Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree programme of Computer Science and Engineering

Antti Tapio

Person-to-Person Identification on Modern Communication and Collaboration Environments

Master's Thesis
Espoo, June 1, 2010

Supervisor: Professor Tuomas Aura, Helsinki University of Technology
Instructor: Pasi Lindholm M.Sc. (Tech.), NorthID Oy

Aalto University

Aalto University
School of Science and Technology
Faculty of Information and Natural Sciences
Degree Programme of Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author:	Antti Tapio	
Title of thesis:	Person-to-Person Identification on Modern Communication and Collaboration Environments	
Date:	June 1, 2010	**Pages:** 15 + 85
Professorship:	Data Communications Software	**Code:** T-110
Supervisor:	Professor Tuomas Aura	
Instructor:	Pasi Lindholm M.Sc. (Tech.)	

This thesis describes a method for person-to-person identification on Google Wave networks. The method can also be used for strong authentication on the Wave network.

The solution is based on using a trusted third party. The users must first authenticate themselves to a trusted third party and then prove to it that they control a said Wave user account. After these steps, the trusted third party is then able to identify the users participating in a Wave discussion and report the identification results to the other participants. The users can request the trusted third party to reauthenticate a user if needed. The thesis describes also a federated model for person-to-person identification on the Wave network using multiple trusted third parties.

The method described can be generalized to any communication networks where the origin of messages can be reliably traced on a domain name level.

A proof-of-concept of the identification model was developed and it was used to evaluate the applicability of the model in the real world.

Keywords:	person-to-person, identification, strong authentication, trusted third party, Wave, XMPP
Language:	English

Aalto-yliopisto
Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietotekniikan koulutusohjelma

Aalto-yliopisto

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Antti Tapio	
Työn nimi:	Käyttäjien välinen henkilöllisyyden todentaminen nykyaikaisissa kommunikaatio- ja yhteistyöympäristöissä	
Päiväys:	1. kesäkuuta 2010	**Sivumäärä:** 15 + 85
Professuuri:	Tietoliikenneohjelmistot	**Koodi:** T-110
Työn valvoja:	Professori Tuomas Aura	
Työn ohjaaja:	Diplomi-insinööri Pasi Lindholm	

Diplomityössä kuvataan menetelmä käyttäjien väliseen henkilöllisyyden todentamiseen Google Wave-verkossa. Kuvattua menetelmää voidaan käyttää myös henkilöiden vahvaan tunnistamiseen Wave-verkossa.

Ratkaisu perustuu luotetun kolmannen tahon käyttöön. Käyttäjien tulee ensin tunnistautua luotetulle kolmannelle taholle ja sen jälkeen osoittaa luotetulle taholle omaavansa tietyn Wave-käyttäjätunnuksen. Tämän jälkeen luotettu kolmas taho voi tunnistaa käyttäjät Wave-verkossa ns. Wave-robotin avulla ja kertoa tunnistamisen tulokset muille osallistujille. Tarvittaessa käyttäjät voivat pyytää robotin avulla luotettua tahoa uudelleentunnistamaan käyttäjät. Työssä esitetään myös malli henkilöiden väliseen tunnistamiseen useamman luotetun tahon avulla.

Menetelmä on yleistettävissä käytettäväksi sellaisissa keskusteluverkoissa, joissa voidaan luotettavasti tunnistaa, miltä verkon palvelimelta kommunikaatio on tapahtunut.

Työssä toteutettiin tekninen kokeilu kehitetystä todennusmenetelmästä ja arvioitiin menetelmän soveltuvuutta käytäntöön.

Avainsanat:	käyttäjien välinen todentaminen, vahva tunnistaminen, luotettu kolmas taho, Wave, XMPP
Kieli:	englanti

Acknowledgements

First of all, I would like to thank my instructor M.Sc. (Tech.) Pasi Lindholm. The initial idea of implementing strong person-to-person identification to Google Wave came from him. In addition to the great idea, he provided valuable guidance for me in the process of writing a Master's Thesis.

I would like to thank my supervisor Professor Tuomas Aura for having time for interesting discussions and for giving valuable feedback during my research. I also appreciate his genuine interest in the subject of this thesis.

Last but not least, I thank my family, my parents and Leena for supporting me always!

Helsinki, June 1, 2010

Antti Tapio

Abbreviations and Acronyms

AIM	AOL Instant Messenger
API	Application Programming Interface
ATM	Automated teller machine
B2B	Business-to-Business
BITNET	Because It's Time NETwork
CA	Certification Authority
CC	Common Criteria for Information Technology Security Evaluation
CMS	Cryptographic Message Syntax
CSP	Credential Service Provider
CSS	Cascading Style Sheets
CVV	Card Verification Value
DAC	Discretionary Access Control
DN	Distinguished Name
DNA	Deoxyribonucleic acid
DNS	Domain Name System
EDI	Electronic Data Interchange
ETSI-MSS	ETSI Mobile Signature Service

ETSI	European Telecommunications Standards Institute
GCHQ	British Government Communication Headquarters
GSSAPI	Generic Security Services Application Program Interface
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IBM	International Business Machines
IdP	Identity Provider
IETF	Internet Engineering Task Force
IM	Instant Messaging
IMAP	Internet Message Access Protocol
IP	Internet Protocol
IRC	Internet Relay Chat
ITU	International Telecommunication Union
ITU-T	Telecommunication Standardization Sector of ITU
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
MAC	Mandatory Access Control
MIME	Multipurpose Internet Mail Extensions
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OT	Operational Transformation
OTP	One-time password
OTR	Off-the-Record Messaging

P2P	Peer-to-Peer
PGP	Pretty Good Privacy
PIN	Personal identification number
PKCS	Public Key Cryptography Standard
PKCS#7	Public Key Cryptography Standard #7
PKI	Public Key Infrastructure
POP	Post Office Protocol
PSTN	Public switched telephone network
RA	Registration Authority
RFC	Request for Comments
RFID	Radio-frequency identification
S/MIME	Secure/Multipurpose Internet Mail Extensions
SAML	Security Assertion Markup Language
SASL	Simple Authentication and Security Layer
SCRAM	Salted Challenge Response Authentication Mechanism
SET	Secure Electronic Transaction
SIM	Subscriber Identity Module
SIMPLE	Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions
SIP	Session Initiation Protocol
SLIC	Secure Internet Live Conferencing protocol
SLR	Single-lens reflex
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol

SNS	Social Network Service
SP	Service Provider
SRP	Secure Remote Password
TCSEC	Trusted Computer System Evaluation Criteria
TLS	Transport Layer Security
TTP	Trusted Third Party
UI	User Interface
URL	Uniform Resource Locator
UUCP	Unix-to-Unix Copy
VAN	Value-added Network
VoIP	Voice over IP
X.509	ITU-T Public-Key Infrastructure
XEP	XMPP Extension Protocol
XML	Extensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
XSF	The XMPP Standards Foundation

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background and Motivation

There is a multitude of secure online services on the Internet. For example, many banks have secure online banking services on the web running on Hypertext Transfer Protocol (HTTP) secured by Transport Layer Security (TLS). Almost every service has its own user interface for identifying and authenticating the users. The organizations encourage users to send all secure communication over these proprietary channels.

However, due to the number of different secure channels that a user would have to follow, the user often ends up in using email instead. Even though email is by default not secure, it is still widely used for exchanging all kinds of sensitive information. Even official, valuable and confidential information is transferred using email. The current email architecture does not offer easy ways for secure identification of messaging partners. While several security mechanisms for email have been proposed and even standardized, they integrate poorly to the user experience and none of them has gained wide acceptance.

In 2009, Google introduced a new service called Wave. It is an architecture intended to merge the best features from both non-real-time (email) and real-time Instant Messaging (IM) communication. Since messaging in this service has been redesigned from the ground up, it gives an opportunity to rethink also the security and user authentication side.

The Wave protocol utilizes Extensible Messaging and Presence Protocol (XMPP) as the actual authentication and data transport method. XMPP has a well defined and robust authentication mechanism for servers. Wave uses XMPP in such way that all communication between servers is protected.

The current Wave specification does not define a secure and trustworthy way to enable users to identify each other. In business and governmental environments however, there is a need to identify participants in a trustworthy way to make sure that confidential information is not leaked to outsiders. If Wave is going to be used in these formal environments, it should provide robust user identification methods. Because of the openness and modularity of the Wave architecture and especially because of the modularity of XMPP, there might be a way to implement a method for reliable person-to-person identification.

The interesting question is: Could the Wave architecture offer an easy to use but still secure user-to-user identification method?

1.2 Problem Statement and Scope

In this light, this thesis seeks out to find answers for the following questions:

- How can users identify each other on the Wave network?
- How can a real identity be linked to a Wave address and verified?
- Is there a method for strongly authenticating users on the Wave network?

The scope of this thesis is to research a person-to-person identification method for the Wave network and to validate the presented solution on a technical level by developing a proof-of-concept implementation.

Existing person-to-person communication methods and authentication and identification concepts are surveyed and analyzed. We also discuss the Wave protocols and their relationship to the underlying Extensible Messaging and Presence Protocol (XMPP).

This thesis also includes a basic evaluation of the usability and security of the method.

1.3 Organization of the Thesis

In chapter 2 we describe the different methods for person-to-person communication on the Internet and what kind of methods are used for person-to-person identification. The next, chapter 3, analyzes the XMPP and Wave protocols that are used in our solution. Authentication and identification concepts, architectures and threats are discussed in chapter 4. Chapter 5 describes our research questions in detail and defines a method for person-to-person identification on the Wave network. The actual implementation is presented in chapter 6. Analysis of the presented method and its strengths and weaknesses and lessons learned from the implementation are discussed in chapter 7. Finally, we draw all the strings together and form a conclusion of the research in chapter 8.

Chapter 2

Person-to-Person Communication on Internet

There are numerous networks and protocols on the Internet that can be used for person-to-person communication. We have identified four main types of communication channels, electronic mail, Instant Messaging (IM), Social Networks and voice communication. They are discussed in this chapter.

Person-to-person communication involves discussion about confidential matters. Thus it is important to have a reliable way to identify the other party with whom the discussion takes place. We go through the methods available nowadays in each type of communication channel and find out that most identification methods are based on public-key cryptography that seems to be hard to understand and use for the general population.

Person-to-Person communication can be divided into online and offline usage situations. Online communications happen when both parties are connected to the network at the same time. If the receiver is not connected to the net, the communication is more like non-realtime fire-and-forget-type messaging. Email is the most widely used off-line messaging tool.

2.1 Electronic Mail

Email is the electronic counterpart for real-world letter correspondence and the basis for person-to-person communication on the Internet. Even before there were computer networks, multi-user computer systems like the IBM 7094 allowed sending messages between users (Vleck, 2009).

Email was the most popular service on the ARPANET. Unix-to-Unix Copy (UUCP)[1] and IBM VNET protocols made it possible to send messages between users in different networks. RFC 822 standardized the format of email messages. (Quarterman & Hoskins, 1986)

X.400 is an ITU-T standard for exchanging mails (ITU-T, 1999). It was introduced in 1984, but it never gained widespread popularity on the Internet. X.400 is still used in some Business-to-Business (B2B) Value-added Networks (VANs) for transmitting primarily Electronic Data Interchange (EDI) messages (Silva, 2003).

Simple Mail Transfer Protocol (SMTP)[2], Internet Message Access Protocol (IMAP)[3] and Post Office Protocol (POP)[4] are the standard communication protocols used in the modern email infrastructure. Messages are formatted according to the Multipurpose Internet Mail Extensions (MIME)[5] standards. MIME is an Internet standard that defines how to support international character sets in both headers and bodies, binary attachments and multipart email messages.

Security solutions are widely available for email but not widely used. For Person-to-Person authentication and identification over email the most common methods are Secure/Multipurpose Internet Mail Extensions (S/MIME) and Pretty Good Privacy (PGP). Both are based on public-key cryptography. Neither has gained large-scale popularity due to obstacles in taking either method into use. (Roth et al., 2005)

[1] RFC 976 (Horton, 1986)
[2] RFC 5321 (Klensin, 2008)
[3] RFC 3501 (Crispin, 2003)
[4] RFC 1939 (Myers & Rose, 1996)
[5] RFC 2045 (Freed & Borenstein, 1996) etc.

2.1.1 Secure/Multipurpose Internet Mail Extensions

S/MIME is an extension to MIME that standardizes the structure and processes of cryptographically signing and encrypting messages (RFC 3851). Cryptographic Message Syntax (CMS)[6] is closely related to S/MIME as it defines a S/MIME compatible format for signing and encrypting arbitrary data. S/MIME and CMS are based on Public Key Cryptography Standard #7 (PKCS#7)[7], which uses Public Key Cryptography Standard (PKCS) specifications regarding public and private keys and their usage.

As S/MIME is based on Public Key Infrastructure (PKI), efficient use of S/MIME requires obtaining certificates from a Certification Authority (CA). Otherwise trust chains cannot be formed and person-to-person identification cannot be established. Combined to the fact that setting up the needed certificates and keys is not an easy task, these factors have limited the adoption of S/MIME (Gaw et al., 2006). For details about PKI, see section 4.3.4.

2.1.2 OpenPGP

OpenPGP[8] is the standardized version of PGP, which, like S/MIME, is based on public and private keys. The main difference is that while PKI is based on an infrastructure, OpenPGP uses a concept called Web of Trust (see section 4.3.3). Forming a web of trust requires considerable effort which has limited the use of PGP to only special cirumstances. Usage of OpenPGP in email is defined in RFC 3156.

2.2 Instant Messaging

Email is the electronic version of letter correspondence. In analogy, it could be said that Instant Messaging (IM) is the electronic version of a face-to-face discussion. The key difference to email is that all participants are online during the chat. Nowadays modern protocols, like XMPP, have functionality to store messages that are received while the recipient is offline and to forward them when the user is online again.

[6] RFC 5652 (Housley, 2009)
[7] RFC 2315 (Kaliski, 1998)
[8] RFC 4880 (Callas et al., 2007)

The first IM service was CompuServe's *CB Simulator*, launched in Feburary 21, 1980. It was presented as a electronic version of a Citizens' band radio and thus had 40 channels and commands like *squelch*. It was an instant success, and some people even ended up using the CB Simulator so much that they could not afford to pay their online service bill. (Banks, 2008)

Next year, 1981, Because It's Time NETwork (BITNET) was launched in the United States. It spread quickly to Canada, Europe and Japan, and as of May 1, 1986 there were 1306 hosts in 17 countries. BITNET made almost real-time communication possible. As Quarterman & Hoskins put it, there were *only moderate delays, usually less than eight seconds.*(Quarterman & Hoskins, 1986).

In 1988, Jarkko Oikarinen invented Internet Relay Chat (IRC). An IRC network consists of interconnected servers, and clients each communicating through one server. IRC is still used nowadays with an average of 750.000 simultaneous users connected to just under 5.000 servers (Gelhausen, 2010). (RFC 1459)

ICQ (November 1996) and AOL Instant Messenger (AIM) (May 1997) made IM popular. They were the first networks to organize the discussion primarily between persons, not under topics like the solutions before had done (Nardi et al., 2000; Grinter & Palen, 2002). This proved to be a successful design choice as usage grew and new networks and products copying the same idea were introduced[9]. The underlying communication protocols used in these networks are all closed. (Preece et al., 2003)

In 2000, an open protocol called Jabber was launched. This was later renamed as Extensible Messaging and Presence Protocol (XMPP) and standardized as an Internet protocol by RFC 3920. XMPP is widely used in many IM networks[10]. Detailed discussion about XMPP can be found in section 3.1. In 2009 Google introduced Google Wave that is an extension for XMPP. It is discussed in section 3.2.

Another open IM protocol Session Initiation Protocol for Instant Messaging and Presence Leveraging Extensions (SIMPLE) (RFC 3428) is based on Session Initiation Protocol (SIP) (RFC 3261). It has not gained momentum on the Internet.

[9]Yahoo Pager! (1998), MSN Messenger (1999), IBM Lotus Sametime (1998)
[10]For example Google Chat, Apple iChat and Facebook Chat

According to our research, the IM networks and protocols do not support person-to-person identification in general. The closed protocol networks, like ICQ and AIM, have a centralized architecture where all servers are controlled by one commercial entity. Thus, it is assumed that, if users trust the company, they also trust authentications done by the servers and the user information given by the server. In XMPP, as will be discussed later in section 3.1, server-to-server and client-to-server authentication is solved by the core protocol, but client-to-client is not[11]. There have been attempts to create IM network protocols with proper authentication features but none have been adopted into general use. Secure Internet Live Conferencing protocol (SLIC) is an example of a network protocol offering end-to-end encryption (Riikonen, 2007).

There are authentication and encryption solutions for IM. They are mostly engineered to work on top of the actual IM protocol and integrated to the client software so that the underlying IM protocol has no knowledge of the solution. For example, Off-the-Record Messaging (OTR)[12] and the pidgin-encryption plugin[13] are such solutions. Both use public-key cryptography for end-to-end authentication and encryption. As with email security protocols using public-key cryptography, this causes usability issues that hinder the adoption of these protocols. Usability of OTR has been studied by Stedman et al. (2008). The authors found out that for example creating secure shared keys for the first authentication caused problems for the users. Another usability issue, noted also by Stedman et al. (2008), is that everyone needs to use the same protocol to be able to identify other persons on the IM network.

2.3 Social Networks

Boyd & Ellison (2008) define social networks *"as web-based services that allow individuals to (1) construct a public or semi-public profile within a bounded system, (2) articulate a list of other users with whom they share a connection, and (3) view and traverse their list of connections and those made by others within the system."*

[11] Client-to-client authentication is defined in RFC 3923, but software support seems to be limited

[12] Borisov et al. (2004)

[13] Anonymous (n.d.)

Communities are formed inside social networks as users connect with other users and communicate with them. The largest Social Network Services (SNSs) at the time of writing are Facebook, Twitter, LinkedIn and MySpace. All these SNSs have also private messaging features so that users can send person-to-person messages in a way similar way to email or IM. (Boyd & Ellison, 2008)

As already noted in section 2.2, convergence between IM and social networks has begun. Facebook opened its chat service Facebook Chat to the outside world using XMPP on February 10, 2010 (Reiss, 2010). International Business Machines (IBM) Project Vulcan is another example of a plan to combine consumer social networks with enterprise social networking and real time communication and collaboration (Brill, 2010). It seems that person-to-person communication is becoming ubiquitous and converging into services that all have similar online and off-line communication and collaboration characteristics.

