticas binarias, como las que pueden estar contenidas en un certificado de clave pública [50], el problema se agrava con la decodificación ASN.1 y con la expresión mediante OIDs (Object Identifiers). Las políticas escritas en XML son más accesibles dado que permiten incluso la lectura directa. Estas últimas pueden ser recorridas mediante sentencias XPath [103], utilizando DOM (Document Object Model) e incluso XML-DB, que permite usar SQL para realizar búsquedas en documentos XML.

Estos mecanismos de exploración de ficheros XML no son adecuados para nuestros propósitos dado que el motor de decisión tendría no solo que recorrer las diferentes políticas que afectan a los recursos, cada vez que se produjese un cambio en la negociación, sino que debería ser capaz de procesar diferentes lenguajes, complicándose excesivamente y dificultando su extensión a otros lenguajes de políticas.

Asumiendo que es necesario independizar el motor de decisión de los diferentes

lenguajes de políticas, podemos dar por sentado que dicho motor dispone de la información en forma de **recursos**, **requisitos** y **policy items**. Sin embargo, sobre esos datos, pueden utilizarse otros algoritmos muy adecuados para la toma de decisiones, que son aquellos usados para crear árboles de decisión, como ID3 [104] o C4.5 [105] (que mejora el rendimiento de ID3).

El problema fundamental de los árboles de decisión, que es además ortogonal a nuestros objetivos, consiste en la imposibilidad de comparar entidades entre si. Por otro lado, tampoco ayudan en la presentación del problema al usuario. Si utilizamos ID3 para generar árboles de decisión[1] obtenemos árboles simples, aunque no necesariamente los más pequeños, dado que utiliza un algoritmo heurístico basado en el principio de la Navaja Occam. Estos árboles se calculan utilizando como criterio para la elección del mejor atributo, aquel que proporciona mayor ganancia de información, por ello el orden en la aplicación de los requisitos no obedece a criterios de sensibilidad y no sería útil para conseguir una liberación gradual de la política.

Pese a ello, podría considerarse cambiar el criterio de elección del atributo de "aquel que proporciona mayor ganancia de información" a "aquel que tiene menor sensibilidad", consiguiendo así una liberación gradual de la política; pero por otro lado, no permitiría el análisis en conjunto de la situación, sino que **sería necesario generar un árbol por recurso y calcular el resultado para cada uno de ellos por separado**. Además los árboles se complican excesivamente con el incremento de unos pocos parámetros. Vamos a analizar un pequeño ejemplo, en la tabla 5.1 tenemos el resultado de aplicar todas los posibles valores a una expresión de requisitos que controla el acceso a un recurso.

La figura 5.3 representa el árbol resultante de aplicar el algoritmo ID3 a la tabla 5.1. Como se puede ver en dicha tabla, se han asumido tres valores para la variable de contexto, dado que no podrían utilizarse valores continuos a menos que se utilizaran umbrales. Como puede observarse, el árbol es bastante complejo para la tratarse de un ejemplo de tan solo cuatro atributos, lo que lo hace poco adecuado para comunicar riesgos al usuario o para que pueda entender el espacio de decisión.

De modo que, para analizar el riesgo, nos vemos obligados a recurrir a técnicas de procesado de datos que no requieran un modelo común y sean capaces de trasladar conceptualmente las similitudes y diferencias de los elementos involucrados a un espacio dimensionalmente acotado, para poder compararlos sin que en el proceso se alteren dichos conceptos. Esto es posible dado que MDS permite una traslación conceptual a espacios con un menor número de variables.

[1] no resta generalidad a la discusión el hecho de analizar únicamente ID3, dado que C4.5 es una versión extendida de ID3 con mejor rendimiento

R1	R2	R3	entorno	Resultado
0	0	0	a	denegar
0	0	1	a	permitir
0	1	0	a	denegar
0	1	1	a	permitir
1	0	0	a	permitir
1	0	1	a	permitir
1	1	0	a	permitir
1	1	1	a	denegar
0	0	0	b	permitir
0	0	1	b	permitir
0	1	0	b	permitir
0	1	1	b	denegar
1	0	0	b	permitir
1	0	1	b	denegar
1	1	0	b	permitir
1	1	1	b	permitir
0	0	0	c	denegar
0	0	1	c	permitir
0	1	0	c	permitir
0	1	1	c	denegar
1	0	0	c	denegar
1	0	1	c	permitir
1	1	0	c	permitir
1	1	1	c	denegar

Tabla 5.1: Tabla de valores para la construcción de un árbol de decisión ID3. La tabla recoge el resultado de aplicar todas las posibles combinaciones de entrada a una expresión que gobierna el acceso a un recurso.

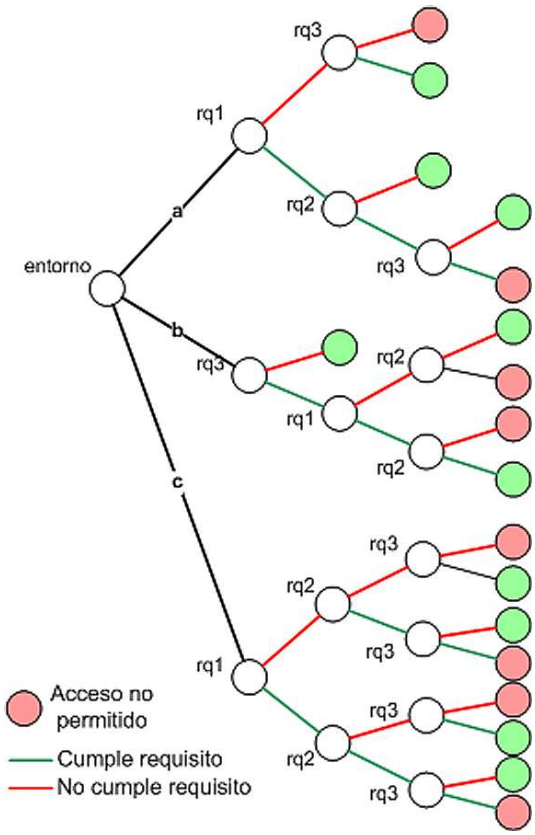


Figura 5.3: Árbol ID3 obtenido. Muestra que al tener en cuenta tan solo una variable de contexto con varios valores posibles, se complica mucho el árbol de decisión.

Procesado de los atributos

Para poder evaluar el riesgo asociado a liberar una recurso, y así determinar si se concede el acceso o no, procederemos al cálculo de las similitudes entre todos los elementos participantes en la negociación. Dependiendo del grado de similitud entre dos entidades, calculado sobre el conjunto de todos sus atributos, dichas entidades estarán más o menos próximas. Por esa razón, si se establece como punto de referencia un elemento que recoja el estado actual de la negociación, el riesgo se puede medir como una distancia entre puntos. El concreto, el riesgo se medirá como una distancia sobre un plano de dos dimensiones respecto a un punto de referencia. Ese punto de referencia es el **estado de la negociación**.

Lógicamente, los atributos del punto de referencia (estado de la negociación) varían durante la negociación, recogiendo los avances de la misma, como pueden ser los requisitos que se van satisfaciendo o los cambios en los atributos que referencian el

contexto. Por tanto, siendo capaces de calcular las similitudes entre elementos, podemos determinar qué recursos están cerca del **estado de la negociación** y cuáles están lejos conceptualmente. Una vez se ha realizado ese calculo, se puede afirmar que **cuanto más cercano** se encuentra un recurso al elemento que describe el estado actual de la negociación, **menor riesgo entraña su acceso**.

Las (di)similitudes entre pares de elementos se pueden calcular con las siguientes ecuaciones, dependiendo de los atributos usados para realizar la comparación:

$$\delta_{i,j,\alpha} = \frac{|u_{i,\alpha} - u_{j,\alpha}|}{max(u_\alpha) - min(u_\alpha)}, \text{ para cuantitativos} \tag{5.1}$$

$$\delta_{i,j,\alpha} = \frac{|rank(u_{i,\alpha}) - rank(u_{j,\alpha})|}{max(rank(u_\alpha)) - 1}, \text{ para ordinales} \tag{5.2}$$

$$\delta_{i,j,\alpha} = \begin{cases} 0 : u_{i,\alpha} = u_{j,\alpha} \\ 1 : \text{resto} \end{cases}, \text{ para pertenencia} \tag{5.3}$$

donde $u_{i,\alpha}$ es el valor del α^{esimo} atributo para el elemento i. Se pueden considerar diferentes tipos de datos: cuantitativos, como valores de confianza [10] y distancias [87]; ordinales para QoS y diferenciación de servicios; pertenencia a grupos (para distinguir los tipos de credenciales).

Además de las expresiones anteriores, que son de uso habitual en MDS, nos vemos en la obligación de proponer otras que permitan la inclusión de atributos con características especiales. Estos atributos los clasificaremos en dos grupos:

- **Atributos con umbral**: Como ya se discutió en el capítulo 3, algunos atributos necesitan compararse con un umbral antes de compararse con el mismo atributo de otra entidad. Este tipo de atributos se definen y formulan en la sección 3.5.

- **Atributos consistentes en expresiones lógicas**. Los requisitos que afectan a un recurso se combinan mediante operadores lógicos, por ejemplo, $(Rq1\&Rq2)\&(Rq3|Rq4)$. Con independencia de dicha asociación lógica, cada uno de los requisitos se considera como un atributo separado, por lo que debe ser posible establecer similitudes entre entidades en base a un requisito.

  Como es lógico, no es adecuado calcular la dependencia simplemente en base a si las entidades dependen o no de dicho requisito. En su lugar se tendrá en cuenta no solo el hecho de que el acceso al recurso dependa de un requisitos en concreto, sino del resultado de la evaluación de todos los atributos combinados mediante operadores lógicos. Estas combinaciones lógicas de requisitos, como el

lector habrá podido suponer, no permiten una evaluación directa para el cálculo de similitudes entre entidades, mediante las ecuaciones habituales de MDS.

De hecho no permiten el cálculo de la similitud, a menos que se contextualicen, dando valores a las variables que componen las expresiones. Por ese motivo definimos las siguientes ecuaciones y las integramos en el motor de decisión incorporando un nuevo tipo de atributo.

$$\delta_{i,j,Rq_\alpha} = \begin{cases} 0: si\ f_{i,Rq_n} \neq f(Rq_\alpha) \\ 0: si\ f_{j,Rq_n} \neq f(Rq_\alpha) \\ eval(f_{i,Rq_n})\ \&\ eval(f_{j,Rq_n}) : otros \end{cases} \tag{5.4}$$

donde f_{i,Rq_n} es la función lógica que combina los diferentes requisitos a satisfacer para acceder al recurso i. Cada requisito, se considera separadamente como un atributo, con independencia de su asociación lógica con el resto de requisitos. Por tanto, para determinar si dos atributos pueden compararse en base a la satisfacción de un requisito en concreto, Rq_α, verificamos si se cumple $f_{i,Rq_n} \neq f(Rq_\alpha)$, es decir, verificamos si f_{i,Rq_n} contiene en su expresión al requisito Rq_α.

Si f_{i,Rq_n} contiene al requisito Rq_α, que es el atributo sobre el que se están evaluando las similitudes, necesitamos calcular el valor de esta función y para ello la evaluamos utilizando $eval(f_{i,Rq_n})$ que devuelve el resultado de evaluar la función lógica usando como parámetros los requisitos que constan como satisfechos en el elemento **Neg**, que recoge el estado actual de la negociación.

Por lo tanto, dos elementos i y j son idénticos respecto al atributo Rq_α si en un determinado momento, su acceso depende del requisito Rq_α y además la evaluación de sus requisitos ($eval(f_{i,Rq_n})$), utilizando los datos contenidos en el estado de la negociación en ese momento, arroja el mismo resultado para ambos.

Cálculo de los pesos

Otro elemento clave en el modelo son los vectores de pesos, con los que se consigue una mayor particularización de los datos. El procedimiento para ponderar las disimilitudes se describió en la sección 2.4.

En este caso existe una diferencia importante respecto al mecanismo de selección de red. En la selección de red, los pesos indicaban la importancia que la política seleccionada para el dominio actual daba a los diferentes atributos. En este caso los pesos se utilizan para maximizar las diferencias entre entidades. Vamos a ver dos ecuaciones para calcular los pesos:

1. Se ponderarán tanto los atributos que corresponden a requisitos, como aquellos que corresponden a variables de entorno, pero con ciertas particularidades. La expresión matemática es la siguiente:

$$w_\alpha = \begin{cases} \alpha\ es\ requisito: \begin{cases} 0 : si\ \alpha\ satisfecho \\ \frac{k}{ua} : si\ \alpha\ no\ satisfecho \end{cases} \\ \alpha\ es\ contexto: \begin{cases} 0 : si\ \alpha\ indefinido\ en\ estado \\ \frac{k}{ua} : si\ \alpha\ definido\ en\ estado \end{cases} \end{cases} \tag{5.5}$$

   donde w_α es el peso para el atributo α^{esimo}. El sistema en la primera ronda, utiliza el mismo valor para los atributos no especificados (aquellos indefinidos). Es necesario matizar qué es un atributo no especificado: una variable de contexto puede estar no especificada, pues no existe un modelo común; pero un requisito no, puesto que de ello depende el acceso al recurso. Si un requisito no ha sido satisfecho hasta el momento, se le da valor 0 y se considera indefinido para el cálculo de pesos.

   El sistema mantiene constante la suma de los pesos, de forma que, a medida que la otra parte proporciona credenciales y la negociación comienza a definirse, podamos distribuir uniformemente el valor entre los atributos restantes, maximizando diferencias de la siguiente forma (siendo k una constante y ua el número de atributos sin definir):

   - Si el atributo se refiere a un requisito y no ha sido satisfecho no se puede proporcionar acceso al recurso. Por esa razón le damos al peso correspondiente el valor de $\frac{k}{ua}$.
   - En cambio, si el atributo representa un requisito que ha sido satisfecho, el peso que se le da es de 0.
   - Al resto de los atributos, que no son requisitos, es decir representan variables de contexto, se les da el valor de 0 si no están definidos y $\frac{k}{ua}$ en caso contrario.

   Esta expresión considera las variables de contexto como **informativas**. Al proporcionarles el valor de 0 en los casos en los que no están definidas, se relajan las restricciones del contexto no especificadas en el estado de la negociación, entendiendo que el sistema las desprecia en favor de otras. Como si la información de contexto fuera accesoria.

2. En este otro caso se ponderarán tanto las variables de contexto definidas, como las que no lo están. La expresión matemática es la siguiente:

$$w_\alpha = \begin{cases} 0 : \text{si} \alpha \text{definido en el estado o es un requisito satisfecho} \\ \frac{k}{ua} : \text{si} \alpha \text{ indefinido en el estado o es un requisito no satisfecho} \end{cases} \quad (5.6)$$

donde w_α es el peso para el atributo α^{esimo}. El sistema en la primera ronda, utiliza el mismo valor para los atributos no especificados, sean requisitos o no, de forma que se maximicen las diferencias. El sistema mantiene también constante la suma de los pesos.

Esta expresión considera las variables de contexto como **vinculantes**, de forma que en algunos casos, la indefinición de determinadas variables de contexto, pueda denegar el acceso. k una constante y ua el número de atributos sin definir.

Procesado de los datos en el tiempo

Una vez las disimilitudes se han calculado, éstas se ponderan para obtener una matriz de disimilitudes ponderada mediante las expresiones descritas en la sección 2.4. El algoritmo consta de dos fases bien diferenciadas que son: el proceso de inicialización, que prepara el sistema; y el proceso estacionario, que va recalculando las disimilitudes y pesos para poder tomar decisiones.

El proceso de inicialización se ejecuta cuando se utiliza el sistema por primera vez o cuando varían las políticas, dado que el sistema debe recoger dichos cambios. El proceso de inicialización debe ser realizado en su mayoría por los Motores de Control de Acceso o (ACEs). Durante el proceso de inicialización (figura 5.4, se realizan las tareas que se enumeran a continuación:

1. Extracción de información: Los ACEs extraen los requisitos, policy items e información de contexto de las políticas, como se detalla en la sección 5.3.

2. Determinación de la sensibilidad de los policy items: Para conseguir liberar de forma gradual los policy items, los ACEs deben proporcionar detalles acerca de la sensibilidad de los mismos, de modo que el motor de decisión pueda determinar la precedencia.

3. Registro de la información en el sistema.

4. Pasar a modo de espera: en este modo lo ACEs se quedan a la espera de ser consultados y sólo actuarían para determinar si un requisito se ha cumplido o no. El motor de decisión es el encargado de formular estas consultas a los ACEs. Si

se realizan cambios en las políticas, los ACEs deberán registrar los cambios en el motor de decisión.

Figura 5.4: Tareas del proceso de inicialización del motor de decisión para negociación de confianza.

El proceso estacionario se ejecuta ante cualquier evento detectado en el sistema y que afecte a la toma de decisiones. Estos eventos comprenden: recepción de solicitudes de acceso a servicios desde fuera o desde dentro del dispositivo (para que sea gestionado por el motor); cambios en los requisitos satisfechos durante una negociación; cambios en la información de contexto. Este proceso se lleva a cabo por el motor de decisión, el cual redirige a los ACEs las consultas que sean necesarias, ya que no es su cometido verificar que los requisitos se hayan cumplido. Las tareas que se realizan en este proceso (figura 5.5) se enumeran a continuación:

1. Inclusión del estado: El motor de decisión incluye un elemento adicional que incluya el estatus de la negociación, así como datos de contexto.

2. Generar los pesos: dependiendo de la configuración del motor se generan los pesos como se discutió en la sección 5.4.

3. Generación de la matriz de disimilitudes.

4. Ejecutar MDS ALSCAL resolviendo para dos dimensiones (proporciona buenos resultados de ajuste).

5. Actuar o esperar reacción de la otra parte: dependiendo de cómo estén dispuestos los elementos en el espacio de decisión, se podrían realizar diferentes acciones, desde solicitar más información a la otra parte; enviar credenciales para satisfacer demandas de la otra parte; o simplemente esperar reacción de la otra parte.

Figura 5.5: Tareas del proceso estacionario del motor de decisión para negociación de confianza.

Consideraciones acerca del riesgo

Cuando se ha realizado la traslación conceptual a un espacio visible mediante MDS podemos presentar al usuario el espacio de decisión. Mediante una representación gráfica el usuario es consciente del riesgo que entrañan ciertas decisiones, como se ha discutido con anterioridad en este capítulo. Por otro lado, es necesario definir un límite para el riesgo aceptable, que puede representarse mediante un círculo que llamaremos **circulo de riesgo**, cuyo centro es el punto que representa el estado y cuyo radio depende del contexto.

Se puede permitir el acceso a los recursos dentro del circulo ya que el riesgo se considera aceptable. ¿Pero como definimos ese límite? Claramente un enfoque muy restrictivo obligaría a permitir el acceso a los recursos sólo si su distancia al punto es 0.0. El problema es que, pudiendo estar algunos de los atributos de la negociación sin especificar, sería complicado conseguir que la distancia fuera 0.0.

Una posible solución sería considerar que **el riesgo varía** dependiendo del contexto de la siguiente manera: cuanto menos definida está una negociación, más atributos están sin definir, por lo que mayores son las disimilitudes y por tanto, las distancias entre los recursos y el status son mayores, razón por la cual mayor riesgo puede ser asumido (y por lo tanto mayor puede ser el circulo) ya que será más difícil encontrar puntos dentro del circulo. Para calcular el radio del circulo de riesgo, se propone la siguiente ecuación, que será analizada durante la validación:

$$radius = \begin{cases} 0 \; si \; ua < uaMin \\ (\frac{ua}{attrNum}) * (\frac{maxDist}{attrNum}) \; otros \end{cases} \tag{5.7}$$

donde $maxDist$ es la distancia máxima, $attrNum$ el número de atributos, ua el número de atributos sin especificar y $uaMin$ el mínimo número de atributos que de-

ben estar especificados para considerar un radio diferente a 0.

Aunque se considere una cierta tolerancia al riesgo en aras de una mayor flexibilidad, nunca jamás pueden comprometerse recursos. Se pueden plantear dos escenarios básicos para gestionar el riesgo:

1. En situaciones en las que muchos de los atributos están indefinidos, ya sea por ausencia de mecanismos captadores de dicha información o bien porque el sistema no considera oportuno utilizarla, se puede recurrir a la utilización de **círculos de riesgo**. Al estar el contexto muy indefinido, existe el riesgo de que el motor de decisión no pueda distinguir entre dos recursos cuyo acceso requiera satisfacer los mismos requisitos pero cuyas restricciones respecto al contexto sean muy diferentes, permitiendo el acceso a los dos.

   Para evitarlo, el motor de decisión, puede utilizar una lista que llamaremos de **"elementos alcanzables"** donde se excluyan aquellos cuyas condiciones de contorno sean, por ejemplo, muy distintas al más cercano al estatus, para evitar que se comprometan recursos hasta una revisión del sistema.

2. En las situaciones en las el problema esté muy bien especificado salvo, por ejemplo, uno o dos atributos, puede ocurrir que existan dos recursos prácticamente idénticos en todo los atributos especificados, pero que se diferencien radicalmente en los no especificados. En ese caso, si utilizáramos la segunda manera de calcular los pesos (sección 5.4), al incrementarse el peso de dichos atributos no especificados, se impediría el acceso a los dos recursos. En estos casos, se puede pedir al usuario que ayude a definir el contexto para evitar la **disonancia cognitiva**[2].

[2] El concepto de disonancia cognitiva, en Psicología, hace referencia a la tensión o desarmonía interna del sistema de ideas, creencias, emociones y actitudes (cogniciones) que percibe una persona al mantener al mismo tiempo dos pensamientos que están en conflicto, o por un comportamiento que entra en conflicto con sus creencias. Es decir, el término se refiere a la percepción de incompatibilidad de dos cogniciones simultáneas.

CAPÍTULO

6 Validación mediante simulación del motor de decisión para negociación de confianza

En este capítulo mostraremos el resultado de simulaciones que demuestran la validez de la solución. Las simulaciones se han realizado mediante scripts de Matlab, utilizando software compilado con C/C++ para acelerar algunas tareas y obtener medidas de rendimiento reales.

En la simulación, mostramos cómo diferentes requisitos, extraídos de diferentes políticas escritas en diferentes lenguajes, pueden ser combinados utilizando un único motor de decisión.

Para ilustrar el mecanismos utilizaremos un ejemplo: un caso en el que los ACEs extraen cuatro policy items, que son la expresión de los requisitos, que llamaremos P_n con n variando de 1 a 4. El dispositivo tiene varios requisitos llamados Rq_n con n variando de 1 a 4, que se corresponden con los policy items. Los policy items expresan requisitos por lo que se pueden enviar a la otra entidad para que ésta pueda averiguar qué tiene que hacer para satisfacer ese requisito. En este ejemplo, los recursos se nombran como C_n con n de A a E si se trata de credenciales (son también recursos) y como R_n con n de A a H para el resto de recursos. La tabla 6.1 ilustra el ejemplo. Por otro lado, los policy items P_n que expresan requisitos, son también considerados como recursos y protegidos por requisitos, obligando así a que se liberen gradualmente.

Como se discutió en el capítulo anterior, a medida que nos movemos a un mundo más cambiante, con mayor movilidad y disponibilidad de servicios, se hace más necesario considerar otros atributos para el control de acceso además de los propios de seguridad. En este capítulo demostraremos cómo es posible combinar toda esa información en un solo motor de decisión. Por ello, en el modelo no solo se consideran atributos de seguridad sino otras propiedades de los recursos.

En la tabla 6.1 podemos ver varios elementos que siguen la notación utilizada al comienzo de esta sección, donde distinguíamos entre recursos como tal, requisitos y

	Requisitos	**P/W**	**RT**	**EC**
Neg	Variable	V	V	V
P_1	none	U	U	U
P_2	Rq_1	1	U	U
P_3	Rq_2	1	U	U
P_4	Rq_2&Rq_3	0	U	U
C_A	Rq_1	0	U	U
C_B	Rq_1|Rq_2	U	U	U
C_C	Rq_2	1	U	U
C_D	Rq_1	0	U	U
C_E	Rq_1|Rq_2|Rq_4	0	U	U
R_A	Rq_1&Rq_2	1	0	0
R_B	Rq_1	0	3	0
R_C	Rq_1|Rq_4	1	3	0
R_D	Rq_1&Rq_3&Rq_4	1	0	0
R_E	Rq_1&Rq_3&Rq_4	0	2	0
R_F	(Rq_1|Rq_2)&(Rq_3|Rq_4)	1	2	0
R_G	Rq_1|Rq_3	1	2	0
R_H	Rq_1 & Rq_2&Rq_3	0	0	0

Tabla 6.1: Valores de los atributos (propiedades) los elementos del ejemplo de negociación de confianza. U:No especificado, V:Variable

policy items, que también son considerados recursos. En la columna “Requisitos” de la tabla encontramos la relación de requisitos que gobiernan el acceso a cada recurso, con independencia de su naturaleza y esto incluye a las políticas (policy items). El acceso a los recursos está gobernado por una combinación lógica de requisitos. Cualquiera que pretenda acceder a un recurso deberá primero satisfacer todos los requisitos establecidos en la columna “Requisitos”.

Cada requisito, por ejemplo Rq_1, es expresado formalmente por un trozo de política o por varias políticas (P_1). En el ejemplo de la tabla 6.1 el requisito Rq_n está expresado formalmente por el policy item P_n. Como además los P_n se consideran recursos y se protegen con requisitos, se establece un acceso a los recursos de forma monótona creciente. Analicemos la tabla para comprenderlo mejor, para acceder a un recurso que necesite satisfacer Rq_2, es necesario conocer P_2; pero como se puede apreciar en la tabla, antes de poder acceder a P_2 es necesario haber satisfecho Rq_1 y para conseguirlo es necesario conocer P_1, que es accesible por defecto.

Se puede decir que P_1 es el **punto de partida** en la negociación, y que bien podría

corresponder a una autenticación básica de cliente en un dominio. En la tabla se utilizan los requisitos indistintamente para acceder a los recursos de trabajo y personales. Esta situación no tendría por qué darse en la realidad, lo más lógico sería que existiesen otras rutas que pudieran tener distintos puntos de partida, dado que los dominios donde se realiza la autenticación o autorización, no tienen por que coincidir para temas personales con temas de trabajo. Esto es una **separación de roles**. En el ejemplo sólo se utiliza un juego de requisitos, que muestran una serie de rutas con el mismo punto de partida dado que el objetivo del ejemplo es ilustrar el funcionamiento del sistema y no le resta generalidad a la validación.

Para mostrar cómo afecta el contexto a las decisiones, para cada uno de los recursos, se incluye información de contexto, en concreto la siguiente:

- el atributo **P/W** diferencia los recursos que sólo pueden ser utilizados por motivos personales (0) de los del trabajo (1).
- **RT** clasifica los recursos por tipo. Hemos elegido las categorías de web-services (0), ftp (1), ficheros compartidos (2) y agenda (3).
- el atributo **Loc** especifica la localización relativa del recurso: 1 si está localizado en el dispositivo móvil y 0 si no (cuando el dispositivo hace de proxy).
- el atributo **EC** expresa el consumo de batería variando entre 0 y 1 (0 para consumo nulo). En el caso de que el dispositivo ofrezca servicios, una caída en la tensión de la batería puede denegar el acceso para conservar otros de mayor importancia o que se consideren vitales, por ejemplo, desactivar un posible servicio de streaming de audio para evitar perdida de la telefonía.

6.1. Calculando las disimilitudes y los pesos

Para la evaluación del riesgo procederemos al cálculo de las similitudes entre todos los elementos participantes en la negociación y un elemento que represente el estado actual. Este elemento es el **estado de la negociación, estatus** o **Neg** como aparece en la tabla. Lógicamente, los atributos de dicho elemento varían durante una negociación, recogiendo los avances de la misma como pueden ser los requisitos que se van satisfaciendo o bien los cambios en los atributos que tienen que ver con el contexto. Las (di)similitudes entre pares se calculan como se ha explicado en el capítulo 5, incluyendo las expresiones lógicas.

