REPORTS
IN
INFORMATICS

ISSN 0333-3590

Next Generation Internet Banking in Norway

**Kjell J. Hole, Thomas Tjøstheim, Vebjørn Moen,
Lars-Helge Netland, Yngve Espelid, and
André N. Klingsheim**

REPORT NO 371 **February 2008**

Department of Informatics

UNIVERSITY OF BERGEN

Bergen, Norway

This report has URL
http://www.ii.uib.no/publikasjoner/texrap/pdf/2008-371.pdf

Reports in Informatics from Department of Informatics, University of Bergen, Norway, is available at http://www.ii.uib.no/publikasjoner/texrap/.

Requests for paper copies of this report can be sent to:

Department of Informatics, University of Bergen, Høyteknologisenteret,
P.O. Box 7800, N-5020 Bergen, Norway

Next Generation Internet Banking in Norway

Kjell J. Hole* Thomas Tjøstheim Vebjørn Moen
Lars-Helge Netland Yngve Espelid André N. Klingsheim

NoWires Research Group
Department of Informatics
University of Bergen

February 28, 2008

Abstract

The Norwegian banking industry has introduced a new security infrastructure for web applications, including Internet banking. The infrastructure, called BankID, has the potential to increase the security of today's web applications and facilitate new business opportunities. The authors consider BankID from the customers' point of view, analyze the risk the customers take when using BankID, and discuss how to mitigate the risk.

1 Introduction

Many countries, including Norway, Sweden, Denmark, Finland, Estonia, Austria, Belgium, and Canada, have introduced large-scale security frameworks implementing Public-Key Infrastructures (PKIs). In general, a PKI is a collection of hardware, software, processes, and people providing applications with security services based on public-key cryptography. The Norwegian banking industry has introduced a PKI called BankID. While BankID mostly authenticates Internet banking customers at the time of writing, the Norwegian banking industry wants BankID to become a national ID infrastructure used by government agencies and commercial companies to authenticate individuals and to provide legally binding digital signatures with a high degree of non-repudiation.

Two earlier papers [1], [2] analyzed the Norwegian Internet banking and Automatic Teller Machine (ATM) systems. This paper applies elements of risk management [3] to BankID. Since the Norwegian banking community declined to share technical information about BankID, we were forced to evaluate BankID exclusively from the customers' point of view. The evaluation was completed in March of 2007 and was only based on publicly available descriptions of the BankID architecture and design, as well as personal use of the system.

In the remainder of the paper, we first determine how BankID differs from a typical X.509 PKI, before carrying out a risk analysis of the end-user authentication, non-repudiation service, and customer privacy. The risk to the customers is found to be significant. We therefore suggest steps to mitigate the risk.

2 PKI primer

This section introduces a typical X.509 PKI [4]–[7].

*Contact address: Professor K. J. Hole, Department of Informatics, University of Bergen, PB. 7800, N-5020 Bergen, Norway. E-mail: Kjell.Hole@ii.uib.no, Mobile: +47 920 38 164, Web: www.nowires.org.

2.1 Keys, signatures, and certificates

In general, each end-user/customer in an X.509 PKI generates his own pair of asymmetric cryptographic keys—one *private key* and one *public key*. The public key is available to all end-users in the PKI, while the private key is only known to its owner. To secure the private key, it is stored in an encrypted file on the end-user's computer, or better, on a tamper-resistant smart card. As an example, all end-users can utilize Alice's public key to encrypt a message to her, but only she can decrypt the message because nobody else has access to her private key.

Bob can use his private key to *digitally sign* a message, or document. During the cryptographic signing procedure a value, denoted the *digital signature*, is calculated over the message. Alice verifies Bob's digital signature by applying a cryptographic procedure which takes Bob's public key, the received message, and the signature as input. If somebody tampered with the message after it was signed, then the verification will fail, else Bob must have signed the message since only he had access to the private key.

An X.509 *certificate* binds a public key to an identifier, e.g. a personal name, an assigned user number, or a web address. The identifier points to the end-user or web site with the corresponding private key.

Commercial PKIs, including BankID, utilize different pairs of keys for digital signatures and encryption/decryption. For simplicity, we do not always differentiate between these pairs.

2.2 PKI architecture

An end-user requests a certificate from a *Registration Authority* (RA) by providing the information required to issue a certificate. The RA verifies the information and sends a certification request, containing the end-user's public key, to the *Certification Authority* (CA). The CA generates the certificate and signs it with its own private key. A web site owner initiates a similar procedure to obtain a web site certificate.

A CA revokes a certificate when the public key should no longer be used. Certificate revocation is frequently caused by the termination of a customer relation. More importantly, the CA must revoke a certificate immediately if the corresponding private key is compromised. The CA keeps track of revoked certificates. Often, a CA maintains a digitally signed *Certificate Revocation List* (CRL) containing unique references to the revoked certificates, the dates they were revoked, as well as the reasons for revocation.