Regarding person-to-person authentication and identification, SNSs are centralized like the closed IM networks. So for a user to identify another user, he/she has to trust the company operating the SNS to do the actual authentication.

2.4 Voice and Video calls

The idea of voice communication over an IP network has existed at least from the introduction of the first text chat software *CB Simulator*. In 1995, a company called Vocaltec introduced the first Voice over IP (VoIP) product. Video communication followed soon after. (Varshney et al., 2002)

H.323[14] was the early leading standard on VoIP. It actually evolved from video telephony standards. H.323 was in close relation with Public switched telephone network (PSTN) standards, and they were interoperable from the beginning. SIP[15] is an Internet Engineering Task Force (IETF) standard that has gained popularity recently. The key difference between SIP and

[14] Recommendation H.323 (ITU-T, 1996, 2009)
[15] RFC 3261 (Rosenberg et al., 2002)

H.323 is that H.323 requires all connections to be routed by a call control service while SIP allows direct connections to applications and services on the Internet. (Goode, 2002)

Skype is probably the largest voice communication service on the Internet (Beckert, 2009). Skype has its own proprietary protocol that is based on Peer-to-Peer (P2P) technology (Baset & Schulzrinne, 2006). Many IM protocols, like those based on XMPP, allow also voice calls over the network.

VoIP solutions have the same issues as IM regarding person-to-person identification. Services are centralized like Skype and require trusting the company, or identification is based on public-key cryptography making it hard for normal persons to understand. Recognizing a familiar voice might help identifying the other person in personal communications but it is not a secure method for public services and businesses.

2.5 Google Wave

According to Google (2010a), *"Google Wave is an online tool for real-time communication and collaboration. A wave can be both a conversation and a document where people can discuss and work together using richly formatted text, photos, videos, maps, and more."*

Google Wave combines online and off-line participation in communication. Switching from online mode to off-line and vice versa is done seamlessly without any actions needed from an end-user. Communication can be IM style rapid back-and-forth sending of short messages between participants, it can be email like document exchange, or even collaborative editing of a shared document as in wikis.

Google Wave is not just an centralized service, but an open protocol and network. This makes it interesting in regard to person-to-person identification. In the current specifications, there is no identification method defined but, as with XMPP, the architecture of the protocol makes it possible to implement various identification solutions. The details of the network and protocol are described in section 3.2.

Chapter 3

Wave and XMPP

In this chapter, we go through and analyze the technical details of both the Wave protocols and their foundation, XMPP.

3.1 Extensible Messaging and Presence Protocol

RFC 3920 defines XMPP as *"an open Extensible Markup Language (XML) protocol for near-real-time messaging, presence, and request-response services."* XMPP is specified in multiple RFCs of which the core (RFC 3920), IM (RFC 3921) and end-to-end signing and encryption (RFC 3923) specifications are relevant in our context.

XMPP is a client-server architecture as shown in figure 3.1. An XMPP network consists of servers with connections to other servers. Each client is connected to one server, but every client can communicate with any other client connected to the same network via an interconnected server.

In general, XMPP is a protocol for streaming XML messages between clients. XMPP messages are called stanzas. There are three primitive message exchange patterns in messaging: request-response, publish-subscribe and fire-and-forget. The three basic stanzas are: `<presence/>`, `<iq/>` and `<message/>`.

The presence stanza is used for communicating presence information such as online/offline status. It can be generalized as a publish-subscribe method. An example presence stanza is shown in figure 3.2. Request-response com-

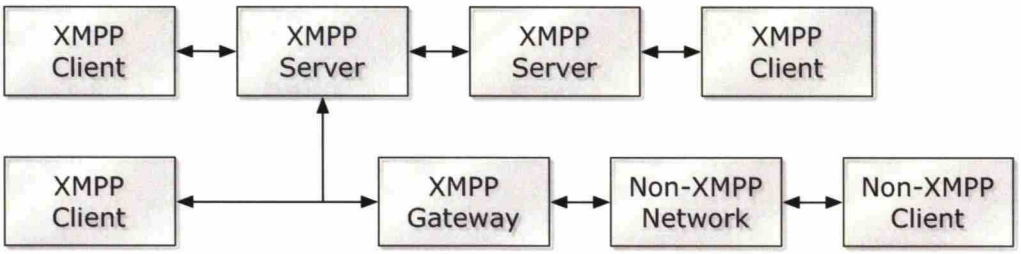

Figure 3.1: XMPP architecture (RFC 3920)

munication, like a client requesting and receiving its list of contacts, also known as roster, from a server, is done using the iq stanza. The basic communication, such as chat messages are transmitted using the message stanza. Messaging is a type of the fire-and-forget pattern.

```
<presence from='user@example.com'
    to='contact@example.org' type='subscribe'/>
<presence to='user@example.com' type='subscribed'/>
```

Figure 3.2: XMPP presence subscription (RFC 3921)

3.1.1 Addressing in XMPP

Addressing on the XMPP network is done using XMPP addresses that are called JabberIDs. An address consists of three parts: a username, a domain part and, finally, a resource identifier (see figure 3.1.1). A JabberID is written as *username@domain/resource* but since the resource part is optional in addressing, JabberIDs can resemble email addresses. Usually, the user is allowed to select the username part freely but the server has the final authority on deciding its value. Typically the username is fixed when creating a new user account to the server, but as XMPP can have anonymous connections (see 3.1.4), this option is included in the specification. The domain portion is determined from the user's server domain address. As a user is allowed to have multiple simultaneous connections to an XMPP server with the same username, a resource identifier is used to identify the separate connections uniquely. As with the username part, a client can propose

a resource identifier but the server makes the ultimate decision in determining the value. The resource identifier should be a random string that is hard to guess to prevent leakage of the user's presence status. An XMPP address with the resource identifier is called a full JabberID and an address without the identifier is called a bare JabberID.

$$\underbrace{john.doe}_{username}@\underbrace{example.com}_{domain}/\underbrace{mobile}_{resource}$$

Figure 3.3: XMPP address structure

The resource identifier of a JabberID is case-sensitive while the username and domain portions are case-insensitive, or more specifically case-folding. Case-folding is defined in RFC 3454. Basically, it specifies which characters are considered to be equal but in different cases, for example, the characters *A* and *a* are considered to be equal in case-insensitive contexts. XMPP user and domain names are not limited to the ASCII character set; rather, almost any Unicode character is allowed. This causes a security issue as many Unicode characters are visually indistinguishable from each other. Case-folding rules might add to the confusion. The fact that the server decides the domain part of the address makes it impossible for the users to fake their XMPP address in the same way as in email. (RFC 3920; RFC 3921)

3.1.2 Server-to-Server Connectivity

Server-to-server connectivity, also known as federation, in XMPP consists of server-to-server authentication and the actual messaging. Server-to-server authentication is discussed in section 3.1.3.

Messaging between XMPP servers consists of exchanging XML based message stanzas. All types of stanzas can be transmitted between servers. (RFC 3920)

3.1.3 Server-to-Server Authentication

In XMPP server-to-server authentication, two methods are used. The Simple Authentication and Security Layer (SASL) EXTERNAL[1] mechanism together with TLS is the secure method. TLS is used to exchange and verify the ITU-T Public-Key Infrastructure (X.509) certificates of both servers and, after verification, the SASL EXTERNAL mechanism is used to link the already verified and known certificates to authorization identities. (Saint-Andre & Millard, 2007)

The other method in XMPP server-to-server authentication is a server dialback protocol. Simplified, in the dialback method the originating server connects to the receiving server and sends a generated dialback key to the receiving server. Then the receiving server performs a Domain Name System (DNS) lookup to determine the IP address of the authoritative server in the originating server's domain. The receiving server then contacts the authoritative server and asks it to verify the dialback key. If the authoritative server verifies the key, the receiving server accepts the connection from the originating server. (RFC 3920)

The Server-to-Server authentication part of XMPP Federation can be classified according to XEP-0238 (Saint-Andre, 2008) into four levels of trust: Permissive, Verified, Encrypted and Trusted.

Permissive federation is not used anymore, as dialback is supported by servers. By definition, *verified federation* means weak verification of the identities of the servers using the dialback protocol. Adding TLS with self-signed certificates to the authentication process results in *encrypted federation*. The connection is encrypted, but the identities of servers are weakly verified. Using TLS, trusted certificates and SASL is known as *trusted federation*. The identities of servers are verified by certificates and the connection between servers is encrypted and protected against modification.

3.1.4 Client-to-Server Authentication

Client-to-server authentication is described in RFC 3920. Clients connect to the XMPP network through an XMPP server. Thus, the server is responsi-

[1] RFC 4422 (Melnikov & Zeilenga, 2006)

ble for the authentication of a client. In XMPP terminology, a connection between a client and a server is called a session. The session is an XML stream where the client can send XMPP stanzas to servers or other clients. Also, both the client and server can exchange XMPP stanzas between each other.

A client finds the server by resolving its IP address with a DNS query. Client-to-server authentication uses the SASL protocol defined by RFC 4422. As in server-to-server authentication, TLS with SASL EXTERNAL can be used in client-to-server authentication too. However, personal X.509 certificates have not gained popularity and this authentication method is rarely used. A username and password authenticator can be used with the SASL PLAIN[2], DIGEST-MD5[3] and SCRAM[4] methods. SASL Generic Security Services Application Program Interface (GSSAPI) mechanism enables authentication with Kerberos V[5]. Finally the SASL ANONYMOUS[6] method can be used for authentication without any credentials. This can be used in such scenarios where the typical usage is one-time only like, for example in customer service use cases where a company offers an XMPP based contact point for its customers. (RFC 4422)

3.1.5 Client-to-Client Authentication

Functionality to enable client-to-client authentication is defined in the XMPP RFC 3923 titled *"End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)"*. It specifies a PKI PKCS#7 based digital signature and encryption protocol for end-to-end messages. The protocol itself provides a secure method for client-to-client authentication, but client software support seems to be limited. To use it, both persons need S/MIME certificates that are issued by a CA trusted by the other party. As is explained in sections 4.3.3 and 4.3.4, both a web-of-trust and PKI are difficult concepts for the general population. This fact hinders the possibility of widespread adoption of this kind of a person-to-person authentication method.

[2]RFC 4616 (Zeilenga, 2006b)
[3]RFC 2831 Leach & Newman (2000)
[4]Newman et al. (2010)
[5]RFC 4121 (Zhu et al., 2005)
[6]RFC 4505 (Zeilenga, 2006a)

Two extensions, "XEP-0250: C2C Authentication Using TLS" (Meyer, 2008) and "XEP-0274: Design Considerations for Digital Signatures in XMPP" (Zeilenga, 2009) discuss also possible solutions for client-to-client authentication. Neither one is in general use. XEP-0250 defines a method for using TLS in end-to-end XML streams where authentication is done by either X.509 certificates, OpenPGP web-of-trust or shared secrets in the form of Secure Remote Passwords (SRPs)[7]. As with RFC 3923, the use of PKI or web-of-trust probably limits the acceptance of this method. On the other hand, SRP solves the person-to-person authentication only in such cases where the users know each other already by some other connection, as they need to somehow exchange the shared secret before using the method defined in XEP-0250. A shared secret cannot be used to prove one's identity to new peers as it does not contain any identification information itself. The identification has to be exchanged by some other method before using a shared secret to prove the identity.

XEP-0274 discusses digital signatures on a use case and requirement level. It does not specify what kind of digital signature format could be used. Again, the use of person-to-person client certificates creates obstacles to popularising XEP-0274 as a method for person-to-person authentication. As will be explained in section 4.3.4, there should be an easy to use method for creating and distributing the authentication certificates before this kind of scheme could gain momentum.

3.2 Wave

As noted before, Wave is an collaboration tool that combines both online and off-line messaging patterns. On technical level, Wave utilizes the XMPP protocol.

3.2.1 Wave Entities: Wave, Wavelet, Participant, Document

The Wave entities are defined in Bekmann et al. (2009). Discussions in the Wave network are called waves. Waves are XML documents that consist of sub-discussions called wavelets. Every wave has a list of participants. As

[7] RFC 5054 (Taylor et al., 2007)

with a wave, each wavelet can have its own list of participants. A wavelet can be marked as private, which means that only the participants in that specific wavelet have access to the contents of the wavelet. It must be noted that each participant's server has access to the user's wavelets, even private ones, as messages go through the server to the client. A wavelet contains documents that can be either text or data. Text documents contain human readable rich text messages. They are also known as *blips*. Data documents contain information mainly intended for machines. For example, tags are stored as data documents.

In short, a Wave is a collection of Wavelets, a wavelet is a collection of documents and documents are a collection of XML elements. The relationships of these wave entities are shown in figure 3.4.

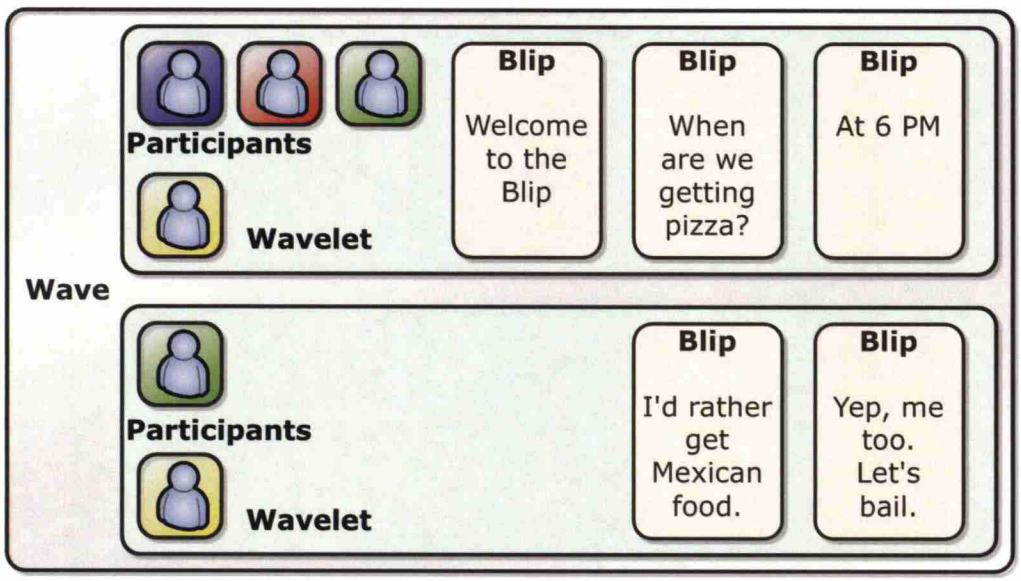

Figure 3.4: Wave entities. (Google, 2010b)

3.2.2 Wave Architecture

Google Wave uses the Google Wave Federation protocol (Baxter et al., 2009) for exchanging messages between servers. The Wave Federation protocol in turn uses XMPP as the messaging protocol. An XMPP extension, XEP-0114: Jabber Component Protocol (Saint-Andre, 2005) is used as the inte-

gration interface between a Wave server and a XMPP server. The architecture is shown in figure 3.5.

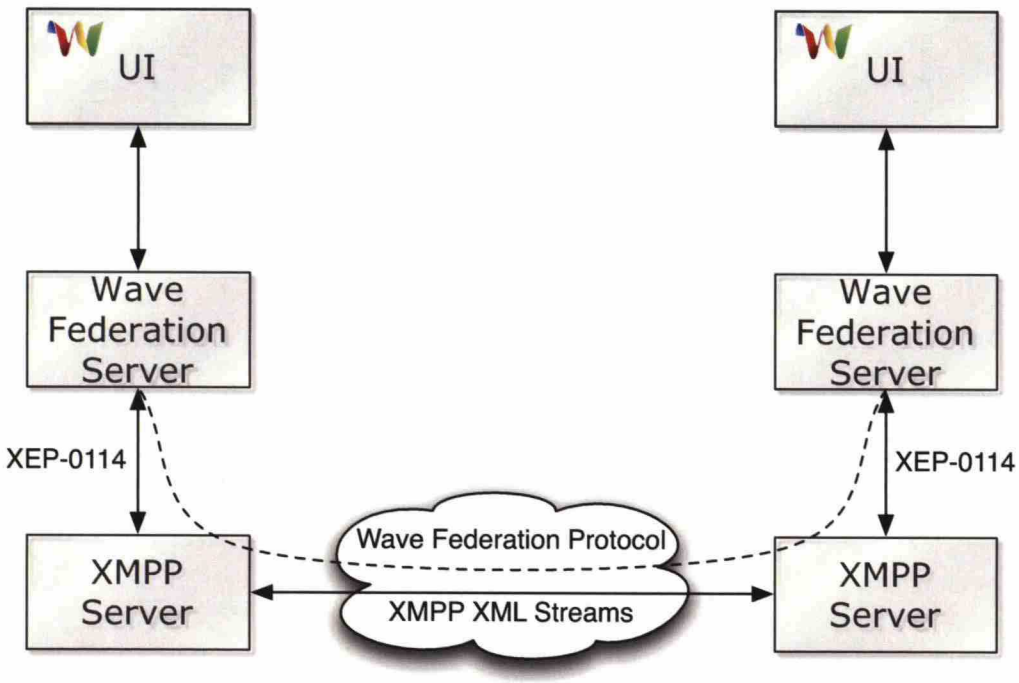

Figure 3.5: Wave architecture (based on Baxter et al. (2009))

All communication is done using the XMPP PubSub (Millard et al., 2008) extension protocol. Wavelet updates are sent using XMPP Message stanzas. All other communication is done by IQ stanzas.

All wavelet update operations are signed. Certificate chains are exchanged between servers before sending signed updates. Wavelet updates are acknowledged by the recipient servers using a method described in XEP-0184 (Saint-Andre & Hildebrand, 2007).

