Robbing Banks with Their Own Software—an Exploit Against Norwegian Online Banks[*]

Yngve Espelid, Lars–Helge Netland, André N. Klingsheim, and Kjell J. Hole

Abstract The banking industry in Norway has developed a new security infrastructure for conducting commerce on the Internet. The initiative, called BankID, aims to become a national ID infrastructure supporting services such as authentication and digital signatures for the entire Norwegian population. This paper describes a man-in-the-middle vulnerability in online banking applications using BankID. An exploit has been implemented and successfully run against two randomly chosen online banking systems to demonstrate the seriousness of the attack.

Key words: Public-key infrastructure, man-in-the-middle attack, online banking

1 Introduction

The Norwegian banking community has created a new infrastructure for secure e-commerce, called BankID.[1] As of October 2007, BankID had more than 700,000 users. This number is expected to approach 2.5 million come 2009. At the time of writing, the infrastructure is mainly used for authentication of Internet banking customers, but BankID is extending into other markets, such as the government sector and e-commerce in general. It has also been used in conjunction with e-voting in some companies. BankID won a European prize, namely the *eema Award for Ex-*

Y. Espelid, L-H. Netland, A. N. Klingsheim, and K. J. Hole
NoWires Research Group
Department of Informatics
University of Bergen, Norway
e-mail: `{yngvee,larshn,klings,kjellh}@ii.uib.no`

[*] A short version of our work was presented at the conference Financial Cryptography and Data Security 2008 (FC '08) [10].

[1] Not to be confused with the Swedish BankID initiative.

cellence in Secure Electronic Business in 2006. Within a few years, the Norwegian banking industry wants BankID to become a nationwide identity system.

No detailed technical information about BankID has been released to the general public. Our request to see in-depth descriptions of the architecture and design was met with a non-disclosure signature prerequisite. Moreover, no publicly available independent third party evaluation of the system confirms that BankID meets a minimum of security and privacy requirements. This is worrisome due to a number of reasons: Firstly, a report by the US National Research Council [20] states that public review is essential when developing a nationwide identity system. The social costs of a poorly thought-out system are simply too high to justify.

Secondly, unlike in the US, liability has historically been assigned to the customer in disputes with Norwegian banks. In a previously analyzed court case [16], two expert witnesses turned the ruling in favor of a bank by claiming that the banking systems were very secure. No technical documentation was provided to support this claim.

Thirdly, the banking industry both owns the BankID infrastructure and provides financial services on top of the framework. It is not clear how potential conflicts of interest, involving the bank as a service provider and operator, will be resolved. Uncontested, the combination of no trusted third party and a security-through-secrecy policy could undermine the legal protection of Norwegian bank customers.

Finally, BankID relies on two-factor authentication with One-Time Passwords (OTPs), similar to earlier online banking systems. In 2005, Schneier warned that this form of authentication failed to address recent security challenges, such as phishing, identity theft, and fradulent transactions [23].

A previous paper [18] describes a risk analysis of the BankID infrastructure. In that study, the authors give an overview of the architecture and design of BankID, and pinpoint several weaknesses. Our work was done in parallel with the mentioned evaluation, and examines the therein suggested Man-in-the-Middle (MitM) attack in detail. A short paper on our work was presented at Financial Cryptography and Data Security 2008 [10].

The remainder of this paper is organized as follows: Section 2 provides a short overview of BankID; Section 3 looks at BankID from an adversary's point of view; Section 4 describes a MitM vulnerability in BankID that has been turned into an exploit; Section 5 explains how to make the attack more effective by capitalizing on a bank customer's trust in BankID; Section 6 describes our disclosure process; Section 7 suggests improvements to BankID; Section 8 presents related work; while Sect. 9 concludes the paper.

1.1 Definitions

Individual authentication, referred to as *authentication* for simplicity, is the process of establishing an understood level of confidence that an individual is who he or she claims to be [21]. If the level of confidence is high, the authentication is said to be strong. *Authorization* is the process of deciding what an individual ought to be allowed to do. A *vulnerability* is a weakness that violates the security goals of an information system. An *exploit* is a practical attack—in the form of detailed instructions or program code—utilizing a vulnerability. The attack must have been implemented and successfully run to constitute an exploit.

2 BankID Overview

BankID is modeled after an X.509 Public-Key Infrastructure (PKI), where the banks themselves own and operate the central infrastructure. PKIs have been studied extensively by the computer security community. In addition, many vendors provide PKI solutions in the commercial marketplace. Hence, there is a strong theoretical foundation for important PKI principles as well as extensive practical experience gained from implementing and running PKIs. A good introduction to PKIs can be found in [1].