A PKI may have multiple CAs. For simplicity, we consider a strict, two-level hierarchy of CAs containing a single root CA on the zeroth level with a special self-signed certificate, i.e., the signature is generated by the root CA's own private key. The root CA generates and signs certificates to the CAs on level one in the hierarchy. These level-one CAs again issue certificates to the end-users in the PKI.

2.3 Transitive trust model

An entity X *trusts* another entity Y when X assumes it knows exactly how Y behaves. As an example, an end-user trusts a CA when she assumes that the binding between the name and the public key in an issued certificate is correct. If X trusts Y and Y trusts Z, then X also trusts Z, e.g., an end-user trusting a root CA also trusts a level-one CA. If she doesn't trust the root CA she will not use the PKI services.

The root CA is the starting point of all trust in a PKI. In a two-level CA hierarchy, certificate path processing extends trust from the root CA to a level-one CA. Consider an instance where Bob and Alice are issued certificates from different level-one CAs. To verify Alice's certificate, Bob first builds a path of certificates back to the root CA. The path consists of Alice's certificate, the certificate of the level-one CA that issued her certificate, and the root CA certificate. Using the root CA's public key, Bob then verifies the signature

of the level-one CA certificate. Next, he extracts the public key from this CA certificate and uses it to verify the signature of Alice's certificate. Because Bob trusts the root CA, he now assumes that the content of Alice's certificate is correct.

2.4 Authentication

Authentication can be defined as the process of establishing an understood level of confidence that an identifier refers to an end-user or a web site. The authentication is said to be *strong* if the level of confidence is high.

Authentication in a PKI is based on certificates, the corresponding private keys, and a source of revocation information, e.g. a CRL. A simplified authentication protocol illustrates the authentication process. When Bob wants to authenticate Alice, he first asks for her certificate and then uses the CRL to verify that the certificate has not been revoked. He also confirms Alice's ownership of the public key by building and verifying a certificate path to a trusted root CA. Bob now asks Alice to sign a random number, denoted the challenge, with her private key. If Bob verifies the signed challenge using Alice's public key, then he assumes he is communicating with Alice. The same protocol is carried out when Alice authenticates Bob.

The strength of the authentication relies on a well tested authentication protocol, such as the Secure Sockets Layer (SSL) protocol, on how securely the private keys are stored, and on the computational difficulty of calculating the private key corresponding to a public key. To maintain the authentication strength over time, the CAs must update their CRLs often and the lists must always be available to both parties during the authentication process.

2.5 Legal view of non-repudiation

Non-repudiation offers a person protection against a false claim by another person that a communication never took place. The two most commonly discussed types of non-repudiation are *non-repudiation of origin* in which a person cannot falsely deny having originated a message, or document, and *non-repudiation of delivery* in which a person cannot falsely deny having received a message.

From a legal point of view, non-repudiation consists of the ability to convince a third party that a specific message originated with, or was delivered to a certain person. Credible evidence is needed to persuade a judge, jury, or arbitrator. Both the quality and the presentation of the evidence determine the level of non-repudiation.

A non-repudiation service utilizing digital signatures can be built on top of a standard PKI. A high level of non-repudiation can be obtained if the basic PKI services are combined with the correct combination of legal and technical non-repudiation protocols. It is also essential that at least one trusted third party collects, validates, time stamps, signs, and stores relevant information [5, Ch. 9], [7, Ch. 4]. The third party must be able to withstand pressure from the communicating parties and present the non-repudiation evidence in an unbiased manner during a conflict.

It is important to note how the burden of proof is on the party wanting to rely on a digital signature. Hence, non-repudiation does not take away a person's legal right to refute a signature. A high level of non-repudiation only implies it will be possible to show, with high probability, that a person digitally signed a particular document even if he later denies it.

3 BankID explained

This section outlines the BankID architecture [8], [9] using the nomenclature established in the PKI primer.

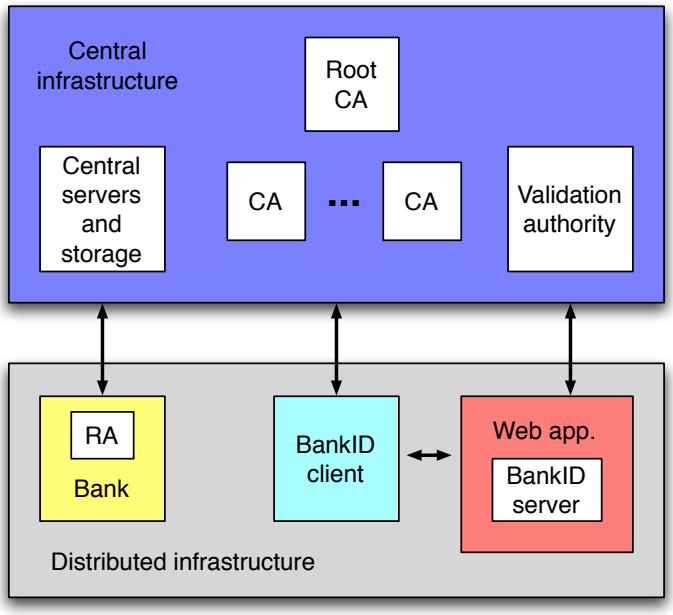

Figure 1: BankID architecture with lines indicating information flow.

