Utilizando Certificados Implícitos para Asignar Identidades en Overlays P2P

Juan Caubet
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: juan.caubet@entel.upc.edu

Jose L. Muñoz
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: jose.munoz@entel.upc.edu

Oscar Esparza
Departamento de
Ingeniería Telemática (ENTEL)
Universitat Politècnica de Catalunya
Email: oscar.esparza@entel.upc.edu

***Resumen*—Desde hace años, la seguridad en las redes P2P estructuradas está siendo cuestionada, y por ello se han propuesto muchos trabajos con el objetivo de proporcionar enrutamiento seguro, sistemas de reputación, control de acceso, confidencialidad de los datos, etc. Sin embargo, el proceso de asignación de identidades se ha dejado casi totalmente olvidado. Estas redes están diseñadas para que cada usuario tenga un identificador único (nodeID), pero la mayoría de los sistemas existentes permiten que los usuarios puedan obtener un conjunto de ellos, e incluso seleccionarlos. Ambas actuaciones provocan problemas importantes de seguridad, ya que gracias a ello los usuarios pueden alterar el adecuado funcionamiento de la red. En este trabajo proponemos un protocolo de asignación de nodeIDs basado en la emisión de *certificados implícitos*. Nuestro propósito es proporcionar servicios de seguridad que permitan luchar contra la mayoría de las amenazas que sufren estas redes, con especial atención a la asignación de identidades. Este protocolo se basa en el uso de certificados y la generación conjunta de nodeIDs por parte la Autoridad de Certificación (CA) y el nuevo usuario.**

***Palabras clave*—Ataque Eclipse, Ataque Sybil, Overlay P2P, Gestión de Identidades, Certificados Implícitos**

I. Introducción

Las redes Peer-to-Peer (P2P) estructuradas, de aquí en adelante overlays P2P, aparecieron hace unos años para resolver problemas de enrutamiento en grandes infraestructuras distribuidas, incluso a nivel de Internet; ya que pueden proporcionar escalabilidad, tolerancia a fallos, auto-organización y baja latencia. Por ello, y según el estudio anual "Cisco Visual Networking Index (VNI) Forecast" [1], el tráfico P2P representaba alrededor del 30 % del tráfico IP global en 2011 y crecerá con una tasa de crecimiento anual compuesto (CAGR) del 23 % de 2010 a 2015. Sin embargo, las overlays P2P no están siendo ampliamente utilizadas por aplicaciones comerciales, ya que presentan importantes problemas de seguridad. Todos sabemos que hoy en día la mayor parte de estas redes funcionan razonablemente bien sin necesidad de profundizar en la seguridad, pero no hay que perder de vista que proporcionan servicios gratuitos; y por ello los usuarios están dispuestos a asumir ciertas deficiencias (no garantizan una calidad de servicio (QoS) mínima), e incluso algún que otro riesgo.

Las aplicaciones P2P de video streaming son un buen ejemplo de aplicaciones comerciales que necesitan especial atención en la seguridad. La transmisión de video en redes P2P surgió como una evolución al intercambio de contenidos multimedia mediante descarga. Su poder para dar cabida a millones de usuarios, junto con su capacidad de resistencia al dinamismo, su fiabilidad y su bajo coste, son algunas de las razones por las que las redes P2P están siendo cada vez más utilizadas por este tipo de aplicaciones. El ejemplo es que las aplicaciones de video bajo demanda (VoD) producirán tres veces más tráfico en 2015 del que produjeron en 2011 [1]. SopCast, PPTV, CoolStreaming, TVUnetworks y Zattoo son algunas de las muchas aplicaciones de streaming de video que se han desarrollado hasta el momento. Sin embargo, la mayoría de ellas son plataformas propietarias, las cuales utilizan la segunda generación de redes P2P o distribuyen contenidos con un control de acceso deficiente, y poca o nula seguridad.

Debido a sus características, las overlays P2P presentan vulnerabilidad ante ciertos ataques, lo cual debe ser solucionado si queremos utilizar estas redes para implementar aplicaciones comerciales (como servicios de VoD bajo suscripción). Además, esta medida ayudaría a que los usuarios tengan más confianza en las redes P2P, ya que a menudo se piensa que son inseguras por naturaleza.