3.2.3 Wave Authentication

As Wave is built on top of XMPP, its server-to-server authentication features are inherited from XMPP. In XMPP, the use of TLS authentication is optional but, in the Wave protocol, it is mandatory. As a result, all servers

in a Wave network have to have X.509 certificates issued by trusted CAs. Servers communicating with each other need to have the other's CA certificate stored in order to authenticate the other server. Server A cannot exchange messages with server B if server B does not have at least the CA certificate of server A's certificate. This leads to the fact that, in order for a server to be able to send messages to all other XMPP servers on the same network, its certificate has to be issued by a CA that all other servers trust. Otherwise some messages cannot be delivered as connections cannot be authenticated and accepted. Connection encryption is recommended but optional in the Wave protocol, similar to XMPP. (Baxter et al., 2009)

Client-to-server authentication is left out of the Wave specifications. Specifially, it is stated by Tirsen (2009) that "Access from individuals to accounts and accounts to addresses is defined and enforced inside each wave provider and not specified in the standard."

3.2.4 Wave Extensions: Gadgets and Robots

Wave extensions (Google, 2010c) make it possible to add supplemental functionality to Wave discussions and Wave clients. Currently there are two types of extensions defined, Gadgets and Robots.

Gadgets are shared components running within Waves. The gadget state is shared between clients. In other words, all operations perfomed to a gadget by participants are transmitted to other participants in the same wave. A gadget's state is serialized within the wave it is attached to.

For example, an interactive map can be displayed as a gadget in a wave discussion. If a person scrolls or zooms the map, everyone else see the same actions performed on the gadget displayed in their own user interface.

Wave Robots are automated wave participants. Robots can perform similar actions as human participants. Robots are connected to the Wave network in the same way as humans, through a server. Robots communicate with a Wave server through a Robot Application Programming Interface (API). A robot can subscribe to various events happening in the Waves where the robot is one of the participants. The server makes HTTP requests to the Robots of the events. The requests contain information about the event and

about the wave. As robots participate in Waves through a Wave server, they can be placed in secured network locations so that they can have access to back end systems behind firewalls.

A robot can, for example, translate text in a wave from a language to another. It first reads text from a wavelet, sends it to a translation service and waits for the service to return the translation. Then, the robot can replace the original text in the wavelet with the translation. Participants see the operation in the same way as if a user had replaced the text with the translation.

3.2.5 Concurrency Control: Operational Transformation

Communication between Wave participants takes place by editing a wave. The edits to a wavelet are sent to other participants as transformations that consist of only the modifications to the wavelet. The transformation operations that can be described are based on an implementation of a theoretical framework known as Operational Transformation (OT). The Google Wave implementation of OT is defined in (Wang & Mah, 2009). It is a client-server model similar to Jupiter (Nichols et al., 1995).

Waves are replicated to every participant's server. The master copy, also known as the authoritative version of a Wave, is kept on the server where the corresponding wavelet has been initiated, i.e., on the original author's server. OT is applied on the Wavelet level. All transformations are sent first to the authoritative server, which then sends them further to other participants' servers.

Waves and wavelet modifications are replicated between servers with a protocol called General Verifiable Federation.

3.2.6 General Verifiable Federation Protocol

The core technology making possible real-time communication and collaboration between users with multiple servers on the wave network is a protocol named General Verifiable Federation (Kissner & Laurie, 2009). Google Wave uses the General Verifiable Federation protocol to distribute modifications made to Waves to all servers having users participating in them.

Six key properties are guaranteed by the protocol. They are described in detail by Kissner & Laurie (2009). In general the protocol ensures that every message can be traced back to the originating server and that the global order of messages is eventually the same between servers. Ordering might be inconsistent at times as the messages reach servers in a different sequence. The protocol does not guarantee that all servers get all messages from all participants. If messages are lost, servers split up into separate groups. The split might not be noticeable by the participants, except if merging of the split groups is attempted. It is suggested that a split mode can be detected by participants communicating directly with each other but possible consequences of splits are not discussed.

3.2.7 Person-to-Person Identification in Wave

As Wave uses XMPP with TLS certificates, a rogue Wave server cannot impersonate other servers in the Wave network, except if there is a trusted CA that has issued malicious certificates. This combined with XMPP addressing details mentioned in section 3.1.1 means that the domain parts of XMPP addresses in messages cannot be forged. Thus, we can assume that users can trust domain parts of addresses on the Wave network.

General verifiable federation (section 3.2.6) ensures that all messages sent over the Wave network can be traced back to the originating server. So, users can also determine in a secure way the server(s) from where modifications to the waves have been made.

As in XMPP, the username part of Wave addresses is determined by the users server. There is no way of ensuring that a server is authenticating its users correctly or that it is mapping the user identities to Wave usernames in a constant way. As Tirsen (2009) puts it: *"Access from individuals to accounts and accounts to addresses is defined and enforced inside each wave provider and not specified in the standard."*

This leads us to the conclusion that there is no direct method for person-to-person authentication and thus no way for persons to determine each other's identities on the Wave network.

Chapter 4

Authentication and Identification

This chapter discusses user authentication, which is also known as identification. The NIST Handbook of An Introduction to Computer Security (Guttman et al., 1995) defines both identification and authentication: *"Identification is the means by which a user provides a claimed identity to the system." "[User] authentication is the means of establishing the validity of this claim."* In other words, a user *authenticates* to a system so that the system can verify the users identity by *identifying* the user.

Both entity and message authentication are defined by the International Organization for Standardization (1999, 1997). Entity authentication is identification. Entity here is for example a person and the goal of identification is to allow the persons identity to be confirmed. The identification happens usually in real time in contrast to message authentication. It is the other form of authentication. Message authentication is verifying the immutability etc. of a data document. It can be done any time after the document is signed.

There are the three means of authenticating a user's identity. Proving that an entity has or knows an identifier is known as an authenticator (O'Gorman, 2003). The authenticators are 1) something the user knows, 2) something the user possesses and 3) something the user is (Guttman et al., 1995; Burr et al., 2006; O'Gorman, 2003). Implementations of these authenticators are described in section 4.1.3.

Authentication is not authorization. Decisions on what an identity is allowed to do or access is authorization.(Burr et al., 2006) After a successful authentication, it is feasible to determine what the authenticated user is allowed to do or access, i.e., does the user have *authorization* to perform the task he/she is trying to do.

4.1 Authentication Concepts

Authentication involves many concepts and details. In this section we go through the most important ones and discuss their importance regarding authentication.

4.1.1 Identifiers, Identities and Entities

A person or an organization is only one. There exists only one *entity* for each person or organization. An *entity* can have multiple *identities* but each *identity* maps to exactly one *entity*. Identities can be considered unique so that no two entities can have same identity. However, an entity can have multiple different identities in different contexts. Each identity has attributes that define the identity. The attributes are known as *identifiers*. Multiple identities can have same kind of identifiers, but the combination of identifiers maps to one identity or more exactly, via identities to one distinct entity. The relationship between identifiers, identities and entities is shown in figure 4.1. (Jøsang et al., 2005)

X.500 defines the concept of a Distinguished Name (DN), which is supposed to be the only name for an entity. In practice, entities have many X.500 names.

4.1.2 Persons and Digital Identifiers

To identify a person, he/she has to have some kind of a unique identifier. There are not many globally unique identifiers, thus many times locally unique, meaning inside one country, identifiers are used.

Names are not unique even locally, nor even DNA as identical twins have the same DNA. Some global identifiers are for example fingerprints and

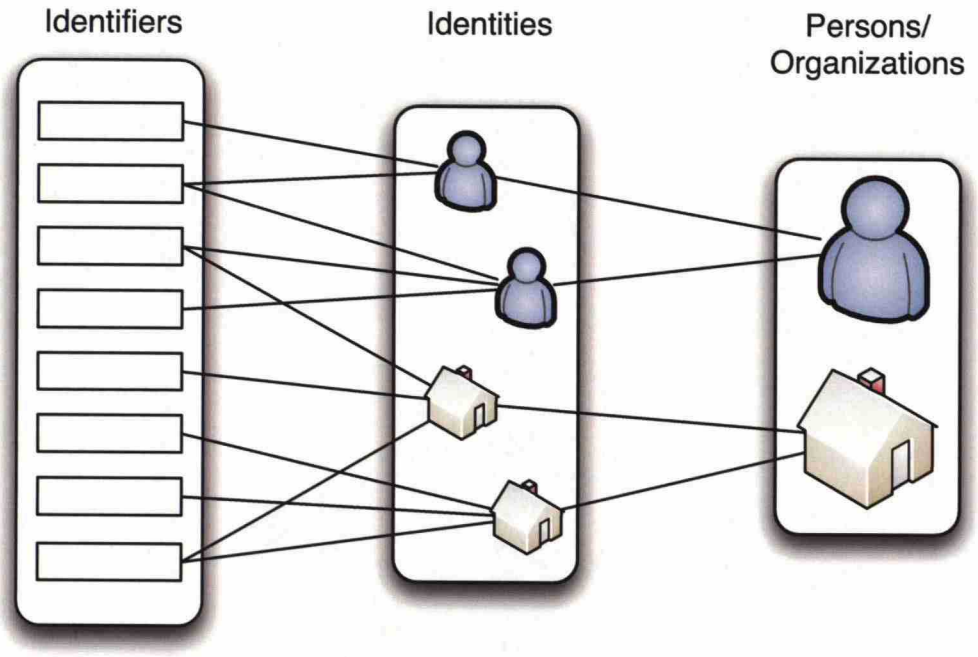

Figure 4.1: Relationship between identifiers, identities and entities (from Jøsang et al. (2005)).

retina scans, but they are difficult to use as the primary identifier because they are analogue and thus standardized ways are needed to store them in digital format.

Otjacques et al. (2007) discuss the identification issues in detail. A common method for identifying persons is to use synthetically generated numbers like national identification numbers. A unique identity given by a government to a person is called a national identification number. The number identifies a person uniquely inside the country and creates a identity for the person. Governmental organizations use the identification numbers for keeping records of people for governmental purposes, like taxation and health care. In general level, national identification numbers are used to transport identification information between systems as the number identifies the person in a unambiguous way. An example of a national identity number is the Finnish Personal Identity Code (Population Register Centre, n.d.).

In some countries, people do not have government-given identifiers. In these countries it is difficult to link a person's identifiers to his/her identity, because no unique identity number cannot be used as the person's identity.

A person might also have multiple national identification numbers for example by having citizenship in two countries. In these cases, it is logical to use the identification numbers according to the country where they were issued and, if needed, link the identification numbers to each other. It must be noted that, if a person does not reveal that he/she has multiple national identification numbers, the person can appear as multiple separate persons in a global system depending on unique national identification numbers as no global registry exists containing information of these dual citizenships.

4.1.3 Authenticators

An authenticator is the tool or instrument used to prove an identity in an authentication event. As described in the beginning of this chapter, authenticators can be divided into three major types[1].

In general, an authenticator consists of credentials linking one or more identity attributes to secret or unique tokens.

[1]They were: What the user knows, possesses or is

One of the simplest authenticators is a username and a password combination. The username is an identity attribute and the password the token. In PKI, the public key certificate, containing identity attributes, is the credential and the private key the token.

One-time passwords (OTPs) are collections of passwords that can be used only once each. The list of passwords can be stored either as a physical list or inside a physical device that displays the correct password when it is needed. An example of an OTP authenticator is S/KEY (RFC 1760). The Finnish Banks' Tupas two-factor authentication system has an OTP authenticator as the second factor method.

In biometric authenticators, the token is some form of biometric data. Biometric data can be pictures of the persons face, fingerprints, DNA, iris and retina scans (Burr et al., 2006). A biometric token can be used in authentication by digitizing it using a known function taking biometric data as input and outputting a digital representation of it (O'Gorman, 2003). This digital representation of a token is then stored to a device or by a Credential Service Provider (CSP). Thus, the CSP is the credential linking a user's biometric data to his/her identity. Examples of biometric authenticators are passports conforming to the ICAO Document 9303 requirements (International Civil Aviation Organization, 2006). A photograph or fingerprints can be stored to a Radio-frequency identification (RFID) readable chip embedded in the passport.

A new authenticator type: Somebody you know

Recently, probably due to the rise of social networking phenomena, there has been research on a possible fourth type of an authenticator. In general, the identity of a user is formed from unique set of his/her name and the people he/she knows.

For example, Brainard et al. (2006) describe a new type that they have given the name *Somebody you know*. As the name implies, the authenticator is based on a user's social network. When a user (Alice) is authenticating, one authentication step is made to another human (Bob), to someone Alice knows. Bob then determines if it is really Alice or not authenticating by for example her voice, or by asking specific questions from Alice. If Bob is

certain that Alice is the user authenticating, he then proceeds to relay this information to the system where Alice is ultimately trying to authenticate. Bob then receives a token that he gives to Alice, so that Alice can prove the successful authentication event with Bob to the system.

This kind of a *fourth factor* authenticator is useful for example in situations where a user has forgotten one of his or her authentication tokens. See figure 4.2 for an example scenario where the fourth factor authenticator could be used.(Brainard et al., 2006)

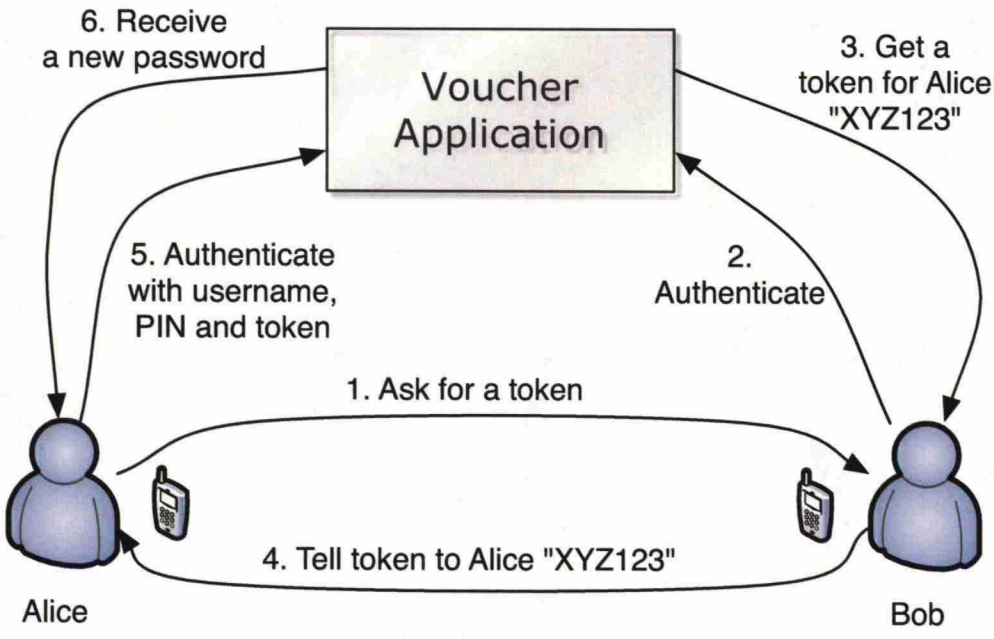

Figure 4.2: Fourth Factor Authentication (based on Brainard et al. (2006) page 172). Alice has forgotten her password, so she authenticates to Bob, who then provides Alice with a token proving the successful authentication.

4.1.4 Authentication Tokens

An authentication token is a thing that can be used to prove ownership of one or more authenticators. Burr et al. (2006) describe four types of tokens: hard, soft, one-time password device, and password tokens.

Hard tokens are hardware devices that contain cryptographic keys. The keys can be used only together with a correct password or a biometric authenticator. Possession of the device is proved by activating the key so that the key can then be used in the authentication event.

Soft tokens are like hard tokens, except that tokens are stored on some kind of media instead of a hardware device. Similarly to hard tokens, the key is activated with a passphrase.

One-time password device tokens have a symmetric key stored on the device. The key is known to both the device and the verifier. With this shared key, the device can generate one-time passwords based on for example the current time. These passwords can then be verified using the same algorithm by the verifier.

Password tokens are passwords. They are usually used together with usernames for proving knowledge of a basic authenticator.

4.1.5 Credentials

Credentials are physical or electronic documents that bind identities to tokens. Physical credentials, like passports, id cards etc. are used to link a set of biometric information to the actual identity of a subject. The use of physical credentials is a two step operation. Firstly, the credential itself is authenticated by physical examination to ensure that it is authentic and has not been tampered with. Secondly the biometric data contained in the credential is compared to the physical characteristics of the person presenting the credential. If a match between these can be reliably made, the authentication is successful. (Burr et al., 2006)

Electronic credentials bind identification information to X.509 public keys for example. Identification information is something that identifies the subject, like the subject's name or a national identity number. (Burr et al., 2006)

4.1.6 Multi-factor authentication

When at least two different authenticator types are combined, it is called multi-factor authentication. All of the authenticators have to be completed successfully in order for the whole authentication event to be deemed valid.

Multi-factor authentication enhances the reliability of the authentication and mitigates the potential risks of misuse of lost authenticators.

There are drawbacks too, as the authentication event gets more complicated when many authenticators are used. Each authenticator has its own way of using it so a person might have to encounter many different user interfaces in a multi-factor authentication event.

Table 4.1 presents Multi-factor authentication combinations and their advantages and drawbacks. (O'Gorman, 2003)

Table 4.1: Security advantages and convenience drawbacks of multi-factor authentication. In the table knowledge-based authenticators are "what you know" authenticators, object-based authenticators "what you have", and ID-based authenticators "who you are". From O'Gorman (2003), page 2024.

Authenticator Combination	**Security Advantage**	**Convenience Drawback**	**Example**
Knowledge- and Object Based	Lost/Stolen token protected by password	Must carry token and memorize password	PIN-enabled bank card
Object- and ID-Based	Lost/Stolen token protected by ID	Must carry token, but not ID if it is a biometric	Photo-ID
Knowledge- and ID-Based	Two factors provide security in case either compromised	Have to memorize password and have ID	Password and biometric for computer access
Knowledge-, Object- and ID-Based	A third factor to provide security in case two other factors are compromised	Have to memorize password, carry token and have ID	Military applications requiring photo-ID checked by guard, plus password

An example of a multi-factor authentication system is the Finnish banks' Tupas certification service. The two authenticators in Tupas are a username and password combination and a list of one-time-passwords. (Federation of Finnish Financial Services, 2008) Burr et al. (2006) considers multi-factor authentication to be level three in a four level classification scheme. More discussion about the levels of authentication are discussed in section 4.1.9.

Mobile authentication, ETSI Mobile Signature Service (ETSI-MSS), (ETSI, 2003) is another example of multi-factor authentication. The two factors of

authentication are a Personal identification number (PIN) code and a PKI private key stored securely on the Subscriber Identity Module (SIM) card.