3.1 Certificate types

There are three different types of 'user' certificates. The first type is a personal certificate for regular end-users, the second type is an employee certificate for end-users representing a company or an organization, and the third type is a certificate for web sites. All these certificates follow Version 3 of the X.509 Recommendation [4]. In addition, CAs, RAs, and other entities in the BankID infrastructure have their own certificates.

3.2 Architectural overview

The BankID architecture is divided into two main parts as depicted in Figure 1. The first part, referred to as the *central infrastructure*, is operated by the Norwegian Banks' Payment and Clearing Centre (also known as BBS). The central infrastructure contains CAs, a certificate *Validation Authority* (VA), central storage facilities for cryptographic keys and certificates, and functionality for digital signing of documents. The *distributed infrastructure* comprises the BankID server, which is part of a web application, the RA at an end-user's bank, and the BankID client running on the end-user's computer.

3.3 Central infrastructure

The central infrastructure includes a two-level hierarchy with one root CA and multiple level-one CAs. The root CA is owned by the Norwegian Financial Services and Saving Banks Associations. While the level-one CAs are part of the central infrastructure, each level-one CA is owned by a separate bank (or group of banks). A bank uses its CA to issue certificates to its own customers. The root-CA certificate is valid for 26 years, while the level-one CA certificates are valid for 12 years [9].

The VA utilizes CRLs from the level-one CAs to determine if certificates have been revoked. This validation service is available to both the BankID server and client in Figure 1.

3.4 BankID server

A web application utilizing BankID, e.g. an online store or an Internet banking site, runs a BankID server. This server software is available in both the C and Java programming languages, and can be incorporated into the web application. The BankID server stores its certificate and private key in a *Hardware Security Module* (HSM) or an encrypted PKCS #12 file [10]. HSMs also provide web applications with dedicated hardware for processor-intensive cryptographic operations.

3.5 Bank RA

Each bank operates its own RA software, which is likely to be integrated into the bank's customer service software. A new BankID customer requests a certificate from the RA using his Internet browser or by showing up in person at the local branch office. The RA sends a certification request to the central BankID infrastructure using the SSL protocol.

3.6 BankID client

The certification request from the bank RA starts the initialization of a new customer record. The central infrastructure first generates (at least) one public-private key pair. The private key is stored in a secure central database and the public key, as well as the request from the RA, are sent to the CA belonging to the customer's bank. The new certificate generated by the CA is stored on the central infrastructure. The customer downloads a Java applet to her Internet browser each time she wants to use the BankID client. No software or information is stored permanently on her computer.

3.7 BankID authentication procedure

The *mutual* authentication between an end-user and a web application is divided into two parts as depicted in Figure 2. The first part, a two-factor authentication procedure (to be explained), authenticates the end-user to the central infrastructure and ensures that the user controls the access to her own centrally stored cryptographic keys. The second part is a challenge-response protocol between the BankID client and server. The protocol is similar to the one described earlier. All communications between the entities in Figure 2 are executed inside SSL tunnels.

The first part of the authentication operates as follows. Since an end-user can have multiple BankID certificates issued by CAs belonging to different banks, she first enters her Norwegian Social Security Number (SSN) into the BankID client. The central infrastructure responds with a list of all her BankID affiliations enabling her to choose the one she wants. Next, the client prompts the user for a one-time Personal Identification Number (PIN) which is verified by the central infrastructure. The one-time PIN is taken from a list of PINs supplied by the end-user's bank, or generated by a hardware token, often called a PIN calculator. A fixed PIN is sometimes needed to activate the PIN calculator. Finally, the client prompts the user for a fixed password used by the central infrastructure to get access to the end-user's private key.

The described procedures are essentially traditional two-factor authentication mechanisms based on something you have (the PIN calculator or list of PINs) and something you know (the fixed password and perhaps a fixed PIN needed to activate a calculator) [6].

The second part consisting of the challenge-response protocol is completed once the central infrastructure has access to the end-user's cryptographic keys. During the protocol execution, the central infrastructure carries out the cryptographic functions for the BankID client. The BankID server uses the VA to verify the certificate from the BankID client and vice versa.