Las overlays P2P han sido analizadas en profundidad para garantizar su escalabilidad y eficiencia. Sin embargo, pocos mecanismos de seguridad se están utilizando en la actualidad. La mayoría de estas redes asumen que los nodos tienen un comportamiento honesto, pero este supuesto no es aceptable en entornos abiertos. La existencia de nodos anónimos y la falta de una autoridad centralizada capaz de controlar (y castigar) a los nodos, hace que estos sistemas sean vulnerables frente a comportamientos egoístas y maliciosos. Y desafortunadamente estos comportamientos no pueden ser evitados únicamente mediante el uso de los servicios básicos de seguridad. Las overlays P2P también deben seguir las primitivas de enrutamiento seguro descritas por Wallach en [2], que son: (1) el mantenimiento seguro de las tablas de enrutamiento, (2) el enrutamiento seguro de mensajes, y (3) la asignación segura de los identificadores de nodo (nodeIDs). Sin embargo, los dos primeros paradigmas dependen directamente del tercero. Si los nodeIDs pueden ser elegidos por los usuarios sin ningún tipo de control, podemos tener problemas de seguridad y funcionamiento. Desafortunadamente, hasta ahora se ha

ISBN: 978-84-9717-323-0

prestado poca atención a la forma en que los nodeIDs deben ser construidos, o cómo hacer los mecanismos de control de acceso más robusto. Al igual que cualquier otra red, las overlays P2P requieren de un control de acceso eficiente para prevenir el acceso de posibles atacantes a la red. Pero además, éstas deberían disponer de un sistema de asignación de identidades robusto, con el fin de mejorar la confianza de los usuarios en estas redes y que así puedan ser mayormente utilizadas por aplicaciones comerciales.

Por ello proponemos el uso de certificados digitales y un nuevo protocolo de asignación de identidades, el cual aprovecha la emisión de estos certificados. Más concretamente, proponemos el uso de *certificados implícitos* [3], [4], los cuales presentan ciertas ventajas sobre los certificados tradicionales (*certificados explícitos*). Los certificados implícitos tienen un menor tamaño, ya que no incluyen la firma de la entidad emisora, y pueden ser verificados más rápido, ya que requiere menos tiempo de cálculo la reconstrucción de una clave pública que verificar una firma digital. Por otra parte, la generación de estos certificados nos permite construir nodeIDs de forma segura. Para ello utilizamos las claves públicas, las cuales son construidas conjuntamente por los usuarios y la autoridad de certificación (CA), y así minimizamos el impacto de los ataques Eclipse [5], entre otros.

El resto del artículo está organizado de la siguiente forma: La sección II explica algunos de los problemas existentes en las overlays P2P relacionados con las identidades. La sección III presenta algunas propuestas que intentan prevenir, detectar y/o limitar los problemas de identidad experimentados en estas redes. La sección IV explica que son los esquemas de compromiso y describe un esquema de certificados implícitos basado en curvas elípticas. La sección V presenta nuestro protocolo de asignación de identidades para una overlay P2P. Y por último, en la sección VI se extraen algunas conclusiones.

II. Problemas de Identidad en Overlays P2P

La mayoría de las overlays P2P están implementadas utilizando tablas de hash distribuidas (DHTs), que almacenan pares $\{clave, valor\}$ junto con los nodeIDs creando un espacio virtual. Un *valor* puede ser un recurso (por ejemplo, un archivo), o la forma de llegar a él (un puntero), y la *clave* asociada indica su ubicación. La DHT se divide en subtablas, las cuales corresponden a una zona determinada del espacio virtual y van siendo asignadas a los diferentes nodos. Así cada nodo es responsable de una zona, y por lo tanto es responsable de los pares {clave,valor} contenidos en esa zona (almacenando mensajes de contenido y enrutamiento). Por lo general, una zona es asignada a un nodo cuyo nodeID es numéricamente cercano a los valores de las claves almacenadas en la subtabla correspondiente. Por lo tanto, la ubicación de los nodos en el espacio virtual está directamente relacionada con sus nodeIDs. Y desafortunadamente, en la mayoría de las overlays P2P actuales estos identificadores son generados por los usuarios localmente, lo que significa que pueden elegir sus nodeIDs y consecuentemente su ubicación en la overlay.

Los usuarios en la red CAN [6] son identificados por la zona que tienen asignada dentro del espacio virtual, zonas seleccionadas por ellos mismos. En las redes Chord [7] y Kademlia [8], los nodeIDs son generados por los usuarios utilizando una función hash sobre sus direcciones IP. En Pastry [9] los nodeIDs son asignados al azar por el software del cliente. Y de manera similar en otras overlays P2P.

Varios problemas relacionados con las identidades surgen de la asignación descontrolada de los nodeIDs: Ataques Sybil, ataques Eclipse, ataques Man-in-the-Middle (MITM), la presencia de whitewashers, etc. A continuación se describen algunas de las amenazas más importantes.