Como no todos los elementos pertenecientes al espacio de decisión pueden describirse utilizando los mismos atributos, a todos aquellos indefinidos se les da el valor de U o NaN, ya que ALSCAL-MDS puede trabajar en ausencia de datos.

t	Rq_1	Rq_2	Rq_3	Rq_4
$t=0$	0	0	0	0
$w[\,]$	1.16	1.16	1.16	1.16
$t=1$	1	0	0	0
$w[\,]$	0	1.4	1.4	1.4
$t=2$	1	1	0	0
$w[\,]$	0	0	1.75	1.75
$t=3$	1	1	1	0s
$w[\,]$	0	0	0	2.33

t	**Loc**	**P/W**	**RT**	**EC**
$t=0$	1	1	Un.	0
$w[\,]$	1.16	1.16	0	0
$t=1$	1	1	Un.	0
$w[\,]$	1.4	1.4	0	0
$t=2$	1	1	Un.	0
$w[\,]$	1.75	1.75	0	0
$t=3$	1	1	Un.	0
$w[\,]$	2.33	2.33	0	0

Tabla 6.2: Resultado del cálculo de los pesos durante la negociación de confianza. Los requisitos no satisfechos son 0. La suma total de los pesos se mantiene, en el instante inicial $K(t=0)=7$.

La 6.2 proporciona los valores de los pesos para el ejemplo en los distintos instantes de tiempo. Se ha utilizado la primera expresión de las descritas en la sección 5.4 para el cálculo de pesos.

Como se puede apreciar en la tabla 6.2, la suma de los pesos se mantiene constante de forma que, a medida que los otros pares proporcionan credenciales satisfaciendo recursos, y la negociación comienza a definirse, podemos distribuir uniformemente el valor entre los atributos restantes maximizando diferencias.

Alcanzado este punto, tenemos todo lo necesario para usar MDS para analizar los datos resolviendo para dos variables. Se ha utilizado la ecuación descrita en 5.4 para calcular el radio de un círculo de riesgo

6.2. Resultados de la simulación

En esta sección presentamos los resultados de la negociación cuyas reglas están definidas en la tabla 6.1. La mecánica de la negociación es la siguiente, la entidad A, que es la que tiene el recurso, revela policy items que expresan requisitos y la entidad B revela credenciales para satisfacer esos requisitos. La negociación evoluciona como

puede verse en la tabla 6.2. Esta tabla recoge tanto los requisitos satisfechos, como la evolución de los pesos para cada instante de tiempo.

En $t = 0$, B trata de acceder a un recurso, en concreto a R_F, en A. El motor de decisión de A tiene registrados varios requisitos para ese recurso. Dicho recurso es revelables sólo a empleados de la compañía, por lo que **P/W** es igual a 1. El parámetro **P/W** es uno de los parámetros que, en lugar de ser especificado mediante política o contexto, podría ser indicado por el cliente que trata de acceder al recurso.

Luego aplica la información de contexto: a *Loc* le da el valor 1 ya que está localizado en el dispositivo y, dado que la batería está completamente cargada, y el usuario no contempla restricciones sobre el tipo de recurso, tanto **RT** como **EC** permanecen indefinidos. Como no se ha satisfecho requisito alguno hasta el momento, los valores para Rq_n son 0. El dispositivo móvil calcula los pesos (ver tabla 6.2) para $t = 0$.

En ese momento se calculan las disimilitudes y se simplifica el problema como puede verse en la figura 6.1 que muestra el espacio de decisión.

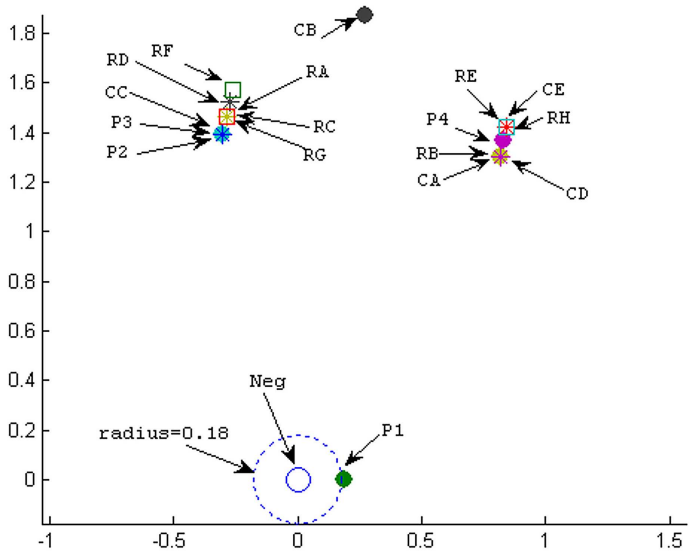


Figura 6.1: Espacio de negociación en $t = 0$, donde se puede apreciar que P1 es el punto de partida de la negociación.

Como puede verse en la figura 6.1, P_1 es el único recurso que entra en el circulo y por ello se permite el acceso a él. El elemento P_1 no coincide exactamente con el ideal, debido a indefiniciones en la mayoría de sus parámetros, pero al ser un **punto de entrada** se podría liberar con independencia de que hubiera o no una tolerancia al

riesgo.

El resto de los elementos forman dos grupos, el primero, con centro aproximado en [0.4,1.5], está compuesto por los recursos que pueden ser liberados por trabajo; el otro por los recursos de acceso personal. Por lo tanto, se puede apreciar cómo afectan las condiciones de contorno o contexto a la decisión.

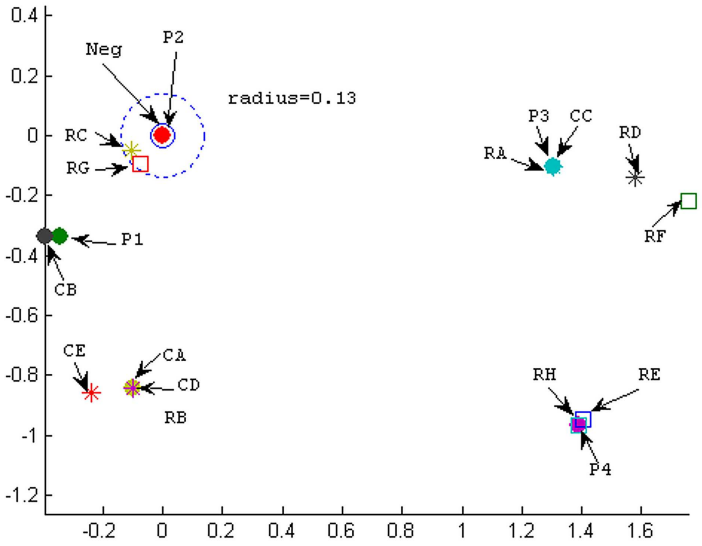


Figura 6.2: Espacio de negociación en $t = 1$. Muestra el espacio de decisión tras satisfacer, la otra parte, el requisito 1.

La figura 6.2 muestra el espacio de decisión en $t = 1$. En $t = 1$ el motor recalcula el espacio de decisión dado que la otra parte ha proporcionado credenciales para satisfacer Rq_1. Por ello A puede permitir el acceso a P_2, R_G y R_C. Estos recursos están ahora disponibles, dado se puede acceder por trabajo y requieren Rq_1. El hecho de que estén disponibles al usuario, quiere decir que si éste lo solicitara, podría acceder; pero en ningún caso se le informaría de ello, salvo que entre ellos se encontrara el recurso al que pretende acceder. Por otro lado, si más adelante el cliente quisiera acceder al recurso, por ejemplo R_C, no tendría que volver a negociar.

Los recursos C_E,C_A,C_D y R_B están separados mucho dado que solo son accesibles por motivos personales. R_A,P_3 y C_C se agrupan puesto que dependen de Rq_2. Por otro lado, puede verse como el motor clasifica el resto de los recursos: R_H,R_E y P_4 son accesibles por motivos personales como C_E,C_A,C_D y R_B, pero estos últimos están más separados **Neg** dado que sus requisitos son mas complejos. R_D y R_F están cerca del grupo de P_3 ya que, pese a ser personales, depende de Rq_2 también.

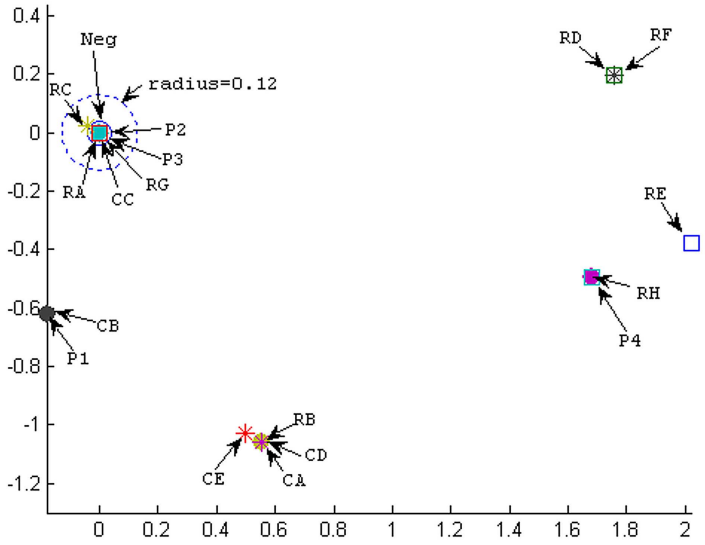


Figura 6.3: Espacio de negociación en $t = 2$. Muestra el espacio de decisión tras satisfacer, la otra parte, el requisito 2. Al definirse mejor el estado de la negociación, aumentan los recursos a disposición del usuario.

El espacio de decisión para $t = 2$ se representa en la figura 6.3. Como Rq_2 se ha satisfecho, R_A, P_3 y C_C están dentro del círculo y por ello son accesibles. El resto permanece lejos del punto **Neg** debido a sus requisitos o a que son de acceso personal.

En la figura 6.4 se puede ver cómo, una vez se cumple el requisito Rq_3, el recurso R_F pasa a estar disponible permaneciendo el resto lejos del centro. El sistema impide el acceso a P_4 dado que es personal y por ello todos los recursos dependientes de Rq_4 no son accesibles.

Finalmente, como se puede apreciar, el cliente ha obtenido acceso al recurso deseado, R_F, mediante el cumplimiento escalonado de una serie de requisitos.

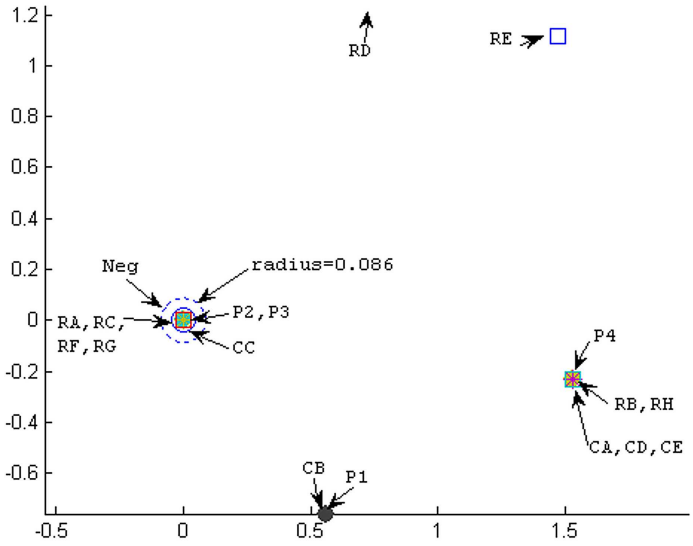


Figura 6.4: Espacio de negociación en $t = 3$. Muestra el estado final del espacio de decisión.

CAPÍTULO

7 Contribuciones a protocolos

Durante los anteriores capítulos se ha puesto de relevancia la conveniencia de realizar negociaciones de confianza. Estas negociaciones permiten alcanzar estados de seguridad de forma gradual y ordenada, permitiendo acceder a recursos con diferentes requisitos de autorización de forma escalonada.

Para alcanzar este objetivo se han propuesto en el capítulo 5 una serie de algoritmos de decisión para el tratamiento de la información que permiten, de una manera global, encontrar solución a un problema de control de acceso complejo, donde intervienen muchas credenciales y políticas. Dichos algoritmos permiten tratar a los participantes en la decisión de forma agnóstica, de manera que, con independencia del lenguaje en que fueron escritos los requisitos o las políticas, éstas puedan combinarse con otras. Por otro lado, se permite que sean los motores de control de acceso los que validen dichas credenciales dado que, pese a que el motor de decisión mantiene la coherencia y el orden en la negociación, no puede verificar las credenciales.

En este capítulo se mostrarán los diseños de nuevas extensiones a protocolos realizadas durante la investigación con objeto de cubrir ciertas necesidades de autenticación y autorización; dar soporte a la emisión de algunas credenciales de autorización, en concreto certificados de atributos; proporcionar una extensión con soporte para la negociación de confianza.

La descripción de las diferentes soluciones se ha realizado en orden cronológico siendo la última de ellas, aquella que soporta negociación de confianza en varios pasos, sustitutiva de la primera. La razón de documentar todos los esfuerzos anteriores a la solución final es la de ilustrar el proceso de mejora continua que se ha realizado en paralelo a las demandas tecnológicas actuales. Por esta razón, se ha pasado de un esquema basado en PKI con soporte de SAML, a un esquema final flexible, compatible con cualquier credencial existente y futura, que recoge las anteriores contribuciones y

corrige sus defectos y vulnerabilidades.

7.1. Elección del protocolo TLS

Pese a que el mecanismo de decisión para negociación de confianza puede orquestar los intercambios de credenciales realizados por cualquier protocolo, siempre que dicho algoritmo esté empotrado en el sistema operativo, es adecuado o al menos simplificaría el despliegue, el hecho de proporcionar extensiones o nuevos protocolos que permitan un intercambio flexible de dicha información de seguridad. Para ello, en este capítulo se proponen modificaciones a un protocolo, en concreto Transport Layer Security (TLS) [43].

Las motivaciones que nos empujan a utilizar TLS como protocolo son las siguientes:

- Motivaciones técnicas: recogen los motivos técnicos por los cuales se considera apto TLS para su utilización en negociaciones de confianza:
  - Proporciona por si solo confidencialidad de los datos intercambiados a través de dicho protocolo.
  - Proporciona opcionalmente autenticación unilateral o bilateral.
  - Permite la confidencialidad sin necesidad de autenticación mediante el uso de generación de claves basado en DiffieHellman [106].
  - Permite negociar la compresión de datos utilizada y los algoritmos de cifrado simétrico que protegen la comunicación extremo a extremo, así como los algoritmos de autenticación de mensaje que garantizan la autenticidad.
  - La funcionalidad del protocolo de negociación puede ampliarse mediante extensiones tal y como se recoge en la recomendación [83].
  - Permite su utilización sobre EAP, lo que permite su uso sobre protocolos utilizados para el acceso a la red.
- Motivaciones de despliegue: El usuario medio confía en TLS dado que su utilización en comercio electrónico es amplia. Es también utilizado para proteger las conexiones a servidores de correo electrónico y otros servicios de uso ubicuo. Estas razones hacen de TLS un protocolo adecuado dado que ya se encuentra desplegado.

Para comprender mejor las extensiones planteadas se recomienda al lector leer la sección 2.3 donde se proporcionan los detalles técnicos de TLS necesarios para comprender el resto del capítulo.

7.2. Extensión de TLS para soporte de autorización

En esta sección describiremos la arquitectura y la sintaxis de las extensiones que, usadas sobre TLS, permiten añadir soporte de autorización al handshake de TLS. Las solución aquí planteada no permite negociación de confianza, dado que solo se soporta autenticación y autorización en un solo paso, es decir:

- Permite negociar el tipo de información de autorización a utilizar pero discerniendo únicamente entre Certificados de Atributos y asertos SAML.
- No permite utilizar varios tipos de autenticación, sino que utiliza la proporcionada por TLS y por tanto la información de autorización debe estar vinculada a la credencial utilizada por el cliente.
- No permite el envío de varios mensajes conteniendo credenciales de autorización sino que permite enviar información de autorización una única vez.
- No permite que el servidor utilice diferentes mecanismos de autenticación.
- No permite al servidor enviar credenciales de autorización.

Las extensiones planteadas soportan por tanto certificados de atributos, que se documentaron en la sección 2.2, y asertos SAML [57]. Por otro lado, se permite la utilización de dichas credenciales en esquemas de "Pull" y "Push", cuyas diferencias pueden apreciarse en la figura 2.6, y que permiten no solo que el servicio al que se tiene acceso busque la información relativa a permisos en repositorios, sino que permite al cliente proporcionar dicha información directamente.

Entidades involucradas y estructura

Como paso previo a la descripción de la arquitectura comentemos las diferentes entidades y roles a considerar en adelante:

- *Proveedor de Identidad (IDP)*, *Origen de Autenticación* o *Source of Authentication (CA)*: es la entidad que asegura la identidad de otra (*Principal*) que puede ser una persona o entidad. En SAML el IDP puede proporcionar también información de autorización, dado que los roles están menos separados. En PKI, en cambio, estas funciones están completamente separadas, correspondiendo el rol de IDP a una Autoridad de Certificación o CA.

- *Origen de Autorización, Source of Authorization (SoA)* o su delegada llamada *Attribute Authority (AA)*: Proporciona información sobre los permisos y privilegios de la entidad utilizando certificados de atributos. El IDP y la SoA pueden ser la misma entidad.
- *Proveedor de Servicio* o *Service Provider*: Es la entidad que consume la información de autorización para realizar el control de acceso.
- *Assertion Consumer* (en SAML) o *backend AAA server* (en otros casos): Es la entidad final a la que son redirigidas las decisiones de autorización dentro del dominio del proveedor de servicio.
- *Protocol Binding*: es una descripción de como utilizar SAML con un determinado protocolo para el acceso a los servicios.

Respecto a la arquitectura, otras soluciones, como se comentará en detalle más adelante, dejan a la capa de Handshake de TLS la tarea de procesar las extensiones. En nuestro caso, añadimos un nuevo módulo, que será cliente de la capa de record *TLS record layer*, para procesar la información de autorización. De esta manera liberamos a la capa de Handshake de la tarea de procesar una información para la que no fue diseñada. La Figura 7.1 ilustra la arquitectura propuesta.

Para poder habilitar el uso de este módulo bajo demanda, respetando así la compatibilidad hacia atrás con servidores que no soporten autorización o simplemente no soporten extensiones, la única modificación necesaria en el protocolo de Handshake es la del soporte de una nueva extensión que afecta a los mensajes de *Client Hello* y *Server Hello*. Este tipo de modificaciones están contempladas por el estándar de TLS permitiendo realizar la modificación sin interferir en servidores que no soporten extensiones.

Descripción de la extensión de TLS para uso de autorización

Como se comentó en la sección 7.2 la extensión de TLS que aquí se plantea permite negociar mecanismos para hacer “push” de la información de autorización; o bien, para proporcionar al servidor todos los datos necesarios de forma que éste pueda recogerla, “pull”, de las fuentes indicadas por el cliente.

Para la definición de nuestra extensión, seguiremos la filosofía del protocolo de Handshake de TLS, que permite la elección, consensuada por ambas partes, de los mecanismos de cifrado y compresión de datos mediante el intercambio de los mensajes de *Client Hello* y *Server Hello*. Por esta razón, hemos definido la extensión de forma que permita al servidor elegir entre los mecanismos proporcionados por el cliente.

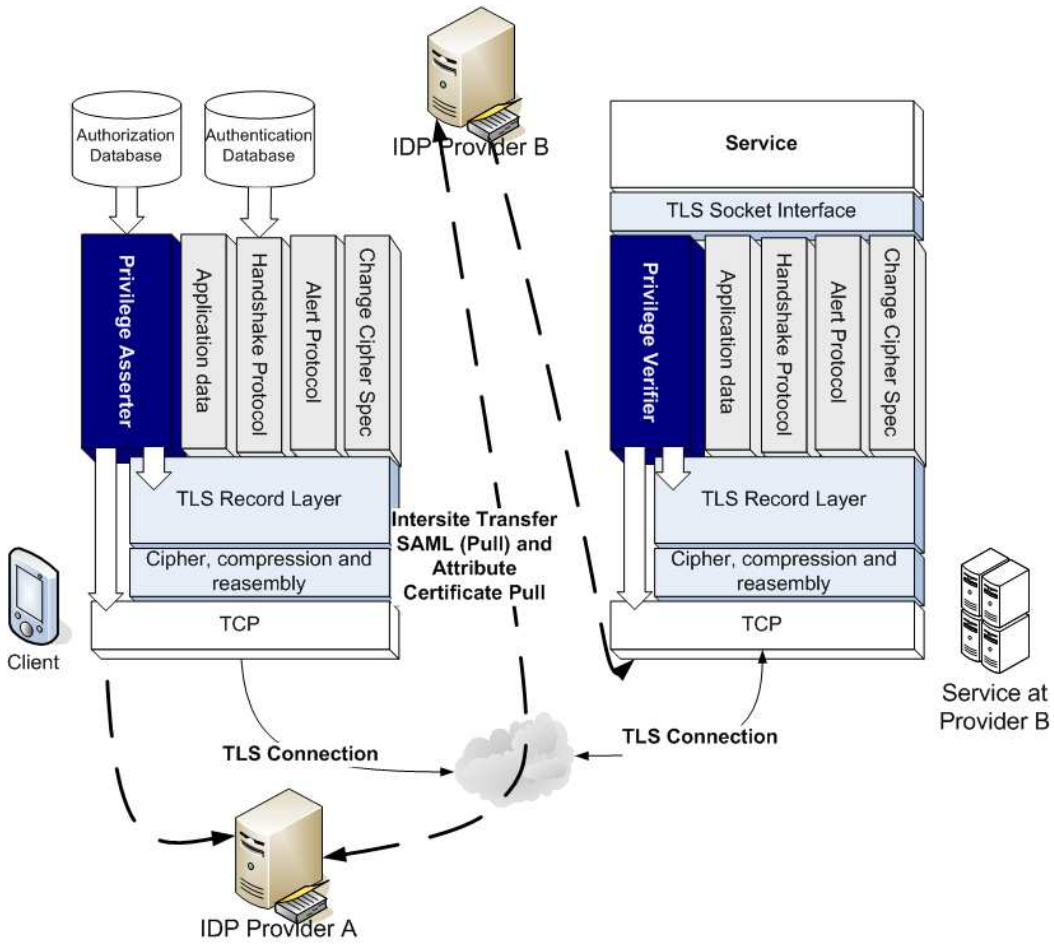


Figura 7.1: Estructura de capas que negocia la extensión de TLS para el soporte de autorización.

Pasos de la negociación de autorización

En esta sección, se describen los pasos en los que se realiza la autorización mediante TLS. Esta sección es un resumen del comportamiento del protocolo, por esta razón, más adelante, en las secciones 7.2 y 7.2 se proporcionarán detalles específicos correspondientes a cada uno de los mensajes intercambiados. Finalmente, en la sección 7.2 se comparará nuestra solución con otras desarrolladas al mismo tiempo.

La autorización se realiza por acuerdo mutuo: el cliente y el servidor tratan de acordar los mejores mecanismos de autorización a utilizar. La negociación como tal se realiza durante el proceso de handshake de TLS y una vez se ha establecido el canal seguro se intercambian las credenciales. También es posible negociar e intercambiar sobre

canal seguro. Durante la negociación se acuerdan los siguientes detalles:

- Tipo de mecanismo de autorización: pull o push.
- Tipo de credencial a utilizar: SAML o Certificados de Atributos.

La estructura de los mensajes de negociación permiten al cliente y al servidor proporcionar información más o menos detallada dependiendo del entorno. En cualquier caso, las credenciales como tal, se intercambien siempre sobre un canal seguro. Dado que la información de negociación se intercambia sobre un canal inseguro (durante el Handshake de TLS todavía no hay canal seguro), en determinadas circunstancias, por ejemplo, en presencia de cifrado a nivel de enlace, puede ser interesante, para aplicaciones que requieran baja latencia, proporcionar más detalles en la negociación. Así dichas aplicaciones pueden aumentar la probabilidad de elegir el mecanismo de autorización adecuado pese a que el canal sea todavía inseguro. En cambio otras aplicaciones, que no tengan requisitos de latencia, pueden ser menos específicos y requerir que la negociación de los mecanismos de autorización se realice sobre un canal seguro, es decir, una vez el handshake de TLS haya finalizado. El intercambio de mensajes propuesto se muestra a continuación, siendo las extensiones aquellas cuyo nombre aparece entre signos de mayor/menor <**Extensión**>; las partes opcionales, fuera del protocolo de handshake estándar, se distinguen por estar seguidas de un *:

```
Client                                              Server

ClientHello
<AuthorizationAdvertising*>  -------->
                                                ServerHello
                                   <AuthorizationAdvertising*>
                                                Certificate*
                                          ServerKeyExchange*
                                         CertificateRequest*
                             <--------      ServerHelloDone
Certificate*
ClientKeyExchange
CertificateVerify*
[ChangeCipherSpec]
Finished                     -------->
                                          [ChangeCipherSpec]
                             <--------             Finished
AuthorizationAdvertising*
(secure exchange only)       -------->
```

```
                                    AuthorizationAdvertising*
                                <-------(secure exchange only)

Cancel*
(if no match)                   ------->

                                <-------  AuthorizationRequest
AuthorizationPush*
(if push mechanism)             ------->

                                <-------  Ready or NotEntitled

Application Data                <------->     Application Data
```

A continuación se describe el proceso de autorización, es decir, los intercambios de extensiones y mensajes que complementan al handshake original de TLS:

1. El cliente, o *Privilege asserter*, puede solicitar al servidor realizar la negociación de mecanismos de autorización sobre un canal seguro (una vez finalice el handshake, sobre canal seguro, por privacidad) o proporcionar en ese momento la lista de mecanismos que soporta. Para ello se utiliza la extensión *AuthorizationAdvertising* como parte de un mensaje de *Client Hello*. La sintaxis de la extensión se describe en la sección 7.2. Del mismo modo, dado que por su estructura SAML puede ser usado con anonimato, el cliente puede especificarlo en la extensión.

   De esta manera se cumplen varios propósitos, en primer lugar se informa a la otra parte de que se soportan extensiones, en concreto mecanismos de autorización; en segundo lugar, se indica cómo ha de realizarse la negociación de mecanismos: utilizando la información contenida en la extensión o más adelante, una vez se haya establecido un canal seguro.

2. Si el servidor soporta extensiones leerá la extensión. Si además es capaz de procesar ésta en concreto, él redirige la información al verificador de privilegios o *Privilege Verifier*. En caso contrario, simplemente continuará con el Handshake como si de una conexión de TLS tradicional se tratara.

   - Si el cliente demanda un intercambio seguro, el servidor, en concreto el verificador de privilegios o *Privilege verifier*, responde copiando el contenido de la extensión recibida en una extensión que acompañará al mensaje *Server Hello*.

- En otro caso, el *Privilege verifier* selecciona un mecanismo de autorización entre aquellos especificados por el cliente. Para informar al cliente de su decisión, enviará una extensión de tipo *AuthorizationAdvertising* acompañando al mensaje *Server Hello* que contenga únicamente el mecanismo seleccionado.
- En el caso de que el servidor no soportara ninguno de los mecanismos indicados por el cliente, responderá con una extensión *AuthorizationAdvertising* que contiene una lista vacía. En ningún caso se podrá finalizar la conexión **debiendo continuarse el protocolo de handshake hasta el final** para prevenir, como se explicará más adelante, ataques de denegación de servicio.