Multi-factor authentication is closely related to strong authentication which is discussed next.

4.1.7 Strong Authentication

Authentication is divided into weak and strong authentication depending on the level of reliability of the authentication to provide a true identification result.

Menezes et al. (2001) classify username/password-based authenticators as weak authenticators because passwords as authenticators have many inherent weaknesses and possible attacks against them. Menezes et al. (2001) call passwords time-invariant authenticators and count PINs as such also. Passwords tend to be changed infrequently so the same password can be used as an authenticator for a long time. Passwords can be searched for by guessing, trying all words and word pairs, triplets etc. from dictionaries or even by exhaustively going through all possible combinations of valid password characters to a certain length.

Strong authentication is defined by Menezes et al. (2001) as using challenge-response authenticators with time-variant parameters. Non-repeating values and unique numbers in every authentication event are considered time-variant in distinction to time-invariant passwords. A key point in strong authentication, according to Menezes et al. (2001), is that the actual secret[2] is never revealed to the verifier.

In some contexts strong authentication is considered to be multi-factor authentication described in the previous section. For example the Finnish law on strong electronic authentication and digital signatures (Finnish Acts of Parliament, 2009) rules that at least two different types of authenticators have to be used in order to consider the authentication strong.

We suggest strong authentication to be the latter as proper multi-factor authentications include a Menezes et al. (2001) defined strong authenticator as at least one of the used authenticators.

[2]For example the secret key in a PKI scheme

4.1.8 Out-of-Band Authentication

An authentication event, where the authenticator data[3] is transmitted via separate channels or networks, is called out-of-band authentication. Wu et al. (2004) present a simple Short Message Service (SMS) based out-of-band authentication method. First a user supplies his/her username to a web site. Then the web site sends a unique session name to the user via both the Internet and mobile networks. The user then verifies that the session name is the same through both channels, to make sure that for example no man-in-the-middle attack is attempted at the same time.

An out-of-band authenticator can be used in a multi-factor authentication scheme. For example, as shown in figure 4.3, if a user authenticates himself/herself to a web page using a username and password combination, the second authentication could be accomplished by sending an SMS challenge and requiring the user to respond to it by replying to the SMS (Jøsang et al., 2007). This way, one authentication is done over internet and the other over a mobile network. Security is increased as an attacker would have to have access to both networks in order to implement for example a man-in-the-middle attack.

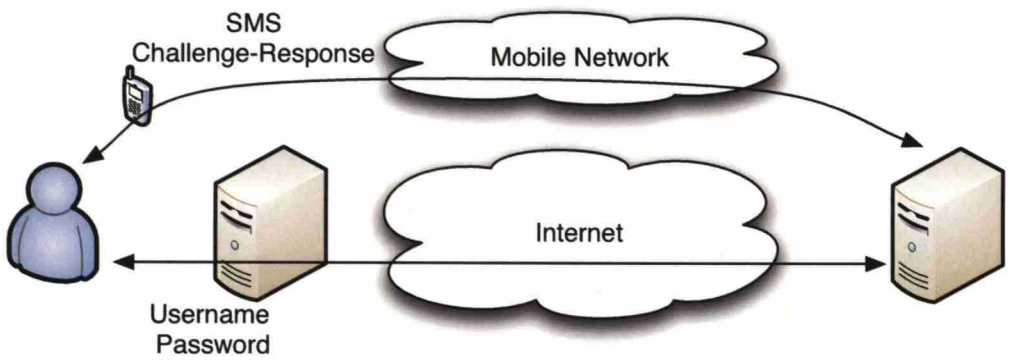

Figure 4.3: Two-factor authentication with SMS challenge-response as an out-of-band authenticator (adopted from Jøsang et al. (2007) page 148 and Wu et al. (2004)).

[3]For example a username and password combination

4.1.9 Levels of Authentication

One way of measuring the quality requirements of authentication is defined by National Institute of Standards and Technology (NIST) (Burr et al., 2006). Authentication quality requirements are divided into four levels, from lowest Level 1 to highest Level 4, based on consequences what happens if there is an error in the authentication or someone misuses authentication credentials.

Level 1, the most basic level, has minimal requirements. The authentication method needs to provide some assurance that the person is who he/she claims to be, but no identity proofing is required. Plaintext passwords and secrets are disallowed on all levels but, on level 1, simple challenge-response protocols are allowed.

Level 2 requires the use of a single-factor authentication method. The method has to prove the person's identity somehow. Passwords and PINs can be used as authentication tokens. The used protocol has to prevent three types of attacs that are eavesdropping, replay and online-guessing.

On level 3, multi-factor authentication is required. Verification of both authentication material and information has to be proven by the user. Passwords and PINs are not allowed as authentication tokens, but soft cryptographic tokens, hard cryptographic tokens and one-time password device tokens need to be used. The tokens have to be protected by a password or by a biometric that is used to unlock the actual token when using it for authentication. In addition to the three attacks on level 2, the protocols need to prevent also verifier impersonation and man-in-the-middle attacks.

Level 4 is similar to level 3, but only hard cryptographic tokens that cannot be copied are allowed.

4.1.10 Types of Authentication

Authentication can also be classified according to the entities participating in the authentication process into user-to-user authentication, user-to-machine, machine-to-user and machine-to-machine authentication (O'Gorman, 2003).

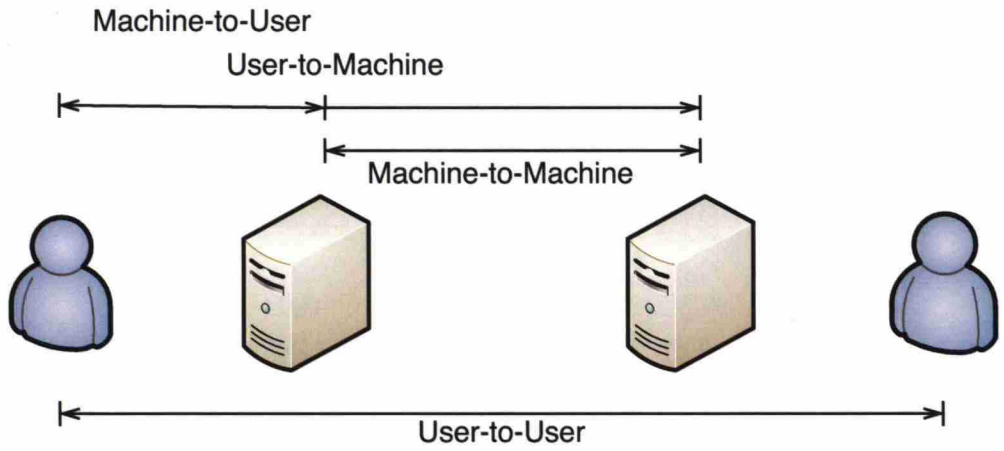

Figure 4.4: Different types of authentication (adopted from O'Gorman (2003) page 2021).

As figure 4.4 shows, user-to-user, also known as person-to-person, authentication happens between two users. Typical authentication events take place in real life situations in which the person authenticating displays some form of a physical identification token and the verifier then uses the identification token to make the actual identification. Passport control on border stations is an example of this kind of authentication. On the net, user-to-user authentication events occur for example in email conversation using S/MIME or other PKI based signatures. A notable difference in user-to-user authentication in relation to other authentication types is that a human verifying the event ultimately makes the decision whether the identification has been made or not. A human can even decide to only partly trust the authentication credentials. On a philosophical level, it can be said that an absolute decision is never made; rather, the verifier establishes a level of trust in the identification he/she has made. User-to-user authentication is discussed in detail in section 4.1.11.

User-to-machine authentication is ubiquitous on the Internet. Most network resources, like email servers and discussion forums require successful authentication before allowing access. In the real world, ATM withdrawals and automated border control checks using machine readable passports are examples of user-to-machine authentication events. Typically, users prove

their identification using authenticators (see section 4.1.3). A concept called trusted path is especially relevant in the user-to-machine authentication context. Trusted path is discussed in section 4.4.5.

Machine–to-user authentication is the opposite to user-to-machine authentication. TLS certificates proving identity of websites to users are examples of machine-to-user authentication.

Machine-to-machine authentication is also common on the net. As described in section 3.1.3, XMPP servers authenticate each other when establishing a link between them. In machine-to-machine authentication, the authenticators have to be stored on the machines because the actual authentication is occuring without user intervention.

4.1.11 Mutual Authentication

Mutual authentication means two-way authentication where both parties in an authentication event obtain reliable identification of the other party. In addition to the user authenticating to a machine, mutual authentication can be achieved by the user also verifying the identity of the machine. For example a user can identify a web server by validating the web server certificate that is issued by a CA the user trusts. Gajek et al. (2008) describe one method for an advanced certificate based authentication. Mutual authentication is also known as two-way authentication. (Otway & Rees, 1987)

Mutual authentication can also be extended to include person-to-person identification if the goal is for both persons to authenticate each other. For example, if both persons trust each others public key certifcate, they can authenticate each other. S/MIME in email is a method for mutual authentication. The difficulty in person-to-person authentication is that mutual authentication usually needs infrastructure both individuals can trust and use easily. In other words, some kind of a Trusted Third Party (TTP) architecture is required for user-to-user identification.

Person-to-person authentication can be achieved also by having a process for a user to initiate the identity verification of the other user.

4.2 Authentication Model

A basic authentication model is described in Burr et al. (2006). An authentication model or process consists of three parts. First a person is identified by a Registration Authority (RA). Then the person is issued a token with a credential to prove his/her identity to a Credential Service Provider (CSP). Lastly the person uses his/her credentials to authenticate to someone by using the CSP.

4.2.1 Identity Proofing

The first step, establishing the identity, is called identity proofing. The identification is done using existing authenticators. For example, a passport or another trusted document can be used. The probability of a successful identification is dependent on the quality of the authenticators used. This first authentication is very important as the RA must be absolutely sure of a correct identification; otherwise an authenticator will be given to the wrong person. (Burr et al., 2006)

It can be argued that the quality of a new authenticator is, at maximum, the same as the quality of the best original authenticator used. Using multiple authenticators does not increse the quality over the best original authenticator.

4.2.2 Issuing Authentication Tokens

When a positive identification has been achieved, a CSP can issue or register a token together with a credential to the person. The credential links the token to the identity attributes determined by the RA. (Burr et al., 2006)

4.2.3 Authentication Using Tokens

After the person has a token, he/she can use it to prove his/her identity to someone by authenticating to the CSP. The CSP then relays the result to the recipient, also known as the verifier end. The role of a CSP is critical in secure authentication. A CSP can be inside the service where the person is

authenticating or it can be an external entity. When a username and password combination is used as the credentials, it is common that no external service is needed. If the CSP is an external entity, it has to be trusted by both the person authenticating to it and by the other person or service wanting to verify the identity of the person authenticating. Thus a CSP can also be called a Trusted Third Party (TTP) (see section 4.3.2). (Burr et al., 2006)

4.3 Authentication Architectures

A good introduction to authentication architectures and comparison between them can be found from Blaze et al. (1996).

4.3.1 Public Key Cryptography

Public key cryptography is cryptography where asymmetric keys are used. The keys are divided into public and secret private keys. Public key cryptography is an fundamental technology in authentication and identification. Diffie & Hellman (1976) published the first assymmetric-key cryptosystem and Rivest, Shamir, & Adleman (1978) the first public key encryption and signature algorithm later named RSA.[4]

Public key cryptography is used for digital signatures to prove immutability and authenticity of a message. In digital signatures, messages are signed with private keys and verified using public keys. When a message has been encrypted with a public key, it can be decrypted only with the corresponding private key, thus ensuring confidentiality of the message.

For example X.509, PGP and TLS are based on public key cryptography.

4.3.2 Trusted Third Party

Trusted Third Party (TTP) is an organization in which persons and other organizations trust. (Blaze et al., 1996) A good example of TTP is a X.509

[4]Actually, in 1969 James Ellis from the British Government Communication Headquarters (GCHQ) discovered the same idea as Diffie & Hellman (1976) and in 1973 his colleague Clifford Cocks found out a similar public key cryptographic algorithm as Rivest, Shamir, & Adleman (1978), but neither was allowed to publish the findings. (Jankvist, 2008)

CA as (ITU-T, 2005) defines it as *"An authority trusted by one or more users to create and assign public-key certificates."*

Jefferies et al. (1995) define a set of requirements that are needed from a TTP architecture. In short, the architecture should provide visible benefits for the user, it should allow international operation, the architecture should be public and based on well known techniques and it should support variety of encryption algorithms, both in hardware and software. In addition, it should not be bound to any specific electronic communication channel, officials should be allowed to access the messages passed through the TTP but not fabricate any messages, and any kind of abuse should be detectable by the other parties.

4.3.3 Web of Trust

Web-of-Trust was first introduced in PGP. There is no central authority that everybody trusts. In other words, there is no TTP. Instead persons display their trust in other people by signing the public keys of these people with their own private key. The signatures then form links between public keys and thus form a web of trust between individual keys. For example, if Alice trusts Bob, she signs Bob's public key. Bob then sends his key to Charlie, who happens to know Alice. By checking that Alice has signed Bob's key, Charlie, if he trusts Alice, can trust Bob's key via Alice. Charlie could then sign Bob's key and enlarge the web. (Abdul-Rahman, 1997)

The web of trust model has been criticized of not having clear trust relationships. For example, there might be an unbreakable chain of ten signed public keys, but there is no guarantee that any one of the ten links has been verified properly by the signer. The person evaluating the trust chain also probably does not know personally all of the signers. Also, if the recipient does not have a trust path to the sender, the recipient does not have any method for verifying the sender's key, except by contacting him/her directly. This is not a reliable mechanism for commerce. (Blaze et al., 1996)

The opposite to a web of trust is a coordinated trust relationship architecture, which is known as PKI.

4.3.4 Public Key Infrastructure (PKI)

PKI consists of a PKI policy[5], Certification Authority (CA), Registration Authority (RA), a certificate repository and a distribution system and applications supporting the PKI. (Hunt, 2001)

The policy describes specifications for example on how the CA operates, how certificates are issued and revoked and how keys are stored. The CA issues and revokes certificates binding identities to public keys. A RA is the interface between users and the CA. It authenticates the users and provides the identities to the CA. The roles of a CA and RA are often combined into one organizational unit. The certificate repository provides access to the public keys signed by the CA. (Hunt, 2001)

X.509 certificate infrastucture form a well known PKI. Websites use certificates with TLS to prove their identity. The certificates are issued by CAs who first identify the website owner and then sign the website certificate with their CA key. Web browsers have lists of trusted root CA certificates that are then used when verifying the website's certificate chain. The trust in the root certificates is established by using third party audits of the PKI policies and by verifying that the policies conform to expected set of requirements of CAs.

PKI has received criticism too. Especially trusting CAs is seen problematic, as PKI models offer no easy way to authenticate the identity of a CA. Also the complexity of understanding PKI concepts is seen as too difficult for the general population. For example, users are taught to trust trust chain verification made by browsers, but the lists of CA certificates in web browsers have been chosen by the browser manufacturers, not by the users. The lack of proper identification when issuing certificates has raised questions. It is possible to get a certificate trusted by major web browsers with only an email address verification [6]. (Ellison & Schneier, 2000)

[5] An example PKI policy can be found from `http://www.apple.com/certificateauthority/`

[6] `http://www.startssl.com`

4.3.5 Three Domain Model

In the 1990s, SETco developed a set of security protocols called Secure Electronic Transaction (SET) (SetCo, 1997) for securing credit card transactions over insecure networks like the Internet, where the card is not present at the location of the actual transaction.

In SET, the merchant asks for the users name, address, credit card number and a Card Verification Value (CVV)[7] code. They are then verified by the credit card company. In 3-D Secure, an additional step is added to the online payment transaction (Visa, 2006). The user is shown a form inside an IFrame HyperText Markup Language (HTML) page of his/her bank where the user has to authenticate to his/her bank before the credit card transaction is accepted.

As the name implies, 3-D Secure is based on a three-domain model: Acquirer Domain, Issuer Domain and the Interoperability Domain. The acquirer represents the merchant or the recipient of the money, the issuer the user's bank paying the money, and the interoperability domain the credit card company whose card is used in the transaction. A diagram of the architecture is show in figure 4.5. Implementations of 3-D Secure are known as Verified by Visa and MasterCard SecureCode. (Visa, 2006)

The technological design and implementation of the 3-D Secure has been critized widely (Meunier, 2007; Internet Retailer, 2005; Murdoch & Anderson, 2010).

Murdoch & Anderson (2010) describe a number of security weaknesses. Security cues are hidden because 3-D Secure is implemented by embedding the payment processors page inside an online shopping page using an IFrame. Browsers will not display any additional information about the connection security and certificates in the IFrame. Activation of 3-D Secure is also problematic. A password has to be registered in order to use 3-D Secure. The password can be activated during the shopping process. The user is authenticated using weak authenticators by asking for birth dates, for example. This teaches the user bad habits, as online shopping pages should not be asking such personal details. According to Murdoch & An-

[7] To be precise, CVV2, as there are two CVVs, CVV1 which is encoded on the magnetic strip and CVV2 which is only printed on the card.