II-A. El ataque Sybil

La gestión de múltiples nodeIDs (Sybils) por parte del mismo nodo se conoce como ataque Sybil [10]. Llevando a cabo este ataque, un usuario malintencionado puede aumentar su presencia dentro de la overlay simulando artificialmente la existencia de varios nodos. Por lo tanto, el atacante que puede manejar un grupo de nodos puede alterar el funcionamiento de la red, o simplemente mejorar su reputación.

II-B. El ataque Eclipse

El ataque Eclipse [5] pretende alterar la información de enrutamiento de un nodo (o grupo de nodos) objetivo para aislarlo del resto de la overlay. El atacante interceptará los mensajes dirigidos a dicho nodo (o grupo) mediante un conjunto de nodos confabulados (o Sybils) que se encuentran cercanos al objetivo con el fin de controlar sus comunicaciones.

II-C. El Ataque Man-In-The-Middle (MITM)

Como su nombre indica, en este ataque el atacante se sitúa entre dos nodos con el propósito de espiar sus comunicaciones, o incluso manipularlas. Por lo general, en las redes P2P, el objetivo de estos atacantes es robar nodeIDs y/o generar información falsa. Por lo tanto, si tenemos en cuenta el tipo de enrutamiento de estas redes y permitimos que los nodeIDs sean seleccionados por los usuarios sin ningún control, no hay duda de que estas redes son extremadamente vulnerables a ataques MITM.

II-D. Otros Problemas

La eficiencia de los algoritmos de enrutamiento se basa en la uniforme distribución de los nodeIDs. Por lo tanto, el rendimiento de una overlay puede ser globalmente degradada si la mayoría de los nodeIDs pertenecen a una sola zona del espacio virtual. Y desafortunadamente, si los nodeIDs pueden ser seleccionados por los usuarios, nadie tendrá la seguridad de que los identificadores van a estar distribuidos uniformemente.

Otra amenaza a la seguridad relacionada con los nodeIDs es la presencia de whitewashers (nodos que intencionadamente abandonan la red y vuelven a entrar en ella con un nuevo nodeID con la intención de limpiar su mala reputación [11]). Los sistemas de reputación pueden ser utilizados para prevenir comportamientos maliciosos y promover la colaboración entre los nodos. Sin embargo, la eficacia de estos sistemas depende de la estabilidad de los nodeIDs.

III. Estado del Arte

Douceur [10] fue el primero en tratar el ataque Sybil en overlays P2P y comentar la imposibilidad de saber si dos nodos son gestionados por dos usuarios diferentes, o si en realidad lo hace uno solo; incluso recabando información de otros nodos de la red. De esta forma concluye que una entidad de confianza que certifique los nodeIDs es la única solución para evitar por completo el ataque Sybil en estas redes. Sin embargo, también sugiere el uso de métodos que añadan un coste computacional al proceso de obtención de nodeIDs para mitigar el ataque. Siguiendo esta línea, hasta la fecha se han propuesto muchas alternativas.

En [12], Castro et al. proponen dos formas centralizadas de generar nodeIDs. La primera de ellas es delegar el problema a un conjunto de entidades de confianza, las cuales firman los certificados vinculándolos con un nodeID aleatorio, una clave pública y la dirección IP del usuario. La segunda propuesta consiste en cobrar dinero por los certificados, u obligar a los usuarios a vincular su identidad real con los nodeIDs. Srivatsa y Liu proponen el uso de certificados con un tiempo de vida limitado y emitidos por una CA, el cual también vincula los certificados a nodeIDs aleatorios [13]. En [14], Butler et al. consideran el uso del encriptado basado en identidad (IBE), donde las claves públicas son derivadas directamente de los nodeIDs. Los nodeIDs son generados aleatoriamente por una CA y la autenticación de los nodos se lleva a cabo a través de un proceso de *callback* utilizando la dirección IP del usuario. En [15], Aiello et al. proponen, por una parte introducir la interacción humana en la fase de autenticación utilizando el protocolo OpenID, y por otra parte utilizar una entidad de confianza que vincule la identidad real del usuario con su clave pública y con un nodeID aleatorio para generar un LikirID. En [16], Rowaihy et al. han propuesto un mecanismo de puzzles criptográficos para limitar el ataque Sybil. Proponen un sistema de control de admisión utilizando una estructura jerárquica autoorganizada de nodos y una cadena de puzzles criptográficos. Ellos explotan dicha estructura jerárquica para distribuir la carga y aumentar la capacidad de resistencia a los ataques dirigidos, y actualizan los puzzles con frecuencia para así evitar la precomputación. En [12], [17], [18], sus autores también utilizan puzzles criptográficos para limitar el ataque Sybil.