3. En este punto de la negociación, pueden ocurrir dos cosas:

   - Si el cliente especificó el uso de SAML con anonimato, el servidor deberá enviar su certificado en el mensaje *Certificate* (y no podrá utilizar el mensaje *ServerKeyExchange* proporcionando únicamente el material criptográfico necesario para generar una clave usando DiffieHellman, para evitar ataques MITM). Los parámetros del certificado del servidor pueden ser utilizados por el cliente como información de *Holder* dentro de una hoja SAML.
   - En el resto de los casos, el intercambio de TLS continúa hasta que el canal es seguro, teniendo en cuenta que el cliente deberá identificarse, a petición del servidor (*Certificate Request*), usando un Certificado. De esta manera, el servidor podrá comprobar, mediante una prueba de posesión de clave privada, que el cliente ostenta determinados derechos asociados al *Holder* de dicho certificado.

   Es decir, salvo en el caso de utilizar SAML con anonimato, el cliente deberá proporcionar un certificado como autenticación previa al proceso de autorización.

4. Una vez el protocolo de handshake ha finalizado y el canal seguro ha sido establecido, el tráfico de aplicación se bloquea y sólo se podrán intercambiar mensajes con origen en el *Privilege Asserter* (lado del cliente) y destino en el *Privilege Verifier* (lado del servidor). Para hacer esto, es necesario cambiar el campo *ContentType* en los mensajes del protocolo de record dándole el valor[1] PRIVILEGE_LAYER, que identifique a la capa de gestión de privilegios (*Privilege Management layer*). Así se consigue distinguir el tráfico generado por las capas de gestión de privilegios del generado por el protocolo de Handshake, Alert o ChangeCipherSpec, que son los otros clientes de *TLS Record Layer*. A partir de este punto,

[1] Los números deberán ser asignados por la IANA.

hasta terminar el proceso de autorización, todos los mensajes que no correspondan al protocolo Alert serán intercambiados especificando un ContentType de tipo PRIVILEGE_LAYER.

5. Si el cliente solicitó un intercambio seguro de credenciales, una vez se ha creado el canal protegido, éste deberá enviar al servidor un mensaje conteniendo los detalles sobre los mecanismos de autorización que no proporcionó durante el Handshake. Por esta razón, deberá enviar un mensaje conteniendo la misma estructura descrita en la extensión *AuthorizationAdvertising*. El servidor deberá responder de la misma manera que en los pasos (1) y (2), seleccionando el mecanismo apropiado, con la salvedad de que en lugar de hacerlo sobre una extensión, deberá hacerlo en un mensaje con ContentType PRIVILEGE_LAYER. Todos estos mensajes de negociación estarán protegidos por la clave generada durante el handshake.

6. En cualquiera de los dos casos, intercambio seguro o no, si no se ha podido negociar satisfactoriamente ningún mecanismo de autorización, llegado este momento y nunca antes, se puede terminar la conexión unilateralmente enviando un mensaje de tipo *Cancel*.

7. Una vez se ha negociado un mecanismo de autorización entre las dos partes, ya sea durante el Handshake o una vez establecido el canal seguro, el *Privilege Verifier* envía al *Privilege Asserter* un mensaje de tipo *AuthorizationRequest* indicándole que debe ejecutar el mecanismo de autorización (este mensaje se describe en la sección 7.2).

   La única particularidad que debe ser precisada en este punto es el uso de los campos opcionales del mensaje *AuthorizationRequest*. El servidor deberá enviar el mensaje *AuthorizationRequest* al cliente, utilizando el campo opcional *service_assertion_consumer*, únicamente si el cliente solicitó la utilización de un mecanismo de tipo "pull" usando SAML. Esto se debe a que es necesario especificar la URL del *Assertion Consumer* del dominio del servidor, donde se recogerá el resultado de ejecutar SAML. Sin esta información, difícilmente el *Privilege Asserter* podrá ejecutar el mecanismo SAML. El IDP en SAML puede mantener una sesión en activo para un cliente y proporcionar información de autenticación a distintos servicios dentro de un dominio de identidad federada, pero para ello, el cliente debe informar al IDP qué servicio va a utilizar a continuación (mediante el *Intersite Transfer Service*), proporcionándole además la URL del *Assertion Consumer* de ese servicio. Así el IDP puede enviar toda la información de autenticación y autorización a dicho servicio (a través de su *Assertion Consumer*) particularizando la hoja SAML para ese servicio.

El lector deberá darse cuenta que si no se especifica el *Assertion Consumer*, en ausencia de una prueba de posesión de clave privada, una hoja SAML podría utilizarse para realizar un ataque tipo Reply.

8. Si el cliente seleccionó un mecanismo de tipo push, el *Privilege Asserter* deberá enviar la credencial apropiada utilizando un mensaje de tipo *AuthorizationPush*. En el resto de los casos, se procede de la siguiente manera:

   - En el caso de pull con certificado de atributos, el *Privilege Asserter* esperará hasta que el servidor descargue la credencial de un servidor HTTP o LDAP.
   - En el caso de pull con SAML, el cliente ejecutará su parte del "binding", por ejemplo: el *Privilege Asserter* hará login en su *InterSite Transfer Service* (GET) y enviará la hoja SAML (POST) a la URL del *Assertion Consumer* indicada previamente por el *Privilege Verifier*. Puede verse un ejemplo de utilización en la figura 7.2.

9. Por último, el *Privilege Verifier* verificará la credencial y enviará un mensaje de tipo *ready* al *Privilege Asserter* para indicar, en caso de éxito, que puede proceder a enviar datos de aplicación.

   En caso contrario, el servidor enviará un mensaje de tipo *notEntitled* si el cliente no está autorizado, indicando opcionalmente la razón o los detalles de la política. Si un mecanismo de autorización no termina su ejecución en un tiempo dado, el servidor responderá con un mensaje *timeout* descartando silenciosamente cualquier mensaje enviado por la otra parte.

La tabla (7.1) resume los posibles casos. Debe tenerse en cuenta que la autenticación de cliente dependerá del mecanismo de autorización a utilizar. Dado que cabe la posibilidad de que el cliente solicite intercambio seguro; en esa situación, el servidor deberá solicitar, siempre que sea posible, el certificado de cliente. En estas circunstancias, si el cliente no necesita autenticarse ante el servidor, para hacer valer sus credenciales de autorización, puede enviar un mensaje *Certificate* vacío. Si el servidor recibe un mensaje *Certificate* de cliente vacío es decisión suya finalizar o no la conexión según consta en la recomendación de TLS.

Descripción del formato de la extensión AuthorizationAdvertising

La extensión *AuthorizationAdvertising* permite indicar si el cliente solicita negociar la autorización sobre canal seguro o, en caso contrario, qué mecanismos de autorización soporta. Por otro lado, permite indicar dichos mecanismos con mayor o menor

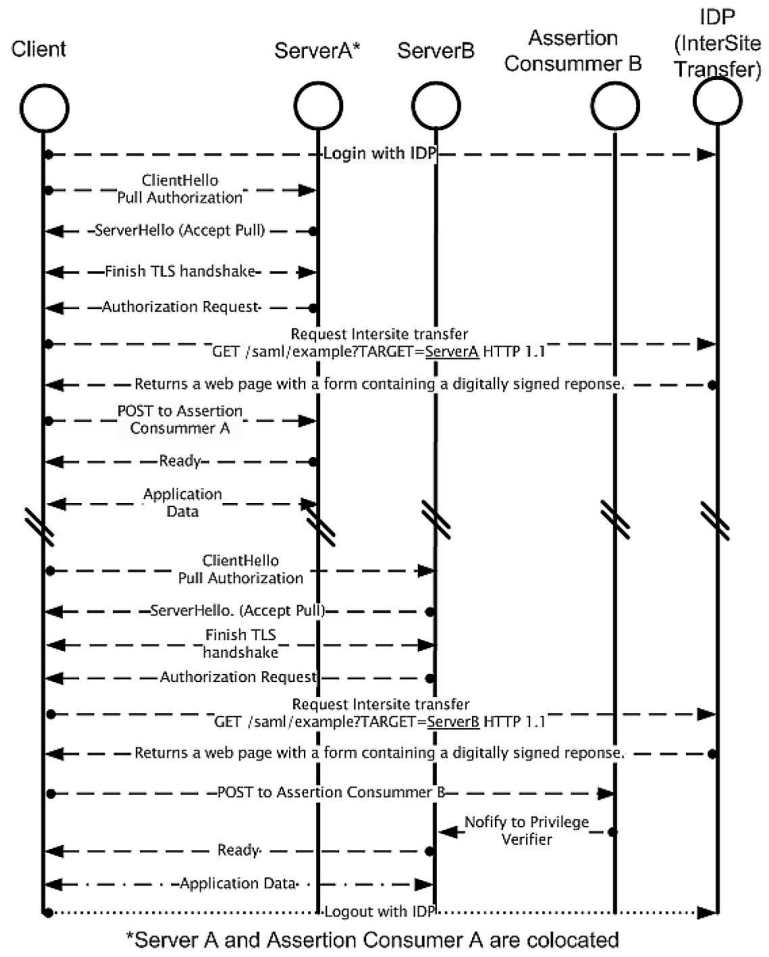


Figura 7.2: Ejemplo de handshake utilizando SAML pull (SAML Browser POST profile).

detalle dependiendo de los requisitos de la aplicación, en cuanto a celeridad de la negociación; dependiendo de si se encuentra o no en un entorno seguro; o si el medio de acceso es compartido o no.

En los siguientes párrafos, cuando hablemos de negociación, hablaremos de negociación entre el *Privilege Asserter* en el lado del cliente y el *Privilege Verifier* en el lado del servidor pero, por simplificar, usaremos cliente para referirnos al *Privilege Asserter* y server para referirnos al *Privilege Verifier*.

En lo que resta de sección, plantearemos la sintaxis de la estructura y argumentaremos el por qué de las decisiones tomadas.

Uso de la extensión por parte del cliente

Autorización cliente ⇒ Autenticación Servidor⇓	AC	AC
Tipo de negociación⇒	**sobre canal seguro**	**handshake TLS**
TLS con PKC	Durante el handshake el cliente deberá proporcionar un certificado de PKI vinculado a la información de autorización a utilizar, sobre el canal seguro, sea mediante pull o push. En ausencia de dicha autenticación de cliente, el servidor no puede autenticar ni autorizar al cliente mediante certificados de atributos.	En este caso, la negociación del algoritmo de autorización se realiza durante el handshake, por tanto, el servidor puede determinar que necesita autenticación de cliente. En ausencia de dicha autenticación de cliente el servidor finaliza la conexión tras el intercambio de mensajes Finished.
TLS con DH	Si el servidor utiliza un certificado Diffie-Hellman (DH) puede proceder igual que si de un certificado RSA se tratase, es decir, puede solicitar al cliente autenticación y éste proporcionarla para usar ACs. En cambio, si el servidor utiliza una clave DH a través del mensaje ServerKeyExchange para utilizar un mecanismo DH anónimo, no puede solicitar autenticación al cliente. Por tanto, en este caso, no sería posible el uso de ACs como autorización.	Este caso es similar a la negociación sobre canal seguro con una salvedad: Si el servidor dispone de dos posibles autenticaciones válidas, con y sin certificado, dado que conoce los mecanismos soportados por el cliente, puede utilizar aquella que le permita solicitar autenticación de cliente. Si el servidor no dispone de certificado y el cliente pretende utilizar ACs, romperá la conexión tras el intercambio de los mensajes finished.
Autorización cliente⇒ Autenticación Servidor⇓	SAML	SAML
Tipo de negociación⇒	**sobre canal seguro**	**handshake TLS**
TLS con PKC	Push: la hoja SAML debe estar vinculada a un certificado de clave pública del cliente y el cliente deberá autenticarse. Si el cliente no se autentica, el servidor no debe aceptar la hoja SAML. Pull: En este caso no es necesario que el cliente se autentique dado que el IDP del cliente generará una hoja SAML vinculada a la sesión TLS y al certificado del servidor. El cliente autentica al servidor pero la confianza por parte del servidor se establece entre el Assertion Consumer del servidor y el IDP del cliente. El cliente únicamente deberá autenticarse ante su IDP.	El funcionamiento es similar al de la negociación sobre canal seguro. La diferencia está en que el servidor puede determinar si debe o no requerir autenticación de cliente.
TLS con DH	Si el servidor utiliza un certificado con clave DH, el comportamiento es idéntico al de TLS con PKC para SAML. Si el servidor no dispone de certificado, sino de una clave DH para intercambio DH anónimo (DH_anon), no es posible admitir una hoja SAML mediante push ni pull. El anonimato en SAML corresponde al cliente, si el servidor no se autentica (DH_anon), cabe la posibilidad de que ocurra un ataque de man-in-the-middle.	Este caso es indistinguible de su homólogo sobre canal seguro.

Tabla 7.1: Tabla con las posibles interacciones.

La sintaxis de las diferentes partes de la extensión[2] así como la explicación, se proporciona a continuación:

```
struct {
    Boolean secureExchangeRequired;
    AuthorizationMechanism avail_mech_list<0..2^16>;
}AuthorizationAdvertising

struct{
    AuthorizationType auth_type;
    ExchangeType      exchange_type;
    select (exchange_type)
    {
        case push: ExchangePush;
        case pull: ExchangePull;
    }extended_information;
} AuthorizationMechanism;

enum{
    push(0), pull(1)
} ExchangeType;

enum{
    ATTRCERT(0), SAML(1),...
}AuthorizationType;
```

Las credenciales de autorización tal y como están definidas en [50] tienen carácter público, es decir, los certificados de atributos, al igual que los certificados de clave pública podrían ser compartidos en repositorios públicos. De esta manera los servidores podrían realizar búsquedas en esos repositorios para autorizar a un cliente una vez éste se ha identificado mediante un certificado de clave pública.

Las fuentes de autorización o SoAs firman los certificados de atributos que enlazan de forma unívoca certificados de clave pública con privilegios, ya sea mediante el nombre de directorio del certificado; mediante un hash de la clave pública del certificado o un hash del certificado completo. Por lo que en principio se puede suponer que la parte privada, que en este caso corresponde a la clave privada, se mantiene protegida en poder del usuario, pudiéndose hacer público el resto de la información.

[2] versión 3.0

Los asertos de SAML pueden ser intercambiados sobre HTTP plano ya que para darles validez deben estar firmados por una autoridad (IDP) y dicha firma los protege de posibles modificaciones. Los asertos SAML se vinculan a una identidad, ya sea esta un certificado X.509 o información de claves DiffieHellman.

Con independencia de la visión de aquellos que diseñaron los estándares, por motivos de privacidad y para evitar ataques de Man-In-The-Middle, las credenciales que contengan privilegios no debe exponerse dado que, aunque éstas no pueda ser utilizada por otros de forma sencilla, protegiendo las credenciales se evitan posibles ataques; se evita exponer información innecesaria al exterior que pudiera ser utilizada para realizar algún ataque, como robo u otros de ingeniería social. Esto no significa que, en determinadas circunstancias, no sea atractivo, por motivos de rendimiento, proporcionar detalles acerca de las credenciales, que luego se intercambiarán sobre canal seguro, como puede ser el emisor (SoA o IDP) o determinados valores de algunos atributos. Esto podría acelerar el acceso al servicio o reducir tiempo que se tarda en determinar si se puede acceder o no al servicio.

La extensión *AuthorizationAdvertising*, cuya estructura se ha mostrado con anterioridad, permite indicar los mecanismos de autorización de que dispone el cliente, utilizando diferentes niveles de detalle, de forma que se adecúe al entorno y por tanto a los requisitos de privacidad impuestos por el mismo o por la aplicación que hará uso del servicio. En aquellos casos en los que se requiera un **alto grado de privacidad** o en los que no sea necesario acceder al servicio en un tiempo dado, el cliente deberá asignar el valor “verdadero” al campo *secureExchangeRequired*, indicando su intención de negociar los mecanismos sobre un canal seguro.

Por otro lado, en los casos en los que el tiempo sea clave para el acceso a los servicios, por ejemplo, si se está haciendo roaming y se quiere mantener la continuidad de un servicio sensible a la latencia (como video bajo demanda); pueden proporcionarse algunos datos sobre los mecanismos en sí directamente en la estructura *AuthorizationAdvertising* usada durante el handshake.

En ambos casos, durante el handshake o establecido el canal seguro, el cliente debe rellenar una lista de mecanismos en el campo *avail_mech_list*. Un mecanismo se describe mediante una estructura de tipo *AuthorizationMechanism*. Para cada una de las estructuras de tipo *AuthorizationMechanism*, el cliente indicará si puede enviar credenciales al servidor (push) o si requiere que el servidor las obtenga del lugar indicado (pull).

El cliente deberá también indicar el tipo de credenciales de autorización a utilizar, certificados de atributos o SAML como se verá a continuación:

```
struct {
```

```
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT:  AttrCertInformation attr_cert_info;
        case SAML:  AssertionsInformation assertions_info;
    }
} ExchangePush;

struct{
    AttrCertChainType attr_cert_chain_type;
    opaque            issuer<0..2^16-1>;
    Attributes        attributes<0..2^16>;
}AttrCertInformation;

enum{
    individual_certs(0), ACPathData(1), (??)
} AttrCertChainType;

struct{
    opaque OID<0..2^16-1>; // Opcional
    opaque value<0..2^16-1>; // Opcional
}Attributes;

struct{
    opaque     IDP_uri<0..2^16-1>;  // URI Identity provider
    Assertions assertion_list<0..2^16-1>; //Opcional
} AssertionsInformation;

struct{
    opaque XMLNameSpace<0..2^16-1>;;
    opaque Element<0..2^16-1>;
}Assertions;
```

En el caso de mecanismos de push, el cliente deberá completar en *Authorization-Mechanism* la estructura de tipo *ExchangePush*. Para indicar que se utilizará un tipo de credencial u otro basta con completar el campo de tipo *AuthorizationType*. Las estructuras subyacentes para cada tipo se completan especificando sólo aquello que sea necesario en función del grado de detalle que se quiera dar.

Hasta el momento sólo se ha especificado la información necesaria para continuar,

que consiste en el tipo de mecanismo (pull/push) y el tipo de credencial (AC/SAML). Si por las razones de rendimiento que se han comentado se quisiera proporcionar más información de la indicada hasta el momento se podrían utilizar el resto de las estructuras que podrían ayudar al servidor a seleccionar el mecanismo más adecuado.

- En el caso de certificados de atributos, la estructura a utilizar es *AttrCertInformation*, que puede proporcionar datos de la credencial como los que se muestran a continuación en orden de perdida de privacidad:

  - Tipo de cadena que se va a utilizar: el campo *attr_cert_chain_type* permite indicar si se utilizará un solo certificado de atributos o una cadena de pares "certificado de clave pública - certificado de atributos".
  - La autoridad de atributos que lo ha emitido: si el servidor puede conocer de antemano el emisor de la credencial sabrá si la puede aceptar o no. Además, esta información acompañada del tipo de cadena ayuda al servidor a determinar si será capaz de procesar la cadena de certificados o no. Debe tenerse en cuenta que si un privilegio ha sido delegado en varias ocasiones puede ser complicado que el servidor construya la cadena completa si no dispone de toda la información.
  - Información sobre los atributos contenidos en el certificado: El tipo y naturaleza de los atributos contenidos en un certificado de atributos puede ser variado, por esta razón, un servidor puede requerir un determinado tipo de atributo para tomar una decisión de control de acceso. Lógicamente al llegar a este nivel de detalle, la privacidad decae. sin embargo, para aquellas aplicaciones que, por política, estén autorizadas a revelar esta información, se proporciona el campo *Attributes*.

    La estructura *Attributes* permite especificar desde el identificador de objeto del atributo, con la intención de identificar el tipo, hasta el valor del atributo en sí. No es recomendable en ningún caso, salvo para aplicaciones que requieran resolver la autorización muy rápido, proporcionar este nivel de detalle. Se debe limitar el detalle por motivos de privacidad y para evitar que el servidor, que podría ser un atacante, recopile información que no necesita.

- Si se utiliza SAML con un mecanismo de push, la estructura que se debe utilizar es *AssertionsInformation*. Esta estructura permite comunicar detalles sobre la credencial SAML al servidor incluyendo tantos como el cliente o su política estime oportuno:

- URI que identifique al Proveedor de Identidad que emitió el aserto SAML. El campo *IDP_uri* permite al cliente indicar el IDP que emitió el aserto; por tanto, el servidor puede determinar si puede aceptarla o no.
- Una lista de los atributos contenidos en la hoja SAML. El campo *assertion_list* permite opcionalmente proporcionar más detalles sobre la credencial. Debe tenerse en cuenta que el grado de privacidad disminuye a medida que se aumentan los detalles.

  Dicho campo permite, mediante la estructura *Assertions*, especificar el espacio de nombres que identifica el tipo de atributo. Se puede especificar además el valor, lo que reduce drásticamente la privacidad. Esto solo debería estar permitido siempre que no se exponga información comprometedora y sólo para aplicaciones tipo make-before-break, que requieran alta velocidad.

```
 struct {
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT: URLAndOptionalHash url_hash<1..2^16-1>;
        case SAML: BindingAndEndPoints binding_info<1..2^16-1>;
    }
} ExchangePull;
```

En aquellos casos en los que el cliente no tenga acceso local a las credenciales de autorización, podrá indicar al servidor dónde obtener dichas credenciales mediante la estructura *ExchangePull*.

La estructura *ExchangePull* permite proporcionar esta información en los campos *URLAndOptionalHash* para certificados de atributos y en *BindingAndEndPoints* para SAML.

```
struct {
     opaque url<1..2^16-1>;
     Boolean hash_present;
     select (hash_present)
     {
        case false: struct {};
        case true: SHA1Hash;
     } hash;
```

```
} URLAndOptionalHash;


struct {
    Boolean anonymous; //SAML anónimo
    opaque saml_binding<0..2^16-1>; //URI (protocol binding)
    opaque idp_url<0..2^16-1>; //URI Identity provider
} BindingAndEndPoints;
```

En el caso de utilizar un mecanismo de tipo push conjuntamente con certificados de atributos, la estructura *URLAndOptionalHash* proporciona detalles sobre el servidor HTTP, FTP o LDAP a utilizar para descargar el certificado. Para ello basta con especificar la URL en el campo *url*. Opcionalmente se puede proporcionar un hash por razones de seguridad, como se razona en [83].

Como ya se comentó con anterioridad, los mecanismos de tipo pull que utilizan SAML son complejos. Por esta razón la url del IDP, proporcionada en el campo *idp_url*, por si misma, no es suficiente para completar el intercambio. Por esta razón es necesario además especificar el tipo de binding utilizado en el campo *saml_binding*. Además, como ya se comentó al comienzo de la sección, si SAML se combina con anonimato, las claves debe intercambiarse mediante Diffie-Hellman durante el Handshake de TLS. Los parámetros Diffie-Hellman pueden utilizarse por un IDP de SAML para referenciar al holder, de cara al servicio, sin revelar su identidad; pero proporcionando los atributos necesarios para una autorización satisfactoria.

Uso de la extensión por parte del servidor

Para permitir la compatibilidad hacia atrás, un servidor TLS que no pueda manejar extensiones en general, o esta extensión en concreto, debe ignorar cualquier extensión como se explica en las recomendaciones [82, 52, 83].

Si el servidor fuera capaz de procesar la extensión, la respuesta adecuada sería responder con la misma extensión como se explica a continuación. Si el cliente solicita explícitamente un intercambio sobre canal seguro, el servidor simplemente copia la extensión recibida, tal cual, y la envía acompañando al mensaje de Handshake *Server Hello*.

En los casos en los que no se requiera un intercambio seguro o una vez sobre el canal seguro, el servidor debe responder al cliente, seleccionando uno de los mecanismos proporcionados por el cliente. Para ello devuelve una extensión idéntica a la anterior, en la que se han eliminado todos los mecanismos proporcionados por el cliente menos el seleccionado por el servidor. De esta forma se selecciona unívocamente el mecanismo a utilizar:

- AC push: El servidor responde con el mecanismo seleccionado rellenado el campo *exchange_type* y las estructuras *ExchangePush*, *AttrCertInformation* y *Attributes* utilizando el mismo nivel de detalle utilizado por el cliente.
- SAML push: El servidor responde rellenando el campo *exchange_type* y las estructuras *ExchangePush*, *AssertionsInformation* así como *Assertions* con el mismo nivel de detalle que el cliente proporcionó en la extensión.
- AC pull: El servidor responde utilizando el campo *exchange_type* y las estructuras *ExchangePull* y *URLAndOptionalHash* tal y como las proporcionó el cliente.
- SAML pull: Al igual que en los casos anteriores, el servidor debe responder especificando el campo *exchange_type* y las estructuras *ExchangePull* y *BindingAndEnpoints*. El servidor debe proporcionar exactamente el mismo detalle que proporcionó el cliente. Más adelante, el servidor proporcionará la información relativa al *Assertion Consumer*.

Consideraciones de seguridad

Como ya se ha comentado con anterioridad, si el servidor no soporta ninguno de los mecanismos proporcionados por el cliente, éste debe responder con una estructura vacía que indique la imposibilidad de utilizar alguno de los mecanismos de autorización proporcionados. Esto podría motivar una desconexión por parte del cliente, dado que en el instante en el que recibiera dicha estructura vacía, sabría que no podrá realizar el proceso de autorización.

Dado que la información de negociación se intercambia sobre un canal inseguro, cabría la posibilidad de que un atacante, situado en el medio, realizara un ataque de denegación de servicio basado en proporcionar siempre una estructura vacía como respuesta. Para evitar este tipo de ataques es necesario finalizar completamente el handshake, aunque finalmente no se pueda realizar el proceso de autorización. La razón es la siguiente:

Como se explica en la recomendación de TLS, el handshake termina cuando ambas partes cambian de un contexto inseguro a uno seguro (*Connection State*) negociado durante el handshake. Esto se realiza enviando el mensaje *ChangeCipherSpec* y posteriormente el mensaje *Finished*. Por tanto, el mensaje *Finished* es el primero en enviarse sobre el canal seguro, protegido con la clave generada durante el handshake. La sintaxis del mensaje *Finished* de TLS, según figura en la recomendación es la siguiente:

```
struct {
          opaque verify_data[12];
} Finished;

verify_data  PRF(master_secret, finished_label,
             MD5(handshake_messages) +
             SHA-1(handshake_messages)) [0..11];
```

Como puede verse, sobre el canal seguro, se envía una combinación de hashes MD5 y SHA-1 de los mensajes de handshake. Con esta información, ambas partes pueden comprobar la integridad de los mensajes de handshake simplemente ejecutando la función PRF sobre el la combinación de hashes de las trazas que conserva la capa de record.

Dado que, según la recomendación, los datos de la extensión deben incluirse en el cálculo de hashes del handshake, ambas partes deberán esperar al fin del handshake antes de desconectar. Así podrán saber si se trata de un ataque de DoS. Éste puede considerarse el mejor esfuerzo para minimizar el impacto de este tipo de ataques.

Nuevos mensajes de protocolo para autorización

El hecho de añadir estos nuevos mensajes al protocolo no afecta a los servidores TLS estándares, dado que únicamente serán utilizados si previamente ambas partes lo han negociado mediante extensiones.

Todos los nuevos mensajes de protocolo son gestionados por la capa de gestión de privilegios de forma que, se descarga al protocolo de handshake de la responsabilidad de procesar la información de autorización.

Negociación de la autorización sobre canal seguro

Como se describió en la sección 7.2, una vez se ha establecido el canal seguro, si el cliente solicitó una negociación sobre un canal seguro, ambas partes deben intercambiar mensajes de protocolo con el campo *ContentType* adecuado para que la capa record los distinga de sus otros clientes. Estos mensajes utilizarán la sintaxis de *AuthorizationAdvertising* para así alcanzar un acuerdo respecto al mecanismo de autorización a utilizar. La forma de proceder es la misma que se explicó con anterioridad con la salvedad de que se negocia sobre canal seguro y en lugar de extensiones, se utilizan mensajes de protocolo.

Mensaje AuthorizationRequest

El servidor envía el mensaje de protocolo *AuthorizationRequest* al cliente para indicar que está preparado para recibir las credenciales, en el caso de push; o para indicar que procede a descargar las credenciales en el caso de pull. Parece obvio que el cliente debe ser informado en los casos en los que él tiene que enviar las credenciales al servidor, pero también debe ser informado para los mecanismos de tipo pull dado que algunos bindigns de SAML requieren intervención del usuario.