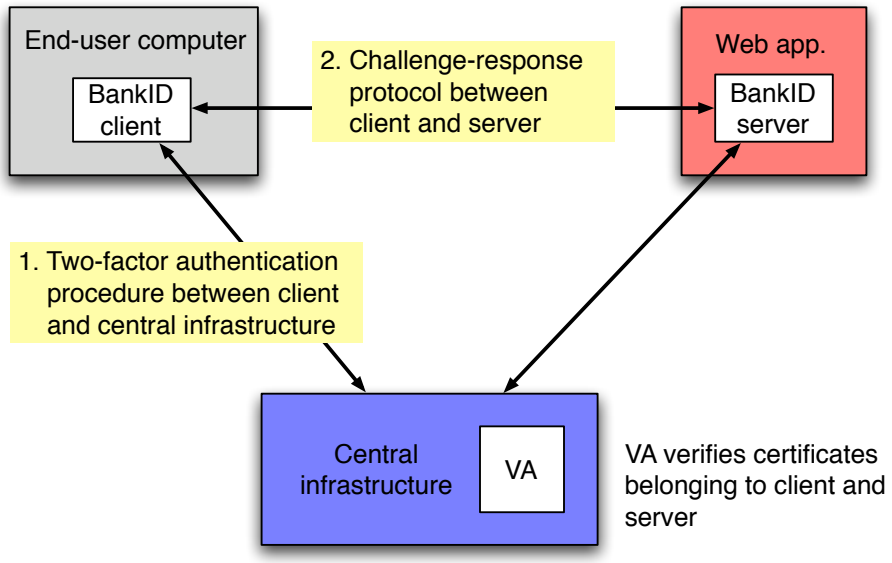

Figure 2: Overview of the BankID authentication procedure.

The reader should note that BankID differs from a typical X.509 PKI because the central storage of asymmetric keys and the central execution of cryptographic operations make it necessary for a BankID client to transmit a one-time PIN and a fixed password over the Internet to the central infrastructure.

3.8 Non-repudiation

BankID aims to provide both non-repudiation of origin and delivery. Unfortunately, nearly all information about the legal and technical non-repudiation protocols and storage of non-repudiation information are kept secret. It is known, however, that signed documents contain the signed data, the signatures of the parties, and the results of the VA requests at the time of signing [8]. BankID again differs from a typical X.509 PKI since no trusted third party is used to establish non-repudiation information for dispute resolution [8], [9].

4 Risk of authentication service

It is convenient to define *risk* as the possibility of suffering harm or loss. There are several types of risks [3]. On one hand we have pure risk associated with natural accidents such as fires, floods, hurricanes, and earth quakes. Speculative risk on the other hand is related to man-made failures, e.g., terrorism and misuse. We'll only consider speculative risk, denoted 'risk' for simplicity.

Let a vulnerability be a flaw in BankID and let a threat be an adversary with the capabilities and intentions to exploit a vulnerability. The risk taken by BankID customers is a function of the exploitable vulnerabilities and the danger from the threats [3]. This section discusses the risk associated with the BankID authentication procedure and suggests how to mitigate the authentication risk.

4.1 Exploiting the BankID authentication procedure

The BankID client gets access to the cryptographic functionality in the central infrastructure after the end-user has provided the correct SSN, one-time PIN, and fixed password. Experiments have shown that if the customer enters correct SSN and a wrong PIN three times, then the access to the central infrastructure closes down and the customer must contact his bank to reopen the account. (Access is also denied if the end-user first types in correct SSN and PIN and then enters a wrong password enough times.) The described process can be automated. In a simple proof of concept, an executable script starts Internet Explorer 7, downloads the BankID client, and simulates a user logging in with the correct SSN and wrong PIN three times.

Because SSNs have a well-defined structure it is possible to generate a large set of SSNs containing SSNs belonging to BankID customers. (The set may also contain SSNs not belonging to customers. See [1] for details.) A small program can then run through the set of SSNs and close down the customers' accounts as described above. The attack can be spread over many computers to construct a Distributed Denial-of-Service (DDoS) attack at the application layer, which is very difficult to stop if the number of computers is large ($\geq 10,000$) [1], [11]. Unlike DDoS attacks at lower network layers, it is not necessary to transmit a large amount of dummy traffic for a long time to close down the accounts, it is only necessary to try to log into each account a small number of times. Because this efficient DDoS attack on the authentication procedure has the potential to deny all users access to BankID, it can be said that the authentication procedure represents a *single point of failure*.

During the fourth quarter of 2006, about 600,000 of the 2.3 million Internet banking customer in Norway had moved to BankID. If most of the remaining customers also move to BankID during 2007, as envisioned by the Norwegian banking industry, then BankID may well get more than two million end-users in the near future. Because they'll all rely on the same infrastructure, a successful DDoS attack will have severe economic consequences. Roughly two million end-users will not be able to access their accounts, transfer funds, or pay bills, and some of the online stores using BankID will not be able to sell goods. If BankID also becomes a national ID used by government agencies, then the consequences of a DDoS attack will be even more severe.