IV. Background

IV-A. *Esquemas de compromiso (Commitment Schemes)*

Un esquema de compromiso es un protocolo interactivo entre dos participantes (Emisor y Receptor), destinado a ocultar temporalmente un valor que ya no debe ser cambiado. Es decir, el Emisor se compromete a utilizar un valor, el cual ha de permanecer temporalmente oculto para el Receptor. Estos sistemas suelen consistir de dos fases:

1. Fase 1 (Compromiso): el Emisor se compromete a utilizar un determinado valor.
2. Fase 2 (Revelación): el Emisor prueba al Receptor que el valor no ha sido cambiado desde entonces.

Estos esquemas son primitivas muy útiles en criptografía y siempre deben cumplir con dos propiedades: *Vinculación* y *Ocultación*. La Vinculación asegura que en la fase de Revelación un compromiso sólo pueda revelar con éxito un valor (unicidad). La Ocultación garantiza que la fase de Compromiso no revela ninguna información sobre el valor oculto (secreto prefecto). Tanto la Vinculación como la Ocultación pueden ser garantizadas (estadística o computacionalmente) en función de la potencia de cálculo necesaria para romperlas. Estos esquemas son aplicados en protocolos tales como las pruebas de conocimiento cero (Zero-knowledge), la computación multiparte, las subastas digitales o el comercio electrónico.

En este artículo nosotros definimos un nuevo esquema de compromiso basado en la criptografía basada en curvas elípticas (ECC) con el fin de mejorar la seguridad en un protocolo de emisión de certificados implícitos.

IV-B. *Certificados Implícitos*

Un certificado estándar contiene explícitamente la clave pública del usuario y la firma de la CA que ha emitido dicho certificado, junto con otra información adicional (número de serie, período de validez, identidad del emisor, identidad del usuario, etc.). Un certificado implícito [3], [4] no contiene la clave pública del usuario ni la firma de la CA. En lugar de ello contiene la información necesaria para calcular su clave pública asociada, un parámetro de reconstrucción.

Por lo tanto, un certificado implícito es simplemente un par (I, Z), donde I denota la información incluida en el certificado y Z denota el parámetro de reconstrucción. Los certificados implícitos tienen una longitud más corta que los explícitos y proporcionan así una alternativa más eficiente.

Antes de validar la firma de un emisor, cualquier receptor de un certificado implícito debe reconstruir la clave pública asociada utilizando Z y la clave pública de la CA emisora. De la misma manera que con los certificados explícitos, el receptor debe confiar en la CA y disponer de su clave pública para tener así la seguridad de que la clave reconstruida ha sido emitida por dicha CA. Con los certificados explícitos, el receptor verifica la firma del certificado con la clave pública de la CA, y a partir de ese momento puede estar seguro de que la clave contenida en el certificado pertenece a un determinado usuario y ha sido emitida por esa CA. Sin embargo, validar únicamente el certificado no es suficiente para autenticar a un usuario. Por lo tanto, para autenticar a un usuario, éste debe demostrar el conocimiento de la clave privada asociada utilizando un protocolo criptográfico seguro. Y lo mismo aplica a los certificados implícitos, donde la autenticación de una clave pública y la autenticación de pertenencia a un usuario no son separables.

La figura 1 ilustra el esquema de emisión de certificados implícitos "Elliptic Curve Qu-Vanstone" (ECQV) [4], propuesto por el Standards for Efficient Cryptography Group (SECG). En él un usuario X solicita un nuevo certificado enviando un punto aleatorio dentro de una curva elíptica (N_X), el cual es utilizado por la CA para generar el parámetro de reconstrucción de su nueva clave pública ($Z = N_X + N$). Una

vez calculado Z, la CA calcula el valor del hash del certificado ($h = H(I||Z)$) y la firma (s). Finalmente X recibe su nuevo certificado (Z, I) y su firma (s), genera su nuevo par de claves criptográficas (la clave privada d_X y la clave pública Q_X) utilizando la clave pública de la CA (Q_{CA}).