Este mensaje será enviado inmediatamente después del mensaje de *Finished* salvo en los casos en los que se haya negociado el mecanismo de autorización sobre canal seguro, en cuyo caso se enviará inmediatamente después del ultimo mensaje de negociación.

En la sección 7.2 se ha demostrado que, alcanzado este punto, no es posible estar involucrado en un ataque de tipo main-in-the-middle y que el impacto de los ataques de DoS se ha minimizado. La sintaxis del mensaje es la siguiente:

```
struct{
    AuthorizationType auth_type;
    ExchangeType      exchange_type;
    opaque service_asertion_consumer_url<0..2^16-1>; -- Opcional
}AuthorizationRequest;
```

Mensaje AuthorizationPush

El mensaje de protocolo *AuthorizationPush* es el utilizado por el cliente para enviar las credenciales cuando se haya seleccionado un mecanismo de tipo push. Este mensaje se envía tras la recepción de un mensaje de tipo *AuthorizationRequest* enviado por el servidor y únicamente si se trata de un mecanismo push. Para enviar la(s) credencial(es) al servidor, el cliente debe utilizar la siguiente estructura:

```
opaque ASN1.Cert<1..2^24-1>; opaque SAMLBase64Data<1..2^24-1>;

struct{
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT:  ASN.1 asn.1certs<0..2^24-1>;
        case SAML:  SAMLBase64Data saml_base64_data<1..2^24-1>;
    }
```

```
} AuthorizationPush;
```

Como podrá recordar el lector, el formato del campo *asn.1certs*, que puede ser una cadena de certificados de atributos individuales o una cadena de pares "certificado de identidad - certificado de atributos", ya ha sido establecido con anterioridad.

Mensaje Cancel

Este mensaje se utiliza para cancelar el intercambio de autorización y puede ser enviado por cualesquiera de las partes, en cualquier momento, una vez finalizado el handshake:

```
struct{
    AlertDescription description;
    opaque displayMessage<0..2^16-1>;
}Cancel;
```

El campo *description* puede tomar cualquiera de los valores definidos para el protocolo de Alert en la recomendación [53], menos el de *access_denied*, ya que para indicar acceso denegado se utiliza otro mensaje diferente. Los mensajes de cancel atienden a otros motivos de cancelación. Un valor posible para el campo description puede ser, por ejemplo, *decode_error*. El campo *displayMessage* puede contener información para mostrar al usuario o para propósitos de log.

Mensaje NotEntitled

Este mensaje se utiliza por el servidor para indicar al cliente que la información de autorización proporcionada no es suficiente para concederle acceso al servicio o recurso. Este mensaje lo puede enviar el servidor únicamente si, tras verificar las credenciales, el usuario no está autorizado para utilizar el servicio. La estructura es la siguiente:

```
enum{
    attribute_descritor_ac(0), role_descriptor_ac(1),
    XACML_policy(2)...
} DataType;

struct{
    DataType  dataType;
```

```
    opaque data<0..2^16-1>;
    opaque displayMessage<0..2^16-1>;
}NotEntitled;
```

El servidor puede proporcionar información adicional en el campo *data*, así como un mensaje para mostrar al usuario en *displayMessage*. En *data*, el servidor puede proporcionar la siguiente información:

- Un certificado de atributos de tipo "descriptor de atributos" que consiste, como consta en [50], en un certificado que contiene las políticas aplicables así como las reglas de utilización de un atributo en concreto, identificado mediante su identificador de objeto (OID).

  Este atributo se emite por una SoA para que los *Privilege Verifiers* dispongan de las reglas adecuadas para procesar un atributo en concreto.

- Un certificado de atributos de tipo "descripción de rol" que contiene los privilegios, políticas y reglas aplicables a un rol identificado por su nombre.

- Una política o trozo de política (policy item) que deniegue el acceso al recurso.

El contenido del campo *data*, puede ser utilizado como policy item de forma que el cliente pueda determinar qué necesita para acceder al servicio y solicitar, a quien corresponda, las credenciales.

El campo *displayMessage* permite enviar un mensaje de texto al usuario.

Mensaje Ready

Este mensaje lo envía el servidor para indicar que la autorización ha sido satisfactoria. Este mensaje será enviado por el servidor una vez las credenciales se hayan verificado y sean adecuadas. La estructura es la siguiente:

```
struct{
    opaque displayMessage<0..2^16-1>;
}Ready;
```

Contiene, opcionalmente, un mensaje en el campo *displayMessage* para ser mostrado al usuario. El mensaje puede ser algún mensaje incluido en el certificado de atributos utilizado, por ejemplo, un mensaje incluido en una extensión de tipo *User notice extension* (id-ce-userNotice), o cualquier tipo de mensaje de bienvenida sin mayor relevancia.

Mensaje TimeOut

Este mensaje se utiliza para indicar que la conexión se cerrará debido a que el tiempo máximo de respuesta ha sido superado. La estructura es la misma que la del mensaje *Cancel* con la salvedad de que el valor utilizado para el campo *description* deberá indicar timeout[3].

Comparación con trabajos relacionados

En [84], S. Farrel, define un enfoque diferente para dar soporte al problema de la autorización en TLS. Farrel introduce cambios en los mensajes de protocolo de TLS añadiendo algunos nuevos sin que su utilización esté sujeta a una negociación previa mediante extensiones. Esta forma de proceder no es la adecuada puesto que no permite compatibilidad hacia atrás. Su utilización exigiría el cambio de todas las implementaciones de TLS dado que los servidores TLS estándar no serían capaces de continuar ante la aparición de un mensaje no estándar y desconocido. Por otro lado únicamente soporta certificados de atributos y sólo en modo push.

En [85], M. Brown and R. Housely, describen una modificación de TLS mediante extensiones similar a la presentada en esta sección. Ambos trabajos, el de Brown y el descrito en esta sección, se desarrollaron al mismo tiempo. El draft fue enviado a la IETF por primera vez en marzo de 2006, y la primera versión de las extensiones aquí descritas fueron enviadas al "ACM Workshop in wireless security" en Julio de 2005 como work in progress. El trabajo de Brown-Housley es similar, dado que utiliza extensiones, pero tiene varias diferencias respecto al aquí descrito:

1. El trabajo descrito en el draft no identifica los módulos de software que deben manejar la autorización, sino que se deja al protocolo de handshake la tarea de gestionar tanto la autenticación como la autorización.

2. Para negociar la autorización sobre un canal seguro, en el draft se propone la utilización de un doble handshake. El primero de ellos establece un canal seguro, por lo que no se utilizarán extensiones de autorización en él. En el segundo handshake se utilizarán extensiones de autorización, ahora sobre un canal seguro.

   Un handshake doble exige el intercambio del doble de mensajes para obtener el mimo resultado que obtenemos nosotros. El doble handshake exige el intercambio de al menos 12 mensajes más, en el caso de que se exija autenticación mutua que, como es lógico, se exigirá.

[3] El valor debería ser proporcionado por la IANA

Por otro lado debe generarse dos veces una clave y todo ello antes de que el cliente pueda conocer o no si se aceptan sus credenciales. Por otro lado no queda claro que ocurriría si el certificado utilizado por el servidor, en el primero de los handshakes, fuera diferente al segundo handshake.

Además el hecho de recurrir a un doble handshake puede afectar a aplicaciones multimedia, sin mencionar la energía desperdiciada en los cálculos criptográficos duplicados.

3. En su propuesta se permite el uso de SAML pero con diferencias sustanciales. En primer lugar, no se proporciona toda la información necesaria para completar la ejecución de un binding dado que no se puede especificar ni el tipo de binding, ni la URL del Assertion Consummer. El mecanismo de pull para SAML se limita a una URL de donde descargar la hoja XML, lo que podría desembocar en ataques de Reply.

En la propuesta descrita en esta sección, identificamos los módulos encargados de gestionar la autorización, no requerimos un doble handshake y permitimos una mayor flexibilidad en la negociación. Por otro lado, si las aplicaciones lo requirieran, éstas podrían especificar durante el handshake información extendida acerca de sus mecanismos, de forma que, puedan conocer mucho antes si podrán acceder o no al servicio.

El soporte dado en nuestra propuesta a SAML es mucho más completo, dado que permitimos especificar no solo el binding de SAML a utilizar, sino también la dirección del assertion consumer endpoint. Además soporta SAML anónimo.

Comparación con las necesidades de negociación de confianza

La solución presentada en la sección 7.2 es una primera aproximación a la negociación de confianza. Si bien es cierto que sólo proporciona soporte a la autorización, la propuesta permite una negociación flexible del mecanismo de autorización a utilizar.

Las carencia fundamental de esta solución y de las similares (sección 7.2) es que sólo permiten la utilización de un mecanismo de autenticación, que es el utilizado en TLS. Ya sea mediante certificados de clave pública, mediante certificados de curvas elípticas o mediante Diffie-Hellman... toda credencial de autorización, que pretenda ser utilizada con el mecanismo de autorización seleccionado, deberá estar vinculada a la credencial utilizada durante la autenticación ya que, en otro caso, el servidor no podría verificar correctamente la información de autorización.

Por otro lado, la autorización no se puede hacer de forma escalonada salvo que se ejecute el proceso de handshake varias veces y, en cualquier caso, sólo permite el

uso de certificados de atributos o SAML. Además el servidor sólo puede expresar sus preferencias mediante la selección del mecanismos preferido.

Esta solución podría considerar relativamente ingenua (naive), tal y como se razonó en el capítulo 5, dado que el cliente muestra todas sus cartas y el servidor selecciona aquella que más le conviene, revelando así el cliente más información de la que debería.

7.3. Extensiones para soporte de emisión de certificados de atributos

Esta sección presenta el formato de mensaje para la emisión de certificados de atributos así como una descripción de su uso con TLS de forma que se pueda automatizar la emisión de credenciales de autorización.

Escenarios y motivaciones para la emisión dinámica

Esta sección presenta el formato de mensaje para una solicitud de certificado de atributos, en inglés **Attribute Certificate Request Message Format** y cuyas siglas son ACRM. La propuesta aquí expuesta está basada en la recomendación RFC2511 [107], que describe el formato de mensaje para la solicitud de emisión de certificados de clave pública, con cambios para el soporte de certificados de atributos. Además se han considerado tres tipos diferentes de autorización:

- Emisión directa: una entidad **A** solicita un certificado de atributos directamente a la fuente de autorización, o *SoA*, o en otro caso, a través de una autoridad de registro, *Registration Authority (RA)*. La *RA* es la entidad que debe comprobar la petición antes de que el certificado de atributos (AC) sea emitido por la *SoA*. Obviamente, la *SoA*, deberá confiar en la *RA* para aceptar solicitudes provenientes de la *RA*. En aquellas circunstancias en las que el acceso a un servicio, y por tanto la emisión de una credencial que autorice su uso, requiera el pago de una cantidad, la *RA*, previo paso a la emisión, deberá comprobar que el pago ha sido realizado: el pago es un requisito y la credencial el ticket, entrada o pase que permite acceso al servicio.
- Emisión indirecta: una entidad **B** solicita un certificado de atributos a una *SoA*, o a través de una *RA*, en beneficio de otra entidad **A**. Este tipo de delegación indirecta puede ser útil para dispositivos limitados que no pueden realizar complejas operaciones criptográficas o no son capaces de construir la solicitud.

- Delegación: se produce cuando una entidad **A** solicita un certificado de atributos a otra entidad **B** con la intención de poder acceder a un servicio al que **B** puede acceder. Si **B** acepta, se dice que "delega" el privilegio de acceder al servicio a **A**.

La RFC2511 describe el formato de mensaje utilizado para solicitar un certificado de clave pública a una Autoridad de Certificación, en adelante CA. En dicha recomendación, el cliente solicita un certificado de clave pública vinculado a una clave pública, que ha generado y al que sólo él tiene acceso. El cliente demuestra que tiene acceso a la clave privada mediante una prueba de posesión, *Proof of Possession o PoP*, de la clave privada asociada a dicha clave pública. El vínculo entre la identidad y la clave pública se establecerá mediante la inclusión de la clave pública en el campo *Subject Public Key Information* del certificado. Si el certificado identifica a una persona, para su uso ante la administración, otro requisito pudiera ser requerir la presencia física de dicha persona ante una entidad que determine que "es quien dice ser".

La prueba de posesión puede realizarse de varias maneras, dependiendo del tipo de clave: usando firma, si las claves lo permiten; mediante el uso de un mensaje de autenticación (MAC) protegido por contraseña, si de claves de intercambio sin posibilidad de firma se trata; o con un intercambio de mensajes de reto y respuesta al reto. La *RA*, o *CA* en el caso de los certificados de clave pública, deben verificar la prueba de posesión contenida en el mensaje de solicitud. Dado que un certificado de atributos vincula privilegios a identidades, según el estándar [50], el ACRM deberá contener dicho certificado de identidad PKI. Por tanto, el ACRM deberá contener una prueba de posesión de la clave privada, asociada a la clave pública contenida en el certificado PKI, de forma que se pueda determina si se otorgan los privilegios a la entidad correcta.

Aunque el formato de ACRM, como se verá a continuación, es muy similar al propuesto en la RFC2511 para certificados de clave pública, proponemos cambios sutiles que permiten cubrir el **carácter dinámico de la autorización**. El formato de los mensajes de solicitud de certificados de clave pública, proporciona la expresividad necesaria para que el usuario demuestre la posesión de la clave privada en el mensaje, pero no se especifica cómo el usuario envía la solicitud a la RA o CA. Esto se hace de otra manera, por ejemplo, a través de una página web u otros mecanismos, dado que en pocas ocasiones se producen cambios en la identidad del sujeto.

En cambio, para soportar correctamente la dinamicidad de la autorización, en esta propuesta describimos mecanismos no solo para comunicar una solicitud a una *RA*; sino para que las entidades intermedias puedan firmar y envolver una petición comunicando así su conformidad a la autoridad de registro *RA*. De esta manera, un sistema de pago puede dar su conformidad, si se ha ingresado la cantidad correspondiente; otra entidad dar su aprobación respecto a la delegación; o permitir, que cualquier entidad involucrada en la toma de decisiones, exprese su conformidad. La figura 7.3 muestra

un ejemplo.

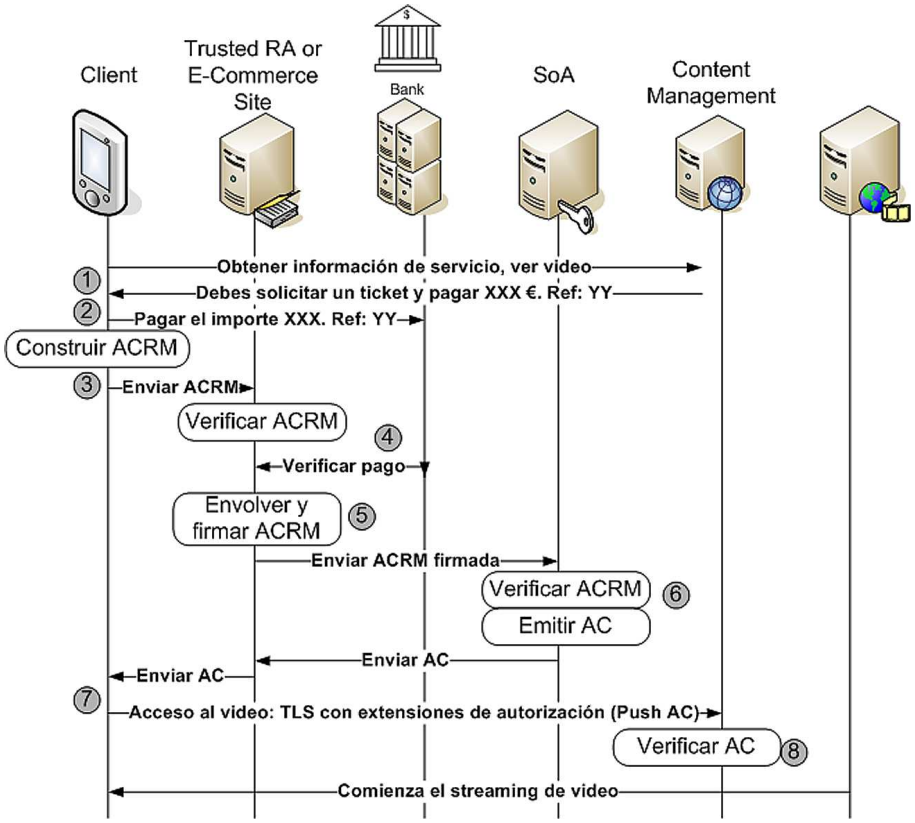


Figura 7.3: Esquema de acceso a un video bajo demanda que muestra las distintas entidades involucradas en la emisión de la autorización, el proceso de solicitud y acceso al servicio.

El ejemplo de la figura 7.3 ilustra la idea. Para una mejor comprensión del objetivo de esta propuesta, se describe a continuación el intercambio de mensajes en el orden que marcan los círculos numerados de dicha figura:

1. El cliente obtiene mediante descubrimiento de servicios; a través de una política (o policy item) durante una negociación de confianza; gracias a otros medios... información sobre los requisitos necesarios para acceder. El servicio requiere un certificado de atributos cuya emisión está supeditada al pago de una cantidad de dinero.

2. El cliente realiza el pago del servicio usando algún tipo de referencia que le haya proporcionado el servicio.

3. Construye una solicitud de certificado de atributo (ACRM) que asocia a su identidad el privilegio de acceder al servicio y la envía a la autoridad de registro (RA).

4. La RA verifica el ACRM y el pago.

5. Al estar conforme con el ACRM, y comprobar que se ha realizado el pago, la RA expresa su conformidad envolviendo la ACRM y firmándola. Posteriormente la envía a la SoA.

6. La SoA, que mantiene una relación de confianza con la RA, verifica la firma de ésta, emite un certificado de atributos que autoriza al cliente al uso del servicio y lo envía al cliente.

7. Finalmente el cliente, utilizando, por ejemplo, las extensiones descritas en la sección 7.2, se autentica y autoriza ante el servicio.

8. El servicio verifica las credenciales y da acceso al cliente.

Attribute Certificate Request Format

La sintaxis en ASN.1 del **"Attribute Certificate Request Message"** es la que se presenta a continuación[4]:

- La estructura *AttCertReqMessages* contiene una secuencia de solicitudes de certificado de atributos. Cada una de esas solicitudes puede estar envuelta y firmada por entidades intermedias para expresar su conformidad.

```
AttCertReqMessages ::= SEQUENCE SIZE (1..MAX) OF
AttCertSignedRequestMsg
```

- La estructura *AttCertSignedRequestMsg* contiene una solicitud o una solicitud envuelta y firmada una o varias veces. Permite encadenar firmas, de diferentes entidades, que envuelvan al ARCM.

[4]la sintaxis en ASN.1 expuesta utiliza tagging implícito e importa todas las definiciones importadas en la RFC 2511 además de PKIXAttributeCertificate : iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) pkix(7) id-mod(0) id-mod-attribute-cert(12)

```
AttCertSignedRequestMsg ::= SEQUENCE {


    reqMsg  CHOICE {
          attCertReqMsg [0]  AttCertRequestMsg,
          attCertSignedReqMsg [1] AttCertSignedRequestMsg
          }

    authInfo CHOICE {
          signer        [0] GeneralName,
          delegationAttCertPath   SEQUENCE SIZE (1..MAX) OF ACPathData
          } OPTIONAL,

    signatureAlgorithm   AlgorithmIdentifier OPTIONAL,
    signatureValue       BIT STRING OPTIONAL

    } -- fin AttCertSignedRequestMsg
```

Dentro de la estructura, encontramos los siguientes campos:

- *reqMsg*: Contiene la solicitud como tal, es decir, una estructura de tipo *attCertReqMsg* o *attCertSignedReqMsg*. Como puede apreciarse, la definición es recursiva. Esta definición recursiva de *AttCertSignedRequestMsg* permite a otras entidades firmar y envolver una solicitud cuando están de acuerdo con el contenido.
- *authInfo*: es un campo opcional que debe ser rellenado cuando se quiera encadenar una firma. Por ejemplo, antes de que la RA envíe una solicitud de otra entidad a la SoA, la RA deberá incluir sus datos en esta estructura, indicando su identidad en el campo *signer*; posteriormente aseverará la solicitud mediante la inclusión de una firma y sus detalles en los campos *signatureValue* y *signatureAlgorithm*.

  Cuando la emisión se solicita por motivos de delegación, se crea una solicitud especificando el(los) permiso(s) en concreto. Aquel que delega su(s) permiso(s), debe incluir en *delegationAttCertPath* la cadena de certificados de atributos que contiene(n) dicho(s) permiso(s).

  Esta forma de proceder en la delegación es de vital importancia. Consideremos el siguiente supuesto: una entidad dispone de un certificado de atributos en el que se expresa un conjunto de permisos y desea delegar **uno** de esos permisos a otra entidad. Para ello, construye

una solicitud de emisión para ese permiso en concreto y se incluye en *reqMsg/authInfo/attCertReqMsg*. Posteriormente, la entidad que delega el permiso, deberá incluir en *delegationAttCertPath*, la cadena de certificados de atributos que contiene al menos al permiso que se delega y en *reqMsg/authInfo/signer* la información del certificado de clave pública con el que firmará. Finalmente, deberá firmar toda la estructura, que delega el permiso, e incluir dicha firma en *signatureValue*.

La estructura *ACPathData*, cuyo formato puede verse bajo estas líneas, contiene una cadena de pares "certificado de clave pública - certificado de atributos".

```
ACPathData  ::=  SEQUENCE {
    certificate                [0]  Certificate  OPTIONAL,
    attributeCertificate       [1]  AttributeCertificate  OPTIONAL
}
```

- Los campos *signatureValue* y *signatureAlgorithm* recogen los datos de la firma cuando proceda.

Consideremos un ejemplo en el que se solicita una emisión indirecta. La entidad que delega indirectamente un privilegio debe incluir su identidad, es decir, su certificado de clave pública; firmar y envolver la solicitud para aseverar su contenido; e incluir la cadena de certificados de atributos que demuestre que ostenta dicho privilegio o privilegios. La figura 7.4 puede ilustrar la idea.

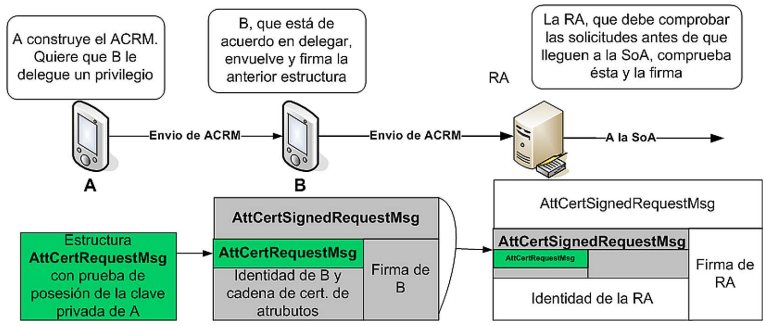


Figura 7.4: Ejemplo de solicitud indirecta de certificado de atributos. Muestra como un ACRM se envuelve y firma por las diferentes entidades involucradas en el proceso.

- La estructura *AttCertRequestMsg* contiene un mensaje con la solicitud de certificado de atributos (*attCertReq*); una prueba de posesión (*pop*), que es opcional dado que podría realizarse mediante otros mecanismos; un campo para información de registro (*regInfo*), que puede ser utilizado para información de pago,

publicidad de la credencial... Dicha estructura de información de registro existe en la RFC 2511 y debe ser considerado en los mismos términos que en dicha recomendación. A continuación se presenta el formato de la estructura y una breve explicación de sus campos:

```
AttCertRequestMsg :: = SEQUENCE {
    attCertReq          AttCertRequest,
    pop                 ProofOfPossesion OPTIONAL,
    regInfo             SEQUENCE SIZE (1..MAX) of AttributeType And Value OPTIONAL
    -- regInfo: can contain information of publishing, payment...
}
```

- La estructura *AttrCertRequest* contiene, en orden, un identificador de solicitud; un certificado de clave pública, al que vincular los permisos, que es opcional, dado que pueden vincularse a un objeto identificado por un hash [50]; un template para la descripción del certificado de atributos a emitir; y un campo de control.

  ```
  AttrCertRequest ::= SEQUENCE {
      attCertReqID        Integer,
      boundCert           Certificates OPTIONAL,
      -- PKI certificate of entity
      attCertTmpl         AttCertTemplate,
      controls            Controls OPTIONAL
  }
  ```

- El campo *boundCert* incluye el certificado de cliente y opcionalmente una ruta de certificación.

  ```
  Certificates ::= SEQUENCE {
      userCertificate     Certificate,
      certificationPath   CertPath OPTIONAL
  }
  ```

- La estructura del template, de tipo *AttCertTemplate*, incluye toda la información necesaria para la emisión del certificado. Las diferencias con la RFC 2511 son la sintaxis de los campos *Holder*, *AttCertIssuer* y *Attribute*, cuya sintaxis es la misma que se describe en [50] y [108]. El resto de los campos deben ser tratados como en la recomendación RFC 2511.

  ```
  AttCertTemplate ::= SEQUENCE {
     version      [0]  Version OPTIONAL,
  ```

```
    serialNumber [1]  Integer OPTIONAL,
    signatureAlg [2]  AlgorithmIdentifier OPTIONAL,
    holder       [4]  Holder OPTIONAL,
    issuer       [3]  AttCertIssuer OPTIONAL,
    -- syntax for issuer and holder
    validity     [5]  Validity OPTIONAL,
    attributes   [6]  SEQUENCE SIZE(1..MAX) of Attribute OPTIONAL,
    extensions   [7]  SEQUENCE SIZE(1..MAX) of Extension OPTIONAL
}
```

- En la RFC 2511 se considera la posibilidad de enviar la clave pública a la autoridad que emite el certificado. En nuestro caso, el certificado a emitir no contiene clave pública sino una referencia a un certificado de clave pública. Por esa razón, simplemente una firma o una demostración de intercambio de claves es suficiente para demostrar la posesión de la clave. Esta demostración de posesión de clave es necesaria dado que la entidad utilizará dicha clave para autenticarse (identificarse) y así poder hacer uso de los privilegios vinculados a la identidad.

  La prueba de posesión, en adelante PoP, dependerá del tipo de clave cuya posesión quiere ser probada. Por ejemplo, con una clave RSA, que permite tanto firma como intercambio de claves, se puede utilizar cualquier tipo de PoP: *signature* o *keyAgreement*. Sin embargo, para todas aquellas claves que sólo pueden usarse para intercambio de claves, sólo se puede utilizar *keyAgreement*. En ese caso, el campo *thisMessage* contendrá el resultado de ejecutar un código de autenticación de mensaje (MAC), protegido por contraseña, sobre el campo *attCertReq* de la estructura *AttCertReqMsg* codificado en DER. La clave utilizada para la MAC se basa en una clave derivada de los parámetros privados de la clave Diffie-Hellman de la entidad y de los públicos de la CA. El mecanismo para generar dicha clave puede encontrarse en la recomendación RFC 2511 [107].

  ```
  ProofOfPossession ::= CHOICE {
      signature         [0] POPOSigningKey,
      keyAgreement      [1] POPOPrivKey
  }

  POPOSigningKey ::= SEQUENCE {
      algorithmIdentifier     AlgorithmIdentifier,
      signature               BIT STRING
  -- signature MUST be computed on the DER-encoded
  -- value of attCertReq
  ```

```
    }


POPOPrivKey ::= CHOICE {
    thisMessage       [0] BIT STRING,
    subsequentMessage [1] SubsequentMessage,
-- possession will be proven in a subsequent message
    dhMAC             [2] BIT STRING }
```

El campo *subsequentMessage* se utiliza para indicar que la prueba de posesión se realizará en un mensaje posterior. Este campo se incluye también en la RFC 2511 y debe ser tratado de la misma forma.