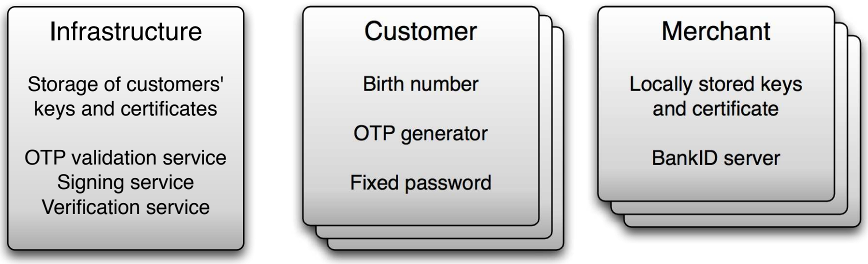

Fig. 1 Entities in BankID

Useful insights into BankID can be obtained from a white paper released by the BankID project [25], and by enrolling as a customer of the PKI. The system is built around three general entities: a central *infrastructure*, *customers*, and online *merchants*. The individual parties, their credentials, and duties are summarized in Fig. 1. An Internet bank is one example of a BankID merchant. In one of the participating banks, new customers sign up for BankID on the Web. Shortly after becoming a member, the customer receives an OTP generator and a fixed password by unregistered mail. When logging into the bank with BankID, customers use their Norwegian birth number[2] for identification, and an OTP in combination with the fixed password for authentication. People who sign up with more than one bank can choose freely among their OTP tools when using BankID. The PKI functionality provided by BankID is transparent to customers, as their private-public key pairs

[2] Norwegian birth numbers uniquely identify Norwegian citizens. These are similar to the US Social Security numbers.

are stored and controlled by the infrastructure. PKI services, such as creating and verifying digital signatures, are performed centrally on behalf of the users. Merchants store and control their own cryptographic keys and rely on server software distributed by the BankID project.

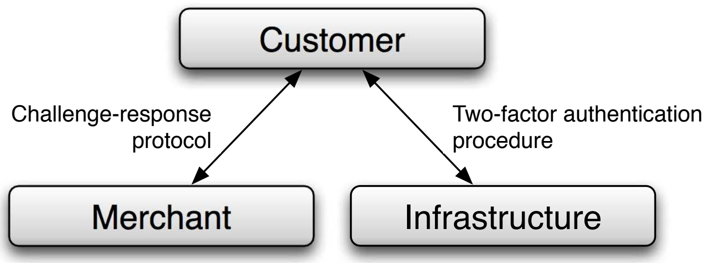
Customer
Challenge-response protocol
Two-factor authentication procedure
Merchant
Infrastructure

Fig. 2 BankID authentication procedure from the customer's point of view

The BankID design differs from a typical X.509 PKI, which requires private keys to be solely available to the entity identified in the matching public-key certificate [1]. Compared to textbook PKIs, BankID offers lower operational costs as the service providers don't have to roll out and maintain relatively expensive cryptographic hardware, but the design makes it harder to argue convincingly that a given private key can only be accessed by its rightful owner. The decision to store the customers' private and public keys on the infrastructure also results in an untraditional authentication protocol, that appears to be a hybrid of previous Internet banking schemes in Norway and X.509 PKI based authentication. Prior to BankID, authentication typically involved bank clients presenting the three customer credentials given in Fig. 1 to the banking system. The new design, depicted in Fig. 2, involves a longer protocol:

- The customer presents her birth number, OTP, and fixed password to the central infrastructure. This action unlocks PKI functionality on the infrastructure.
- The customer engages in a challenge-response protocol with the merchant. The infrastructure handles all PKI operations on behalf of the user.

A closer look at BankID's architecture and design can be found in [18].

2.1 The BankID Applet

A Java applet [24] is central in the authentication procedure depicted in Fig. 2. The applet is readily available from the central infrastructure.

The following scenario describes a typical BankID session involving bill payment for Internet banking customers. First, the client visits her bank's log-in page, which instructs her browser to download the applet from the central infrastructure.

The applet initiates the previously described authentication procedure. Upon successful authentication, an HTTPS session loads in the customer's browser. Now, the client can fill out payment details. Upon submitting the bill, the applet is reinitialized in the customer's browser, prompting her for credentials to sign the transaction. After submitting a fresh OTP and the private-key password, the transaction is processed. Next, the customer can continue the HTTPS session or terminate the bank session by logging out.