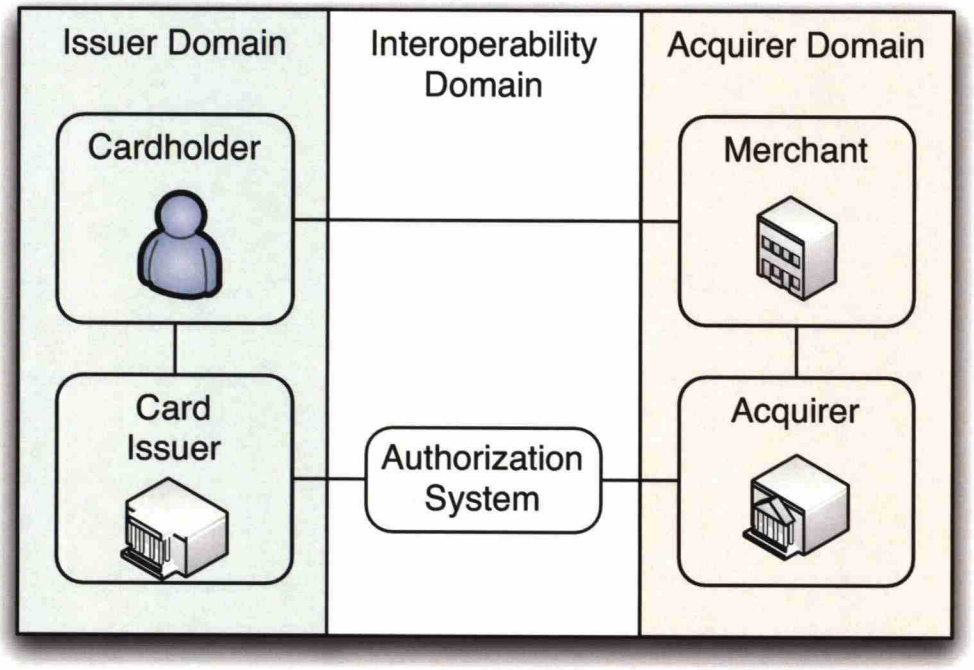

Figure 4.5: 3-D Secure Architecture and the three domains (based on Visa (2006)

derson (2010) this has already been used in phishing scams. Authentication method inconsistency is also noted as a weakness. In the specification, there is no definition of what a 3-D Secure authentication should look like; thus the actual implementations vary. The user has no way of verifying by using visual clues whether a site is real or not. This also has been used in phishing scams.

4.3.6 Federated Identity Management

Federated Identity Management is a model where a user's identity is stored in multiple identity management systems that trust each other on some level. The systems which provide user identity information are called Identity Providers (IdPs).

The federated identity domain model can be further divided into submodels depending on how many IdPs there are and how the trust relationships are arranged. Two notable submodels are the Centralized SSO Identity Domain Model and the Federated SSO Identity Model. In the Centralized SSO Identity Domain Model, there is only one IdP that every Service Provider (SP) uses as the actual identity source. Figure 4.6 presents a simple example of this model. In the Federated SSO Identity Model, SPs share identities in so called circles of trust. (Madsen et al., 2005)

A Centralized SSO Identity Domain Model is suitable for providing strong identification data needed in the first authentication when authenticators are created for the user. The Finnish Tupas (Federation of Finnish Financial Services, 2008) service can be seen as a sample model having the described architecture.

The Federated SSO Identity Domain Model is suitable for sharing identities between SPs having relatively equal level of trust. OpenID is an example of this kind of model. (Recordon & Reed, 2006)

4.4 Authentication Threats

Good authenticators together with secure authentication protocols mitigate threats that someone can fake his/her identification by misusing authen-

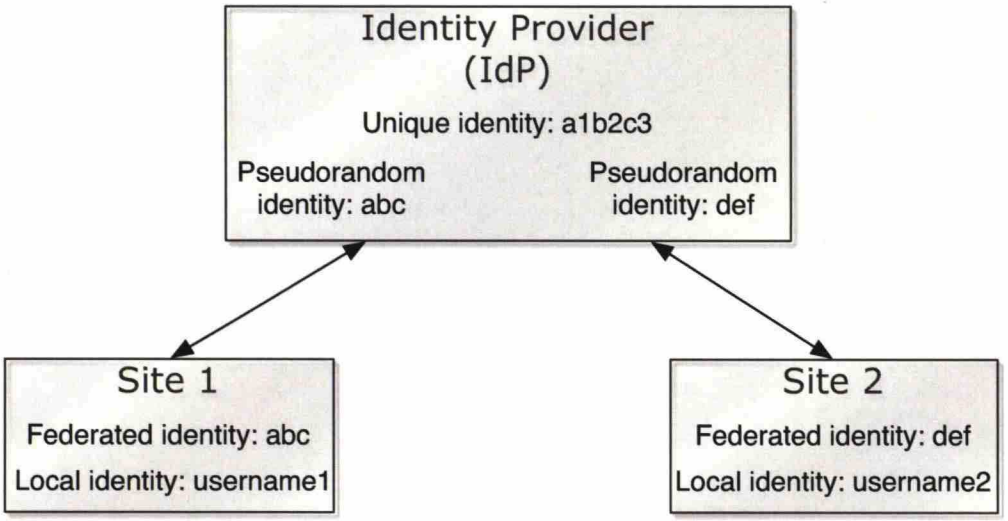

Figure 4.6: Identity Federation (based on Madsen et al. (2005) page 79).

ticators. Quality of an authenticator can also be measured by how easy various attacks are to execute in the authentication context.

There are several types of attacks and threats. Burr et al. (2006) divide threats into four categories: token threats, authentication protocol threats, registration threats and other threats.

4.4.1 Token Threats

Token threats, as the name implies, affect the trustworthiness of the token as a valid authenticator. Burr et al. (2006) classify attacks according to which type of the authenticator[8] they concern.

Something you know, for example passwords and PINs, can be disclosed in many ways. A person might be forced to reveal his/her password[9], or an attacker might guess a weak password. Something you have, as in a OTP list or a physical token, might be stolen or cloned. Something you are can also be replicated. For example, fingerprints can be cloned with some photo copier powder, copper plate and gelatin as described by Sten et al. (2003).

[8] something you know, something you have and something you are
[9] The legendary rubber-hose attack

According to Burr et al. (2006) token threats can be mitigated by multi-factor authentication, physical security mechanisms, complex passwords and system and network security controls.

4.4.2 Registration Threats

During the registration phase of establishing an authenticator (see section 4.2) there are at least two possible attacks: impersonation of a claimed identity and repuditation of registration.

An attack where the person registering for a authenticator presents false credentials or other misleading identification attributes and so claims an incorrect identity is called a impersonation of a claimed identity.

Repudiation of registration means a situation where a person does not admit that he/she has at some point registered a specific token to him/herself.

Risk of registration threats can be reduced by increasing the quality of identification sources and authenticators used during the registration phase. (Burr et al., 2006)

4.4.3 Authentication Protocol Threats

Authentication protocol threats are threats that result from imperfections in the protocol design. Burr et al. (2006) lists three major types of protocol-caused threats.

Eavesdropping means recording an authentication protocol transaction for later analysis. The protocol messages are then analysed with the goal of obtaining valid authentication tokens.

Masquerading attacks are based on acting as a user wanting to authenticate to a CSP or acting as a CSP to a user authenticating. Posing as a CSP might trick a user to give his/her credentials and tokens to the imposter. Acting as a user authenticating to a CSP enables the attacker to, for example, test the validity of guessed authenticator tokens.

When an authenticated session has been established, that is, a user has authenticated successfully and a CSP has verified the validity of tokens, an attacker can try to hijack the session. This way, an attacker could possi-

bly learn sensitive information exchanged in the authenticated channel or present invalid information to either party as being from the other party.

4.4.4 Other Threats

There are also various other threats that do not fit into the above categories. Burr et al. (2006) describe the following threats.

In a malicious code attack, evil code is injected to the users computer. The aim of the code is to gain knowledge of an authentication token by, for example, sending it to the attacker over the Internet. Details and ways to prevent malicious code attacks are discussed in section 4.4.5.

Trying to get valid authentication credentials or tokens by intruding or cracking into the system of the CSP of either side participating in the authentication event is called an intrusion attack.

Insider threats are caused by persons having access to the systems used for authentication inside a CSP, for example.

An attacker might also be able to fool a user to use an insecure protocol in such a way that the user does not even notice it and thinks that he/she is still using a secure protocol.

The person holding authentication credentials can also be a risk. Intentional repudiation occurs if the person gives his/her credentials to someone else. There is no straightforward method to prevent this except for biometric authenticators, but there are ways at least to inform the user that his/her credentials are used or to require the user to authenticate through separate channels. For example an out-of-band channel can be used to confirm authentication transactions (see section 4.1.8).

4.4.5 Trusted Computing Issues

Every authenticator that involves the use of a computer has security issues. By default, every computer system is considered untrusted because software can be run on it without verifying that the software does not contain malicious code.

DOD Orange Book (1985) defines criteria to evaluate trusted computer systems know as Trusted Computer System Evaluation Criteria (TCSEC). Trusted systems can be divided into four divisions depending on the level of trust that can be placed on the system. The divisions are A, B, C and D which are further divided into classes "beyond A1", A1, B1, B2, B3, C1 and C3, where D is the weakest class and "beyond A1" the highest possible level of trust. The classification is described in further detail in table 4.2.

There have been reports of attacks where malicious code injected to a computer has hidden the real authentication event from the end user and caused the user to authenticate and authorize transactions he/she has not wanted to do.

One example of how to prevent users from becoming victims of these attacks is a concept known as Trusted Path. It is a mechanism which guarantees that a user has a method for initiating an authentication event in such way that only trusted software interacts with the user. In other words, a trusted path prevents any malicious software from impersonating the real software. A real world implementation of a trusted path is Microsoft Windows authentication initiation. A user initiates an authentication to the operating system by pressing the Control-Alt-Delete key combination. The operating system prevents all other software from grabbing the event and reacting to it and, thus, the key combination forms a trusted path. (Loscocco et al., 1998)

Laurie & Singer (2008) propose an interesting way of solving this issue. The authors suggest that the actual authentication event is redirected from the untrusted system to a separate physical device that has been verified as trusted. This way, a trusted path can be established for the actual authentication part. The device should contain a display that would tell the person authenticating what he/she is actually authenticating. No part of the untrusted environment could interfere with the device; thus it would be impossible to, for example, alter the text displayed on the device.

Table 4.2: TCSEC criteria. Based on DOD Orange Book (1985)

Division		**Class**		**Features**
D	Minimal Protection			Evaluated, but does not meet requirements.
C	Discretionary Protection	C1	Discretionary Security Protection	DAC, authentication and identification.
		C2	Controlled Access Protection	Audit trails, accountability.
B	Mandatory Protection	B1	Labeled Security Protection	Data sensitivity labels, MAC
		B2	Structured Protection	Separated into protection-critical and non-protection-critical elements.
		B3	Security Domains	Minimized complexity, automated intrusion detection.
A	Verified Protection	A1	Verified Design	Formally verified.
		Beyond A1		Exceeds A1 requirements.

Chapter 5

Person-to-Person Identification Method for Wave

In this chapter we describe our research questions in detail and answer them based on the findings from previous chapters. The method we describe is a set of existing technologies and processes combined in a novel way.

5.1 Research Questions

Our main research question is *How can users identify each other on the Wave network?* Secondary questions are *How can a real identity be linked to a Wave address and verified?* and *Is there a method for strongly authenticating users on the Wave network?*

We define identifying each other to mean whether the persons can reliably determine whether the other person is the same person as before. That is, can a person notice if the other user he/she has communicated with before on the Wave network is the same person as with whom the user is communicating at the moment of the identification event.

The secondary questions focus on process methods for integrating strong authentication providers to the Wave identification process. As we have strong authentication providers in some countries, like Tupas in Finland, are there feasible methods for using them also on the Wave network.

The research questions relate to each other and have interdependencies between them. The actual person-to-person identification in the Wave network can be split into two distinct problems. The first part is how to link a Wave address to the real identity of a person. The second part is how a Wave user can prove his/her identity to another Wave user and how this verifier can verify the identification. The answer to these two problems should provide us with a solution to the research questions.

5.2 Research Methods

A literature survey and a prototype construction were used as research methods in this thesis. The literature survey was presented in the previous chapters.

The solution was designed by describing the scenario from the users' perspective. The scenario was refined to use-case-based stories that formed the initial solution. The feasibility of the proposed solution was validated by implementing a proof-of-concept person-to-person identification service. The design and implementation was done in iterative cycles.

5.3 The Identification Method

The obvious solution would have been to base the method on public-key cryptography. It would have required a client-based implementation, but the current Wave architecture and especially the lack of clients limits the usefulness of a client based solution. Existing clients offer no extendability, which means that there is no easy way to add functionality to them. Thus, identification based on public-keys similar to S/MIME, cannot be implemented at the moment. As more clients emerge, it is probable that some of them will make it possible to solve person-to-person identification with client-based public-key solutions.

Public-key cryptography as a technology has proven its usefulness in applications requiring digital signatures and encryption. PKI with its CA-based trust model is a viable way for establishing identities of network machines and even persons. But as we found out in chapter 2, public-key Person-to-

Person identification methods have been perceived as hard-to-understand among non-professional computer users.

Our research goal was to find an understandable visual method for users to identify each other on the Internet. Real-world identification is done using physical identity cards. An analogous method for the electronic world seems as a good starting point. Electronic identity cards that resemble physical identity cards have been emerging as services on the Internet[1]. In our solution, we use the NorthID OnlineID service as the electronic identifier.

Under the hood, used as the technical solution, public-key cryptography and PKI are feasible methods for electronic identity cards as they can be both used in machine-level authentication and identification and for making sure the actions can be later verified and non-repudiated.

5.3.1 Example Scenario

To make the design and implementation of person-to-person authentication and identification easier to understand, we'll define a scenario that we will use as an example when describing and solving our problem.

Let's start with three persons, Alice, Bob and Charlie. Alice has found out that Bob is selling his expensive Single-lens reflex (SLR) camera and contacted Bob using the Wave network and made an offer to buy his camera. Alice and Bob are living far apart of each other and thus they cannot meet in person. Bob would want Alice to pay the camera in advance or at least have some kind of identifying information about Alice before actually sending the camera to her. Alice is reluctant to pay the money before knowing who Bob really is. Our third person, Charlie, is trying to either get Alice to pay him instead of Bob or Bob to send the camera to him instead of Alice. He does ths by impersonating Alice or Bob. The scenario is illustrated in figure 5.1.

Wave provides reliable server-to-server authentication by using XMPP and its TLS and certificate-based authentication methods. However, the current

[1]NorthID OnlineID, `https://www.nettihenkkarit.fi/`, Trufina Verified ID `http://www.trufina.com/`, Honesty Online `ttp://www.honestyonline.com/` to name a few.

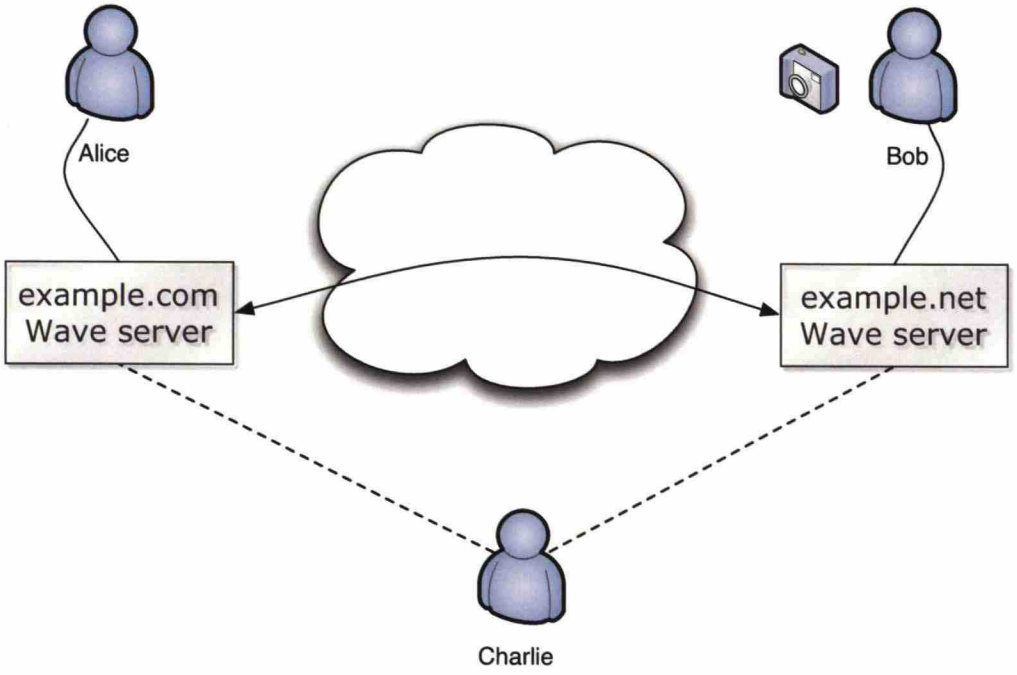

Figure 5.1: Illustration of the example scenario.

Wave protocol specifications do not specify a solution for person-to-server identification. Specifically, it is stated in the Wave specifications that client-to-server authentication is left out of the scope.

A Wave user can trust that another user comes from a certain domain but not the name of the other user or that the other user is the same person as before.

Alice, chatting with a user called bob@example.net, can be certain that "Bob" is from example.net, but not that bob@example.net is really called Bob. Charlie could be impersonating Bob by using the Wave address bob@example.net. Also, even if Alice has has communicated before with bob@example.net, Alice cannot know that she is talking with the same real person called Bob every time.

5.4 Establishing True Identities

To establish real identities, users must be identified in a reliable manner.

The authentication and identification methods have to be trustworthy so that users can trust the links between the Wave identities and the persons' real identities. In essence, a trusted third party needs to be established.

The service that acts as the trusted third party is responsible for creating, validating and deleting links between the real identity and other identifiers. Basically, the service stores a verified list of bindings between a person and his/her Wave addresses. To visualize this, the lines connecting addresses to identities and identities to persons in figure 4.1 are what the service tracks.

We will use the basic authentication model described in section 4.2. First of all, the service needs to establish the identity of the person, process known as identity proofing, using strong authentication as described in section 4.1.7.

The Tupas authentication solution is provided by major Finnish banks and is used by the banks, other companies and governmental organizations. It fulfills the Finnish legislative requirements on strong authentication as discussed in chapter 4. We can conclude that, by using Tupas, we can reliably

identify a user and obtain his/her Finnish personal identity code[2]. This code is then used as the persons real identity identifier in the TTP service's list of links. After creating a record of the user in the TTP database, the TTP can issue the user authentication tokens, which the user can use to authenticate to the TTP service.

In our scenario, both Alice and Bob register to the NorthID TTP service. The TTP service redirects Alice and Bob to authenticate to their bank by using their Tupas authenticators. The bank then forwards Alice's and Bob's identity codes to the NorthID TTP which stores the code to the person's record in the user database. Let us assume that Alice's personal identity code is 101070-876U and Bob's is 120464-121C[3]. Alice and Bob then get tokens for later authentication to the NorthID service. Because of the Tupas authentication, Charlie has hard time in trying to authenticate to the NorthID service as Alice or Bob. In order to succeed, Charlie would have to obtain either Alice's or Bob's Tupas account name, password and list of one-time passwords.