Extensión de TLS para el soporte de emisión dinámica

En la sección anterior, se presentaba la sintaxis de la solicitud de certificados de atributos. En esta sección presentamos una extensión que permite el uso de TLS para la emisión de certificados de atributos, de esta manera, una entidad puede conectarse de forma segura a una autoridad de registro (RA) y solicitar un certificado de atributos.

Como un certificado de clave pública, al que se asocia una serie de privilegios, puede contener claves distintas a RSA, esta extensión proporciona cobertura a este tipo de claves de forma que pueda demostrar la posesión mediante mecanismos alternativos a la firma. Las figuras 7.5(a), 7.5(b), 7.5(a) y 7.5(b) muestran cuatro posibles escenarios donde se utiliza esta extensión sobre TLS. En todos los escenarios el cliente comienza el proceso de handshake de TLS enviando el mensaje de clientHello acompañado de una extensión de tipo *AuthzRequestExtension*, cuyo formato se muestra bajo esta líneas.

```
struct{
    Boolean secureExchange,
    Boolean useClientCert,
    opaque  targetSoA -- Optional
}AuthzRequestExt;
```

Mediante el uso de esta extensión, el cliente puede descubrir si el servidor soporta la emisión de certificados de atributos sobre TLS. Si el servidor entiende la extensión, devolverá el mismo contenido al cliente en una extensión, acompañando al serverHello.

Si el cliente solicita que los datos de la solicitud de autorización se envíen sobre un canal seguro, debe marcar el campo *secureExchange* de *AuthzRequestExtension* co-

mo true: indicando así su intención de solicitar un certificado de atributos pero que la información relativa a la solicitud se proporcionará más adelante, una vez establecido el canal seguro. Para ello, más adelante se utilizará un mensaje de tipo Attribute Certificate Request Message (ACRM).

Dado que es decisión del servidor solicitar el certificado de cliente o no, si el cliente desea utilizar **TLS con autenticación mutua** para demostrar la posesión de la clave, éste deberá poner a true el campo *useClientCert*, obligando así al servidor a solicitar su certificado de cliente. De esta manera, si el certificado de atributos que se solicita va vinculado al certificado de clave pública usado durante el handshake por el cliente, la RA no necesitará que el ACRM contenga una prueba de posesión. Esto se debe a que un proceso de handshake de TLS con autenticación mutua que termina de forma satisfactoria, es suficiente para demostrar que el cliente posee la clave.

Por otro lado, para distinguir entre diferentes SoAs utilizadas por la misma RA, se incluye el campo *targetSoA*. Este campo ayuda a seleccionar la SoA adecuada de entre todas las posibles. Si el cliente requiere intercambio seguro no deberá utilizar ningún campo de la extensión salvo el campo *secureExchange*.

Mensajes de protocolo

Una vez el cliente ha determinado que el servidor entiende la extensión y opcionalmente ha proporcionado cierta información a la RA durante el handshake; establecido el canal seguro, puede intercambiar mensajes para transmitir la solicitud a la RA. Para ello se utilizará un nuevo mensaje de protocolo que se detalla a continuación:

```
struct{
    ACReqType acReqType,
    opaque    ACRMs<0..2^16-1>
}ACRequest;

enum{ SendACRM(0), ForwardApprovedACRM(1),
 RequestKeyExchange(2),..
}

struct{
    opaque payload<1..2^16-1>
}challenge_Response;

struct{
    opaque  ASN1.AC<1..2^16-1>,
```

```
    Boolean encrypted
}ACResponse;
```

El mensaje *ACRequest* puede tener tres usos diferentes dependiendo del valor del campo *acReqType*:

- *SendACRM*: El mensaje se utiliza para enviar un mensaje ACRM con motivos de solicitud.
- *ForwardApprovedACRM*: Lo usa la RA para enviar un mensaje a la SoA conteniendo una ACRM aprobada y firmada por la RA. Otras entidades que desee aseverar el contenido pueden hacer uso de este tipo de mensajes (emisión indirecta o delegación).
- *RequestKeyExchange*: Solicita intercambio de claves para demostración de posesión de clave.

El mensaje *ACResponse* contiene el certificado de atributos solicitado. Opcionalmente, el certificado puede estar cifrado. El tipo de cifrado y codificación puede negociarse mediante el uso de SubsequentMessage (ACRM).

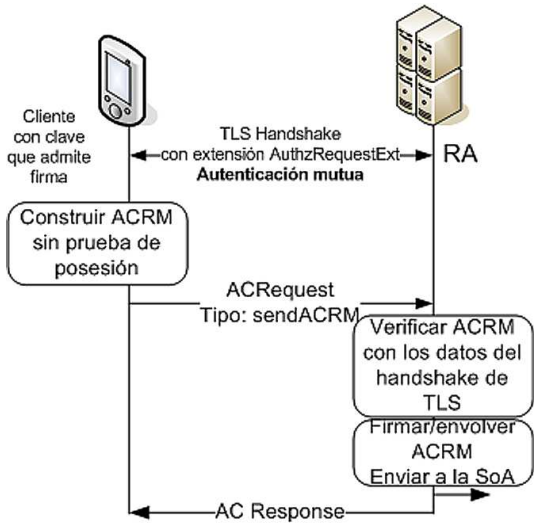
Ejemplos de utilización

La figura 7.5(a) muestra un intercambio de mensajes sobre TLS para emisión de certificados de atributos. En este caso, la posesión de la clave (clave con posibilidad de firma) se prueba durante el handshake de TLS, lo que obliga a que el servidor solicite al cliente un certificado durante el handshake. Por esta razón, no es necesario incluir una prueba de posesión en el ACRM.

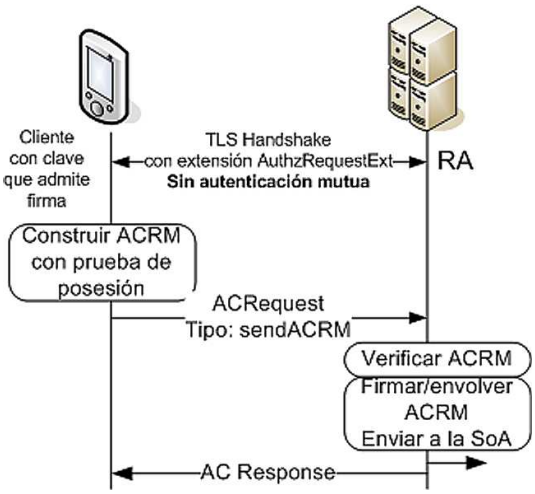
La figura 7.5(b) muestra un intercambio de mensajes en los que el cliente, ya sea por privacidad u otros motivos, no demuestra la posesión de la clave durante el handshake y por esta razón no se realiza autenticación mutua en TLS. La prueba de posesión (clave con posibilidad de firma) se incluye en el ACRM mediante una firma.

La figura 7.5(a) muestra un escenario donde un cliente envía un mensaje de tipo *ACRequest* conteniendo un ACRM vacío. El tipo especificado para el *ACRequest* es *RequestKeyExchange*. El cliente solicita, por tanto, una prueba de posesión basada en intercambio de claves, dado que su clave es de tipo Diffie-Hellman.

Inmediatamente a este mensaje, el cliente envía un *ClientKeyExchange* de TLS usando *ClientDiffieHellmanPublic* [53]. A continuación, el cliente envía un mensaje *Finalize* de TLS y espera respuesta del servidor. Estos mensajes permiten establecer



(a) Solicitud de certificado de atributos en la que se utiliza como prueba de posesión la autenticación mutua realizada durante el handshake de TLS



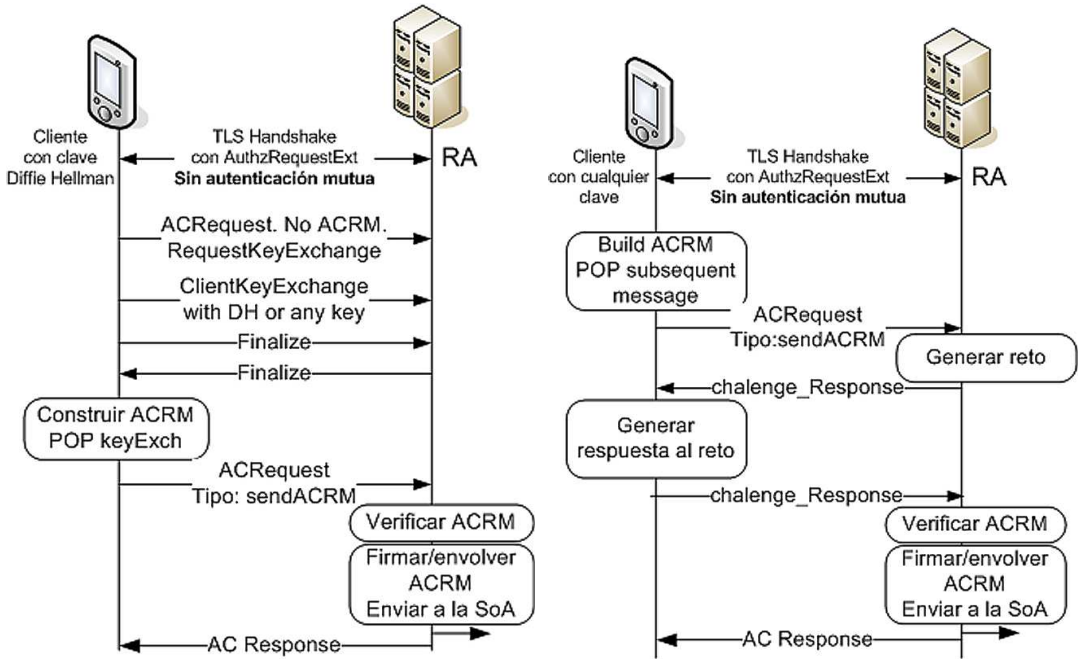
(b) Solicitud de certificado de atributos sobre TLS sin autenticación mutua. El cliente incluye la prueba de posesión directamente en la ACRM en lugar de durante el handshake por cuestiones de privacidad

una clave entre ambos lados mediante el uso de Diffie-Hellman. El proceso está descrito ampliamente en la recomendación de TLS [53].

Es necesario hacer unas precisiones respecto al uso de los mensajes de *ClientKeyExchange* y *Finalize*. Dichos mensajes pueden ser intercambiados fuera de orden dado que previamente ambas partes **han negociado el uso de dichos mensajes mediante extensiones**. Por otro lado, esos mensajes no los consumirá la capa de handshake de TLS, sino una capa de gestión de privilegios; pese a ello, el intercambio de claves usará los datos aleatorios intercambiados por ambas partes durante el handshake. El procedimiento para el cálculo de la clave maestra (*master secret*) usando Diffie-Hellman con los datos del handshake de TLS se describe en [52]. Este tipo de prueba de posesión deberá ser usado únicamente por clientes que no dispongan de capacidad de firma en sus claves.

Finalmente, el cliente puede solicitar un proceso de reto-respuesta usando el campo *SubsequentMessage* del ACRM (ver figura 7.5(b)). El mensaje de reto-respuesta es *challenge_Response*. Además, el campo *SubsequentMessage* se puede utilizar para solicitar que el certificado de atributos se devuelva cifrado. La prueba de posesión, en este caso, se realiza enviando un hash del certificado recibido una vez descifrado, cosa que sólo puede hacer aquel que tiene la clave correcta. Para transmitir dicho hash del

certificado descifrado a la RA, el cliente puede utilizar el mensaje *challenge_Response* conteniendo el hash en su carga útil.



(a) Solicitud de AC con TLS en la que se utiliza un mecanismo de intercambio de clave como prueba de posesión

(b) Solicitud de AC con TLS en la que se utiliza un mecanismo de Challenge-Response como prueba de posesión

7.4. Extensión de TLS para soporte de negociación de confianza

Una vez discutida la necesidad de soportar una amplia variedad de credenciales durante un proceso de negociación (capítulo 5); tras no satisfacer completamente los requisitos de negociación de confianza en la sección 7.2; vamos a proponer una variación, sobre la solución propuesta en la sección 7.2, que soporte multiples credenciales.

Los sistemas de negociación de confianza descritos en la literatura, como SAML, SPKI o KeyNote, no proporcionan un protocolo dedicado, o en su defecto, una extensión de un protocolo conocido para el soporte de intercambio de credenciales. Para cubrir la ausencia de un protocolo que permita autenticación y autorización múltiple; en múltiples pasos; así como negociación de confianza; proponemos en esa sección

una extensión a TLS [53] que lo permite. Los cambios aquí descritos pueden ser incorporados además a DTLS [81] y a TLS sobre SCTP.

Las modificaciones aquí propuestas permiten al cliente enviar credenciales al servidor o especificar dónde pueden ser obtenidas. Además definiremos una serie de mensajes de protocolo agnósticos, en el sentido de que no están diseñados para una credencial en concreto. Estos mensajes permitirán liberar de forma gradual credenciales; y liberar gradualmente, así como combinar mediante operadores lógicos, distintos trozos de política o policy items, sin importar su naturaleza. El espacio de nombres, ontología y por tanto el significado de los operadores puede ser negociado también.

Para proporcionar esta funcionalidad sobre TLS sin romper la compatibilidad hacia atrás se usarán extensiones que permitan negociar el uso de nuevos mensajes de protocolo.

Arquitectura propuesta

Al igual que en la sección 7.2, utilizamos extensiones en los mensajes del protocolo de handshake *Client Hello* y *Server Hello*. Además, del mismo modo en que se procedió en la mencionada sección, se define una nueva capa, cliente del protocolo de record de TLS, que bautizamos como *Privilege & trust negotiation Layer*, PTN Layer (PTNL) o capa PTN para abreviar. Esta capa consume un nuevo conjunto de mensajes de protocolo que permiten no solo el intercambio de información de negociación, sino la negociación de confianza. La figura 7.5 muestra la arquitectura propuesta.

El proceso de autorización y negociación de confianza sobre TLS comprende tres pasos:

1. El soporte de autorización y de negociación de confianza se negocia mediante extensiones. La motivación del uso de estas extensiones es la de detectar si el otro lado comprende y soporta esta extensión, así como conocer los mecanismos en concreto que soporta.

2. El segundo paso es opcional y sólo debe ser utilizado por las aplicaciones que no deseen negociar nada, ni siquiera los mecanismos, en el protocolo de handshake. Este paso obliga a reproducir la negociación de mecanismos una vez el canal seguro está establecido.

3. El tercer paso corresponde al intercambio de mensajes consumidos por las capas PTN de ambos extremos que serán utilizados para intercambiar credenciales de autorización o negociar confianza.

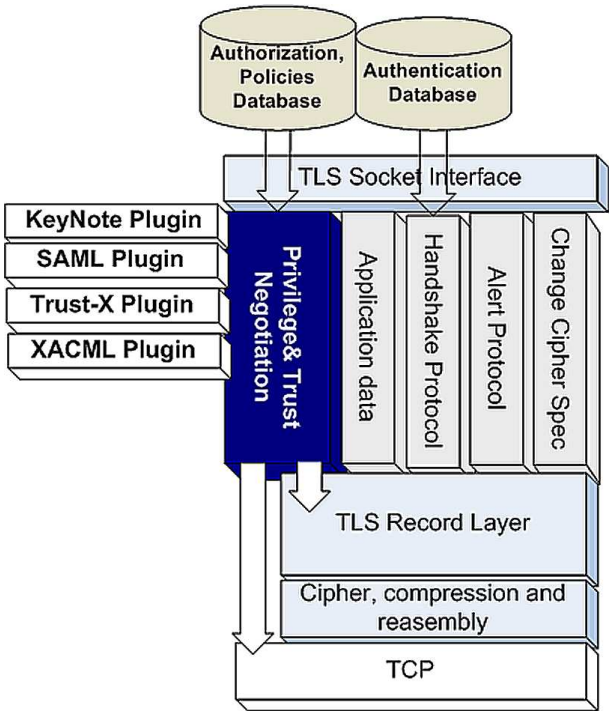


Figura 7.5: Arquitectura de TLS para Negociación de confianza, que incluye una nueva capa para la gestión de los mensajes de negociación de confianza.

Extensión de TLS para negociación de confianza y autorización

La extensión de TLS aquí descrita permite negociar mecanismos para enviar información de autorización al servidor, mecanismos "push"; o para indicar al servidor dónde debe recoger dicha información, mecanismos "pull". La extensión permite además negociar la confianza entre las dos partes. En realidad esta propuesta no es nueva en si misma, es decir, comprende ciertos cambios que habilitan la negociación de confianza, pero mantiene los mecanismos de pull y push ya comentados por una razón muy concreta: ¿Qué ocurre cuando ya se ha negociado previamente con una entidad?

La idea general es la de utilizar la funcionalidad de negociación de confianza cuando no se conoce a la otra parte o cuando se necesitan escalar privilegios. Una vez que se ha conseguido llegar a un estado de seguridad dado, en el que se garantizan una serie de privilegios, tras varias autenticaciones y procesos de autorización, se debe tratar de obtener un **"Trust Ticket"**. La función de este ticket es la de ahorrarnos la negociación en futuras interacciones con la entidad como ya se discutió en la sección 2.2. Otra razón es la de utilizar dicha credencial en otros servicios del mismo dominio si soporta Single Sign On o Identidad Federada.

Pero ¿Qué es un "Trust ticket"? En general un trust ticket es información de autorización que expresa una serie de privilegios alcanzados y que, por seguridad, deberá estar vinculada a la autenticación más rigurosa, de mayor calidad o más segura, de las usadas durante la negociación. Para determinar a que credencial de autenticación se debe vincular dicha información se puede utilizar una medida de calidad obtenida en términos inversos de riesgo, es decir, la credencial más adecuada es aquella que permite alcanzar el recurso de mayor riesgo de entre los alcanzables.

Por lo tanto, la extensión permite negociar o bien utilizar los resultados de una negociación previa, sin perjuicio de que posteriores utilizaciones del servicio requieran negociación. La entidad puede requerir volver a negociar por cambios en la política, condiciones de contorno o por incremento de la calidad solicitada.

Al igual que en la sección 7.2, las extensiones aquí descritas permite al servidor seleccionar el mecanismo de autenticación más adecuado. En este caso, se amplía la negociación para recoger la negociación de políticas, operadores y tipos de credenciales a utilizar. Del mismo modo, se permite negociar tanto sobre canal seguro como durante el handshake de TLS, de forma que la aplicación pueda encontrar el equilibrio adecuado entre privacidad y latencia, evitando siempre, como en la otra solución, que las credenciales como tal se envíen durante el handshake de TLS.

El orden de intercambio de mensajes en esta solución es equivalente al de la solución descrita en la sección 7.2 con las siguiente apreciaciones:

- Si el protocolo se utiliza únicamente para autorización, el intercambio de mensajes es el mismo. La única diferencia en este caso es que, además del soporte para certificados de atributos y SAML, se proporciona una **estructura genérica que permite la utilización de cualquier credencial**.
- Si el protocolo se utiliza para negociación de confianza, que no excluye la autorización, el intercambio de mensajes entre las capas PTN de ambos extremos no se reduce a un par de mensajes; sino que puede comprender tantos intercambios de mensajes como sean necesarios hasta llegar a un estado satisfactorio para ambas partes, o hasta concluir que no es posible la interacción.

Sintaxis de la extensión PTNL

La extensión *PTNL* comunica los mecanismos de autorización soportados por el cliente así como los detalles de los objetos que el cliente puede utilizar para la negociación de confianza. Al igual que en la propuesta de la sección 7.2, se deja al cliente la decisión de negociar sobre canal seguro, así como la decisión de proporcionar mayor o menor detalle respecto a la descripción de los elementos involucrados.

La sintaxis de la extensión, en la que puede apreciarse los cambios, es la que sigue:

```
struct {
    Boolean secureExchangeRequired;
    AuthorizationMechanism Authz_avail_mechs<0..2^16>; //--Opcional
    TrustNegotiationMechanisms TrustN_avail_mechs<0..2^16>; //--Opcional
}PTNLExtension
```

La estructura principal de la extensión muestra un cambio respecto a la otra propuesta. En este caso se incluye un campo opcional de tipo *TrustNegotiationMechanisms* que proporcionará la información necesaria para la negociación de confianza.

Por otro lado, respecto a la información de autorización, puede apreciarse bajo estas líneas que se incluye una estructura genérica, *GeneralAuthzInformation*, que da soporte a cualquier tipo de credencial de autorización o "Trust ticket".

```
struct{
    AuthorizationType auth_type;
    ExchangeType      exchange_type;
    select (exchange_type)
    {   case push: ExchangePush;
        case pull: ExchangePull;
    }extended_information;
} AuthorizationMechanism;

enum{ push(0), pull(1) } ExchangeType;

enum{ ATTRCERT(0), SAML(1), OTHER(3)... }AuthorizationType;

struct {
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT:  AttrCertInformation attr_cert_info;
        case SAML:  AssertionsInformation assertions_info;
        case OTHER: GeneralAuthzInformation generic_info;
    }
} ExchangePush;

 struct {
```

```
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT: URLAndOptionalHash url_hash<1..2^16-1>;
        case SAML: BindingAndEndPoints binding_info<1..2^16-1>;
        case OTHER: GeneralAuthzInformation generic_info<1..2^16-1>;
    }
} ExchangePull;

struct{
    AuthzType     authz_type;
    opaque auth_payload<0..2^16-1>;
} GeneralAuthzInformation;

struct{
    opaque OID<0..2^16-1>;
    opaque URI<0..2^16-1>; /* Optional*/
}AuthzType;
```

Por tanto, ya sea para pull o push, el cliente debe indicar, en el campo *auth_type*, el tipo de credencial de autorización que utilizará. El soporte para certificados de atributos y SAML es formalmente idéntico al de la anterior propuesta. En este caso, en cambio, cualquier otra credencial puede ser utilizada, identificándola mediante un identificador de objeto (OID) o mediante una URI. Se permite además, la inclusión de datos adicionales que puede expresar cualquier tipo de particularidad sobre la credencial, desde estructura; hasta detalles de bindings; URLs donde conseguirla; o directamente una descripción mediante WSDL de un servicio web que la proporcione. Lo único se exige para su procesado es que se identifique correctamente mediante una URI o OID de forma que la otra parte pueda determinar si puede o no procesarla. El resto de las estructuras no se incluyen, dado que son idénticas a las utilizadas por la otra solución.

```
struct {
    TrustObjectType trust_object_type;
    opaque     Uri<0..2^16-1>;
    opaque     OID<0..2^16-1>;
} TrustNegotiationMechanisms

enum{
    policy(0), credential(1), strategy(2),...
```

```
} TrustObjectType;
```

La lista de objetos para negociación de confianza, campo *TrustN_avail_mechs*, está compuesta de estructuras de tipo *TrustNegotiationMechanisms*, que permiten expresar qué políticas, credenciales e incluso estrategias u operadores se soportan. El cliente, de esta manera, puede expresar, por ejemplo, que puede procesar políticas escritas en XACML; credenciales de tipo X.509, SAML y KeyNote; y que comprende los operadores lógicos recogidos en un determinado espacio de nombres identificado mediante una URI.

El identificador de objeto OID o URI debe identificar unívocamente al objeto. Mediante esta estructura agnóstica, se pueden soportar nuevas credenciales, políticas... sin necesidad de modificar la sintaxis.

Consideraciones de seguridad en el Handshake

Las mismas consideraciones para evitar los ataques de man-in-the-middle y minimizar el impacto de los ataques de denegación de servicio (DoS), discutidas en la sección 7.2, pueden aplicarse a esta propuesta.

Nuevos mensajes de protocolo para autorización

Los mensajes de protocolo utilizados en esta propuesta que son: *AuthorizationRequest*, *AuthorizationPush*, *Cancel*, *NotEntitled*, *Ready* y *TimeOut*, son idénticos a los descritos en la propuesta de la sección 7.2, salvo el mensaje de *AuthorizationPush* que permite además el envío de cualquier otra credencial de autorización distinta de certificados de atributos y asertos SAML.

```
struct{
    AuthorizationType auth_type;
    select(auth_type)
    {
        case ATTRCERT:  ASN.1 asn.1certs<0..2^24-1>;
        case SAML:  SAMLBase64Data saml_base64_data<1..2^24-1>;
        case OTHER: opaque<1..2^24-1>
    }
} AuthorizationPush;
```

Nuevos mensajes de protocolo para negociación de confianza

La negociación de confianza se entiende como un intercambio de información entre dos entidades para lograr alcanzar un estado de confianza mutua. Para simplificar, llamaremos "servidor" a la entidad que dispone del recurso y "cliente" a aquel que accederá al recurso.

La estructura de los mensajes es la siguiente:

```
struct{
    Message assertion_list<0..2^16-1>;
    Message requirement_list<0..2^16-1>;
    Message information_list<0..2^16-1>;
}TrustNegotiationMessage;

struct{
    Type type;
    opaque      PayLoad<0..2^16-1>;
    Parameter   Parameters<0..2^16-1>;
}Message;

struct{
    Type type;
    opaque  PayLoad<0..2^16-1>;
}Parameter;

struct{
    opaque OID<0..2^16-1>;
    opaque URI<0..2^16-1>;
}Type;
```

El servidor liberará detalles de la política o policy items utilizando la lista de requisitos *requirement_list* y se los enviará al cliente, el cual tratará de satisfacerlos mediante el envío de credenciales utilizando la lista de asertos *assertion_list*.

En cualquier momento, como se razonó en la sección 2.2, el cliente puede intercambiar el rol con el servidor. De esta manera, el cliente puede solicitar al servidor el cumplimiento de determinado requisito como medida de protección ante abusos. Por lo tanto, ambas partes podrán solicitar y enviar datos a la otra. También se permite el envío de información, como texto, que pueda ser mostrada al usuario o almacenada en un log.

Consideremos un ejemplo: Bob y Alice están involucrados en una negociación de confianza. Bob dispone de un recurso al que Alice desea acceder. Alice envía a Bob, durante el handshake, una lista de los lenguajes de políticas que entiende, una lista de los tipos de credenciales que es capaz de gestionar así como los operadores que soporta. Bob selecciona, de entre la información proporcionada por Alice, aquellos lenguajes de políticas, tipos de credenciales y operadores o estrategias que comprende o soporta. Entonces, Bob envía sobre el canal seguro un mensaje de tipo *TrustNegotiationMessage* con tres argumentos: el primero, según su URI, corresponde a una política XACML, que llamaremos P1 y contiene un parámetro. El parámetro indica "predicado" en una de las ontologías o espacios de nombres previamente negociados y contiene una carga en XML que indica NOT. El segundo mensaje identifica, mediante una URI, un operador lógico con una carga en XML que indica OR. El tercer parámetro indica una política o policy item de otro tipo que llamaremos P2. Por lo tanto Bob requiere de Alice: $!P1|P2$.

La capa PTN de Alice procesa la información utilizando pluggings de diferentes tipos de políticas y determina que dispone de un par de credenciales que pueden ser utilizadas para satisfacer los requisitos de Bob. Por tanto, Alice envía a Bob un mensaje con dos argumentos en la lista *assertion_list*: una credencial KeyNote y una SAML.

Finalmente, Bob envía a Alice un requisito para que Alice genere un par de claves pública/privada y solicita también que Alice le envíe la clave pública. Una vez Bob recibe la clave pública de Alice, genera un certificado de atributos asociado a dicha clave, con unos atributos que la autorizarán directamente, hasta su vencimiento, a acceder al recurso sin tener que negociar otra vez.

Comparación con trabajos relacionados

Respecto a los trabajos relacionados en la sección 7.2, en esta propuesta se aporta además la capacidad de negociar confianza utilizando expresiones genéricas (agnósticas), lo que permite que la solución no se cierre a unos tipo de credencial en concreto.

Respecto a soluciones que soporten negociación de confianza, en [101] se describe un sistema de negociación de confianza sobre TLS. Dicho trabajo introduce al menos un doble handshake; no identifica la capa de TLS que debe procesar esos mensajes, dejando el trabajo a la capa de Handshake; y sólo permite un intercambio, tras cada handshake, de credenciales. Por otro lado, sólo soporta certificados PKI y utiliza mensajes estándares de tipo CertificateVerify fuera de orden, que son consumidos por el protocolo de handshake, lo que requeriría cambios profundos en el funcionamiento del protocolo actual.