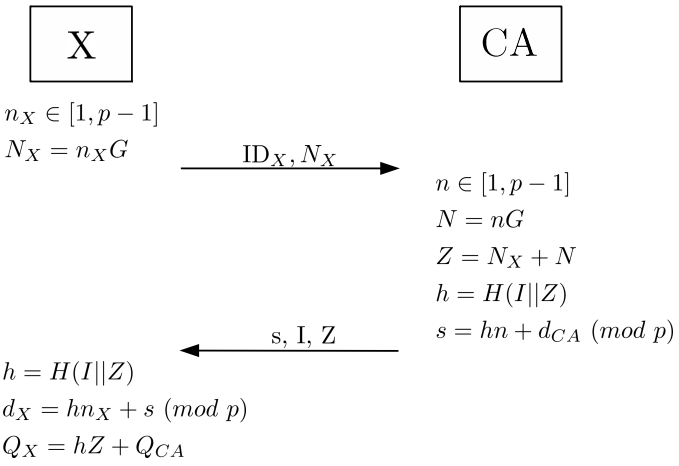

Figura 1. Protocolo de Emisión de Certificados Implícitos ECQV.

En este artículo proponemos una serie de modificaciones sobre este sistema de emisión de certificados con el fin de proponer un nuevo sistema de gestión de identidades seguro.

V. Protocolo de Asignación de Identidades

Nuestro protocolo asigna nodeIDs de forma segura y eficiente aprovechando la emisión de certificados implícitos. Estos certificados proporcionan identificación digital para autenticar usuarios, soporte a la criptografía de clave pública, etc.; pero también presentan ciertas ventajas sobre los certificados tradicionales. En este protocolo, los NodeIDs se calculan utilizando una función de hash sobre la clave pública de los usuarios, pero a diferencia de otras propuestas, estas claves públicas son generadas bajo la supervisión y participación de una CA, la cual no conoce las claves privadas asociadas. El esquema de emisión de certificados ECQV [4] ha sido modificado con el fin de garantizar que ninguna de las dos partes involucradas en el proceso tenga la capacidad de elegir el valor de la clave pública emitida.

V-A. Suposiciones y Clarificaciones

Con el fin de adaptar el esquema ECQV a nuestras necesidades, hemos definido un nuevo esquema de compromiso basado en curvas elípticas. Este esquema ha sido construido inspirándonos en el cifrado Exclusive-OR (XOR) y suponiendo que un emisor S posee una clave privada d_S y una clave pública $Q_S = d_S G$, donde G es el generador de la curva elíptica. La figura 2 describe el protocolo en detalle, donde u es un número aleatorio, U es el punto de la curva elíptica asociado a u, c es el valor de compromiso y v es el valor elegido por S. En la primera fase, S se compromete a utilizar un valor v enviando los valores c y U a R. Y en la segunda fase, S revela el valor del número aleatorio utilizado (u) y R chequea que u realmente fue utilizado para generar U. Finalmente R calcula el valor de v utilizando c.

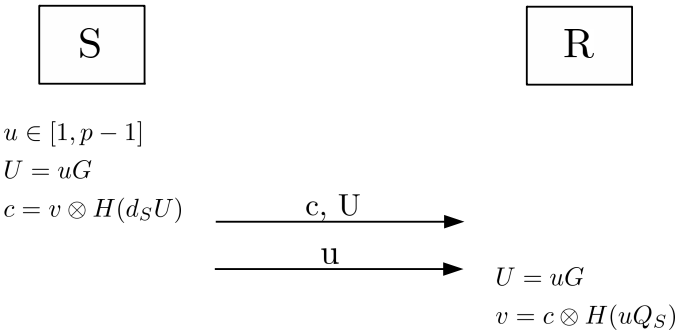

Figura 2. Nuevo Esquema de Compromiso.

En este esquema de compromiso se cumplen las dos principales propiedades de seguridad que requieren este tipo de protocolos: *Vinculación* y *Ocultación*. La Vinculación es segura, ya que dos valores diferentes de u no pueden dar como resultado el mismo valor correcto de v. Y la Ocultación es computacional, ya que dados los valores c, U y G, un atacante puede resolver el problema del logaritmo discreto en una curva elíptica (ECDLP) para obtener el valor aleatorio u, para luego para calcular v.

La modificación realizada en el esquema ECQV ha consistido en añadir el esquema de compromiso anteriormente explicado. Con ello evitamos que la CA pueda elegir el valor del parámetro de reconstrucción (Z) una vez conocido el valor enviado por el usuario (N_X) durante el proceso de generación de certificados. Esto implica que ahora la CA debe seleccionar N y enviar un compromiso (c, U) al usuario antes de recibir N_X. Así, una vez que el usuario ha recibido su nuevo certificado, él será capaz de verificar que el certificado se ha generado utilizando el valor al que se comprometió la CA.

V-B. Especificaciones

En esta sección describimos el protocolo en detalle; la información intercambiada, la forma de intercambiarla, los mecanismos de seguridad utilizados, etc. La figura 3 muestra la información intercambiada por ambas partes y las operaciones llevadas a cabo, pero sin tener en cuenta las operaciones de cifrado y firma. La tabla I presenta un resumen global de la notación utilizada a lo largo de esta sección.