It has been argued that the likelihood of a DDoS attack against an online banking system in Norway is small because there is no group of people with strong enough motivation to carry out an attack. On the other hand, BankID will become a national "monoculture" [12] for online banking, which makes it easy to attack two million customers at the same time. As long as the BankID authentication procedure is not changed, it will not be difficult to repeat a DDoS attack at the application layer. While traditional crackers may be reluctant to attack BankID because they fear the inevitable investigation by Norwegian government agencies concerned with national security, other groups wanting to attack the Norwegian financial system to create chaos may find BankID a tempting target.

Examples of potential threats are terrorist groups in strong opposition to the Western society in general, and Norway in particular. Cyber terrorists [13] are both able and willing to disrupt critical information structures to cause harm in order to advance their own political or religious agendas. Other terrorist groups without the needed skill set can hire or coerce crackers to carry out attacks. The danger of terrorist attacks in Europe, the increasing usage of BankID, and the increasing number of DDoS attacks against businesses in Norway reported by the Norwegian National Security Authority, lead us to conclude that the likelihood of a DDoS attack against BankID will grow during the coming years.

Observation 1 *Because the authentication procedure in BankID utilizes SSNs and denies an end-user access after a few wrong login trials, it is particularly vulnerable to DDoS attacks—just like the authentication procedures in the other Norwegian Internet banking systems. The potential DDoS attacks represent a growing risk to end-users and web site owners.*

4.2 Strength of end-user authentication

Let us consider the strength of the end-user authentication in BankID consisting of a traditional two-factor authentication procedure followed by (part of) a request-response authentication protocol as depicted in Figure 2. Assume that an adversary has obtained an end-user's PIN calculator or list of PINs. Furthermore, the adversary knows the end-user's fixed password, and—if needed—the fixed PIN used to activate the calculator. The adversary can then download the BankID client and give it access to the cryptographic functionality at the central infrastructure. The client can now complete the request-response authentication protocol with the BankID server. Hence, if the two-factor authentication is compromised, then the authentication of the end-user is compromised. Because BankID-member banks utilize different two-factor authentication mechanisms, the exact strength of the end-user authentication varies between the banks.

Observation 2 *The end-user authentication in BankID is no stronger than the two-factor authentication used in many older Internet banking systems.*

Surprisingly, the fixed password provided by the end-user during the first part of the BankID authentication procedure doesn't always strengthen the end-user authentication. One particular BankID-member bank, which provides each end-user with a PIN calculator requiring an activating PIN, lets any end-user ask for a new password by simply providing his SSN and a one-time PIN. The new password is displayed in the user's browser. Hence, if an adversary gets hold of the calculator and the activating PIN, he can also obtain a new valid password without knowing the old password!

4.3 Phishing/man-in-the-middle attacks

Each time an end-user downloads the BankID client, he also downloads HTML code containing parameters to the Java applet. While the applet itself is signed, the HTML code is not signed and changes to the parameters will not be detected by the user's Internet browser. Two parameters specifying URLs can be altered to make the BankID client communicate with the BankID server and central infrastructure through a proxy (server) controlled by a cracker. The proxy can be realized by software running on some computer.

To initiate this Man-in-the-Middle (MitM) attack, a phishing attack may first be used to trick a BankID customer into downloading modified HTML code together with the unaltered BankID client. When the customer starts a new session, the BankID client then connects to the proxy, which again connects to the BankID server and central infrastructure. We have developed proof-of-concept code to show that the proxy can steal the session after the user has authenticated himself to the central infrastructure and the BankID server.

MitM attacks are a well established form of deceit. In 2006, several Norwegian Internet banks, not based on BankID, were victims of MitM attacks. The likelihood of MitM attacks on BankID will increase as BankID gets more users and the system becomes a national monoculture.

Observation 3 *Combined phishing/MitM attacks can be used to steal sessions initiated by BankID customers because it is possible to change the addresses to which the BankID client connects.*

4.4 DDoS/phishing attacks

A DDoS attack closing down BankID customers' accounts can be followed by a phishing e-mail attack to let the same customers "know" how they again can use their accounts. It is clear from Observation 1 that it is particularly easy to execute a DDoS attack to close down accounts. An e-mail can then notify each customer about the account closure, which the customer can easily verify, and then entice the customer into entering a one-time PIN and a

password at a fake banking site to open the account. The attacker must either call the bank to try to reopen the account, or wait for the bank to do it, and then use the stolen one-time PIN and fixed password to access the account before the legitimate owner.

No such combined DDoS/phishing attack has yet been reported in the press as far as we know. However, both DDoS attacks and phishing attacks are increasing in frequency, and it may only be a question of time before combined attacks occur.

Observation 4 *BankID is potentially vulnerable to combined DDoS/phishing attacks where customers are tricked into entering one-time PINs and passwords on fake Internet banking sites.*

4.5 Exploiting the BankID server

A web application using the BankID infrastructure incorporates the BankID server software. The private key and the certificate belonging to the web application are stored in an encrypted PKCS #12 file or an HSM. At least one version of the BankID server does not support the use of an HSM. In this case, the web site owner provides a fixed password to give the BankID server access to the decrypted file. If an outside cracker or rogue insider gets hold of the file and the password, then the web site owner's private key is exposed. An attacker with the private key can pass himself off as the web site owner. The seriousness of this threat is determined by how hard it is to get hold of the PKCS #12 file and how difficult it is to determine the owner's fixed password.