En la propuesta descrita en esta sección, se describe un mecanismo que permite

negociar en primer lugar los espacios de nombres, tipos de políticas y credenciales a utilizar, durante el handshake o una vez establecido el canal seguro. Una vez se ha establecido el canal seguro se permite el intercambio de todos los mensajes que sean necesarios para lograr el objetivo. Esta negociación no requiere un nuevo handshake. Además, la sintaxis flexible de la estructura permite utilizar y combinar cualquier política, policy item, credencial u operados, escrito en cualquier lenguaje y codificado en ASN1, XML o cualquier otro.

7.5. Uso de las extensiones para acceso a la red

En las secciones anteriores de este capítulo se han propuesto diversas extensiones a TLS que permiten el uso de este protocolo en tareas de autorización, incluyendo la emisión de credenciales y negociación de confianza.

La elección de TLS se justifica en la sección 7.1 aduciendo diversos motivos de estructura y despliegue, entre los que cabe destacar la posibilidad de usar TLS sobre EAP.

TLS sobre EAP permite el uso de TLS sobre protocolos de acceso a la red de nivel superior como, por ejemplo, 802.1x y PANA. En la sección 2.3 se repasa ampliamente el protocolo PANA, dado que se encuentra en proceso de definición y aún quedan problemas por resolver.

TLS sobre EAP, y a su vez sobre PANA, permite autenticación mutua punto de acceso-cliente, así como generar claves. La posibilidad de generar claves permite protección contra los ataques descritos en [23], que afectan a los mecanismos de autenticación que no generan claves que son ejecutados sobre túneles. Con las extensiones presentadas en las secciones anteriores, una entidad podría, si dispusiese de la infraestructura mostrada en la figura 7.6, por ejemplo, realizar las siguientes tareas:

1. Si la entidad que accede a la red, en adelante cliente, es desconocida, no encuentra un operador con el que tenga acuerdo o simplemente necesita unas características de tráfico, que ningún otro puede le proporcionar, podría usar la negociación de confianza para obtener las credenciales adecuadas.

   Supongamos en ese caso que el cliente es desconocido en primera instancia para la red; utiliza EAP-TLS sobre PANA e intercambia las credenciales y políticas necesarias para acceder a un recurso; el recurso es la red; se dan unas determinadas condiciones de contexto; y se demandan unas calidades específicas. Si el cliente puede obtener acceso mediante negociación de confianza pese a que no tenía acuerdo previo.

2. Para que en otra ocasión el proceso no sea tan lento, es decir, no implique el intercambio de un gran número de credenciales entre las partes, el cliente puede solicitar la emisión de una credencial que resuma la negociación anterior. Dicha credencial podría usarse con otros puntos de acceso del mismo operador para acelerar futuros accesos.

   Para ello, una vez ambas partes han negociado y alcanzado un estado de seguridad satisfactorio, el cliente conecta de nuevo con el punto de acceso, mediante EAP-TLS sobre PANA o conecta con un servidor de autenticación/autorización que hace las veces de autoridad de registro. Durante el handshake utiliza la extensión de TLS que permite la emisión de certificados de atributos. De esta manera, el cliente genera una ACRM y recibe de la red una credencial (certificado de atributos) que le permite acceder a la red sin necesidad de negociar otra vez.

3. El cliente se desplaza y descubre otro punto de acceso del operador con el que negoció previamente y al que solicitó un certificado de atributos.

   El mecanismo de selección de red del cliente busca una red adecuada para las necesidades actuales y determina que el punto de acceso descubierto cubre las necesidades. Entonces, el cliente, utilizando EAP-TLS sobre PANA, conecta con el punto de acceso y mediante las extensiones descritas proporciona al punto de acceso una credencial, mediante push, que le autoriza al uso de la red.

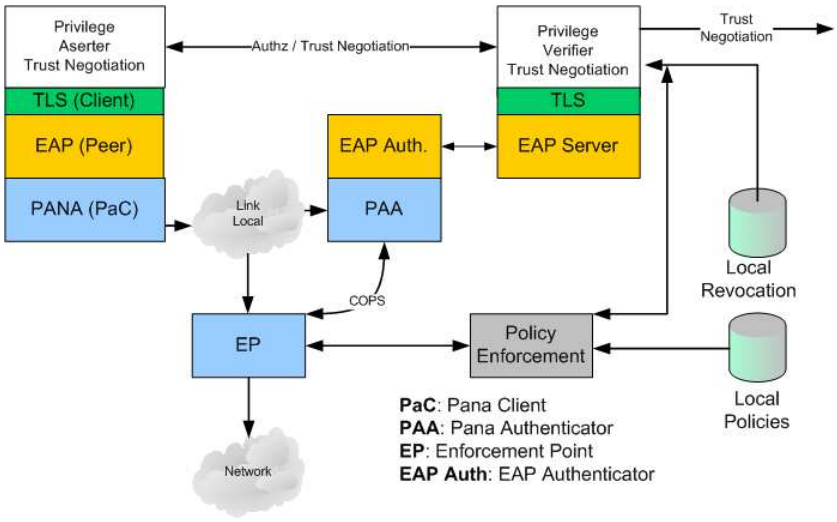


Figura 7.6: Estructura, entidades involucradas y protocolos en un sistema de acceso a la red flexible, que permite autorización y negociación de confianza.

El uso de un protocolo extensible como TLS, sobre un protocolo, también extensible como EAP, permite gran libertad y flexibilidad en tareas de acceso a la red. Sin duda,

este tipo de esquemas presentan ventajas de escalabilidad que no tienen parangón si las comparamos con los más cercanos al nivel de enlace como WEP.

CAPÍTULO

8 Conclusiones y líneas de investigación futuras

La computación ubicua es una realidad cotidiana: cada vez existen más dispositivos con capacidad de cómputo que nos rodean y nos facilitan la vida de forma nunca antes pensada. Por otro lado las tecnologías radio ha evolucionado dotando a todos estos dispositivos de capacidad de comunicación avanzada encontrándose, en dichos dispositivos, muy habitualmente, más de dos interfaces de red de diferente tecnología.

Las economías de escala han abaratado los costes de producción del hardware programable, permitiendo esta diversidad de tecnologías de red. En estos momentos nos encontramos cerca de dar el salto hacia lo que serán las futuras tecnologías de red como, por ejemplo, Long Term Evolution, Ultra Wide Bandwidth y Wimax. Sólo el tiempo dirá qué tecnologías serán más usadas o adoptadas comercializarse por operadores; y cuáles serán relegadas al uso doméstico. Con independencia de esto, al aumentar de forma tan considerable el ancho de banda y el alcance, nos encontraremos en una situación diferente a la que hemos vivido hasta el momento.

Dadas las grandes areas de cobertura proporcionadas por las tecnologías y la convergencia de los operadores tradicionales hacia núcleos de IP, se puede dibujar un futuro con cuatro grandes actores: el usuario, el proveedor de acceso a la red o NAP, el proveedor de transporte y el proveedor de servicio. Pese a que en muchas ocasiones se producirán integraciones aguas abajo o aguas arriba, de forma que la misma marca comercial instancie varios roles, es de suponer que serán necesarios nuevos mecanismos de selección de red, decisión y apoyo al control de acceso, así como modificaciones de protocolos, que permitan orquestar todo este sistema con altísima entropía.

En esta tesis se ha contribuido en varios escenarios de dicho sistema en aras de una integración natural que no podría realizarse con los mecanismos actuales por problemas de escalabilidad. Se ha planteado una estructura o cadena de proveedores de diferentes servicios, entre los que pueden existir relaciones cliente-proveedor dos a dos; se

han analizado y descrito las perspectivas que todos los involucrados pueden tener respecto al control de acceso; se han analizado protocolos en proceso de formalización y otros de sobrada solvencia para acceso a la red; se han analizado mecanismos de simplificación, conceptualización y comunicación de decisiones y riesgos de forma que, rozando la interpretación psicológica, se consiga la integración de varias tecnologías, dotando de visión global al control de acceso.

8.1. Principales contribuciones

Contrastando el resultado de la investigación con los objetivos planteados a su inicio, se puede afirmar que se han conseguido alcanzar dichos objetivos de forma satisfactoria. A continuación enumeramos las contribuciones de esta tesis.

Contribuciones técnicas

Las principales contribuciones de la tesis tienen carácter técnico, las de tipo estudio son: estudio de las perspectivas de control de acceso distribuido en un sistema multiproveedor estructurado; estudio de las credenciales, políticas y negociación de confianza; estudio de los protocolos de autenticación y autorización para el acceso a redes y servicios; estudio de los algoritmos de procesado y simplificación de información multivariable.

Por otro lado, enumeramos a continuación aquellas consistentes en propuestas y validaciones: propuesta de mecanismos para la mejora del control de acceso en terminales; validación del mecanismo de selección de red con modelo teórico y caso práctico; diseño de un mecanismo agnóstico de asistencia a la decisión en negociación de confianza; validación del mecanismo de asistencia a la negociación de confianza; diseño de una extensión de un protocolo estándar para el soporte de autorización; diseño de una extensión de un protocolo estándar para el soporte de emisión de credenciales de autorización; diseño de una extensión de un protocolo estándar para el soporte de negociación de confianza.

A continuación pasamos a describir las contribuciones enumeradas en el párrafo anterior:

1. **Estudio de las perspectivas de control de acceso distribuido en un sistema multiproveedor estructurado**

Los escenarios en los que se ha basado la investigación se han caracterizado teniendo en cuenta la evolución de las telecomunicaciones y el incremento de la capacidad de cómputo de los dispositivos. En estos escenarios se supone:

- Una gran población de dispositivos móviles o dispositivos personales.
- Una gran cantidad de dispositivos integrados o empotrados en la arquitectura que ofrecen servicios.
- Demandas intermitentes de conexión, variables en calidad, ancho de banda. La cobertura a la movilidad también puede variar, dependiendo de la velocidad del terminal.
- Múltiples tecnologías de red integradas en los dispositivos.
- Amplia oferta de acceso a la red a través de diferentes tecnologías.
- Multitud de aplicaciones corriendo en el dispositivo y demandando acceso a la red.

En estos escenarios distinguimos cuatro actores:

- Equipo de usuario: aquellos dispositivos personales que acompañan al usuario como la PDA o el teléfono móvil, el ordenador portátil, ropa inteligente...
- Proveedores de acceso a la red o Network Access Provider (NAP): proporcionan acceso capilar a la red mediante una o varias tecnologías. Hay que tener en cuenta que el servicio proporcionado no tiene por qué ser universal.
- Proveedores de transporte o proveedores de servicios de Internet (ISPs): estos proveedores utilizan varios NAPs para lograr acceso capilar al equipo de usuario, proporcionando el transporte entre el equipo de usuario y los servicios.
- Servicios de terceros: aquellos servicios de datos, multimedia u otros que utilizan las redes de los ISPs, e indirectamente la de los NAPs para llegar al usuario.

Cada uno de los actores descritos anteriormente tiene una visión diferente del control de acceso dado que tienen diferentes recursos que proteger.

En el capítulo 2, en concreto en la sección 2.1, se identifican:

- Los requisitos de control de acceso del equipo de usuario estableciendo una división en dos grupos. El control de acceso de fuera hacia dentro, que protege al equipo de usuario de ataque externos y el control de acceso de dentro hacia fuera, que protege al equipo de usuario del uso indebido de los recursos radio.

- Las amenazas externas de las que debe protegerse el terminal, como son el acceso de dispositivos no confiables, posibles ataques de DoS, ataques de recopilación de información sensible o de localización...
- Las amenazas internas, es decir las provenientes del interior del terminal, como son la utilización de redes inseguras por parte de las aplicaciones, revelación de información sensible, gestión ineficiente del roaming, consumos de batería inecesarios...

En el mismo capítulo, en la sección 2.1, se razona sobre las ventajas e inconvenientes de los mecanismos de autenticación y autorización concluyendo que es adecuado realizar la autenticación a niveles superiores de la torre de protocolos.

En dicha sección, se caracterizan dos escenarios antagónicos, en uno de ellos se carece por completo de flexibilidad, requiriendo relaciones de confianza fuertes entre proveedores para el soporte de roaming, lo que desemboca en el uso de unas credenciales establecidas de antemano. En el otro, se permite una gran flexibilidad y todo ha de negociarse, de forma que se requiere un gran intercambio de paquetes entre las partes involucradas para el acceso a la red. Este último entorno no requiere relaciones de confianza fuertes entre proveedores, sino capacidad para alcanzar acuerdos de forma dinámica. La solución pasa por implantar mecanismos balanceados que permitan la negociación entre entidades, hasta alcanzar un acuerdo, a partir del cual se genera una credencial que se utilizará para agilizar el proceso en posteriores interacciones.

En la sección 2.2 se introduce la negociación de confianza para llegar a acuerdos entre entidades de forma justa. También se propone el uso de los "Trust Tickets" para agilizar el proceso en posteriores interacciones dado que se emite una credencial una vez se ha negociado satisfactoriamente. Más adelante, en el capítulo 5 se razona sobre las necesidades de la negociación de confianza en escenarios complejos como el descrito.

En la sección 2.1 se discute sobre la necesidad de autenticación y autorización extremo a extremo con independencia de que las comunicaciones subyacentes estén protegidas.

Asimismo, en el resto de los capítulos se analizan los requisitos pormenorizados de estos entornos, a la vez que se proponen soluciones que den cobertura a las necesidades de todos los actores.

2. **Estudio de las credenciales, políticas y negociación de confianza**

   En la sección 2.1, se define el control de acceso como el proceso desencadenado para dar una respuesta adecuada a la pregunta: ¿puede el usuario o entidad acceder a un recurso en concreto dadas unas ciertas restricciones?

En general se puede decir que todo control de acceso tiene siempre varios procesos asociados que resultan de dividir la pregunta subpreguntas: ¿qué usuario trata de acceder al recurso?,¿qué privilegios tiene asignados dicho usuario?, conocidos los derechos del usuario, ¿se pueden considerar suficientes para permitir el acceso al recurso?

El proceso que da respuesta a la primera pregunta es el de autenticación o identificación. La segunda pregunta se responde mediante autorización y la tercera por intervención de una política.

Durante la investigación se han utilizado numerosas credenciales, analizado su comportamiento y su eficiencia en diferentes entornos. Se ha utilizando OpenSSL [109], PKCS#11 y cryptoAPI como APIs criptográficos. En la sección 2.2, se resumen las credenciales más utilizadas en la actualidad y aquellas que tienen relevancia para la investigación, atendiendo a su naturaleza. Salvando las diferentes codificaciones utilizadas por las credenciales, como ASN1 o XML, éstas se pueden clasificar atendiendo a cómo identifican a las entidades o cómo asignan privilegios a las mismas, dado que todas necesitan un espacio de nombres global. En la sección 2.2 se discute sobre las ventajas e inconvenientes de utilizar diferentes espacios de nombres globales, concluyendo que todas las credenciales de uso extendido, son eficientes en algún escenario concreto y que no es posible encontrar una credencial que pueda sustituir eficientemente a todas las alternativas actuales.

Por otro lado, las políticas, que son las encargadas de informar de los requisitos que afectan a los recursos, y por tanto, de comunicar la sensibilidad de un recurso, son analizadas en la sección 2.2 conjuntamente con la negociación de confianza. En dicha sección se concluye que:

- Existe una gran diversidad de credenciales. Es difícil prescindir de algunas o encontrar una que cubra todas las necesidades.
- Existe una gran cantidad de lenguajes de políticas, capaces de describir las sensibilidades más complejas, así como de integrar la información de contexto en el proceso de decisión, siendo complicado el uso de un único lenguaje.

Por tanto, es necesario recurrir a sistemas de negociación de confianza que permitan el uso de multiples mecanismos de autenticación y autorización, de forma ordenada y escalonada. Mediante la utilización de sistemas de negociación de confianza, logramos:

- Permitir el acceso a desconocidos o aquellos que no tienen relaciones de confianza preestablecidas.

- Establecer calidad de servicio en la autenticación y la autorización.
- Realizar una acceso escalonado sólo dependiente de las necesidades de las aplicaciones en cada momento.

3. **Estudio de los protocolos de autenticación y autorización para el acceso a redes y servicios**

   Como se mencionó anteriormente, existen multitud de tipos de credenciales cada una de ellas con un ámbito de aplicación. En muchas ocasiones, determinadas credenciales se popularizan por su idoneidad para determinados servicios.

   Con los protocolos para el acceso a la red y servicios ocurre algo similar. Mientras que para el acceso a los servicios, extremo a extremo, se emplea TLS, que suele depender de PKI, los protocolos de acceso a la red dependen típicamente del hardware subyacente y de claves simétricas.

   En la sección 2.3 se discute acerca de la conveniencia de realizar la autenticación en diferentes niveles de la torre de protocolos. Mientras que las soluciones más cercanas al nivel de enlace abaratan costes, las mas cercanas a la capa de red o de transporte presentan una mayor flexibilidad.

   Claramente, en un escenario en el cual no existen muchas tecnologías de acceso a la red, la autenticación a niveles inferiores resulta más atractiva, dado que es un problema abordable por el usuario medio (éste debe aprender a configurar un número limitado de redes y es más barato de fabricar). Pese a ello, la distribución de parches para subsanar errores, de implementación o de diseño, es complicada debido a su cercanía al nivel físico. Dentro de estos protocolos podríamos encontrar WEP y WPA o los utilizados por Bluetooth para emparejar dispositivos.

   Por otro lado, en un escenario donde existen varias tecnologías de acceso a la red, el problema deja de ser abordable por el usuario medio. En este tipo de escenario es más recomendable el uso de mecanismos extensibles basados en EAP, como 802.1x, EAPOL, EAP-TLS. Esta soluciones son menos dependientes del hardware, por lo que es más sencillo de mantener, permite la integración con servidores AAA (authentication, authorization and accounting). Además, aquellos basados en EAP puede ser extendidos a otros algoritmos de autenticación más robustos.

   En la sección 2.3, se describe ampliamente PANA, un protocolo que funciona sobre UDP y proporciona transporte a EAP. Este protocolo es el candidato ideal, como se razona en dicha sección, para entornos en los que se encuentran disponibles muchas tecnologías de acceso a la red. La idea fundamental es que, dado que el factor común a toda conexión es IP, la utilización de PANA puede ayudar a simplificar el problema de la autenticación para el acceso a la red, ya que puede ser utilizado por cualquier tecnología de conexión.

En conclusión, cuanto mayor independencia presenta un protocolo frente al hardware, más útil será en entornos donde se hace uso de diferentes tecnologías de red para el acceso a los servicios.

En la sección 2.3 se analizan los protocolos de autenticación para acceso a los servicios. En concreto se analiza en detalle el protocolo TLS dado que es el más extendido para autenticación extremo a extremo. En dicha sección, se describe la facilidad de TLS para incorporar nuevas funcionalidades mediante extensiones.

Más adelante, en la sección 2.3, se analizan algunas propuestas de extensiones realizadas a TLS para añadir soporte de autorización dado que no existen esfuerzos importantes en dicha materia sobre otros protocolos. Dichas modificaciones de TLS, que habilitan el soporte conjunto de autenticación y autorización, pueden ser utilizadas incluso para el acceso a la red utilizando EAP-TLS sobre PANA o cualquier otro protocolo capaz de transportar EAP.

Más adelante, en el capítulo 7, se razona ampliamente sobre el tema, justificando convenientemente la utilización de TLS para autorización y para negociación de confianza.

4. **Estudio de los algoritmos de procesado y simplificación de información multivariable**

   Uno de los objetivos a cumplir por el control de acceso del equipo de usuario, de un servicio o de cualquier elemento de red, es el de analizar la información para la evaluación del riesgo así como para la toma de decisiones.

   Por esta razón, en la sección 2.4 se analizan las familias de algoritmos más utilizados en estadística multivariante entre los que se encuentran:

   - Multidimensional Scaling: ampliamente utilizado en ciencias del comportamiento, sicología y econometría. El algoritmo se usa para análisis cualitativo y cuantitativo de similitudes entre entidades.
   - Análisis factorial, dentro del que se encontraría el análisis de componentes principales. Este tipo de algoritmo está diseñado para el análisis de las correlaciones de las variables de las observaciones obtenidas midiendo aspectos del comportamiento o propiedades de un conjunto de entidades, mientras que MDS es una técnica orientada a analizar relaciones entre las entidades y no entre las variables.
   - Análisis de clusters: permite agrupar la información, pero no permite medir, como tal, las diferencias entre entidades o grupos.

   De entre todos, el algoritmo seleccionado es MDS, que permite no solo analizar los datos manteniendo las diferencias conceptuales, sino que, al estar orientado

a simplificar y visualizar problemas complejos, cumple el objetivo de representar el espacio de decisión y los riesgos, de forma comprensible, al usuario.

En los capítulos 3, 4, 5 y 6 se analiza en profundidad el algoritmo MDS, se proponen nuevos tipos de atributos y se estudia su capacidad de ajuste en diferentes modelos.

5. **Propuesta de mecanismos para la mejora del control de acceso en terminales**

   En la sección 2.1 se mostró la perspectiva que un equipo de usuario tiene del control de acceso. Los equipos de usuario deben ser protegidos desde fuera hacia dentro, así como de dentro hacia fuera, como en el caso de un gestor de conexiones basado en preferencias de usuario.

   Para ello se diseño una serie de mecanismos que permitieran una protección activa del dispositivo de usuario, descrito y formalizado en el capítulo 3. Con este conjunto de mecanismos, se ha intentado cubrir todas las necesidades de control de acceso en los terminales. Para ello en dicho capítulo:

   - Se describe el problema en detalle, identificando las necesidades específicas a cubrir.
   - Se propone una organización basada en dominios, que son grupos de dispositivos, y se establece un mecanismo para reconocerlos. Se propone una manera de caracterizar cada dominio a partir de las redes y servicios disponibles.
   - Se describe un mecanismo de valoración colaborativo de dominio que puede ser utilizado opcionalmente para obtener una medida de la confianza de un dominio.
   - Se propone un mecanismo de selección de red/par independiente de la tecnología, capaz de analizar similitudes ente entidades sin necesitar un modelo común de datos.
   - Se propone una estructura que permite la aplicación de políticas, embebiendo en el sistema operativo los puntos de aplicación; evitando así que ninguna aplicación pueda evitar las restricciones de la política.

A continuación valoraremos los resultados obtenidos frente a los objetivos planteados en el capítulo 1.2:

- Autónomo: el sistema de caracterización de dominio utiliza los interfaces de red como monitores, por lo que no depende de la red para obtener información. Además, no revela la posición ni la presencia del dispositivo.

- Independiente de la tecnología: el sistema de selección de red es capaz de procesar la información proporcionada por el sistema de caracterización de dominio, con independencia de que las entidades se describan con distinto número de atributos.
- Sin estado o estado mínimo: un dominio se puede reconocer mediante los dispositivos no móviles contenidos en él. Por otro lado, decisiones pasadas no tienen por qué afectan a la selección de red, por lo que no se considera estado alguno.
- Sensible al contexto: la información de contexto se procesa adecuadamente. Para ello se establecen las diferencias entre entidades respecto al contexto por la similitud al un ideal establecido que recoge las restricciones impuestas por el contexto.
- Colaborativo bajo demanda: se propone un mecanismo de marcado o valoración colaborativo, que puede ser utilizado por el terminal para determinar la confiabilidad de un dominio.
- Sensible a preferencias de usuario: las preferencias del usuario y del sistema, recogidas en forma de políticas, afectan a las decisiones dando mayor importancia a unos atributos frente a otros.

6. **Validación del mecanismo de selección de red con modelo teórico y caso práctico**

   El mecanismo de selección de red se ha validado mediante simulaciones con Matlab. El capítulo 4 recoge dos simulaciones que demuestran la utilidad del mecanismo. La primera es una demostración de concepto en la que se exageran los atributos de forma que se pueda apreciar la traslación de conceptos y la capacidad de trasmitir al usuario, de forma comprensible, los detalles del espacio de decisión.

   En la segunda simulación se aborda un caso real. En esta simulación no solo se demuestra la capacidad del algoritmo de seleccionar la red más adecuada a las necesidades, sino que se demuestra que el mecanismo se ajusta correctamente al modelo.

   En la simulación del caso real se simula la selección de varias redes con diversos requisitos de tráfico: redes confidenciales con una velocidad y calidad dadas, redes para el ocio, así como redes para realizar una llamada de emergencia.

   Para valorar el ajuste del modelo se estudia la variación de los parámetros de ajuste del algoritmo respecto al número de entidades consideradas. Para la valoración se recurre también al análisis de los autovalores y a una supervisión sobre

los resultados que demuestra cómo las preferencias del usuario, así como las políticas aplicables, permiten modular la selección adecuándola a las preferencias y al contexto.

Por último, se mide el rendimiento del algoritmo, con un resultado de complejidad de $O(n^{2,65})$.

7. **Diseño de un mecanismo agnóstico de asistencia a la decisión en negociación de confianza**

   De acuerdo a los objetivos planteados en el capítulo 1.2, en el capítulo 5, se propone un sistema para la asistencia a la negociación de confianza instanciable tanto por un dispositivo móvil, como por cualquier entidad involucrada en el control de acceso.

   La necesidad de este sistema se razona en la sección 2.2 y se detalla en el capítulo 5.2. Los grandes problemas que se tratan de resolver con este sistema son, a grandes rasgos:

   - Gestión de un creciente número de credenciales: Al aumentar los proveedores, los sistemas de control de acceso y la disponibilidad de servicios, es sencillo encontrarse en un maremágnum de credenciales, con distintos formatos, de difícil manejo por parte del usuario medio. Consensuar a todas las partes involucradas para imponer el uso de un único tipo de credencial es muy complicado. Por otro lado, cada credencial es adecuada en su ámbito de aplicación y sería complicado encontrar un único tipo de credencial que satisficiera todas las necesidades.
   - Gestión de los requisitos con elevado número de políticas: En los casos en los que el número y tipo de políticas que gobiernan un sistema crece, es imposible aplicar, de forma eficiente, todas las políticas en cada interacción. Habría que recorrer secuencialmente todos los motores de control de acceso para determinar si tienen algo "que decir". Por otro lado, tampoco es viable imponer un único tipo de política.
   - Necesidad de negociación: Existen situaciones en las que, pese a la falta de una relación de confianza previa entre dominios, se puede llegar a establecer un estado de seguridad o de confianza tras la presentación de varias credenciales. Es decir, si para acceder a un recurso es necesario disponer de la credencial A, es posible que, aquellos que no disponen de la credencial A como tal, puedan presentar, digamos, las credenciales B, C y D o las credenciales E y F. Esto se debe a que en algún lugar de la política se considera que: $A \approx (B\&C\&D)|(E\&F)$.

     Por ello una entidad, que en primera instancia se considera no confiable, puede llegar a un estado de seguridad o confianza, que le permita acceder

a un determinado recurso o conjunto de recursos mediante el intercambio de varias credenciales.

- Inclusión de información de contexto: Existen muchos lenguajes de políticas que permiten incluir información de contexto. Pero en los casos en los que multiples políticas, gestionadas por diferentes motores de control de acceso, se aplican sobre un recurso, es necesario considerar la información de contexto en global, atendiendo a las restricciones de las políticas, pero permitiendo al sistema que lo gestiona todo establecer o medir el riesgo del conjunto y tomar decisiones en consecuencia.
- Ayuda a la comprensión de riesgos por parte del usuario: se tratará en lo posible de comunicar el riesgo al usuario de forma comprensible, gráficamente, logrando así una mejor comprensión de los resultados de determinadas acciones. Para ello se deberá presentar al usuario el espacio de decisión de forma comprensible para él.