V-B1. Paso 1: Cada vez que un usuario quiere unirse a la overlay P2P contacta con la CA enviando un "HELLO MESSAGE", el cual contiene las identidades (ID_X y ID_{CA}), una marca temporal (t_X), también utilizada como identificador de petición, y el certificado del usuario (C_X). Este mensaje es firmado por el usuario utilizando su clave privada en el mundo real, y cifrado utilizando la clave pública de la CA.

$$HELLO\ MESSAGE,\ X \rightarrow CA:$$
$$\{C_X, \{ID_X, ID_{CA}, t_X\}_{d_X}\}_{Q_{CA}}$$

Tabla I
NOTACIÓN

p	El orden del cuerpo finito $\mathbb{F}_p$.
G	El generador de la curva elíptica definida sobre $\mathbb{F}_p$ ($E(\mathbb{F}_p)$).
ID_X	La identidad del nuevo usuario X.
ID_{CA}	La identidad de la CA.
P_X	El seudónimo de X dentro de la overlay (nodeID).
C_X	El certificado digital de X en el mundo real.
d_X	La clave privada de X en el mundo real.
Q_X	La clave pública de X en el mundo real.
d_{Xo}	La clave privada de X dentro de la overlay.
Q_{Xo}	La clave pública de X dentro de la overlay.
d_{CA}	La clave privada de la CA.
Q_{CA}	La clave pública de la CA.
t_X	La marca temporal generada por el nuevo usuario (identificador de petición).
I	La información incluida en el certificado de X.
Z	El parámetro de reconstrucción de la clave pública de X.
h	El resumen del nuevo certificado ($I\|\|Z$).
s	La firma de la CA para el nuevo certificado de X.
$n,\ u,\ n_X$	Parámetros privados de la petición de X ($\in [1,\ p-1]$).
$N,\ U,\ N_X$	Parámetros públicos de la petición de X ($\in E(\mathbb{F}_p)$).
c	El valor de compromiso.
$H(m)$	Una función de hash sobre un mensaje m.
$i \rightarrow j:$	El envío de un mensaje de la entidad i a la entidad j.
$\{m\}_Q$	El texto cifrado de un mensaje m utilizando la clave pública Q.
$\{m\}_d$	La firma sobre un mensaje m utilizando la clave privada d.

V-B2. Paso 2: Cada vez que un nuevo usuario contacta con la CA, ésta genera dos parámetros privados ($\{n,\ u\} \in [1,\ p-1]$), sus respectivos puntos en la curva elíptica ($N = nG$ y $U = uG$) y calcula $c = N \otimes H(d_{CA}U)$. Finalmente envía c y U a X, todo firmado y cifrado junto con las identidades y la marca temporal.

$$ACCEPT\ MESSAGE,\ CA \rightarrow X:$$
$$\{\{ID_{CA}, ID_X, t_X, c, U\}_{d_{CA}}\}_{Q_X}$$

V-B3. Paso 3: X recibe el "ACCEPT MESSAGE" y genera un parámetro privado $n_X \in [1,\ p-1]$ y su punto asociado $N_X = n_X G$. Después envía N_X a la CA, firmado y cifrado junto con las identidades y la marca temporal.

$$REQUEST\ MESSAGE,\ X \rightarrow CA:$$
$$\{\{ID_X, ID_{CA}, t_X, N_X\}_{d_X}\}_{Q_{CA}}$$

V-B4. Paso 4: La CA recibe el "REQUEST MESSAGE", calcula el parámetro de reconstrucción ($Z = N_X + N$) y el valor de hash de ese parámetro concatenado con I ($h = H(I||Z)$). Después firma el certificado ($s = hn + d_{CA}\ mod\ p$) y le proporciona a X los valores $\{s, r, I, Z\}$, todos firmados y cifrados junto con las identidades y la marca temporal.

$$RESPONSE\ MESSAGE,\ CA \rightarrow X:$$
$$\{\{ID_{CA}, ID_X, t_X, s, u, I, Z\}_{d_{CA}}\}_{Q_X}$$

Note que una vez que Z y h han sido calculados, si $Q_{Xo} = hZ + Q_{CA} = \mathcal{O}$, la CA le pide al usuario que le mande un nuevo parámetro y repite el proceso.