The encrypted PKCS #12 file has a known structure, which makes it possible to locate on a computer. When a cracker has access to this file, he can run a brute-force or dictionary attack to try to recover the accompanying password [14]. On the other hand, "social engineering" or "shoulder surfing" may be all that is needed. The attacker can also try to install a hidden camera or a hardware keylogger to obtain the password.

In the case where a cracker has no physical access to the computer, he may still be able to employ malicious software, or *malware*, to obtain the PKCS #12 file. The cracker can then run a dictionary attack to try to determine the password and obtain the private key. Alternatively, the malicious software can try to both copy the file and sniff the password using a software keylogger.

A cracker can implement a DoS attack by developing self-propagating malware which spreads to many computers and deletes the PKCS #12 files, thus, preventing web applications from utilizing BankID. Since it is only possible to detect malware by scanning for known patterns, the risk associated with the described DoS attack cannot be completely eliminated.

Observation 5 *A BankID server utilizing an encrypted PKCS #12 file to store the private key is potentially vulnerable to well-known password attacks to decrypt the file, and DoS attacks where malware deletes the file.*

4.6 Mitigating authentication risk

From Observation 1, the policy of using SSNs to identify customers and denying them access after a few wrong login trials must be changed because it enables efficient DDoS attacks on the application layer, potentially affecting more than two million customers in the near future. According to Observation 3, transactions should be authenticated for BankID to become more robust against combined phishing/MitM attacks, and from Observation 4, passwords and PINs should not be transmitted from a BankID client to the central infrastructure since this solution is vulnerable to combined DDoS/phishing attacks. Passwords and PINs should only be used locally by the end-user to give the client access to the end-user's PKI credentials. A new end-user authentication solely based on the end-user's

public-private key pair will increase the strength of the authentication beyond what is possible with traditional two-factor authentication. From Observation 5, all web applications using BankID should employ HSMs to store private keys.

5 Risk of non-repudiation service

In the following we first discuss three vulnerabilities in the BankID architecture with the potential to limit the degree of achievable non-repudiation. We then consider how the strength of the end-user authentication influences the degree of non-repudiation. Finally, we discuss how to mitigate the non-repudiation risk.

5.1 No trusted third party

BankID does not employ a trusted third party to achieve non-repudiation despite the fact that this is required by most non-repudiation protocols described in the literature [7]. On the contrary, the BankID-member banks have strong financial relationships with both end-users and merchants owning web sites. In particular, when the web sites are Internet banking sites, the banks own the sites as well as the complete BankID infrastructure, giving the banks a large amount of control over financial operations requiring non-repudiation.

The following scenario illustrates the problem with the non-existent third party. Assume that a BankID customer and his bank have both digitally signed a document. At some later point in time there is a conflict between the bank and the customer during which the bank claims it didn't sign the document. It is then up to the customer to show that the bank did in fact sign. Since the bank controls the Internet banking application and BankID is controlled by the Norwegian banking community, the bank has access to a wealth of technical and judicial information, whereas the customer has only a copy of the digitally signed document on his computer. Furthermore, the bank has readily access to BankID experts, while the customer has only access to security experts without any inside knowledge of BankID. Consequently, since no third party has collected non-repudiation information to assist the customer during the conflict, the customer and his lawyers will find it very difficult to convince a judge that the bank really signed the document when it denies having done so.

Observation 6 *The non-repudiation service in BankID gives a bank an advantage over its customers during conflicts involving repudiation of digital signatures because the customers cannot rely on help from a trusted third party.*

5.2 Insecure local key storage on BankID server

A high degree of non-repudiation requires that it is very difficult for an outside cracker (or rogue insider) to obtain a web site's private key. Unfortunately, in some cases the private key is stored in an encrypted PKCS #12 file. A cracker may be able to obtain the fixed password used to decrypt the file since the password is vulnerable to dictionary and social engineering attacks. Once the cracker has the private key in the decrypted file he can sign documents without the knowledge of the key's rightful owner.

Observation 7 *Only web sites utilizing HSMs to store and use private keys can support a high level of non-repudiation.*

5.3 Central key storage

An end-user's private key is stored on the central BankID infrastructure controlled by The Norwegian Banks' Payment and Clearing Centre. According to [8], this key is only used inside an HSM. Since private keys must *only* be available to end-users [4, pp. 52, 93, 156] to

achieve a high degree of non-repudiation, the central key storage raises several questions. How are the private keys generated on behalf of the customers and placed in the HSM without anyone possibly learning the value of the keys? How can customers provide the fixed password to activate the authentication and signing functionality without anyone else obtaining the password? Is a network connection from a customer made directly to the HSM to avoid MitM attacks from rogue insiders?