3 An Adverserial View into BankID

Attackers use a variety of tactics to break the security of software systems. A common strategy is to start by gathering information about the target. A detailed profile on an application allows attackers to apply their resources in the potentially most rewarding places. Upon mapping out the target, adversaries are wise to consider common vulnerabilities, as studies show that systems often fail for the same reasons. Common techniques used by attackers have been documented at length by the software security community, e.g. [4, 15, 19].

In terms of BankID, an interesting observation is the centralized key storage that contradicts advice given in the security literature. The resulting authentication protocol should draw the attention of attackers, because the development of new secure cryptographic protocols is a difficult undertaking. So much so that security experts strongly discourage the practice of "rolling your own" cryptography [26].

3.1 Reverse Engineering the Authentication Protocol

An inspection of merchant web pages reveals that the BankID applet is initialized by HTML parameters. One parameter specifies the address to the infrastructure server running the two-factor authentication procedure. Another parameter controls the location of the merchant server carrying out the challenge-response protocol. Consequently, all merchants can use the same applet by configuring these initialization parameters.

The applet initialization parameters can be altered so that the applet communicates with BankID software through a proxy controlled by an attacker. This allows adversaries to produce a blueprint of the BankID authentication procedure.

A walk-through of the customer authentication is provided by Fig. 3. The broken vertical lines represent the three main entities' lifelines. When entities interact, activation bars overlay their lifelines. The customer's activation bar represents the browser running on her computer. The grey and partially overlapping activation bar symbolizes the applet running in her browser.

Steps 1 through 4 in the figure show how merchants bootstrap the authentication process. The customer downloads an HTML page from the merchant, which

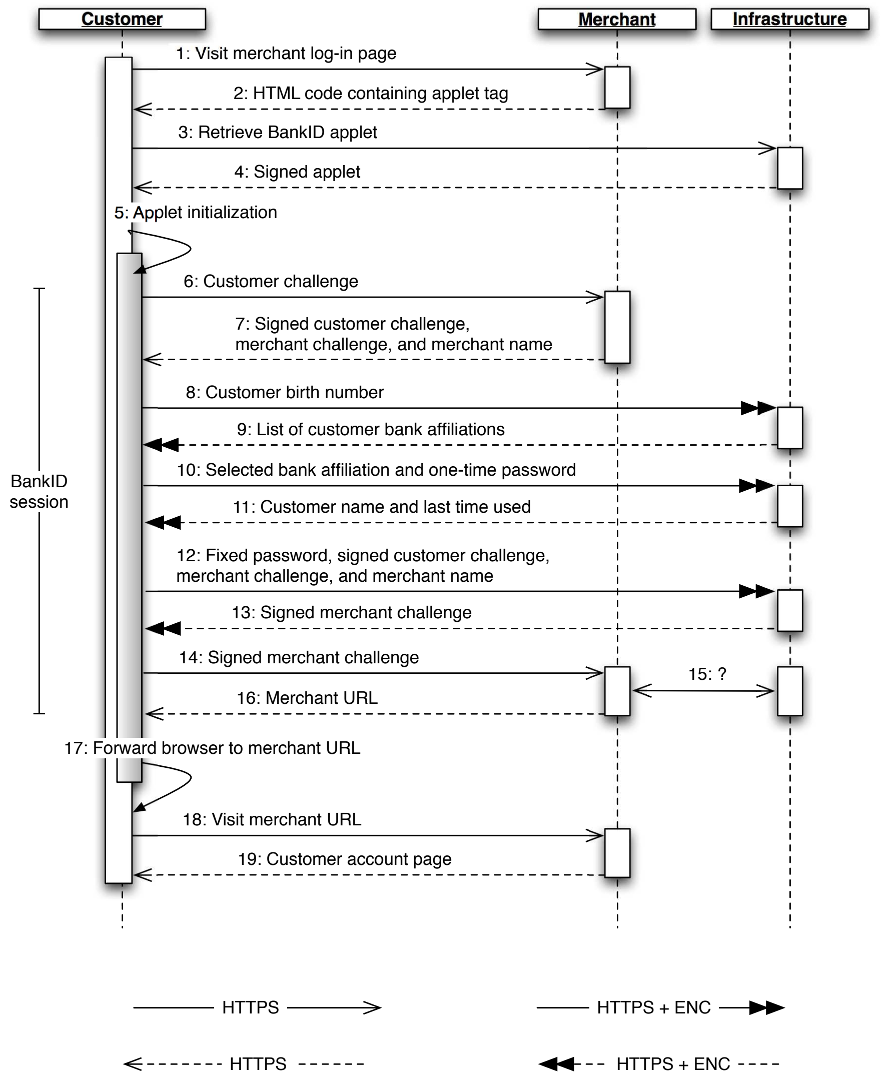
Customer
Merchant
Infrastructure
1: Visit merchant log-in page
2: HTML code containing applet tag
3: Retrieve BankID applet
4: Signed applet
5: Applet initialization
6: Customer challenge
7: Signed customer challenge, merchant challenge, and merchant name
8: Customer birth number
9: List of customer bank affiliations
10: Selected bank affiliation and one-time password
11: Customer name and last time used
12: Fixed password, signed customer challenge, merchant challenge, and merchant name
13: Signed merchant challenge
14: Signed merchant challenge
15: ?
16: Merchant URL
17: Forward browser to merchant URL
18: Visit merchant URL
19: Customer account page
BankID session
HTTPS
HTTPS + ENC
HTTPS
HTTPS + ENC