V-B5. Paso 5: X recibe su nuevo certificado y la firma de la CA ("RESPONSE MESSAGE"), y calcula $U' = uG$ (y compara éste con U), $N = Z - N_X$ y $N' = H(uQ_{CA}) \otimes c$ para verificar que la CA ha utilizado el valor inicial n; si no es así cancela el proceso. Después X genera su clave privada $d_{Xo} = hn_X + s\ mod\ p$ y su clave pública $Q_{Xo} = d_{Xo}G$, y calcula su nuevo nodeID como el valor de hash de Q_{Xo} ($P_X = H(Q_{Xo})$). Finalmente envía su nodeID junto con las identidades y la marca temporal, todo firmado y cifrado.

$$CONFIRMATION\ MESSAGE,\ X \rightarrow CA:$$
$$\{\{ID_X, ID_{CA}, t_X, P_X\}_{d_X}\}_{Q_{CA}}$$

V-B6. Generación de la Clave Pública: Cada vez que un usuario recibe un mensaje, éste necesita generar la clave pública del emisor para poder autenticarlo y verificar la firma del mensaje. Para ello utiliza su certificado implícito, el cual incluye la información del certificado (I) y el parámetro de reconstrucción (Z). Finalmente sigue los siguientes pasos:

1. Calcula el parámetro $h = H(I||Z)$.
2. Genera la clave pública del emisor $Q_{Xo} = hZ + Q_{CA}$.
3. Verifica la firma del mensaje utilizando Q_{Xo}.

Nótese que dicha verificación se cumplirá porque:

$Q_{Xo} = d_{Xo}G = hn_XG + sG = hn_XG + hnG + d_{CA}G = hN_X + hN + Q_{CA} = h(N_X + N) + Q_{CA} = hZ + Q_{CA}$.

V-B7. Validación del nodeID: Cada vez que un nodo recibe información de otro nodo (contenidos o información de enrutamiento) debe validar su nodeID. Para ello, el nodo sólo tiene que calcular el hash de su clave pública ($P_X = H(Q_{Xo})$) y compararlo con el nodeID utilizado.

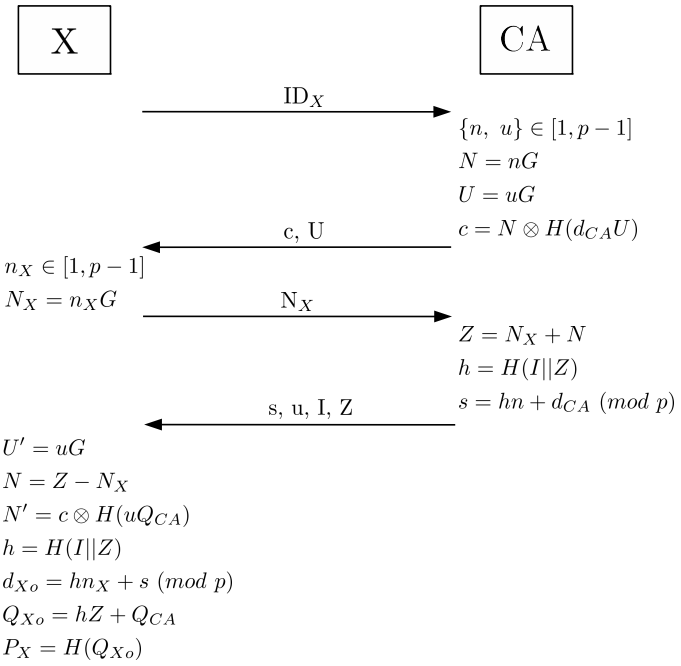

Figura 3. Esquema de Generación de nodeIDs/Certificados.

VI. Conclusiones

La vulnerabilidad a ciertos ataques es un fuerte obstáculo para el desarrollo de aplicaciones comerciales en las overlays P2P. En este artículo se ha propuesto un esquema seguro de asignación de identidades con el objetivo de resolver algunas de estas vulnerabilidades y convertir estas redes en una potente plataforma para aplicaciones comerciales. Nuestro protocolo genera nodeIDs de forma segura y anónima, sin afectar al funcionamiento actual de la red. Éste no permite ni que los usuarios seleccionen sus nodeIDs ni que la CA pueda seleccionarlos por ellos, y garantiza que los nodos sean ubicados en el espacio virtual de forma pseudo-aleatoria (uniformemente). Finalmente, hay que tener en cuenta que cualquier sistema de seguridad implica un compromiso entre el nivel de seguridad y el rendimiento de la red. Pero en nuestro caso, y teniendo en cuenta que un usuario sólo ejecutaría el protocolo la primera vez que quiere unirse a la red, la calidad experimentada por el usuario (QoE) no se verá afectada. En cuanto a la seguridad, nuestra propuesta sólo tiene una debilidad; debemos confiar en la CA. Pero hay que tener en cuenta que utilizar una CA es la única forma de evitar 100 % ciertos ataques (ataque Sybil, ataque Eclipse, etc.). El trabajo futuro se centrará en proponer un sistema de gestión de identidades que proporcione trazabilidad de usuarios y revocación de certificados y nodeIDs.