During a dispute a bank must provide a judge with convincing answers to these questions. In particular, a bank must explain why a rogue insider isn't able to exploit the HSM's application programming interface, modify existing code, install new malicious code, and/or modify server configurations to get access to keys.

The banks have already been able to convince the Norwegian Post and Telecommunications Authority how only the end-users can grant the central infrastructure access to private keys inside an HSM. However, the authority has refused to share its reasons for accepting central key storage. This is unfortunate since the lack of information makes it very difficult for an end-user (or his lawyer) to challenge a bank's claims about the non-repudiation during a dispute.

Not surprisingly, the Norwegian banking community has also refused to share any information with us about the system they use for non-repudiation. As far as we know, the non-repudiation protocols have not been analyzed by independent security experts or tested in Norwegian courts.

Observation 8 *The security-through-secrecy policy of the BankID-member banks gives them an advantage over their customers during conflicts because the customers and their lawyers have no access to technical information about the non-repudiation service.*

5.4 End-user authentication limits degree of non-repudiation

If the end-user authentication in a PKI is too weak then it is possible for a skilled cracker to steal a user's digital identity. The cracker can then digitally sign a document using the victim's identity. It can be difficult for the unfortunate user whose digital identity was misused, to show that he didn't sign the document. If the PKI offers a non-repudiation service, this will only increase the problem for the unfortunate user because the other signee has access to "credible evidence" showing that the user did sign when in fact it was the cracker who misused the user's identity. Hence, strong authentication of users is needed to achieve a high degree of non-repudiation.

According to Observation 2, the strength of the end-user authentication in BankID is limited by the strength of the two-factor authentication utilized by the BankID-member banks. Furthermore, Observations 3 and 4 point out that the the two-factor authentication is vulnerable to well-known attacks. As a result, it is possible to steal users' identities.

Observation 9 *The end-user authentication in BankID limits the degree of non-repudiation. The strength of the authentication should be increased before customers digitally sign contracts concerning large-valued assets.*

5.5 Mitigating non-repudiation risk

From Observations 6 and 8, the non-repudiation service gives a bank an advantage over its customers because the customers cannot obtain technical information about the service or rely on help from a third party during a conflict. The Norwegian banks should release information about the technical and legal non-repudiation protocols. In particular, the banks need to publish their dispute resolution procedures. Only then will it be possible for the users to get an understanding of the true risk associated with the non-repudiation service. It follows from Observation 7 that web site owners wanting to use the service must invest in HSMs to store cryptographic keys. From Observation 9, the strength of the authentication should be increased to improve the level of non-repudiation.

6 Privacy risk

The meaning of the term *privacy* depends on the context in which it is used. We define privacy as the right of an individual to decide when and how sensitive personal information should be revealed. To get an understanding of the privacy risk associated with BankID, we'll consider how the system may be used to build customer profiles.

Let us first consider the situation where an end-user wants to authenticate a web site. During the authentication process the BankID client depends on the central infrastructure to verify the web site's X.509 certificate. Because the end-user has to authenticate to the central infrastructure and the certificate contains the web address of the site, both the end-user and the web site are uniquely identified by the central infrastructure.

The CAs in the central infrastructure have access to the personal information provided by all individuals wanting to become BankID end-users. Once they start using BankID, it follows from the above observation that the central infrastructure can record e.g. where end-users shop and which government agencies they have dealings with.

Because the end-users must enter their SSNs during the authentication process, it is possible to link information from BankID with information in other commercial and governmental systems such as credit reporting agencies and taxation agencies. Hence, assuming BankID becomes the prevalent tool for online authentication of individuals in Norway, it will be possible to build increasingly *detailed profiles* over time revealing business and personal relationships of more than 2 million customers.

The US National Research Council has published a report [15] stating that individuals should know what information about them is stored in a national computer system, how this personal information is made available to third parties, how the information is updated, and most importantly, how they can prevent disclosure of the information. Neither of these requirements are fulfilled by BankID.

Observation 10 *The Norwegian banking community controls an ID system with the potential to build detailed profiles of roughly half of the Norwegian population as long as the BankID authentication utilizes X.509 certificates and SSNs. The BankID customers don't know how their personal information is utilized.*

We remark that the BankID client leaks information. It can be downloaded by anyone with a computer. If you enter the SSN of a BankID customer into the client, then the client will list the banks for which the customer uses BankID for authentication.

6.1 Mitigating privacy risk

The BankID system should be reviewed by independent privacy experts before it is allowed to become a de facto national ID system. Identified weaknesses in the privacy protection should be carefully examined and mitigated. In the long run a new authentication procedure, not using X.509 certificates and SSNs, should be introduced to minimize the system's negative effect on the end-users' privacy.