Fig. 3 Authentication process

instructs the customer's browser to retrieve the signed applet from the central infrastructure. These interactions run over HTTPS. The browser automatically verifies the applet signature, and prompts the user to trust the applet.

On customer acceptance, the applet is initialized in step 5. The applet is the catalyst for the authentication protocol and manages the BankID session, depicted as steps 6 through 16. Two stages, numbered in accordance with Fig. 3, initiate the challenge-response protocol:

6. The applet generates and sends a challenge to the merchant.
7. The merchant signs the customer challenge and generates a challenge to the customer. These challenges, along with the merchant name, are sent to the customer.

Steps 8 through 13 make up the two-factor authentication procedure necessary to unlock the customer's PKI credentials on the central infrastructure and sign the challenge. This communication has an additional layer of encryption, denoted ENC in Fig. 3, which denies the proxy the possibility to determine the content of that part of the protocol. Hence, the following dialogue is partially guesswork based on observations of network activity and the applet's response to customer input:

8. The customer inputs her birth number for identification.
9. The infrastructure returns a list of her BankID affiliations.
10. The customer chooses a bank and enters an OTP.
11. On valid OTP, the infrastructure returns the customer's name and the time when BankID was last used.
12. The signed customer challenge, the merchant challenge to sign, the name of the merchant, and the fixed customer password, are sent to the infrastructure.
13. The infrastructure verifies the merchant signature and uses the customer's private key to sign the merchant challenge, which it returns to the customer.

This concludes our guesswork. The applet now completes the challenge-response protocol with the merchant:

14. The signed merchant challenge is returned to the merchant.
15. We assume that the merchant and the infrastructure communicate to determine the customer's identity and verify the signature.
16. If the challenge-response protocol is successful, the merchant sends a URL to the customer.

Step 16 completes the authentication protocol, and the customer continues an HTTPS session with the merchant as illustrated by steps 17 through 19.

3.2 Reverse Code Engineering

Java byte code is easily reverse engineered and can reveal a program's inner workings to an attacker. When reverse engineering the BankID applet, we found the additional layer of encryption in steps 8–13 in Fig. 3 intriguing and made some interesting observations. As it turns out, three public keys belonging to the infrastructure are hardcoded in the applet. These are linked to different infrastructure services, namely OTP validation, signing, and verification.

In a customer request to the infrastructure, the applet generates a symmetric key used to encrypt the query. This key is encrypted with the service's public key and appended to the request. Using its private key, the service can decrypt the symmetric

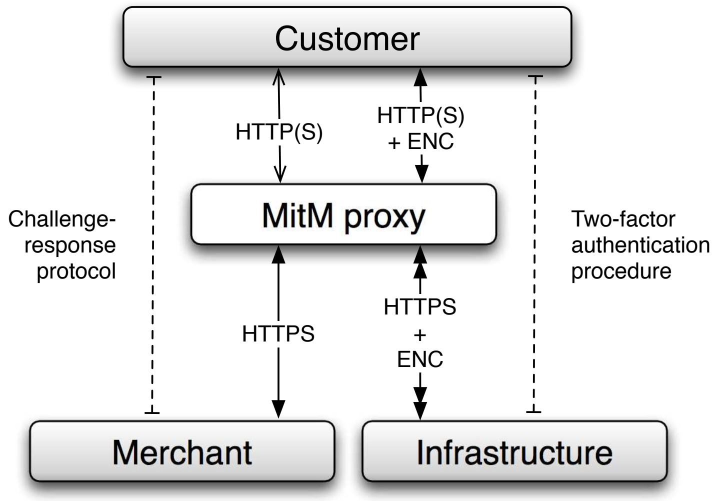

Fig. 4 The MitM proxy in the authentication protocol

key, and in turn decrypt the query. The same symmetric key is used to encrypt the response to the customer. A new key is created for each request.