Reconocimientos

Este trabajo ha sido parcialmente subvencionado por la Secretaría de Estado de Investigación, Desarrollo e Innovación bajo los proyectos SERVET TEC2011-26452 y CONSOLIDER CSD2007-00004 (ARES), y por la Generalitat de Catalunya bajo la ayuda 2009 SGR-1362 para grupos consolidados.

Referencias

[1] Cisco Systems, Inc, "Cisco Visual Networking Index: Forecast and Methodology, 2011-2016," http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-481360_ns827_Networking_Solutions_White_Paper.html.

[2] D. S. Wallach, "A Survey of Peer-to-Peer Security Issues," in *Proceedings of the Mext-NSF-JSPS international conference on Software security: theories and systems*, ser. ISSS'02. Tokyo, Japan: Springer-Verlag Berlin, Heidelberg, 2002, pp. 42–57.

[3] D. R. Brown, R. Gallant, and S. A. Vanstone, "Provably Secure Implicit Certificate Schemes," in *Financial Cryptography*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2002, vol. 2339, pp. 156–165.

[4] C. Research, "Standards for Efficient Cryptography 4 (SEC 4): Elliptic Curve Qu-Vanstone Implicit Certificate Scheme (ECQV)," November 2013, version 1.1.

[5] R. Fantacci, L. Maccari, M. Rosi, L. Chisci, L. M. Aiello, and M. Milanesio, "Avoiding Eclipse Attacks on Kad/Kademlia: An Identity Based Approach," in *Proceedings of the IEEE International Conference on Communications*, ser. ICC'09. IEEE Press, june 2009, pp. 983–987.

[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, San Diego, CA, USA, 2001, pp. 161–172.

[7] I. Stoica, R. Morris, D. R. Karger, M. F. Kaashoek, and H. Balakrishman, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," in *Proceedings of the ACM Conference on Applications, Technologies, Architectures and Protocols for Computer Communication (SIGCOMM)*, San Diego, CA, USA, 2001, pp. 149–160.

[8] P. Maymounkov and D. Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," in *Proceedings of the 1st International Workshop on Peer-to Peer Systems*, ser. IPTPS'02, march 2002, pp. 53–65.

[9] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems," in *Proceedings of the IFIP/ACM International Conference on Distributed Systems Platforms*, 2001, pp. 329–350.

[10] J. R. Douceur, "The Sybil Attack," in *Proceedings of the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS'02. London, UK: Springer-Verlag, 2002, pp. 251–260.

[11] S. Marti and H. Garcia-Molina, "Taxonomy of Trust: Categorizing P2P Reputation Systems," *Computer Networks*, vol. 50, no. 4, pp. 472–84, 2006.

[12] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D. S. Wallach, "Secure Routing for Structured Peer-to-peer Overlay Networks," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 299–314, december 2002.

[13] M. Srivatsa and L. Liu, "Vulnerabilities and Security Threats in Structured Overlay Networks: A Quantitative Analysis," in *Proceedings of the 20th Annual Computer Security Applications Conference*, december 2004, pp. 252–261.

[14] K. R. Butler, S. Ryu, P. Traynor, and P. D. McDaniel, "Leveraging Identity-Based Cryptography for Node ID Assignment in Structured P2P Systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 12, pp. 1803–1815, december 2009.

[15] L. M. Aiello, M. Milanesio, G. Ruffo, and R. Schifanella, "An identity-based approach to secure P2P applications with Likir," *Peer-to-Peer Networking and Applications*, vol. 4, pp. 420–438, 2011.

[16] H. Rowaihy, W. Enck, P. McDaniel, and T. L. Porta, "Limiting Sybil Attacks in Structured P2P Networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications*, Anchorage, Alaska, USA, may 2007, pp. 2596–2600.

[17] W. L. D. C. Cordeiro, F. R. Santos, G. H. Mauch, M. P. Barcelos, and L. P. Gaspary, "Identity management based on adaptive puzzles to protect P2P systems from Sybil attacks," *Comput. Netw.*, vol. 56, no. 11, pp. 2569–2589, july 2012.

[18] C. Lu, "Detection and Defense of Identity Attacks in P2P Network," in *Advances in Computation and Intelligence*, ser. Lecture Notes in Computer Science. Springer-Verlag Berlin Heidelberg, 2009, vol. 5821, pp. 500–507.