7 Conclusions

In April of 2007—after the risk analysis was completed—we were informed by a BankID representative that changes had been made to the system to mitigate the risk associated with MitM attacks (Observation 3). These changes may also increase the level of non-repudiation (Observation 9).

At the time of writing, late April 2007, the risk to BankID customers is still significant because (i) BankID is particularly vulnerable to DDoS attacks at the application layer, (ii) combined DDoS/phishing attacks may empty out customers' accounts, (iii), the lack of

an independent third party in the non-repudiation service and the banks' security-through-secrecy policy gives them an advantage over their customers during conflicts involving repudiation of digital signatures, and (iv) the customers don't know how BankID utilizes their personal information. We recommend that steps are taken to mitigate (i)–(iv).

7.1 Who should own the remaining risk?

Even after a risk mitigation process is carried out there still is a residual risk associated with BankID. Hence, it is interesting to observe that if an outside cracker or rogue insider is able to empty out an account in a BankID-member bank, then the bank's responsibility is limited to one-hundred thousand Norwegian Kroner (about sixteen thousand US dollars) [9, p. 15].

If a bank is grossly negligent, then the limit doesn't apply. During a dispute it is up to the customer or his lawyers to establish that the bank has been grossly negligent. Experience shows that this is close to impossible since the Norwegian banks have long refused to share any technical information about their systems (see [2] for a further discussion of this point).

Since only the banks can strengthen the security of BankID, they should take the remaining risk and, thus, be liable for any loss caused by crackers and rogue insiders. It is then up to the banks to determine the best balance between investing in better security and simply covering the loss caused by fraud.

7.2 General recommendations

We make four general recommendations to limit the risk customers take when using a national PKI such as BankID. First, traditional secrets, e.g. PINs and passwords, should not be transmitted from a client to the central PKI infrastructure. Hence, two-factor authentication should only be used locally to give the client access to the end-user's PKI credentials.

Second, if a PKI is found to be vulnerable to well-known attacks such as phishing and MitM attacks, then immediate steps should be taken to mitigate the risk to the end-users. This is of particular importance when the end-users don't have a convenient alternative to the services offered by the PKI.

Third, the non-repudiation service in a national PKI should not be sanctioned by any government before independent lawyers and security experts have evaluated the legal and technical protocols and determined the true level of non-repudiation. The results of such an evaluation should be made public.

Four, no citizen in any country should be forced to use a national ID system before an analysis of the system's privacy implications is made public and any discovered weaknesses in the privacy protection are corrected. Robust safeguards against profile building should be in place before any ID system is accepted by a nation's government.

References

[1] K. J. Hole, V. Moen, and T. Tjøstheim, "Case Study: Online Banking Security," *IEEE Security & Privacy*, vol. 4, no. 2, 2006, pp. 14–20. 1, 4.1

[2] K. J. Hole, V. Moen, and A. N. Klingsheim, "Lessons from the Norwegian ATM system," *IEEE Security & Privacy*, vol. 5, no. 6, 2007, pp. 25–31. 1, 7.1

[3] A. Jones and D. Ashenden, *Risk Management for Computer Security*, Elsevier, 2005. 1, 4

[4] C. Adams and S. Lloyd, *Understanding PKI*, 2nd Edition, Addison-Wesley, 2003. 2, 3.1, 5.3

[5] W. Ford and M. S. Baum, *Secure Electronic Commerce*, 2nd Edition, Prentice Hall, 2001. 2.5

[6] S. T. Kent and L. I. Millett, Editors, *Who Goes There?*, National Academies Press, 2003. 3.7

[7] J. Zhou, *Non-repudiation in Electronic Commerce*, Artech House, 2001. 2, 2.5, 5.1

[8] The Norwegian Banks' Payment and Clearing Centre (BBS), "BankID FOI White Paper," Release 2.0.0, 2006 (in Norwegian). 3, 3.8, 5.3

[9] Bankenes Standardiseringskontor, "Norsk BankID sertifikatpolicy for banklagrede kvalifiserte sertifikater til personkunder," Version 1.1, 2005 (in Norwegian). 3, 3.3, 3.8, 7.1

[10] RSA Laboratories, "PKCS 12 v1.0: Personal Information Exchange Syntax," 1999. 3.4

[11] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service*, Prentice Hall, 2005. 4.1

[12] D. Geer, R. Bace, P. Gutmann, P. Metzger, C. P. Pfleeger, J. S. Quarterman, and B. Schneier, "Cyberinsecurity: the cost of monopoly," Sep. 2003; `www.ccianet.org/filings/cybersecurity/cyberinsecurity.pdf`. 4.1

[13] S. C. McQuade, *Understanding and Managing Cybercrime*. Pearson, 2006. 4.1

[14] R. E. Smith, *Authentication*. Addison-Wesley, 2002. 4.5

[15] S. T. Kent and L. I. Millet, Editors, *IDs—Not That Easy: Questions About Nationwide Identity Systems*, National Academies Press, 2002. 6