La solución propuesta en el capítulo 5, cubre las necesidades recogidas en dicho capítulo, proporcionando un mecanismo agnóstico capaz de manejar, sobre un mismo espacio de decisión, todas las entidades involucradas en el control de acceso. De esta forma los riesgos se valoran en conjunto. Permite a su vez aplicar las políticas en orden de sensibilidad, matizando las decisiones en función de la información de contexto de forma global. El sistema permite a su vez mostrar comprensiblemente el espacio de decisión al usuario.

8. **Validación del mecanismo de asistencia a la negociación de confianza**

   La validación se ha realizado mediante simulación con Matlab. El capítulo 6 recoge una simulación que demuestra la funcionalidad del mecanismo. En dicha simulación se muestra un caso hipotético de protección de recursos mediante varias políticas gestionadas por diferentes motores de control de acceso. El sistema de asistencia a la negociación de confianza media entre las entidades involucradas en el control de acceso; orquestando el intercambio de credenciales; dándole al sistema completo una visión global que no es posible hacer desde cada motor de control de acceso, dado que funcionan de forma separada.

   En la simulación se demuestra cómo los detalles de la política, que especifican cada requisito por separado, son liberados de forma gradual, evitando comprometer información sensible. Por otro lado, la simulación permite ver cómo las credenciales y políticas pueden ser tratados como recursos dado que su exposición al exterior implica un riesgo.

   Además, se demuestra cómo, simplemente midiendo distancias de los puntos al estado de la negociación, se puede evaluar el riesgo global de exponer un determinado recurso al exterior. Del mismo modo, permite comparar recursos en

cuanto al riesgo que supone su exposición, de forma que se puede informar al usuario, mediante comparaciones, siendo éstas adecuadas para la comprensión.

9. **Diseño de una extensión de un protocolo estándar para el soporte de autorización**

   Desde el inicio del documento, planteándose en la sección 2.3 y extendiéndose en los posteriores capítulos, se pone de manifiesto la necesidad de proporcionar mecanismos de autorización y protocolos de transporte de credenciales de autorización para incrementar la flexibilidad en el acceso a los servicios y la red. De esta manera, se evita la necesidad de establecer los permisos de forma local en cada servicio.

   Por este motivo, en el capítulo 7, se presentan una serie de extensiones a un protocolo estándar, en concreto TLS, para el soporte de nuevas funcionalidades. En dicho capítulo, en la sección 7.1, se discute sobre lo adecuado de elegir TLS para nuestros propósitos concluyendo: que TLS está suficientemente soportado en los terminales; que la confianza del usuario medio en dicho protocolo es alta, dado que se usa ampliamente en comercio electrónico; que puede ser extendido adecuadamente además de ser utilizado sobre EAP, lo que amplía su alcance como se expone en la sección 7.5.

   Una vez seleccionado TLS, se proponen unas modificaciones, compatibles hacia atrás con servidores que no soportan dichas extensiones y que permiten la utilización de certificados de atributos (ACs), así como SAML, para la comunicación de privilegios extremo a extremo. La razón de dar soporte a certificados de atributos la encontramos en lo cercano de estas credenciales a los certificados de clave pública, que típicamente acompañan al uso de TLS. Por otro lado, el uso de SAML proporciona una gran flexibilidad dado que SAML puede utilizar multitud de credenciales y reflejar complejos conjuntos de atributos para autorización.

   La extensión se propone estableciendo una nueva capa, cuya utilización por ambas partes es negociada durante el primer estadio del protocolo TLS (handshake), y que absorbe las tareas de autorización, liberando a la capa de handshake de una tarea para la cual no fue diseñada. Dado que las nuevas capas son negociadas durante el handshake, mediante el uso de la extensión, se respeta la compatibilidad hacia atrás.

   La extensión permite negociar el tipo de credenciales a utilizar así como cuál sera el mecanismo de transmisión o comunicación de dichas credenciales: proporcionada directamente por el cliente (“push”) o proporcionando el cliente la información necesaria para que el servidor pueda recogerla de otro lugar (“pull”).

Por otro lado, si el mecanismo de autorización es SAML, se proporcionan los detalles necesarios para la correcta utilización del "SAML protocol binding"´y se permite también, el uso de SAML con anonimato.

Por otro lado, en la sección 7.2, se compara esta propuesta con otras similares, resaltando cómo esta propuesta resuelve muchos problemas presentes en la otras propuestas, como pueden ser el uso de un doble handshake y el laxo soporte a SAML. Además, en la sección 7.2, se compara la propuesta con las necesidades de negociación de confianza, llegando a la conclusión de que, pese a que permite la utilización de autorización, que valdría para el uso de "trust tickets" (ver sección 2.2), no soporta negociación de confianza como tal. Por esa razón, más adelante se proporciona otra modificación, que si permite la negociación de confianza.

10. **Diseño de una extensión de un protocolo estándar para el soporte de emisión de credenciales de autorización**

Como ya se ha razonado en varias ocasiones, como en la sección 2.2, la capacidad de negociación de confianza es muy necesaria para entornos complejos, en los que no se pueden establecer relaciones de confianza dos a dos con todas las entidades; debido al gran número de participantes o por la dinamicidad del sistema; con alta tasa de entrada y salida de participantes.

Pese a que la negociación de confianza es muy necesaria, no redunda en una mayor eficiencia dado que negociar en cada interacción es demasiado costoso y resultaría en una mayor latencia en el acceso a la red o a los servicios. Si recurrimos a un esquema más balanceado, en el que una vez se ha negociado se emite una credencial resumen de la negociación, que acelere posteriores interacciones, aumentaría la eficiencia.

Por esta razón, en la sección 7.3, se propone lo siguiente:

- Un formato de mensaje para la solicitud de emisión de certificados de atributos. Este mensaje esta basado en el formato empleado para la emisión de certificados de clave pública descrito en [107] con ciertas modificaciones. Dichas modificaciones permiten envolver una solicitud de emisión, acompañándola de una firma, para que las entidades intermedias, que deban aseverar que dicha solicitud es lícita, puedan hacerlo.
- Una extensión a TLS, que negocia la utilización de una capa intermedia de TLS, que permite la emisión de certificados de atributos sobre un canal seguro.

Por lo tanto, una entidad que ha negociado satisfactoriamente puede utilizar esta extensión para solicitar la emisión de un certificado de atributos que resuma la negociación. Esto es posible dado que se pueden utilizar varias extensiones

durante una conexión con TLS, por esta razón, una entidad puede negociar para más tarde solicitar un certificado de atributos.

La posibilidad de envolver una solicitud y firmarla varias veces permite emisión compleja de credenciales como serían delegación, emisión indirecta, etc.

11. **Diseño de una extensión de un protocolo estándar para el soporte de negociación de confianza**

    En la sección 7.2, se compara la propuesta de la sección 7.2 con las necesidades de negociación de confianza, llegando a la conclusión de que pese a que permite la utilización de autorización, que valdría para el uso de "trust tickets", no soporta negociación de confianza como tal. Por esa razón, en la sección 7.4, se propone otra modificación a TLS que mediante extensiones permite no solo el uso de credenciales de autorización, sino el intercambio de credenciales y políticas para la negociación de confianza.

    La extensión descrita en la sección 7.4 permite además del envío de credenciales de autorización a la otra parte, tal y como se describía en la sección 7.2, liberar de forma gradual credenciales y políticas. Además, permite combinar mediante operadores lógicos las políticas o la especificación formal de requisitos (policy items).

    Esta modificación es compatible hacia atrás porque utiliza una capa cuyo uso se negocia durante el handshake de TLS. La modificación de TLS permite no solo negociar el tipo de credenciales y políticas a utilizar, sino también los operadores, especificando mediante identificadores de objeto (OIDs) o URIs la semántica de los mismos. De esta manera las dos entidades se ponen de acuerdo en todos los parámetros que afectan a la negociación y se evita que se rompa una negociación por la imposibilidad de procesar una determinada credencial o política.

    La arquitectura propuesta sigue la misma filosofía que las anteriores extensiones de TLS dado que negocian la existencia de una capa, que corre en paralelo a la capa de handshake, y que realiza las tareas de negociación de confianza o autorización. El diseño de la extensión permite al cliente solicitar al servidor la negociación sobre canal seguro, al igual que las anteriores extensiones descritas en las secciones 7.2 y 7.3, sin perjuicio de aquellos clientes, que por eficiencia prefieran proporcionar ciertos datos durante el handshake.

    Los mensajes intercambiados para la negociación de confianza, una vez se ha llegado a un acuerdo entre ambas partes, son mensajes de tipo **Assert**, **Require** e **Inform**, que permiten, a cualquiera de las partes, enviar una credencial, formular requisitos o proporcionar información de cualquier tipo.

    En la sección 7.4 se compara esta extensión con otras encontradas en la literatura poniendo de manifiesto que esta extensión:

- No requiere un doble handshake de TLS.
- Permite un intercambio ordenado.
- Soporta otras credenciales además de certificados de clave pública ya que el formato es agnóstico.
- Es compatible hacia atrás con cualquier versión de TLS.

Otras contribuciones

Además de las contribuciones técnicas, en el transcurso de la investigación se han realizado otras tareas tales como:

- **Difusión y diseminación.** Durante el desarrollo de la tesis se ha tratado de difundir los resultados de la manera más efectiva posible, para ello se han publicado artículos en congresos nacionales, internacionales así como en revistas. En la sección 1.1 se detallan los artículos relacionados con la investigación publicados en los diferentes congresos y revistas.

  Por otro lado, se ha colaborado en grupos de noticias, listas de correo de la IETF y foros de seguridad relacionados con el desarrollo de herramientas como OpenSSL. Además se ha contribuido a proyectos europeos como UBISEC, EASY WIRELESS y TRUST-ES; de colaboración con empresa como los realizados con SIDSA y Nokia.

- **Identificar nuevas líneas de investigación.** En la sección 8.3, se identifican posibles líneas de investigación para la mejora y ampliación de la cobertura de las contribuciones de esta investigación.

8.2. Conclusiones

Una vez hemos identificado las principales contribuciones de la tesis, podemos concluir lo siguiente:

- En los escenarios complejos, donde existen variedad de participantes, es necesario establecer una organización y determinar correctamente los diferentes actores y sus respectivos roles. En los escenarios que se analizan en la investigación se establecen los grupos de equipo de usuario, proveedor de acceso a la red, proveedor de servicios de Internet y servicios de valor añadido. Respecto a los roles, es de vital importancia analizar y comprender las perspectivas de los diferentes actores respecto al control de acceso dado que cada uno de ellos tendrá unos requisitos diferentes.

- Actualmente el control de acceso en el equipo de usuario se centra en evitar que entidades no autorizadas accedan a los recursos de dicho equipo. Es decir, se establece una defensa de fuera a dentro. En la investigación identificamos la necesidad de que el equipo de usuario participe en su seguridad estableciendo un control de acceso de dentro a fuera, de forma que las aplicaciones del equipo de usuario, realicen un uso correcto de los recursos internos del mismo, seleccionando adecuadamente las redes, identificando el entorno y obrando en consecuencia. De esta manera se consigue una seguridad proactiva.

- Los mecanismos de autenticación y autorización utilizados para el acceso a la red y a los servicios deben ser flexibles. Deben permitir una negociación escalonada, de manera que gradualmente se alcance un estado de seguridad que permita interactuar incluso con entidades que a priori son desconocidas. Sin embargo, para conseguir una mayor eficiencia se deben diseñar sistemas balanceados que permitan negociar, pero a su vez reutilizar anteriores negociaciones para evitar que el tiempo de acceso se incremente en todas las interacciones. Por lo tanto, los sistemas de control de acceso, deberán permitir negociación de confianza y el uso de "Trust tickets" que aceleren posteriores iteraciones.

- La seguridad extremo a extremo es necesaria. En muchas ocasiones se ha visto en protocolos de autenticación y confidencialidad extremo a extremo, como TLS, una alternativa durante el despliegue de IPSEC para la confidencialidad. Sin embargo, la dependencia actual que tiene el comercio electrónico con TLS lo hace perdurar pese a que no tuvo esta intención en su diseño. Las razones son obvias, la seguridad extremo a extremo es absolutamente necesaria dado que no existe otra alternativa para garantizar a ambos extremos que la información es confidencial en todo su recorrido. Por otro lado, es una razón de peso equivalente a la anterior, el poder autenticar y autorizar al otro extremo con totales garantías.

- El control de acceso es una tarea que comprende varias subtareas de gran importancia como son la autenticación o identificación, la autorización y la acción de cotejar la información proporcionada con la sensibilidad, o lo que es lo mismo, aplicación de una política. Pese a que históricamente algunos sistemas resuelven todo el control de acceso sin distinguir entre tareas, la alta especialización que en el transcurso de la historia han adquirido cada una de las subtareas comentadas, nos obliga a establecer una separación adecuada entre ellas para obtener el máximo rendimiento. Por ejemplo, en ocasiones algunos servicios cotidianos recurren a vincular la identificación o autenticación con los privilegios, lo que limita la escalabilidad, reduciendo el alcance de una credencial a un único dominio. Estableciendo una separación adecuada se incrementan los grados de libertad y por tanto aumenta la complejidad, pero el alcance, usabilidad y flexibilidad del sistema completo es mayor.

- En cuanto a las credenciales, respecto a cómo logran la unicidad que permiten identificar unívocamente a la entidades, algunas credenciales basan la unicidad en espacios de nombres globales como PKI. Otras en la clave en sí, como SPKI o KeyNote, e incluso otras disponen de un nombre global y otro local (SDSI). Salvando la popularidad de unas frente a otras, cada una de ellas suele encajar en un campo de aplicación específico, siendo complicado elegir una de ellas como candidata para todos los posibles sistemas de autenticación y autorización. Respecto a la codificación, aquellas basadas en XML, com SAML, que no es una credencial en si misma sino una colección, permiten la inspección directa por el ser humano, dado que se puede leer; pero su tamaño la hace menos atractiva para pequeños dispositivos, como tarjetas inteligentes, frente a otras mas compactas como certificados de clave pública. Con las políticas ocurre igual, la capacidad de describir las sensibilidades de los recursos o las restricciones del contexto varían de un tipo a otro, pero resulta difícil e ingenuo pensar que con un solo lenguaje pueden cubrirse las necesidades actuales y futuras. Todo esto nos lleva a la necesidad de diseñar sistemas de control de acceso capaces de ser extendidos para soportar cualquier credencial o lenguaje de políticas.

- Los sistemas que asocian los privilegios a la identidad o al resultado la autenticación; que pueden determinar el grupo al que pertenece; o el rol que desempeña una entidad; permiten el acceso directo a todos los recursos una vez la entidad es autenticada y autorizada. En general esta forma de proceder es correcta, pero es mucho mejor conceder el acceso a los recursos de forma escalonada respecto al riesgo que supone su exposición. De esta manera, tras una autenticación y autorización básica, se podría tener acceso a una serie de recursos menos sensibles, requiriéndose una autenticación o autorización avanzada para el acceso a otros más sensibles. Esto se conoce como calidad en autenticación/autorización. Este tipo de técnicas son las más adecuadas para la protección de los recursos sensibles, ya que limita su exposición y los protege de manera más eficiente al requerir una autenticación o autorización de mayor nivel para su acceso. Estas técnicas, que consiguen, durante el transcurso de una sesión, incrementar los privilegios escalonadamente se llaman técnicas de negociación de confianza y permiten: el acceso a desconocidos dependiendo de la calidad de sus credenciales; establecer calidad de servicio en autenticación y autorización; acceso gradual a los servicios y protección de las políticas y credenciales.

- Tras analizar los distintos protocolos de autenticación y autorización para el acceso a la red y los servicios se ha concluido lo siguiente:

    - Los protocolo de autenticación a nivel físico y de enlace abaratan los costes de despliegue, dado que son sencillos de usar y no suelen requerir infraestructura. La distribución de parches para corregir errores es cara, lenta y

compleja y aumenta el riesgo cuando se vulnera un algoritmo, dado que el tiempo de exposición es el mismo que tarda el fabricante en proporcionar una solución.

- Aquellos en los que la autenticación se realiza a niveles superiores son más caros de mantener, requieren en ocasiones infraestructura, pero son más flexibles y reutilizables.
- De todos ellos, los basados en EAP permiten el uso de diferentes mecanismos de autenticación y autorización permitiendo así cambiar los actuales por otros más seguros con el paso del tiempo, cosa que no permiten, de manera sencilla, los cercanos al hardware.
- A medida que aumentan las tecnologías de acceso a la red, si no se utilizan protocolos extensibles, como EAP a niveles superiores, el problema de configuración de la red deja de ser abordable por el usuario medio, dado que requeriría conocimiento específico de cada tecnología.

Es por tanto conveniente, utilizar protocolos a niveles superiores dado que el factor común a todas las tecnologías de acceso es IP. Además, si dichos protocolos son extensibles, permiten la reutilización de credenciales; la utilización de sistemas de identidad federada; y simplifican la configuración. Estos sistemas permiten incluso la utilización de sistemas de negociación de confianza.

- Como una de las tareas fundamentales del software residente en el equipo de usuario destaca la toma de decisiones y la comunicación efectiva de las decisiones y riesgos al usuario. La información que debe analizar el equipo de usuario proviene de multiples fuentes y describe entidades de diferente naturaleza, por tanto, para su análisis se requieren técnicas de estadística multivariante. De entre las analizadas en la investigación se recomienda el uso de Multidimensional Scaling (MDS) que es capaz de simplificar los problemas para una adecuada comprensión por parte de la máquina y del usuario, manteniendo la conceptualización del problema, y permitiendo clasificar, agrupar y comparar datos de diferente naturaleza. Por otro lado, MDS permite representar gráficamente problemas complejos.
- Como resultado de analizar la perspectiva del equipo de usuario respecto del control de acceso, se establecen como requisitos necesarios para una protección activa lo siguiente:
  - Ser capaz de reconocer el entorno que nos rodea, siendo capaces de identificar en dicho entorno en futuras ocasiones.
  - Poder valorar de forma autónoma el entorno que nos rodea por los servicios que ofrece y por los dispositivos cercanos.

  - Permitir la valoración mediante protocolos colaborativos en aquellos casos en los que sea necesario o conveniente.
  - Ser capaz de seleccionar las redes que más se ajusten a las necesidades de conexión, al entorno que nos rodea y a las restricciones impuestas a dicho entorno.
  - Aplicar políticas que tengan en consideración el entorno, la valoración del mismo y el contexto para limitar las interacciones.
  - Empotrar los puntos de aplicación de las políticas en el sistema operativo para que ninguna aplicación pueda soslayar el control de acceso de dentro a fuera.

- Los sistemas de control de acceso actuales no cumplen todos los requisitos necesarios y por esta razón se propone uno en la sección 3 que tiene las siguientes características:

  - Es autónomo ya que utiliza los interfaces de red para escuchar y así determinar qué dispositivos y servicios nos rodean. De esta forma se caracteriza el entorno. Por tanto, no revela la posición ni la identidad del dispositivo.
  - Es independiente de la tecnología, siendo capaz de elegir entre multitud de redes diferentes mediante el análisis estadístico y la comparación con un ideal que recoge las demandas actuales de tráfico.
  - Permite incluir la información de contexto en la decisión adecuándola al entorno actual.
  - Es sensible a las preferencias del usuario y las políticas que gobiernan los recursos del sistema, permitiendo dar mayor importancia a determinados aspectos de la selección.

- Las simulaciones realizadas muestran la capacidad del sistema de control de acceso de tomar decisiones complejas, basadas en muchos parámetros de selección. El sistema es, por otro lado, capaz de representar gráficamente el espacio de decisión. Mediante una representación en una o dos dimensiones se simplifica el problema de elegir entre multitud de redes con diferentes parámetros, a elegir el punto más cercano al ideal. Esto asimila la decisión al comúnmente utilizado modelo mental de distancias o potencia de señal recibida. La representación gráfica mantiene la conceptualización del problema, demostrándose en las simulaciones cómo agrupa adecuadamente elementos conceptualmente similares.

  Además, las simulaciones muestran que el ajuste de los datos es correcto.

- Las soluciones de control de acceso propuestas en la literatura describen sistemas que combinan políticas del mismo tipo. Muchas de ellas incluyen información de contexto. Este tipo de soluciones de control de acceso no se adecúan completamente a las necesidades de los escenarios descritos, dado que en éstos es posible encontrar políticas de diferente tipo, controladas por diferentes motores de control de acceso, protegiendo los mismos recursos. Para una correcta aplicación de dichas políticas es necesario una visión global del problema que no pueden proporcionar cada uno de los motores de control de acceso por separado.

  Por esta razón, se propone un mecanismo de decisión agnóstico que asiste a los diferentes motores de control de acceso, estableciendo un orden de aplicación de las políticas de forma. El sistema permite además que se procese la información de contexto en global, dándole al sistema de una visión global del problema. El mecanismo propuesto cumple con su cometido dado que posee las siguientes características:

  - Trata a todas las entidades participantes en la decisión de la misma manera, es decir, como recursos. Permite, por tanto, proteger recursos, credenciales y políticas. En el caso de estas últimas, lo que hace el mecanismo de asistencia a la negociación de confianza es establecer el orden en que las políticas se aplican. Al establecer un orden sólo se liberarán detalles de la política cuando la otra entidad ha satisfecho con anterioridad una serie de requisitos. Estos requisitos garantizan que revelar la política no supone un riesgo para la seguridad o privacidad.
  - Dado su carácter agnóstico, el sistema permite manejar cualquier credencial, recurso o política, con independencia de su naturaleza.
  - Permite medir el riesgo al aplicar MDS al conjunto de los datos. El mecanismo reduce la dimensionalidad del problema y, dado que la conceptualización del problema se mantiene una vez reducidas sus dimensiones, se puede asociar riesgo a distancia entre puntos.
  - Permite representar gráficamente el problema para que el usuario pueda, mediante comparaciones, comprender el riesgo de determinadas decisiones; o comprender por qué el sistema bloquea el acceso a determinados recursos.

- Las simulaciones realizadas para validar el mecanismo de asistencia a la negociación de confianza, muestran cómo el mecanismo es capaz de combinar diferentes requisitos provenientes de diferentes motores de control de acceso en un mismo espacio de decisión. También muestra la posibilidad de presentar comprensiblemente esta información al usuario.

En la simulación puede comprobarse que el mecanismo libera gradualmente los requisitos y permite el acceso escalonado a los recursos, a la vez que permite valorar el riesgo. Por lo que se concluye que es conveniente recurrir a este tipo de análisis para dotar al sistema de visión global.

- TLS es un protocolo adecuado para el soporte de autorización, emisión de credenciales y negociación de confianza por las siguientes razones:
  - Proporciona confidencialidad.
  - Soporta anonimato, autenticación del servidor o autenticación mutua.
  - Permite negociar los algoritmos de cifrado, compresión de datos y autenticación de mensaje al inicio aumentando sus posibilidades de uso.
  - La funcionalidad del protocolo puede aumentarse mediante extensiones, estando el procedimiento de extensión correctamente recogido en el estándar.
  - Permite su uso sobre EAP.
  - Está desplegado, el usuario confía en él y lo conoce. Además está implementado en la mayoría de los dispositivos comerciales actuales.
- La autorización, como proceso separado de la autenticación, no está soportada en los protocolos de acceso a los servicios como TLS. Típicamente los servidores que utilizan TLS autentican al cliente y posteriormente determinan, de forma local o a través de una base de datos, sus privilegios. Para evitar esto, se proponen unas modificaciones al protocolo TLS que den soporte a la autorización. Dichas modificaciones tiene las siguientes características:
  - Las modificaciones son compatibles hacia atrás con versiones de TLS que no soporten autorización.
  - Permite el uso de certificados de atributos y SAML.
  - Toda la información la gestiona una nueva capa cuyo uso es negociado durante el handshake por lo que no lastra a las capas del protocolo original.
  - Permite que el cliente envíe las credenciales directamente (modo “push”) o bien que proporcione todos los datos necesarios para que el servidor la consiga (modo “pull”).
  - Dispone de expresividad suficiente para dar soporte a bindings complejos de SAML.

  Mediante el uso de esta extensión se permite el envío de información de autorización que podría corresponder incluso a una credencial resumen (trust ticket) de un proceso de negociación de confianza.

- Es necesario balancear los sistemas de negociación de confianza para aumentar su rendimiento. Para ello es necesario que se puedan emitir credenciales resumen o "trust tickets" de forma que, en futuras interacciones, se pueda resumir la negociación evitando intercambiar continuamente una gran cantidad de paquetes. Por esta razón, se propone una formato de mensaje para la solicitud de certificados de atributos basado en el formato utilizado para la emisión de certificados de clave pública. La emisión de dichas credenciales necesita un tratamiento diferente a las de PKI para lograr:

  - Emisión automática: que las entidades puedan, sin necesidad de la intervención de un administrador, realizar todo el proceso que lleva a la emisión de la credencial.
  - Emisión indirecta y delegación: que las entidades afectadas puedan aseverar una solicitud proporcionando los medios necesarios para que la autoridad que emite la credencial pueda comprobar su conformidad.

  Por esta razón es necesario proponer un formato recursivo que permita envolver y firmar una solicitud cuantas veces sea necesario para que todas las entidades que lo necesiten puedan confirmar una solicitud de forma secuencial.

- En la misma línea del punto anterior, se propone una extensión de TLS para dar soporte a la emisión dinámica de credenciales con las siguientes características:

  - Permite negociar el uso de una capa intermedia que asuma las tareas de gestión de la emisión de credenciales.
  - Permite la utilización del material criptográfico intercambiado para el establecimiento del canal seguro como prueba de posesión de claves si se realiza autenticación mutua.
  - Da soporte a pruebas de posesión de claves para claves de tipo Diffie-Hellman y otras que necesiten un mecanismo de reto-respuesta al reto.
  - Permite el envío de las solicitudes y la recepción de las credenciales sobre canal seguro.

- El soporte de autorización basado en certificados de atributos y SAML proporciona cierto grado de libertad al sistema pero no deja de ser muy limitado si se tienen en cuenta los requisitos de los sistemas de negociación de confianza. Por esta razón se propone, en la sección 7.4, una extensión a TLS para el soporte de negociación de confianza que permite:

  - Negociar durante el handshake de TLS tanto del uso de una capa separada para la gestión de la negociación de confianza como las credenciales y

los lenguajes de políticas a utilizar durante la negociación. De esta manera es posible determinar si el cliente o el servidor será capaz de entender las credenciales y requisitos intercambiados.

- Posponer la negociación de las credenciales y políticas a utilizar hasta el establecimiento del canal seguro. Esto no evita que se negocie el uso de la capa de negociación de confianza.
- Acordar el uso de operadores lógicos que podrán ser utilizados para combinar requisitos, provenientes incluso de diferentes políticas o motores de control de acceso, en un único mensaje.
- Intercambiar multiples credenciales de autenticación y autorización logrando así facilitar la calidad en la autenticación/autorización.
- Intercambiar tantos mensajes como sea necesario, pudiendo intercambiar requisitos y credenciales en modo P2P, es decir, sin que sea necesario distinguir al cliente del servidor. También se permite la utilización de dicha capa durante todo el periodo de vida de la sesión, permitiendo así el incremento de privilegios bajo demanda.

- Todas las extensiones propuestas en esta tesis que afectan a TLS pueden ser utilizadas sobre EAP, aumentando el campo de aplicación. De esta forma, no solo se permite su uso para el acceso a los servicios, sino para el acceso a la red siempre sobre protocolos portadores de EAP.

8.3. Líneas futuras

Esta tesis propone nuevos mecanismos para la mejora del control de acceso interno del dispositivo y para la asistencia a la decisión en un proceso de negociación de confianza. Por otro lado, se proponen modificaciones a protocolos para permitir el intercambio de credenciales de autorización, la solicitud de emisión de credenciales e el intercambio de credenciales para la negociación de confianza. No obstante, quedan muchas mejoras y precisiones que hacer a las propuestas aquí planteadas, a la vez que se abren líneas de investigación futuras que permitan que estos sistemas flexibles y dinámicos de control de acceso sean una realidad. Durante la realización de la tesis se han identificado algunas mejoras y líneas de trabajo futuro que se resumen a continuación:

1. **Estudio y definición de un mecanismo de intercambio de información de contexto**: En el capítulo 3 se propone una manera de identificar y caracterizar, de

forma autónoma, anónima y automática, el entorno que nos rodea. La autonomía se rompe cuando opcionalmente, el equipo de usuario, recurre a protocolos colaborativos que le permitan obtener un valor de confianza para el dominio.