After establishing the users' real identities, we are ready to create links between the real identities and other identifiers, such as their Wave addresses.

5.5 Linking a Wave User Id to the Real Identity

On a general level, when linking identities together, the user has to prove that he/she controls both identities. In our case, the user has to prove that he/she controls the Wave addresses that is to be linked to the real identity.

To prove that the same person controls a Wave account, we need to have the user to authenticate on two independent channels in a specialized case of a combination of multi-factor and out-of-band authentication. The details of these methods are described in sections 4.1.6 and 4.1.8 respectively.

The check for the user's control of the Wave addresses can be implemented with a random nonce. The service sends a random number or string to the Wave user address which the user has provided. The user then has to

[2]In Finnish: *Henkilötunnus*, described in Population Register Centre (n.d.)

[3]So, now we know that Alice is a female born on October 10, 1970, and that Bob is a male born on April 12, 1964.

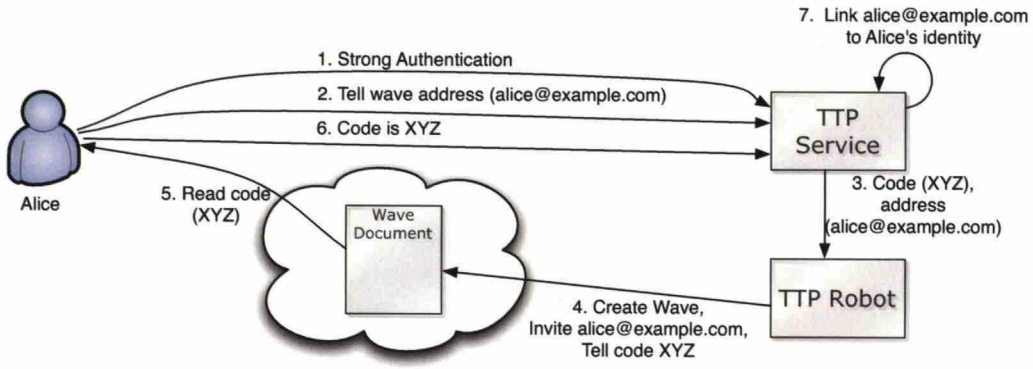

Figure 5.2: Linking a Wave address to the real identity

input the code to the TTP service. If the correct code is provided, the user has access to the Wave account, which means that the Wave address can be linked as a trusted identifier to the user's real identity. Figure 5.2 illustrates the process.

We use a robot to create a new Wave and invite the user to be linked to the Wave and let the robot tell the user a random string. The Wave protocol defines automated participants called Robots as discussed in 3.2.4. On the technical level, the robot sends an HTTP request to a Wave server that creates the Wave containing the code and the server then sends it onwards to the recipient address. A detailed sequence diagram representing the linking process is shown in figure A.1.

5.5.1 Linking Example

In our example scenario, Alice logs into the NorthID TTP service. The service requests Alice to authenticate with Tupas. Next, the service asks for Alice's Wave address. Alice types it (alice@example.com) in. The service generates a random string, for example "LGXYKVXJD6", and sends it to alice@example.com by using a Wave robot. The robot creates a new Wave, invites Alice to participate and tells Alice the random token. Alice then copies the code and pastes it onto a form on the NorthID TTP service. The TTP service then verifies that codes match and, if they do, saves the address alice@example.com to the TTP database as one of Alice's verified

addresses. In order for Charlie to either link his Wave address to Alice's identity or Alice's Wave address to his identity, Charlie would have to gain access to Alice's Wave account or Alice's NorthID TTP account. As the NorthID TTP account requires Tupas authentication, Charlie probably will not gain access to it If Charlie somehow gets Alice's Wave account in his control, he can link his real identity to her Wave account. Fortunately for Alice and Bob, this will not help Charlie in impersonating Alice, because Bob only gains knowledge of Charlie's real identity through Alice's Wave address.

To continue with our scenario, Bob verifies his Wave address in the same manner as Alice. At this point, Alice and Bob are ready to use their verified addresses to actually identify each other.

5.6 Verifying a Wave User's Real Identity

As we discovered in sections 3.1.1 and 3.2.3 about Wave addressing and authentication, Wave users can trust the domain parts of Wave addresses. Faking the domain part is difficult as Wave servers accept messages only from servers that have valid X.509 certificates issued for the domain of the sender Wave address. For example, a server with a certificate for example.com can only send messages that have example.com as the domain part of the sender address. If a user can trust the domain part then a possibility of successful person-to-person authentication method opens up.

With a Wave robot, we can have the TTP participate in a discussion through a robot. This robot can then perform identification tasks in a trustworthy manner as figure 5.3 illustrates. The fact that the user cannot verify the username part of the robot does not matter as the TTP does not issue accounts to regular users. The user can be certain that the robot is from the TTP domain by examining the domain part of the robot address. No other person or organization can appear on the Wave network as the TTP. This is the key idea in our solution.

Users' real identity with verified Wave addresses are stored in the TTP database. If the robot finds that a Wave address has been linked to a real identity, it can present a visual representation of the identification informa-

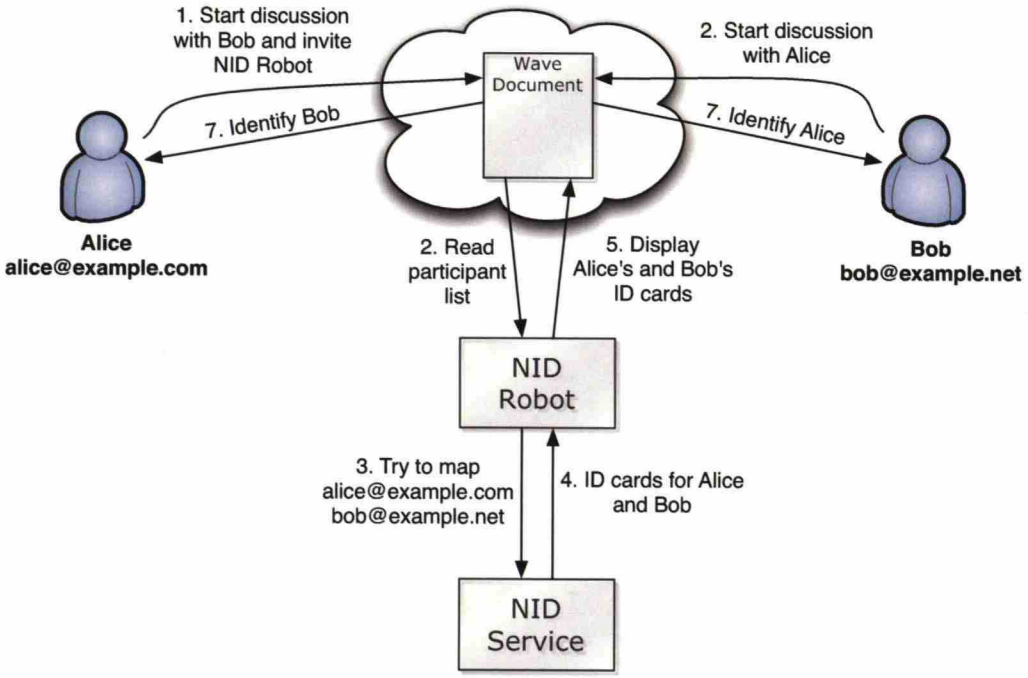

Figure 5.3: Verifying a Wave user's real identity

tion to other participants. One method for this is to create a Wave gadget that displays ID card-like information visually on a Wave discussion.

It must be noted that we cannot be sure that the person controlling the Wave address is the same person who had control of the address when the link was verified. This is because the username parts of Wave addresses cannot be trusted. In order to verify that the link is still valid and increase the level of trust in the identification, extra authentication steps may have to be done.

We have a variety of options to implement the extra authentication step. A natural requirement is that the Tupas authentication should be repeated at the time when the link is used. However, the level of assurance can also be increased by performing authentication bewheen the TTP and the user using a separate channel, as explained in section 4.1.8 about out-of-band authentication. This way, we create a multi-factor authentication scenario with an out-of-band part enhancing the quality of authentication.

As the extra step makes the verification more complex, we can decide the extra validation step to be optional. We let the verifier decide whether the extra verification step is needed or not. Figure A.2 in the Appendix displays the verification process in detail. It includes the extra verification of the identity by an SMS challenge.

The identification result displayed by the robot is valid only for the duration the robot is a participant on the Wave. This is due to the fact that contents of Wave discussions can be modified by every participant. Unauthorized modifications of the identification results might be made by other participants. It is essential that the robot monitors the Wave and notifies participants, if someone modifies the identification results.

5.6.1 Verification Example

Lets jump to our use case scenario again. Alice and Bob are now discussing on the Wave about payment and shipping of the camera as described earlier. Either one of them can now initiate the person-to-person identification. Bob notifies Alice that he is going to invite the NorthID Authentication Robot to the discussion. Bob adds the robot, robot@northid.com, to the discussion.

The robot discovers that there are two persons in the Wave, alice@example.com and bob@example.net. It contacts the NorthID TTP service and determines that both addresses have been linked to real identities. The robot then fetches Alice's and Bob's ID cards and feeds them to the discussion using an ID card Wave gadget. Now Alice and Bob see each other's ID card. If they both trust the NorthID TTP service, they can trust the ID cards too.

Neither user can at this point be sure that Alice is Alice and Bob is Bob because the link between the real identity and Wave address has been verified only at the time of the linking process. Alice, for example, cannot know if Charlie has gotten access to Bob's Wave account and is now pretending to be Bob. To ensure Bob is really Bob and not Charlie, Alice can initiate an extra verification process by clicking on a verify button on the ID card gadget. After Alice pushes the button, the NorthID robot receives an event from the Wave server notifying it that the state of the Wave has changed and that Alice has asked for an extra verification. The robot then signals the NorthID TTP service to initiate a new Tupas authentication or SMS-based extra authentication step for Bob. Bob receives an SMS and replies to it proving that he really is Bob. The NorthID TTP service receives Bob's SMS, verifies it and sends a notification to the robot. The robot then sets Bob's ID card gadget to a state that displays information about the extra verification done. As Alice sees the gadget change, she can be sure that Bob really is Bob. If Bob wants, he can ask Alice to authenticate in a similar way.

Our solution so far has required both Alice and Bob to have their identities verified by the one and same TTP. Next, we describe a method for letting Alice and Bob use separate TTPs and still be able to identify each other on the Wave network.

5.7 Network of Trust

As Wave robots have similar identities on the Wave network as humans, similar principles can be used to authenticate them. If there are two robots from two TTPs, domainA.com and domainB.com, the robots can identify each other as being from a trusted domain. This creates an interesting possibility on forming a network of TTP robots that know each other. Tech-

nically, the identification between robots can be established in the background, for example, by PKI methods. The trusted domains are needed only for humans to have a visual verification method.

The PGP trust model, web-of-trust, has been criticized of being an anarchy as discussed in section 4.3.3. Despite of the criticism, the decentralized trust model itself is an interesting way to construct a network of trust. We apply a modified version of the web-of-trust in our solution to build trust relationships between users.

Our hypothesis is that, if a person trusts one TTP, there is a possibility that his/her trust chain can extend to another TTP. This is possible if his/her own identity provider declares the second TTP as trusted. The trust relationships can be established between TTPs, for example, by auditing their PKI policies and operations as described in section 4.3.4 or by mutually agreeing to follow a set of policies, for example, by forming a business contract between the TTPs.

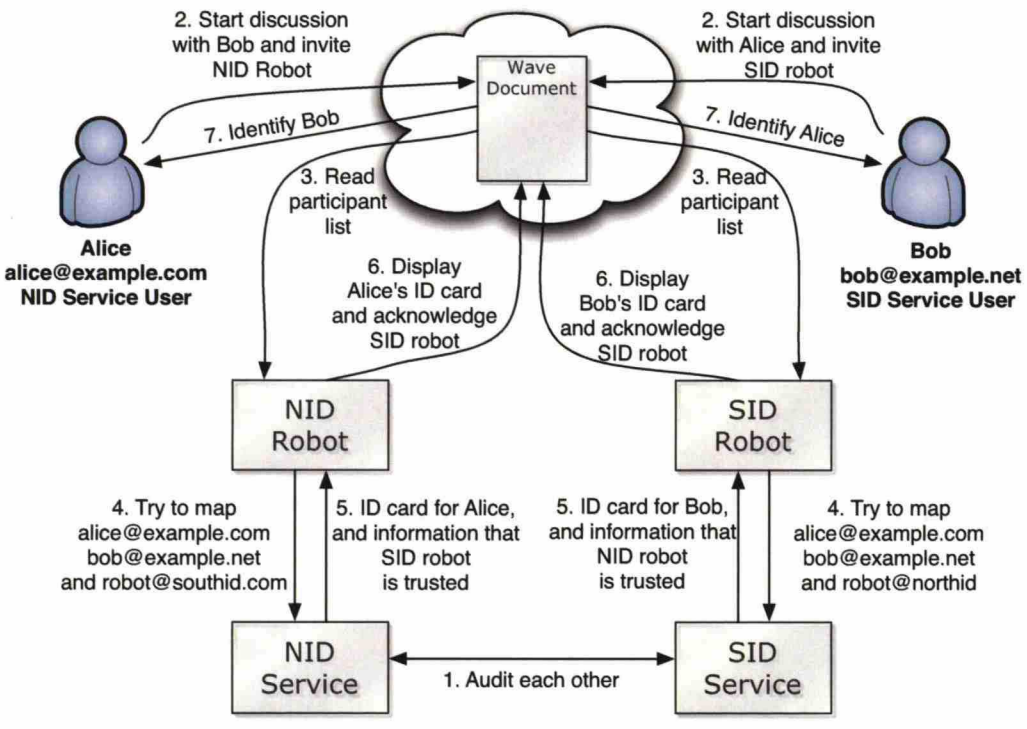

Figure 5.4: Network of Trust

Let us assume that, in our scenario, Alice uses NorthID as before but Bob has chosen SouthID as his identity provider. Alice and Bob are on the same Wave discussion and want to identify each other. Alice invites the NorthID robot and Bob the SouthID one to the discussion. The NorthID robot identifies Alice and displays her ID card. Similarly, Bob's ID card is shown by the SouthID robot. Now, if Alice has never heard of SouthID or Bob of NorthID, they cannot trust the ID cards displayed by the robots. It seems like Alice and Bob cannot identify the other one.

But, if NorthID has determined that SouthID operates according to a set of requirements defined by NorthID, their identification robot has been configured to identify the SouthID robot. In the Wave, the NorthID robot first identifies the SouthID robot using PKI and then displays, for example, an ID card of the robot. This tells Alice visually that the SouthID robot is trusted by the NorthID robot. Now Alice knows that NorthID trusts SouthID, and that the SouthID robot has identified Bob. Alice can be fairly certain that Bob is really Bob. Bob can identify Alice in a similar manner. The scenario is shown in figure 5.4. In order for Charlie to impersonate either one of the users, he could try to find a TTP that gives him a fake identity and that is trusted by either Alice's and Bob's TTP. To prevent this, both NorthID and SouthID need to exercise good judgement in building their webs of trust.

The network of trust requires each TTP to audit the other TTPs, or in some other way, ensure that the operating policies and practices of the other TTP match the requirements of the first TTP. This works for a small number of TTPs, but the number of connections between the TTPs grows quadratically as the network grows. Identity federation architectures that are described in section 4.3.6 could be used to reduce the total number of connections needed between TTPs. There could be a limited set of root TTPs that form the network of trust and audit their sub-TTPs. This is similar to the current X.509 architecture where CAs can be independent, cross-certify each other or form a hierarchy.

Another solution is to form a directory of TTPs operating on the Wave network and providing identification services. The directory could then be used by the TTPs and their robots as an authoritative source of trusted

TTPs providers. To get listed in the directory, a TTP would have to agree to follow a set of predefined policies regarding the identification procedures. The directory could also map domain names to their specific TTPs.

Chapter 6

Proof-of-Concept Implementation

In this chapter, we describe a proof-of-concept implementation of the model presented in the previous chapter.

6.1 General Architecture

The architecture is visualized in figure 6.1. The implementation consists of the NorthID Trusted Third Party service, an identification robot implemented to the Google App Engine infrastructure, and an ID card implemented as a Google Wave Gadget.

In general, the components are implemented in Java. The components communicate with each other using TLS-secured HTTP protocol and exchange JavaScript Object Notation (JSON) formatted messages. In the proof-of-concept implementation, the authentication between the components is based on username and password combinations.

6.2 NorthID Service API

An API was developed to the NorthID service. The API contains two methods for the identification robot to use.

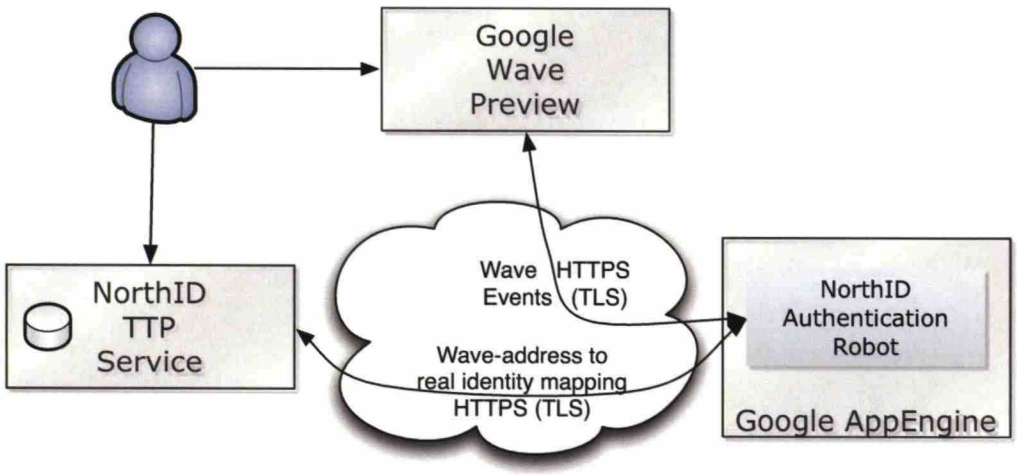

Figure 6.1: Illustration of the actual implementation architecture.