Another observation from the code study is the use of the core Java class `SecureRandom`. This class provides a pseudo-random number generator. A study of the code segment generating symmetric encryption keys revealed a possible vulnerability. The running Java version affects whether the `SecureRandom` instance seeds itself or by an algorithm in the applet. Given Java version prior to 1.4, current time, amount of free memory, and loop counters are used to compute the seed. A cryptanalysis can determine the strength of both the seeding technique and the encryption algorithm. However, we chose to focus on session management issues in BankID.

4 An Exploit Against BankID

As mentioned in Sect. 3, by changing two initialization parameters, the applet willingly communicates—over either HTTP or HTTPS—with the MitM proxy depicted in Fig. 4. The proxy learns the communication between the applet and the merchant, which is sufficient to obtain an authorized session to the merchant. The attack is carried out through the following steps:

1. Trick the user into initializing the applet with malicious parameters.
2. Start the HTTPS session between the MitM proxy and the merchant to obtain a session ID (this identifier is not shown in Fig. 3.)

3. Relay the BankID session until the authentication completes.
4. Seize the HTTPS session to the merchant after the authentication is completed (step 16, Fig. 3.)

The attack can be explained in terms of session management. Conceptually, two sessions exist; a regular HTTPS session between the customer and the merchant, and the BankID session involving the infrastructure, the merchant, and the customer, as shown in Fig. 3. A discrepancy between these sessions enables the MitM attack. First, only the merchant server is authenticated in the HTTPS session, enabling the MitM proxy to initiate a session on behalf of the customer. Then the customer and the merchant are mutually authenticated through the BankID session, which is simply relayed by the MitM proxy. Finally, the authorization granted to the customer in the BankID session is transferred to the HTTPS session controlled by the MitM proxy, and the attack is successful (step 16, Fig. 3).

Note that the attack uses the signed applet from the infrastructure, turning it into an attack tool against BankID.

4.1 Proof of Concept

The previously described vulnerability was turned into an exploit against two randomly chosen Norwegian online banking systems in March 2007. Both attempts gave access to a customer account in these banks. The BankID community claimed to have fixed the problem in November 2007. A slightly modified version of the first exploit was successfully launched against BankID again in December 2007, using a version rollback attack. In short, an old version of the BankID applet was used to sidestep the countermeasures implemented in November 2007. The BankID community then introduced additional security measures in January 2008, thereby stopping our rollback attack.

5 Attack Considerations

The MitM attack can be bootstrapped in multiple ways, using well known attack strategies. Phishing attacks are already plaguing the banking industry, and can be used to trick some users into opening a webpage from the MitM proxy.

Nordic banks have been pestered over the last year by man-in-the-browser attacks [13]—trojan horses installed in web browsers. A trojan could change the parameters to the applet when the customer visits her online bank. This would be extremely difficult to detect for the average user.

5.1 Trust Management Capitalization

Trust can be defined as a positive expectation regarding the behavior of someone or something in a situation that entails risk to the trusting party [9, p. 77]. Risk is simply the possibility of suffering harm or loss. It is important to note that, unlike in the X.509 PKI specification [1], trust in this context is not a binary concept but involves many levels of trust. For any given user, there is a certain amount of trust that is needed to be willing to transact. Let this level be denoted the *cooperation threshold* [22]. In order to get the most out of the attack against BankID, an adversary wants as many customers as possible to reach the cooperation threshold. The BankID design helps achieve this goal.

Assume that an attacker decides to bootstrap the BankID attack with an e-mail phishing scheme. A press release from Gartner indicates that approximately 19% of phishing targets click on a link in a malicious e-mail, and that 3% give financial or personal information to phishers [11]. The numbers illustrate that the success rate relies firstly on the cleverness of the phishing e-mail, and secondly on the trustworthiness of the phishing site. We discuss our attack in conjunction with the latter.