Los protocolos colaborativos pueden ser útiles para, además de obtener un valor de confianza, intercambiar información de servicios, dispositivos y cualquier atributo en general, que permita una mejora de la calidad de la información de contexto.

Para que estos mecanismos colaborativos funcionen correctamente sería necesario recurrir a sistemas de reputación que permitieran asignar una calidad a la información obtenida de esta manera, en función de la reputación de las fuentes.

2. **Estudio, simulación y validación de algoritmos alternativos para la valoración del contexto**: En la sección 3.4 se propone una serie de expresiones matemáticas que combinan los valores de confianza de las diferentes entidades, obtenidos mediante el uso de PTM[4], con las valoraciones que dichas entidades dan al entorno. Los resultados muestran que las expresiones allí propuestas son conservadoras dado que evitan cambios bruscos en el valor de confianza.

   Sería interesante estudiar, simular y validar otras alternativas, de manera que el usuario pudiese elegir el comportamiento que desea que tengan sus dispositivos. Permitiendo incluso que, dependiendo del tipo de entorno o de la presencia de determinados dispositivos o servicios, varíe para adaptarse al entorno.

3. **Evaluación y mejora del consumo de recursos de la preparación de datos**: El cálculo del rendimiento expuesto en la sección 4.3, cuyos resultados son aplicables tanto al mecanismo de selección de red como al de asistencia la negociación de confianza, muestra el rendimiento del algoritmo MDS. Es decir, no determina el rendimiento conjunto de la preparación de los datos, la construcción de la matriz y la posterior aplicación del algoritmo MDS.

   En ambos casos, en la selección de red y en la negociación de confianza, existen mejoras que permiten ahorrar cálculos evitando reconstruir la matriz entera cuando se producen cambios. Por ejemplo, en el caso de la selección de red, la aparición de un nuevo elemento en la decisión podría sugerir el cálculo de una nueva matriz, pero en realidad, bastaría con actualizar la matriz incluyendo una nueva fila y columna que recoja las disimilitudes de éste nuevo elemento con los existentes, sin necesidad de calcular de nuevo la matriz completa. Del mismo modo, si desaparece un elemento, basta con eliminar una fila y una columna.

   En cambio si se produce una variación en alguno de los atributos, puede ser necesario volver a realizar una gran cantidad de cálculos, dado que ese cambio debe propagarse. Es necesario evaluar el impacto en el consumo de los recursos al

actualizar la matriz, tras la variación de los atributos de alguna entidad, y mejorar la preparación de datos de forma que minimize el coste computacional.

4. **Evaluación del coste computacional de empotrar los puntos de aplicación de política en el sistema operativo**: En la sección 3.2 se propone empotrar los puntos de aplicación de las políticas en el sistema operativo de manera que se evite que las aplicaciones se salten las protecciones.

   Se ha trabajado en esa dirección mediante el desarrollo de drivers NDIS para Windows, de minipuerto y de protocolo, que vigilen las interacciones y limiten su alcance. Salvados los problemas de depuración y de compilación a plataformas de tipo Windows CE; el siguiente paso es evaluar el impacto en el coste computacional de tener en funcionamiento estos monitores de comportamiento.

5. **Mecanismos automáticos de extracción de requisitos y policy items de las políticas más comunes**: En la sección 5.3 se muestra el diagrama de comunicación entre módulos en un sistema de negociación de confianza. Una de las tareas, a realizar por los motores de control de acceso, descrita en el capítulo 5, es la de extraer los requisitos y su expresión formal, policy item, de las políticas y registrarlos en el motor de decisión.

   Sería interesante diseñar mecanismos automáticos para realizar esa extracción de requisitos de las políticas más usadas habitualmente, tratando en la medida de lo posible de ordenar sus sensibilidades.

6. **Reproducción y modelado de ataques con el mecanismo de decisión para negociación de confianza**: En el apartado de negociación de confianza, podría ser interesante reproducir situaciones peligrosas y ataques, monitorizando los movimientos de los puntos en el sistema, de manera que se crearan casos base que permitieran detectar posibles ataques por comparación de riesgos.

7. **Estudio de modelos mentales, iconos y mensajes para la comunicación efectiva de riesgos**: En el mecanismo de asistencia a la negociación de confianza, descrito en el capítulo 5 y validado en el capítulo 6, se muestra cómo una representación gráfica de un problema complejo, ayuda a la comprensión, por parte del usuario, de los riesgos; sin embargo, el uso de iconos asociados a modelos mentales mejora todavía más la efectividad de la comunicación de riesgos.

   Por esta razón, se considera interesante recurrir a encuestas que permitan encontrar los iconos de los modelos mentales más apropiados para la comunicación de este tipo de riesgos. Esto se discute también en la sección 2.1.

8. **Estudio de la mejora de la selección de rutas en negociación de confianza**: En el capítulo 5, se propone un mecanismo de decisión para la negociación de confianza. En dicho capítulo, se define el **punto de entrada** como el primer requisito

a cumplir durante una negociación. También se definen las **rutas**, como la secuencia de requisitos desde el punto de entrada hasta el acceso final al recurso. Es posible que para acceder a un recurso existan varias rutas posibles que impliquen satisfacer diferentes secuencias de requisitos; sin embargo, es la otra parte la que decide qué requisito será el siguiente, con el riesgo de tener que volver a atrás si no consigue acceder al recurso.

Un análisis de cómo comunicar a la otra parte las rutas más convenientes, respetando la liberación escalonada de requisitos, podría ser interesante. Es decir, si la otra parte conociese cuál de los posibles "siguientes pasos" es el más acertado, se podría aumentar mucho la eficiencia de los sistemas de negociación de confianza.

9. **Extender TLS para el soporte de nuevas capas y su disposición en niveles**: Las extensiones de TLS propuestas en el capítulo 7, liberan a las capas estándar de TLS de soportar las nuevas funcionalidades, negociando la presencia de una capa en paralelo a las estándar que soporte la nueva funcionalidad; sin embargo, no permiten la utilización de una capa encima de otra requiriendo que la nueva funcionalidad se implemente de forma monolítica.

   Una funcionalidad muy interesante para TLS, sería la de poder negociar capas adicionales y su disposición en niveles, de forma que se pudiera reaprovechar las diferentes implementaciones o establecer un orden de aplicación de las funcionalidades, aumentando mucho más la flexibilidad del protocolo.

10. **Independizar de TLS el protocolo de negociación de confianza para su uso sobre otros protocolos**: La extensión propuesta en la sección 7.4, permite negociación de confianza sobre TLS. Esta extensión define un conjunto de mensajes de protocolo nuevos que permiten el intercambio de requisitos y credenciales entre las partes y se aprovecha de la confidencialidad de TLS para garantizar la seguridad.

    Dichos mensajes de protocolo podrían utilizarse sobre otros protocolos, pero requerirían de la capacidad de generar una clave para proteger el intercambio de mensajes. En TLS no es necesario, dado que TLS protege el canal, y que los mensajes de este protocolo se consumen en capas internas de TLS, por lo que no existe la posibilidad de que se produzcan los ataques descritos en [23]; no obstante, si el protocolo se independiza por completo de TLS, dejando de ser una capa interna del dicho protocolo, sí sería susceptible de sufrir los ataques de man-in-the-middle comentados en [23]. Esto se puede evitar dotando al protocolo de la capacidad de proteger los mensajes con una clave, lo que requiere algunas modificaciones.

11. **Validación con casos reales del rendimiento al utilizar las extensiones de TLS propuestas sobre EAP**: En esta tesis, se ha propuesto la estructura de un sistema flexible de autenticación y autorización que utiliza TLS y EAP para la comunicación de las credenciales y la negociación de confianza.

    Como parte del proceso de validación, sería necesario realizar pruebas reales con hardware comercial; tomar medidas de rendimiento, latencia, segmentación de paquetes, etc.

Bibliografía

[1] Mark Weiser, "The Computer for the 21st Century," *Scientific American*, pp. 94–104, September 1991.

[2] D.F. Bantz, C. Bisdikian, D. Challener, J.P. Karidis, S. Mastrianni, A. Mohindra, D.G. Shea, and M. Vanoveret, "Autonomic personal computing," *IBM Systems Journal*, vol. 42, no. 1, pp. 165–176, 2003.

[3] RSALabs, "Pkcs#11 v2.11: Cryptographic token interface standard," 2004.

[4] Florina Almenares, "Arquitectura de seguridad para entornos de computación ubicua abiertos y dinámicos," Ph.D. dissertation, Department of Telematic Engineering, University Carlos III of Madrid, March 2006.

[5] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "PPP Extensible Authentication Protocol (EAP)," Tech. Rep. RFC3784, IETF Network Working Group, 2004.

[6] Daniel Díaz, Andrés Marín, Florina Alménarez, Carlos García-Rubio, and Celeste Campo, "A framework for authorization and delegation in ubiquitous computing," in *Ubiquitous Computing and Ambient Intelligence conference*. 2006, Thomsom.

[7] Daniel Díaz, Andrés Marín, Florina Alménarez, Celeste Campo, and Carlos Garcia-Rubio, "Securing interactions in emerging environments.," *2nd International Workshop on Ubiquitous Computing and Ambient Intelligence - 2006. Conference proceedings*, june 2006.

[8] Daniel Díaz, Andrés Marín, and Florina Alménarez, "Mecanismo de selección de red sensible al contexto para entornos dinámicos," *Jornadas de Ingeniería Telemática (JITEL). Conference proceedings*, september 2007.

[9] Daniel Díaz, Andrés Marín, Florina Alménarez, Celeste Campo, and Carlos Garcia-Rubio, "Mejorando el control de acceso para dispositivos móviles con un motor de decisión agnóstico para negociación de confianza," *Actas de congreso*, september 2007.

[10] Florina Almenárez, Daniel Díaz, and Andrés Marín, "Secure Ad-hoc mBusiness: Enhancing WindowsCE security," in *1st Conference on Trust Digital Business (TrustBus'04)*, 2004.

[11] Florina Alménarez, Andrés Marín, Daniel Díaz, and Juan Jesús Sánchez, "Developing a model for trust management in pervasive devices," *Third IEEE International Workshop on Pervasive Computing and Communication Security held in conjunction with IEEE PerCom 2006.IEEE PerCom.*, may 2006.

[12] Juan Jesús Sánchez, Daniel Diaz, Jose Alberto Vigo, Natividad Martinez, and Ralf Seepold, "Cards and residential gateways: Improving osgi gateways services with java cards," *Seventh Smart Card Research and Advanced Application IFIP Conference (CARDIS 2006). Lecture Notes In Computer Science*, april 2006.

[13] Daniel Díaz, Andrés Marín, and Florina Almenárez, "A smart card solution for access control and trust management for nomadic users," *Seventh Smart Card Research and Advanced Application IFIP Conference (CARDIS 2006). Lecture Notes In Computer Science*, april 2006.

[14] Daniel Díaz, Andrés Marín, Florina Almenárez, Carlos Garcia-Rubio, and Celeste Campo, "Interaction distance determination with pervsim," *15th IST Mobile and Wireless Communication Summit. Conference proceedings*, june 2006.

[15] Daniel Díaz, Andrés Marín, Florina Almenárez, Carlos Garcia-Rubio, and Celeste Campo, "Context awareness in network selection for dynamic environments," *11th IFIP International Conference on Personal Wireless Communications "PWC06". Lecture Notes In Computer Science*, june 2006.

[16] Andrés Marín, Wolgfran Mueller, Robbie Schaefer, Florina Almenárez, Daniel Díaz, and Max Ziegle, "Middleware for secure home access and control," *IEEE Pervasive Communications (PERCOM) 2007. Conference proceedings/IEEE Library*, march 2007.

[17] Daniel Díaz, Andrés Marín, and Florina Almenárez, "Access control agnostic trust negotiation decision engine," *18th annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications. IEEE.*, september 2007.

[18] Daniel Díaz, Andrés Marín, and Florina Almenárez, "Enhancing access control for mobile devices with an agnostic trust negotiation decision engine," *Personal Wireless Communications. Springer series in Computer Science. ISSN: 1571-5736.*, 2007.

[19] Daniel Díaz, Andrés Marín, Florina Almenárez, and Alberto Cortés, "Dvb-h key management system for umts capable devices," *IEEE International Conference on Consumer Electronics. IEEE*, january 2008.

[20] Celeste Campo, Florina Almenárez, Daniel Díaz, Carlos Garcia-Rubio, and Andrés Marín, "Secure service discovery based on trust management for ad-hoc networks," *Journal of Universal Computer Science. vol. 12*, march 2006.

[21] Andrés Marín, Daniel Díaz, Florina Almenárez, Carlos Garcia-Rubio, and Celeste Campo, "Smart card-based agents for fair non-repudiation," *Special Issue: Advances in Smart Cards. vol. 51, Issue 9, ISSN 1389-1286. Computer Networks.*, june 2007.

[22] Daniel Díaz, Andrés Marín, Florina Alménarez, Celeste Campo, and Carlos Garcia-Rubio, "Context awareness in network selection for dynamic environments," *Personal Wireless Communications Special Issue. Telecommunication Systems Journal*, 2007.

[23] J. Puthenkulam et al., "The Compound Authentication Binding Problem," Tech. Rep., IETF, 2003.

[24] F. UMTS, "Enabling UMTS Third Generation Services and Applications (No. 11 Report from the UMTS Forum)," *UMTS Forum, London*, 2000.

[25] BP Crow, I. Widjaja, LG Kim, and PT Sakai, "IEEE 802.11 Wireless Local Area Networks," *Communications Magazine, IEEE*, vol. 35, no. 9, pp. 116–126, 1997.

[26] "Ieee 802.16 wirelessman standard for wireless metropolitan area networks," Tech. Rep. 802.16e, IEEE, 2005, http://www.ieee802.org/16/.

[27] "Official bluetooth website," 1999, http://www.bluetooth.com/.

[28] J. Mitola, "The software radio architecture," *Communications Magazine, IEEE*, vol. 33, no. 5, pp. 26–38, 1995.

[29] A. Ivers and D. Smith, "A practical approach to the implementation of multiple radio configurations utilizing reconfigurable hardware and software building blocks," *MILCOM 97 Proceedings*, vol. vol.3, pp. 1327–1332, 1997.

[30] Authorization In Wireless, "Greenpass radius tools for delegated," citeseer.ist.psu.edu/664206.html.

[31] Ashutosh Dutta, Tao Zhang, Sunil Madhani, Kenichi Taniuchi, Kensaku Fujimoto, Yasuhiro Katsube, Yoshihiro Ohba, and Henning Schulzrinne, "Secure universal mobility for wireless internet.," in *WMASH*, 2004, pp. 71–80.

[32] B. Aboba, "Certificate-based roaming," Tech. Rep. draft-ietf-roamops-cert-02.txt, IETF ROAMOPS Working Group, 1999.

[33] Walter A. Chudson, "[untitled]," *Political Science Quarterly*, vol. 85, no. 2, pp. 358–360, jun 1970.

[34] L.J. Camp, "Mental Models of Computer Security," *Lecture Notes in Computer Science*, pp. 106–111, 2004.

[35] A. Acquisti and R. Gross, "Imagined communities: awareness, information sharing, and privacy on the Facebook," *6th Workshop on Privacy Enhancing Technologies, Robinson College, Cambridge University, UK*, vol. 10, 2006.

[36] DA. Norman, "Some observations on mental models," *Human-computer interaction: a multidisciplinary approach table of contents*, pp. 241–244, 1987.

[37] A. Bostrom, B. Fischhoff, and M.G. Morgan, "Characterizing mental models of hazardous processes: A methodology and an application to radon," *Journal of Social Issues*, vol. 48, no. 4, pp. 85–100, 1992.

[38] M.A. Kamrin, D.J. Katz, and M.L. Walter, *Reporting on Risk: A Journalist's Handbook on Environmental Risk Assessment*, Produced by Foundation for American Communications and National Sea Grant College Program, 1995.

[39] B. Aboba, "Pros and Cons of upper layer network access," *IEEE documents*, pp. 802–11.

[40] J. Puthenkulam et al., "The Compound Authentication Binding Problem," *draft-puthenkulam-eap-binding-04 (work in progress), October*, 2003.

[41] Kipp Hickman, "The Secure Socket Layer (SSL) protocol," February 1995.

[42] A. Frier, P. Karlton, and P. Kocher, "The SSL 3.0 Protocol," November 1996.

[43] T. Dierks and C. Allen, "The transport layer security (TLS) version 1.0," Tech. Rep. RFC2246, IETF, January 1999, http://www.ietf.org/rfc/rfc2246.txt.

[44] S. Kent and R. Atkinson, "Security architecture for the internet protocol (IPSec)," Tech. Rep. RFC2401, IETF Network Working Group, November 1998, http://www.ietf.org/html.charters/ipsec-charter.html.

[45] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet X.509 public key infrastructure: Part i: Certificate and CRL profile," Tech. Rep. RFC 2459, IETF, January 1999.

[46] Florina Almenárez, Andrés Marín, Celeste Campo, and Carlos García, "Security model for intelligent-agents-based pervasive computing environments," *IEEE Pervasive Computing*, March 2003.

[47] Florina Almenárez and Celeste Campo, "SPDP: A Secure Service Discovery Protocol for Ad-hoc Networks," in *Workshop on Next Generation Networks - EUNICE 2003*, September 2003.

[48] D. Ferraiolo and R. Kuhn, "Role-based access control (RBAC)," in *15th NIST-NCSC National Computer Security Conference*, 1992, pp. 554–563.

[49] International Telecommunication Union, "The directory: Public-key and attribute certificate frameworks," Tech. Rep. X.509, International Telecommunication Union, 2000.

[50] International Telecommunication Union (ITU), "The directory: Public-key and attribute certificate framework," Tech. Rep. X.509, International Telecommunication Union (ITU), 2005.

[51] R. Housley, "Internet x.509 public key infrastructure certificate and crl profile," Tech. Rep. RFC 3280, IETF PKIX Working Group, 2002.

[52] T. Dierks, "The tls protocol," Tech. Rep. RFC 2246, IETF TLS Working Group, 1999.

[53] T. Dierks and E. Rescorla, "The transport layer security (tls) protocol. version 1.1," Tech. Rep. RFC 4346, IETF TLS Working Group, 2006.

[54] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. D. Keromytis, "The KeyNote trust management system," Internet Request for Comment RFC 2704, Internet Engineering Task Force, Sept. 1999, Version 2.

[55] C. Ellison, "Spki certificate theory," Tech. Rep. RFC 2693, IETF SPKI Working Group, 1999.

[56] David Chadwick, Sassa Otenko, and Von Welch, "Using SAML to Link the GLOBUS Toolkit to the PERMIS Authorisation Infrastructure," in *Proceedings of Eighth Annual IFIP TC-6 TC-11 Conference on Communications and Multimedia Security*, Windermere, UK, September 2004.

[57] P. Mishra, "Saml v2.0 oasis standard specification," Tech. Rep. SAML v2.0, OASIS Security Services TC, 2005.

[58] Amir Herzberg, Yosi Mass, Joris Michaeli, Yiftach Ravid, and Dalit Naor, "Access control meets public key infrastructure, or: Assigning roles to strangers," in *SP '00: Proceedings of the 2000 IEEE Symposium on Security and Privacy*, Washington, DC, USA, 2000, p. 2, IEEE Computer Society.

[59] D.W. Chadwick and A. Otenko, "The PERMIS X.509 role based privilege management infrastructure," *Future Generation Computer Systems*, vol. 19, no. 2, pp. 277–289, February 2003.

[60] Rafae Bhatti, Elisa Bertino, and Arif Ghafoor, "An integrated approach to federated identity and privilege management in open systems," *Commun. ACM*, vol. 50, no. 2, pp. 81–87, 2007.

[61] Anna Cinzia Squicciarini, "Trust negotiation systems.," in *EDBT Workshops*, 2004, pp. 90–99.

[62] E. Bertino, E. Ferrari, and A. Squicciarini, "X -tnl: An xml-based language for trust negotiations," *policy*, vol. 00, pp. 81, 2003.

[63] E. Bertino, L. R. Khan, R. Sandhu, and B. Thuraisingham, "Secure knowledge management: confidentiality, trust, and privacy," *Systems, Man and Cybernetics, Part A, IEEE Transactions on*, vol. 36, no. 3, pp. 429–438, 2006.

[64] A. Squicciarini, A. Czeskis, E. Bertino, and A. Bhargav-Spantzel, "Auth-sl - a system for the specification and enforcement of quality-based authentication policies. submitted for publication," 2006.

[65] "Wireless lan meduim access control (MAC) and physical layer (PHY) specifications," 1999, ANSI/IEEE Standard 802.11.

[66] Jim Geier, "802.11 WEP: concepts and vulnerability," June 2002, http://www.wifiplanet.com/tutorials/article.php/1368661.

[67] Wi-Fi Alliance, "Wi-fi protected access (WPA)," February 2003, http://www.wifi.org/OpenSection/pdf/.

[68] J.C. Chen, M.C. Jiang, and Y. Liu, "Wireless LAN security and IEEE 802.11 i," *Wireless Communications, IEEE [see also IEEE Personal Communications]*, vol. 12, no. 1, pp. 27–36, 2005.

[69] "Ieee 802.1x - port based network access control," Tech. Rep. 802.1X, IEEE, 2004, http://www.ieee802.org/1/pages/802.1X-rev.html.

[70] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, and H. Levkowetz, "RFC 3748-Extensible Authentication Protocol (EAP)," Tech. Rep., IETF, 2004.

[71] RFC IETF, "3588"Diameter Base Protocol," *URL: http://www. ietf. org/rfc/rfc3588. txt*.

[72] C. Perkins et al., "IP Mobilty Support for IPv4," Tech. Rep., RFC 3344, 2002.

[73] D. Johnson and C. Perkins, "J. Arkko,"Mobility Support in IPv6," Tech. Rep., RFC 3775, June 2004.

[74] Forsberg D., Ohba Y., Patil B., Tschofenig H., and Yegin A., "Protocol for carrying authentication for network access (pana)," Tech. Rep., IETF, 2007.

[75] L. Morand, A. Yegin, S. Kumar, and S. Madanapalli, "Dhcp options for pana authentication agents," Tech. Rep., IETF, draft, December 2006.

[76] C. Kaufman et al., "Internet Key Exchange (IKEv2) Protocol," Tech. Rep., IETF, 2005.

[77] M. Parthasarathy, "PANA enabling IPsec based Access Control," *draft-ietf-pana-ipsec-07 (work in progress), July*, 2005.

[78] A. Yegin, Y. Ohba, R. Penno, and G. Tsirtsis, "RFC 4058 Protocol for Carrying Authentication for Network Access (PANA) Requirements," *Protocol for Carrying Authentication for Network Access (PANA) Requirements, May*, 2005.

[79] M. Parthasarathy, "Protocol for Carrying Authentication and Network Access (PANA) Threat Analysis and Security Requirements," Tech. Rep., RFC 4016, March 2005.

[80] P. Ferguson and D. Senie, "RFC2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing," *Internet RFCs*, 2000.

[81] E. Rescorla, N. Modadugu, and D.T. Layer, "Datagram Transport Layer Security," Tech. Rep., IETF, April 2006.

[82] B. Aboba, "Ppp eap tls authentication protocol," Tech. Rep. RFC 2716, IETF Network Working Group, 1999.

[83] S. Blake-Wilson, "Transport layer security (tls) extensions," Tech. Rep. RFC 3546, IETF TLS Working Group, 2003.

[84] S. Farrell, "Tls extensions for attributecertificate based authorization," Tech. Rep. draft-ietf-tls-attr-cert-01.txt, IETF Transport Layer Security Working Group, 1998.

[85] M. Brown and R. Housley, "Transport layer security (tls) authorization extensions," Tech. Rep. draft-housley-tls-authz-extns-07.txt, IETF, 2006.

[86] I Borg and P Groenen, "Modern multidimensional scaling, theory and applications," in *IEEE SECON 2004*, New York, NY, USA, 1997, Springer-Verlag.

[87] Yi Shang, Wheeler Ruml, Ying Zhang, and Markus P. J. Fromherz, "Localization from mere connectivity," in *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, New York, NY, USA, 2003, pp. 201–212, ACM Press.

[88] Katrijn Van Deun and Luc Delbeke, "Multidimensional scaling," 2000, http://www.mathpsyc.uni-bonn.de/index.htm.

[89] RÑ Shepard, "The analysis of proximities: multidimensional scaling with unknown distance function part i," in *Psychometrika 27*, 1962.

[90] J B Kruskal, "Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis," in *Psychometrika 29*, 1964.

[91] Y Takane, F W Young, and J. de Leeuw, "Nonmetric individual differences multidimensional scaling: an alternating least squares method with optimal scaling features," in *Psychometrika 42*, 1977.

[92] "eXtensible Access Control Markup Language (XACML)," 2003, http://www.oasis-open.org/apps/org/workgroup/xacml/.

[93] Florina Almenárez, Andrés Marín, Celeste Campo, and Carlos García, "TrustAC: Trust-based access control for pervasive devices," in *2nd International Conference Security in Pervasive Computing (SPC'05)*, 2005.

[94] M. Satyanarayanan, "Pervasive computing: Vision and challenges," *IEEE Personal Communications*, vol. 8, no. 4, pp. 10–17, August 2001, citeseer.nj.nec.com/gennaro99robust.html.

[95] C. Campo, C. García-Rubio, A. Marín, and F.Almenárez, "PDP: A lightweight discovery protocol for local-scope interactions in wireless ad hoc networks," *Computer Networks Journal. Elsevier*, 2006, Pending to be published.

[96] Eiman Elnahraway, Xiaoyan Li, and Richard P. Martin, "The limits of localization using rss," in *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, New York, NY, USA, 2004, pp. 283–284, ACM Press.

[97] Eiman Elnahraway, Xiaoyan Li, and Richard P. Martin, "The limits of localization using signal strength: a comparative study," in *IEEE SECON 2004*, 2004, pp. 406–414.

[98] Florina Almenárez, Andrés Marín, Celeste Campo, and Carlos García, "Managing ad-hoc trust relationships in pervasive environments," in *Workshop on Security and Privacy in Pervasive Computing SPPC'04 at Pervasive 2004*, 2004, http://www.vs.inf.ethz.ch/events/sppc04/program.html.

[99] S. Farell and R. Housley, "An internet attribute certificate profile for authorization," April 2002, http://www.faqs.org/rfcs/rfc3281.html.

[100] Elisa Bertino, Elena Ferrari, and Anna Cinzia Squicciarini, "Trust-x: A peer-to-peer framework for trust establishment," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 7, pp. 827–842, 2004.

[101] A. Hess, J. Jacobson, H. Mills, R. Wamsley, K. Seamons, and B. Smith, "Advanced client/server authentication in tls," 2002.

[102] Romain Laborde and Thierry Desprats, "Dealing with stable environmental conditions in xacml systems," in *Systems and Networks Communications, 2007. ICSNC 2007*, 2007, p. 63.

[103] J. Clark and Steve DeRose, "Xml path language (xpath) version 1.0," Tech. Rep., World Wide Web Consortium (W3C), 1999.

[104] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106.

[105] J. Ross Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[106] Ueli M. Maurer and Stefan Wolf, "The diffie hellman protocol," *Des. Codes Cryptography*, vol. 19, no. 2-3, pp. 147–171, 2000.

[107] M. Myers, C. Adams, D. Solo, and D. Kemp, "Internet x.509 certificate request message format," Tech. Rep., IETF, United States, 1999.

[108] S. Farrell and R. Housley, "An internet attribute certificate profile for authorization," Tech. Rep. RFC 3281, IETF PKIX Working Group, 2002.

[109] "Openssl documentation: Crypto library," http://openssl.org/docs/crypto/.