The first method maps Wave addresses to real identity parameters. When a robot needs to find out if a link between a real identity and a Wave address exists, it calls the mapping method with the Wave address as a parameter. The NorthID TTP service then searches for a linking record in the TTP database based on the Wave address. If a record is found, the service formulates a JSON-based object containing details about the real identity of the owner of the Wave address. The data returned is enough to fill the fields of an ID card gadget.

The second method is a signalling method. The robot calls it whenever it notices that someone has requested additional online verification of a Wave address linked to a real identity. When the method is called, the NorthID service initiates a verification process. The service signals the verification result to the robot by calling a notification method implemented in the robot API.

6.3 Identification Robot

The identification robot was written in Java using the Google Wave Robot API. The robot communicates with the NorthID service using Hypertext

Transfer Protocol Secure (HTTPS). The connections to the service API are authenticated with HTTP Basic Authentication consisting of a username and password combination. The robot uses the NorthID service API to identify Wave users and to request additional strong authentication.

The robot registers itself to the Wave server using the published API and asks for the server to send all `WaveletParticipantsChanged` events to it. This way, the robot is notified of every change in the list of participants in the Waves in which the robot is participating.

When the robot identifies the participants for the first time, it creates a blip for displaying the identification results and appends it to the end of the Wave. The blip is annotated with a unique annotation in order for the robot to efficiently locate it when reacting to events. By default, the server does not include the whole Wave in an event sent to the robot. Thus, the robot might not receive the blip containing previous identification results. This issue was solved by configuring the robot to request the whole Wave content to be included in the event context by the server.

No functionality for the robot to guard its blip from unauthorized modifications was implemented. The proof-of-concept implementation is thus vulnerable to attacks where someone else participating on the same Wave discussion edits the identification results. In the future this problem could be solved if the Wave architecture allowed write access to a blip to be restricted to specific users. Also, if there are only two participants in the Wave (in addition to the robots), Alice and Bob both know that there are no outsiders present who could maliciously modify the Wave.

6.3.1 Robot API for Wave Address Verification

In addition to using the NorthID service, the robot exposes an API for the service. This API is used for sending the random challenges to a Wave address needed in the initial Wave address verification. The strong authentication results are also received by the robot API from the NorthID service. The APIs are implemented using HTTPS as the transport protocol and JSON as the data protocol. This functionality requires using the Active Robot API, as the robot needs to initiate a new Wave discussion rather than just participate in a Wave. During the initial development phases, the

Active Robot API was not yet available, but it was published by Google in March 2010 thus making it possible to implement the needed functionality.

In the current state of the Google Wave preview service, a robot that wants to use the Active Robot API has to register to Google and obtain OAuth tokens for verifying the registration. This process is described in the Robot API documentation.

6.4 ID Card Gadget

The User Interface (UI) of the ID card displayed in the Wave discussions was implemented as a gadget. The gadget uses the Google Wave Gadget API and consists of an XML document containing Javascript, Cascading Style Sheets (CSS) and HTML.

One way of implementing the gadget would be to let the robot generate a gadget XML containing the identification data in a programmatic way each time a positive identification is made. But as gadgets are stored in Waves as Uniform Resource Locators (URLs) to the gadget XML, this would lead to a situation, where an attacker could just copy the gadget to another Wave and fake his/her identity. Thus, the gadget located in a URL cannot contain both the UI and the identification data.

We have solved this issue by implementing the ID card gadget as a generalized ID card component containing placeholders for the actual identification data. The data is fed to the gadget by the identification robot as key-value pairs using the Gadget API. The data is stored as the gadget state in the Wave itself.

6.5 Unimplemented Functionality

Identity proofing with Tupas was designed but not implemented. It was not seen as essential for validating the method presented.

Implementation of the Network of Trust concept was also left out of the proof-of-concept scope as it was seen as an extension to the initial solution of person-to-person identification method using only one TTP.

Figure 6.2: Sample identification of the author and the instructor made by the robot. The ID cards are displayed as a gadgets and the robot is shown as a participant in the Wave.

6.6 Limitations of the Implementation

Due to the fact that Google has not yet made possible to run robots in any other environment than their Google AppEngine, our implementation cannot be absolutely trusted. As the AppEngine limits robots to one *appspot.com* domain, users cannot trust our robot in a strict sense.

In an ideal situation, we would host the robot in a network and on hardware controlled by ourselves to organize everything under our own TTP. As we do not have full control to the environment where our robot is running, we cannot guarantee that the solution is at its current configuration entirely secure.

In the future, when it is possible to run robots in other environments, the implementation can be made more secure and trustworthy by installing our robot in our own environment.

6.7 Summary

The proof-of-concept implementation contains functionality for linking a Wave address to a real identity and for asking a robot to identify participants in a Wave discussion. It does not contain all the security features needed for a full implementation.

The proof-of-concept implementation contains functionality for evaluating the method for person-to-person identification on a general level. The proof-of-concept also shows that the identification process is implementable in the Wave network when the network is opened to all domains and services.

Chapter 7

Discussion

In this chapter, we analyze our findings from the proof-of-concept implementation described in the previous chapter and compare existing person-to-person identification methods to the methods proposed in chapter 5.

7.1 Overall Findings

No reliable person-to-person identification method exists currently on the Wave network. Identification is based on Wave addresses that cannot be reliably mapped to real identities.

This thesis presents a way to link real identities with Wave addresses. If messages sent to a network can be traced back to the sender domain, our method can be used to provide person-to-person identification to the network. Adding strong authentication to such networks is also possible as authentication information can be relayed from strong authentication providers to the network.

The Wave network is an open network formed according to the community principles (Google, n.d.). Not everything is open and available yet, as we found out during the proof-of-concept implementation. The Wave UI has not been released yet, nor is it possible to connect to the Wave network with ones own Wave server. It is expected that the limitations will be lifted in the future. The methods presented in this thesis will become useful after that.

7.2 Comparison to Existing Identification Methods on the Wave Network

As the current Wave network consists of only the Google Wave server, person-to-person identification can be based on trust in Google and its authentication methods. Once Google opens up the Wave network and lets other servers connect to it, the identification quality will lower as, at that point, no one organization will have control over user authentication. From then on, our proposed method could be used to improve the quality of identification. Once non-Google servers are allowed to the network, our solution can be installed under one trusted domain, such as northid.com, making it possible to base trust on the TTP domain as required by our method.

7.3 Comparison to Public Key Based Methods

Compared to person-to-person identification methods requiring knowledge and understanding of public key cryptography, our solution hides all technical details from the end user. The user can rely on visual ID cards for making identification decisions. This is analogous to the real world where identification is often done by visual inspection of identity documents, as described in section 4.1.10.

Methods based on public-key cryptography can be difficult to understand as research, for example, by Stedman et al. (2008) shows. Identification in these methods is also often left as the responsibility of the end user. In order for the user to be able to identify other persons, he/she has to understand the relatively difficult concept of digital signatures and certificates. In our method, the TTP robot verifies the identities, hides all cryptographic concepts from the user, and displays the results to end user in an easy-to understand visual form.

Our method requires registration to a TTP and proving control of a Wave address before identification can be done on the Wave network by a robot. This registration and identity proofing process does not differ from general PKI-based identification methods where users have to obtain trusted

certificates from recognized CAs. It is possible that effort needed for registration and linking limits the rate of adoption of the proposed method, as the same requirement may have been one limiting factor for the adpotion rate of PKI-based methods.

It is improbable that only one TTP operates in the global Wave network. Thus, it is unlikely that everyone would register and link his/her Wave address to the same TTP. This issue was identified during the research and as a solution the Network of Trust concept was proposed. On a general level, the concept seems viable, but the technical and business processes required for implementing it need further analysis.

7.4 Strong Authentication

Another important finding is that strong authentication methods can be linked to the identification process in a way that is simple and intuitive to the end users In other words, a robot together with a TTP can use any kind of external strong authentication provider to authenticate users. This is similar to, for example, the 3-D Secure method, except that in our solution the user can be directed to the strong authentication site completely, not just inside a IFrame, which should improve both user experience and security. As discussed in section 4.3.5, the IFrame architecture has been criticized of having security gaps because browser TLS certificate information is hidden inside the IFrame and not displayed as a familiar lock icon on the browser frame.

7.5 Security Issues

Our research did not include deep analysis of security considerations. More detailed analysis on possible attack vectors and potential weaknesses is needed before the proposed solution is deployed.

Adding verifiable public-key signatures to the data is one of the first concrete steps that should be done in order to make the ID card secure on a technical level. The card data could be signed by the robot so that the data could be verified by validating the signature. This would enable the users to validate the ID cards also cryptographically.

Chapter 8

Conclusions

The goal if this thesis was to study if it is possible to implement person-to-person identification on the Google Wave network. As literature research and analysis of the Wave and XMPP protocols showed, it is possible on a theoretical level because domains can be trusted in these networks, and thus, a TTP can provide trustable identification service to the network. The results of the proof-of-concept implementation showed that the Google Wave network is not yet open enough to actually achieve a robust person-to-person identification method on a concrete level, but that the method seems indeed viable. If and when the whole Google Wave network is opened up to all domains and servers, there should be no limitation to actually implementing fully functional person-to-person identification on the Wave network.

The methods presented in this thesis are a novel way of implementing person-to-person identification on the Wave network. They have the following key characteristics compared to existing solutions.

Google Wave together with the presented person-to-person identification method brings a totally new user experience for strong person authentication. A method for strong authentication of persons was also described. Taking advantage of the trusted identity providers available in many countries, it is possible to strongly authenticate persons. In Finland, Tupas can be used as the trusted identity provider. Thus, with Finnish Internet users, strong authentication is achievable on the Wave network.

Identification information is presented to the user in easily readable visual form and a TTP performs the actual authentication and cryptography behind the scenes. With the presented method, a user can ask for a TTP to identify Wave participants and display the results as visually understandable information compared to the traditional PKI-based method on where the user has to understand public key cryptography concepts.

The method presented is generalizable to other social networks where domains can be trusted. Both the person-to-person identification and strong authentication methods described in this thesis are generalizable to XMPP based networks, and also to all such networks where messages sent to the network are traceable back to at least the first server which injected the message into the network.

Trust chains can be formed by inviting robots that trust each other to the Wave discussion. Persons can identify each other even if they are using different TTPs as their identity providers, as the TTPs can form trust networks and relay trust between them.

The methods we have described have numerous potential usage scenarios. Private persons buying and selling goods and services over the Internet could benefit from reliable person-to-person identification methods. Companies and public administration could offer customer service and collaborative form filling with strong authentication on the Wave network.

8.1 Future work

Altogether, person-to-person identification methods on the Internet seem to be an area where not much existing scientific research has been done.

There are a number of interesting future work areas on the subject of this thesis and, in general, on person-to-person identification on the Internet. The method described in this thesis was tested only on the proof-of-concept level. A thorough security analysis should be done, for example, by creating a full scale implementation and analyzing it.

From a softer viewpoint, user acceptance and understandability of our solution should be studied. Prior, technology based, person-to-person identification methods have failed to gain popularity because of the difficulty

for people to actually understand what is happening during authentication and whether the identification is successful or not. Thus, it would be interesting to know whether the method we propose is easier to comprehend than the previous ones.

Use of digital signatures in the Wave network is another compelling field of future research. Agreements, contracts and other documents needing signatures by the participants could be signed over the Wave network in a Wave discussion. As our solution for identification involves using a TTP robot, could the Wave documents be signed and verified by the same robot?

Bibliography

Abdul-Rahman, A. (1997). The PGP Trust Model. *EDI-Forum: the Journal of Electronic Commerce, 10*(3), 27–31.

Anonymous. (n.d.). *Pidgin-Encryption.* Retrieved April 2, 2010, from `http://pidgin-encrypt.sourceforge.net/`

Banks, M. A. (2008). Evolution. In *On the Way to the Web* (pp. 49–59). Apress. Available from `http://www.springerlink.com/content/m51x604w32312313/`

Baset, S. A., & Schulzrinne, H. G. (2006, april). An Analysis of the Skype Peer-to-Peer Internet Telephony Protocol. In *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings* (p. 1 - 11).

Baxter, A., Bekmann, J., Berlin, D., Lassen, S., & Thorogood, S. (2009, jul). *Google Wave Federation Protocol Over XMPP.* Retrieved March 28, 2010, from `http://www.waveprotocol.org/draft-protocol-specs/draft-protocol-spec`

Beckert, S. (2009, march). *Skype's share of the international long-distance pie on the increase.* World Wide Web electronic publication. Retrieved April 3, 2010, from `http://www.telegeography.com/cu/article.php?article_id=27800`

Bekmann, J., Lancaster, M., Lassen, S., & Wang, D. (2009). *Google Wave Data Model and Client-Server Protocol.* Retrieved April 2, 2010, from `http://www.waveprotocol.org/whitepapers/internal-client-server-protocol`

Blaze, M., Feigenbaum, J., & Lacy, J. (1996, may). Decentralized trust management. In *IEEE Symposium on Security and Privacy, 1996. Proceedings.* (p. 164 -173).

Borisov, N., Goldberg, I., & Brewer, E. (2004). Off-the-record communication, or, why not to use PGP. In *WPES '04: Proceedings of the 2004 ACM workshop on Privacy in the electronic society* (pp. 77–84). New York, NY, USA: ACM.

Boyd, D. M., & Ellison, N. B. (2008). Social Network Sites: Definition, History, and Scholarship. *Journal of Computer-Mediated Communication, 13*(1), 210-230. Available from `http://dx.doi.org/10.1111/j.1083-6101.2007.00393.x`

Brainard, J., Juels, A., Rivest, R. L., Szydlo, M., & Yung, M. (2006). Fourth-factor authentication: somebody you know. In *CCS '06: Proceedings of the 13th ACM conference on Computer and communications security* (pp. 168–178). New York, NY, USA: ACM.

Brill, E. (2010, January). *Lotusphere 2010: IBM Project Vulcan.* Retrieved March 28, 2010, from `http://www.edbrill.com/ebrill/edbrill.nsf/dx/lotusphere-2010-ibm-project-vulcan`

Burr, W. E., Standards, N. I. of, & (U.S.), T. (2006). *Electronic authentication guideline [electronic resource] : recommendations of the National Institute of Standards and Technology / William E. Burr, Donna F. Dodson, W. Timothy Polk* (Version 1.0.2. ed.) [Book, Online]. U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology, Gaithersburg, MD :.

Callas, J., Donnerhacke, L., Finney, H., Shaw, D., & Thayer, R. (2007, November). *OpenPGP Message Format* (No. 4880). RFC 4880 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc4880.txt` (Updated by RFC 5581)

Campbell, B., Rosenberg, J., Schulzrinne, H., Huitema, C., & Gurle, D. (2002, December). *Session Initiation Protocol (SIP) Extension for Instant Messaging* (No. 3428). RFC 3428 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3428.txt`

Crispin, M. (2003, March). *INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1* (No. 3501). RFC 3501 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3501.txt` (Updated by RFCs 4466, 4469, 4551, 5032, 5182)

Crocker, D. (1982, August). *STANDARD FOR THE FORMAT OF ARPA INTERNET TEXT MESSAGES* (No. 822). RFC 822 (Standard). IETF. Available from `http://www.ietf.org/rfc/rfc822.txt` (Obsoleted by RFC 2822, updated by RFCs 1123, 2156, 1327, 1138, 1148)

Department of Defense Trusted Computer System Evaluation Criteria [Computer software manual]. (1985, December). (DOD 5200.28-STD (supersedes CSC-STD-001-83). (Orange Book))

Diffie, W., & Hellman, M. (1976, nov). New directions in cryptography. *Information Theory, IEEE Transactions on, 22*(6), 644 - 654.

Elkins, M., Torto, D. D., Levien, R., & Roessler, T. (2001, August). *MIME Security with OpenPGP* (No. 3156). RFC 3156 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3156.txt`

Ellison, C., & Schneier, B. (2000). Ten Risks of PKI: What You're not Being Told about Public Key Infrastructure. *Computer, 16*(1), 1.

ETSI. (2003, August). *Mobile Commerce (M-COMM); Mobile Signature Service; Security Framework* (Technical Report No. TR 102 206 V1.1.3). European Telecommunications Standards Institute.

Federation of Finnish Financial Services. (2008, June). *The Certification Principles of the Banks' Tupas Certification Service* (Tech. Rep. No. v1.1). `http://www.fkl.fi/modules/system/stdreq.aspx?P=2800&VID=default&SID=758696344345909&A=processocument`: Author.

Finnish Acts of Parliament. (2009, August). *Laki vahvasta sähköisestä tunnistamisesta ja sähköisistä allekirjoituksista 7.8.2009/617.* Available from `http://www.finlex.fi/fi/laki/ajantasa/2009/20090617`

Freed, N., & Borenstein, N. (1996, November). *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies* (No. 2045).

RFC 2045 (Draft Standard). IETF. Available from `http://www.ietf.org/rfc/rfc2045.txt` (Updated by RFCs 2184, 2231, 5335)

Gajek, S., Manulis, M., Sadeghi, A.-R., & Schwenk, J. (2008). Provably secure browser-based user-aware mutual authentication over TLS. In *ASIACCS '08: Proceedings of the 2008 ACM symposium on Information, computer and communications security* (pp. 300–311). New York, NY, USA: ACM.

Gaw, S., Felten, E. W., & Fernandez-Kelly, P. (2006). Secrecy, flagging, and paranoia: adoption criteria in encrypted email. In *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems* (pp. 591–600). New York, NY, USA: ACM.

Gelhausen, A. (2010, March). *IRC Networks – Summary.* World Wide Web electronic publication. Retrieved March 27, 2010, from `http://irc.netsplit.de/networks/summary.php`

Goode, B. (2002, sep). Voice over Internet protocol (VoIP). *Proceedings of the IEEE, 90*(9), 1495 - 1517.

Google. (n.d.). *Community Principles.* World Wide Web electronic publication. Retrieved April 20, 2010, from `http://www.waveprotocol.org/wave-community-principles`

Google. (2010a). *About Google Wave.* Retrieved March 28, 2010, from `http://wave.google.com/about.html`

Google. (2010b). *Google Wave API Overview.* Retrieved March 14, 2010, from `http://code.google.com/apis/wave/guide.html`

Google. (2010c). *Wave Extensions.* Retrieved March 28, 2010, from `http://code.google.com/apis/wave/extensions/`

Grinter, R. E., & Palen, L. (2002). Instant messaging in teen life. In *CSCW '02: Proceedings of the 2002 ACM conference on Computer supported cooperative work* (pp. 21–30). New York, NY, USA: ACM.

Guttman, B., Roback, E., & National Institute of Standards and Technology (U.S.). (1995). *An introduction to computer security [microform] : the NIST handbook* [Book, Microform, Online]. U.S. Dept. of Commerce, Technology Administration, National Institute of Standards and Technology. Available from `http://purl.access.gpo.gov/GPO/LPS72090`

Haller, N. (1995, February). *The S/KEY One-Time Password System* (No. 1760). RFC 1760 (Informational). IETF. Available from `http://www.ietf.org/rfc/rfc1760.txt`

Hoffman, P., & Blanchet, M. (2002, December). *Preparation of Internationalized Strings ("stringprep")* (No. 3454). RFC 3454 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3454.txt`

Horton, M. (1986, February). *UUCP mail interchange format standard* (No. 976). RFC 976. IETF. Available from `http://www.ietf.org/rfc/rfc976.txt` (Updated by RFC 1137)

Housley, R. (2009, September). *Cryptographic Message Syntax (CMS)* (No. 5652). RFC 5652 (Draft Standard). IETF. Available from `http://www.ietf.org/rfc/rfc5652.txt`

Hunt, R. (2001). Technological infrastructure for PKI and digital certification. *Computer Communications, 24*(14), 1460 - 1471. Available from `http://www.sciencedirect.com/science/article/B6TYP-4417950-B/2/dfa2555659bbc311b83eef9fbf0d019b`

International Civil Aviation Organization (Ed.). (2006). *Part 1 - Machine Readable Passport - Volume 2 Specifications for Electronically Enabled Passports with Biometric Identification Capabilities*. International Civil Aviation Organization.

International Organization for Standardization. (1997). *Information technology – Security techniques – Entity authentication – Part 1: General* (ISO Standard No. ISO/IEC 9798-1:1997).

International Organization for Standardization. (1999). *Information technology – Security techniques – Message Authentication Codes (MACs) – Part 1: Mechanisms using a block cipher* (ISO Standard No. ISO/IEC 9797-1:1999).

Internet Retailer. (2005, January). *Verified by Visa security program used as bait in phishing scams.* Retrieved March 13, 2010, from `http://www.internetretailer.com/dailyNews.asp?id=13764`

ITU-T. (1996, nov). *H.323 : Visual telephone systems and equipment for local area networks which provide a non-guaranteed quality of service* (Tech. Rep.

No. H.323 (11/96)). International Telecommunication Union. Available from http://www.itu.int/rec/T-REC-H.323-199611-S/en/

ITU-T. (1999). *Recommendation F.400/X.400* (Tech. Rep.). International Telecommunication Union. Available from http://www.itu.int/rec/T-REC-F.400-199906-I/en

ITU-T. (2005, August). *Information technology – Open Systems Interconnection – The Directory: Public-key and attribute certificate frameworks* (International Standard No. X.509). http://www.itu.int/rec/T-REC-X.509-200508-I: TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU.

ITU-T. (2009, dec). *H.323 : Packet-based multimedia communications systems* (Tech. Rep. No. H.323 (12/09)). International Telecommunication Union. Available from http://www.itu.int/rec/T-REC-H.323/en/

Jankvist, U. (2008). A teaching module on the history of public-key cryptography and RSA. *BSHM Bulletin: Journal of the British Society for the History of Mathematics, 23*(3), 157–168.

Jefferies, N., Mitchell, C. J., & Walker, M. (1995). A Proposed Architecture for Trusted Third Party Services. In *Proceedings of the International Conference on Cryptography: Policy and Algorithms* (pp. 98–104). London, UK: Springer-Verlag.

Jøsang, A., Fabre, J., Hay, B., Dalziel, J., & Pope, S. (2005). Trust requirements in identity management. In *ACSW Frontiers '05: Proceedings of the 2005 Australasian workshop on Grid computing and e-research* (pp. 99–108). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

Jøsang, A., Zomai, M. A., & Suriadi, S. (2007). Usability and privacy in identity management architectures. In *ACSW '07: Proceedings of the fifth Australasian symposium on ACSW frontiers* (pp. 143–152). Darlinghurst, Australia, Australia: Australian Computer Society, Inc.

Kaliski, B. (1998, March). *PKCS #7: Cryptographic Message Syntax Version 1.5* (No. 2315). RFC 2315 (Informational). IETF. Available from http://www.ietf.org/rfc/rfc2315.txt

Kissner, L., & Laurie, B. (2009, may). *General Verifiable Federation.* Retrieved March 28, 2010, from `http://www.waveprotocol.org/whitepapers/wave-protocol-verification/Generallyverifiablewaveprotocol.pdf`

Klensin, J. (2008, October). *Simple Mail Transfer Protocol* (No. 5321). RFC 5321 (Draft Standard). IETF. Available from `http://www.ietf.org/rfc/rfc5321.txt`

Laurie, B., & Singer, A. (2008). Choose the red pill and the blue pill: a position paper. In *NSPW '08: Proceedings of the 2008 workshop on New security paradigms* (pp. 127–133). New York, NY, USA: ACM.

Leach, P., & Newman, C. (2000, May). *Using Digest Authentication as a SASL Mechanism* (No. 2831). RFC 2831 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc2831.txt`

Loscocco, P. A., Smalley, S. D., Muckelbauer, P. A., Taylor, R. C., Turner, S. J., & Farrell, J. F. (1998). The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments. In *In Proceedings of the 21st National Information Systems Security Conference* (pp. 303–314).

Madsen, P., Koga, Y., & Takahashi, K. (2005). Federated identity management for protecting users from ID theft. In *DIM '05: Proceedings of the 2005 workshop on Digital identity management* (pp. 77–83). New York, NY, USA: ACM.

Melnikov, A., & Zeilenga, K. (2006, June). *Simple Authentication and Security Layer (SASL)* (No. 4422). RFC 4422 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc4422.txt`

Menezes, A. J., Oorschot, P. C. van, & Vanstone, S. A. (2001). *Handbook of Applied Cryptography.* CRC Press. Available from `http://www.cacr.math.uwaterloo.ca/hac/`

Meunier, P. (2007, May). *"Verified by VISA" Issues.* Retrieved March 13, 2010, from `http://www.cerias.purdue.edu/site/blog/post/verified-by-visa-issues/`

Meyer, D. (2008, September). *C2C Authentication Using TLS* (No. 0250). XEP-0250 (Deferred). XSF. Available from `http://xmpp.org/extensions/xep-0250.html`

Millard, P., Saint-Andre, P., & Meijer, R. (2008, September). *Publish-Subscribe* (No. 0060). XEP-0060 (Draft). XSF. Available from `http://xmpp.org/extensions/xep-0060.html`

Murdoch, S., & Anderson, R. (2010). Verified by Visa and MasterCard SecureCode: or, How Not to Design Authentication. In *Financial Cryptography and Data Security '10.* Available from `http://www.cl.cam.ac.uk/~rja14/Papers/fc10vbvsecurecode.pdf`

Myers, J., & Rose, M. (1996, May). *Post Office Protocol - Version 3* (No. 1939). RFC 1939 (Standard). IETF. Available from `http://www.ietf.org/rfc/rfc1939.txt` (Updated by RFCs 1957, 2449)

Nardi, B. A., Whittaker, S., & Bradner, E. (2000). Interaction and outeraction: instant messaging in action. In *CSCW '00: Proceedings of the 2000 ACM conference on Computer supported cooperative work* (pp. 79–88). New York, NY, USA: ACM.

Newman, C., Menon-Sen, A., & Melnikov, A. (2010, February). *Salted Challenge Response (SCRAM) SASL and GSS-API Mechanism* (No. draft-ietf-sasl-scram-11). draft-ietf-sasl-scram-11 (Draft). IETF. Available from `http://tools.ietf.org/html/draft-ietf-sasl-scram-11`

Nichols, D. A., Curtis, P., Dixon, M., & Lamping, J. (1995). High-latency, low-bandwidth windowing in the Jupiter collaboration system. In *UIST '95: Proceedings of the 8th annual ACM symposium on User interface and software technology* (pp. 111–120). New York, NY, USA: ACM.

O'Gorman, L. (2003, December). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE, 91*(12), 2021-2040.

Oikarinen, J., & Reed, D. (1993, May). *Internet Relay Chat Protocol* (No. 1459). RFC 1459 (Experimental). IETF. Available from `http://www.ietf.org/rfc/rfc1459.txt` (Updated by RFCs 2810, 2811, 2812, 2813)

Otjacques, B., Hitzelberger, P., & Feltz, F. (2007). Interoperability of E-Government Information Systems: Issues of Identification and Data Sharing. *Journal of Management Information Systems, 23*(4), 29–51.

Otway, D., & Rees, O. (1987). Efficient and timely mutual authentication. *SIGOPS Oper. Syst. Rev., 21*(1), 8–10.

Population Register Centre. (n.d.). *Personal identity code.* Retrieved April 10, 2010, from `http://www.vaestorekisterikeskus.fi/vrk/home.nsf/pages/951E0BDFF6E9FCA7C2257244003FBBE1?opendocument`

Preece, J., Maloney-Krichmar, D., & Abras, C. (2003). History of Emergence of Online Communities. In B. Wellman (Ed.), *Encyclopedia of Community.* Berkshire Publishing Group. Available from `http://www.ifsm.umbc.edu/\~{}preece/paper/6\%20Final\%20Enc\%20preece\%20et\%20al.pdf`

Quarterman, J. S., & Hoskins, J. C. (1986). Notable computer networks. *Commun. ACM, 29*(10), 932–971.

Ramsdell, B. (2004, July). *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.1 Message Specification* (No. 3851). RFC 3851 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3851.txt`

Recordon, D., & Reed, D. (2006). OpenID 2.0: a platform for user-centric identity management. In *Proceedings of the second ACM workshop on Digital Identity Management (DIM)* (pp. 11–16). New York, NY, USA: ACM. Available from `http://portal.acm.org/citation.cfm?id=1179529.1179532`

Reiss, D. (2010, February). *Facebook Chat Now Available Everywhere.* Available from `http://blog.facebook.com/blog.php?post=297991732130`

Riikonen, P. (2007, January). *Secure Internet Live Conferencing (SILC), Protocol Specification* (No. draft-riikonen-silc-spec-09). draft-riikonen-silc-spec-09 (Draft, Expired). IETF. Available from `http://tools.ietf.org/id/draft-riikonen-silc-spec-09.txt`

Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM, 21*(2), 120–126.

Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., et al. (2002, June). *SIP: Session Initiation Protocol* (No. 3261). RFC 3261 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3261.txt` (Updated by RFCs 3265, 3853, 4320, 4916, 5393, 5621, 5626, 5630)

Roth, V., Straub, T., & Richter, K. (2005). Security and usability engineering with particular attention to electronic mail. *Int. J. Hum.-Comput. Stud., 63*(1-2), 51–73.

Saint-Andre, P. (2004a, October). *End-to-End Signing and Object Encryption for the Extensible Messaging and Presence Protocol (XMPP)* (No. 3923). RFC 3923 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3923.txt`

Saint-Andre, P. (2004b, October). *Extensible Messaging and Presence Protocol (XMPP): Core* (No. 3920). RFC 3920 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3920.txt`

Saint-Andre, P. (2004c, October). *Extensible Messaging and Presence Protocol (XMPP): Instant Messaging and Presence* (No. 3921). RFC 3921 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc3921.txt`

Saint-Andre, P. (2005, March). *Jabber Component Protocol* (No. 0114). XEP-0114 (Active). XSF. Available from `http://xmpp.org/extensions/xep-0114.html`

Saint-Andre, P. (2008, March). *XMPP Protocol Flows for Inter-Domain Federation* (No. 0238). XEP-0238 (Informational). XSF. Available from `http://xmpp.org/extensions/xep-0238.html`

Saint-Andre, P., & Hildebrand, J. (2007, September). *Message Receipts* (No. 0184). XEP-0184 (Draft). XSF. Available from `http://xmpp.org/extensions/xep-0184.html`

Saint-Andre, P., & Millard, P. (2007, February). *Best Practices for Use of SASL EXTERNAL with Certificates* (No. 0178). XEP-0178 (Active). XSF. Available from `http://xmpp.org/extensions/xep-0178.html`

SetCo. (1997, May). *SET Secure Electronic Transaction Specification – Book 1: Business Description* (Specification). Visa and MasterCard.

Silva, M. da. (2003). Challenges for EDI Adoption by Small and Medium-size Enterprises (SME). In *IADIS International Conference e-Society, Lisbon, Portugal.*

Stedman, R., Yoshida, K., & Goldberg, I. (2008). A user study of off-the-record messaging. In *SOUPS '08: Proceedings of the 4th symposium on Usable privacy and security* (pp. 95–104). New York, NY, USA: ACM.

Sten, A., Kaseva, A., & Virtanen, T. (2003). Fooling Fingerprint Scanners - Biometric vulnerabilities of the Precise Biometrics 100 SC scanner. In *4th Australian Information Warfare and IT Security Conference 2003.*

Taylor, D., Wu, T., Mavrogiannopoulos, N., & Perrin, T. (2007, November). *Using the Secure Remote Password (SRP) Protocol for TLS Authentication* (No. 5054). RFC 5054 (Informational). IETF. Available from `http://www.ietf.org/rfc/rfc5054.txt`

Tirsen, J. (2009). *Access Control in Google Wave.* Retrieved April 2, 2010, from `http://www.waveprotocol.org/whitepapers/access-control`

Varshney, U., Snow, A., McGivern, M., & Howard, C. (2002). Voice over IP. *Commun. ACM, 45*(1), 89–96.

Visa. (2006, April). *3-D Secure™ Protocol Specification, Core Functions* (Protocol Specification No. 70000-01 v 1.0.2). Author. Available from `https://partnernetwork.visa.com/vpn/global/category.do?categoryId=88&documentId=127&userRegion=1`

Vleck, T. V. (2009, sep). *The IBM 7094 and CTSS.* Retrieved March 28, 2010, from `http://www.multicians.org/thvv/7094.html`

Wang, D., & Mah, A. (2009). *Google Wave Operational Transformation.* Retrieved March 9, 2010, from `http://www.waveprotocol.org/whitepapers/operational-transform`

Wu, M., Garfinkel, S., & Miller, R. (2004). Secure Web Authentication with Mobile Phones. In *DIMACS Workshop on Usable Privacy and Security Software.*

Zeilenga, K. (2006a, June). *Anonymous Simple Authentication and Security Layer (SASL) Mechanism* (No. 4505). RFC 4505 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc4505.txt`

Zeilenga, K. (2006b, August). *The PLAIN Simple Authentication and Security Layer (SASL) Mechanism* (No. 4616). RFC 4616 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc4616.txt`

Zeilenga, K. (2009, September). *Design Considerations for Digital Signatures in XMPP* (No. 0274). XEP-0274 (Experimental). XSF. Available from `http://xmpp.org/extensions/xep-0274.html`

Zhu, L., Jaganathan, K., & Hartman, S. (2005, July). *The Kerberos Version 5 Generic Security Service Application Program Interface (GSS-API) Mechanism: Version 2* (No. 4121). RFC 4121 (Proposed Standard). IETF. Available from `http://www.ietf.org/rfc/rfc4121.txt`

Appendix A

Detailed Sequence Diagrams of the Processes

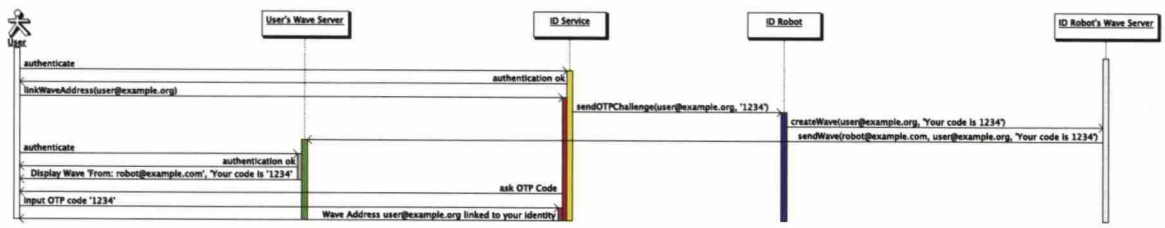

Figure A.1: Linking a Wave address to the true identity.

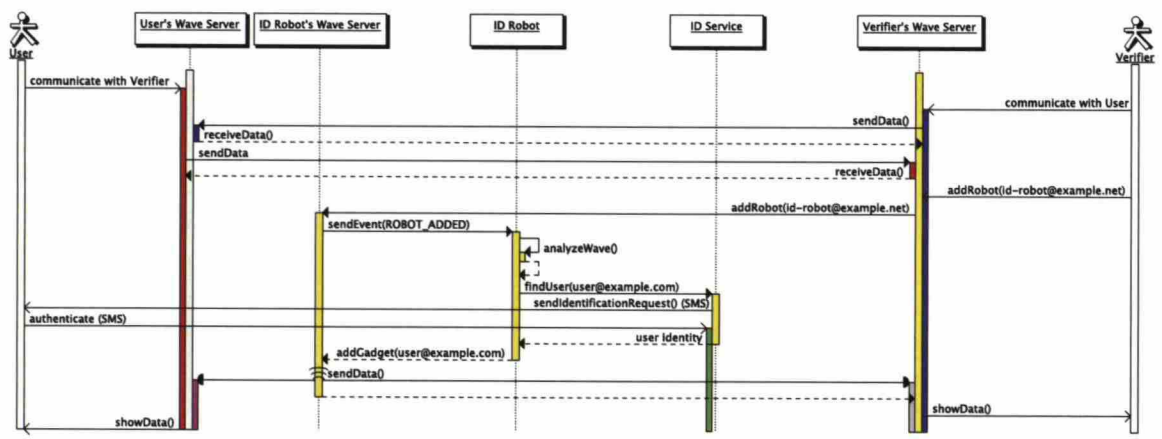

Figure A.2: Identifying a Wave user by using a TTP robot.