Recall that the signed BankID applet is loaded unmodified from the BankID infrastructure. Hence, the user carries out a seemingly regular authentication procedure. The browser successfully validates the applet signature and displays a certificate belonging to the BankID infrastructure. This is ideal in winning the trust of customers, as they are carefully instructed to look for this when using BankID [5]. Furthermore, the user is presented with this information before the webpage is rendered. The user's attention is drawn to the applet—not to the webpage or the MitM proxy's URL in the address bar. Hence, the important first step towards the cooperation threshold is taken before the malicious webpage is shown to the user.

The next step is to present a webpage visually indistinguishable from the merchant's authentic webpage. The only indication of an attack will then be in the address bar of the victim's browser. Phishers use a variety of tactics to manipulate this bar. If a merchant website has a cross-site scripting vulnerability [19], the success rate of the attack could rise further by capitalizing on the customer's trust in the merchant's own website. Upon completing the authentication, the attacker assumes control of the real BankID session. The information sent to the customer after this point can be crafted so that the he still believes that the attack was a legitimate log-in attempt.

Norwegian banks currently use OTPs and fixed passwords to authorize transactions. Therefore, the attacker must collect at least one OTP and the password to transfer money out of the account. This can be achieved by alerting the user at the end of the log-in procedure that the previously entered fixed password and OTP were incorrect, after which the attacker asks for them again. Upon receiving the credentials, the attacker sends the customer one of the bank's standard error messages. This last step is a standard phishing technique. However, the customer has already reached the cooperation threshold and should take the bait.

6 The Disclosure Process

The discovery of the BankID MitM vulnerability and the subsequent exploit urgently called for countermeasures from the BankID community. Building on insights from the BankID risk analysis [18], our team needed less than a month to break into Norwegian Internet banks using BankID. Taking into account that the attack relies on techniques well-known to malicious hackers, it was reasonable to conclude that our attack posed a significant risk for BankID customers.

According to a survey [7], the world's largest software companies encourage some variant of *responsible disclosure* [8] when independent researchers find vulnerabilities in their products. Inspired by responsible disclosure, we informed major stakeholders in the BankID community, namely Bankenes Standardiseringskontor (BSK) and The Norwegian Banks Payment and Clearing Centre (BBS), about the MitM vulnerability in March '07. The Financial Supervisory Authority of Norway (FSAN) was also informed about the problem at this point. On request, BSK also received a technical description of how the BankID vulnerability could be turned into an exploit. They later responded that the vulnerability had been removed in January, i.e. before we developed the proof of concept code.

Unable to convince the system owners about the dangers posed by the exploit, we released the interim report "Next Generation Internet Banking in Norway" on May 16th [18]. This work points out weaknesses in BankID, and explicitly states that we had developed a proof of concept attack against BankID. Our discovery was reported by several media outlets, but did not spark a broad discussion around BankID as a nationwide identity system candidate.

In September and October '07 we demonstrated the MitM exploit for FSAN and a group of security experts with influence on the Norwegian banking industry, hoping that this would get the attention of the BankID community. A month later we distributed an early version of this paper to BSK, BBS, and the BankID coordinator. In November '07 we again told of the exploit in a large Norwegian newspaper [12]. In the subsequent debate, the banks claimed to have addressed our attack in a November patch, and questioned the lawfulness of our security testing. In early December '07 our attack caught the attention of members of the Norwegian Parliament, and was scheduled for a Question Time session.

By using a version rollback attack, the exploit was again successfully run against the previously vulnerable Internet banks on December 18th. Hence, the BankID community had spent eight months coming up with a fix that did not work. During this time period the banks neither seeked our counsel nor asked for another test. Having failed to establish a productive dialogue with the system owners, we decided to look at other alternatives for improving the situation.

Full disclosure of an exploit on a live banking system with close to one million users seemed drastic. Still, banking customers had for at least nine months believed BankID to be secure, while our exploit showed that the user community had been and continued to be exposed to an unnecessary high risk. In the end, we chose to revisit responsible disclosure, but this time with FSAN as a coordinator. In January

’08 we were informed that new countermeasures had been introduced in BankID to prevent our rollback attack. We have not analyzed these security measures in detail.

7 Possible BankID Improvements

Due to lack of complete information on the BankID system we can only give some general recommendations on how to improve the security through changes to the applet itself. In future versions of the BankID applet, the session discrepancy discussed in Sect. 4 should be corrected to mitigate the risk posed by our exploit. The applet needs to properly authenticate its communication peers, enabling it to detect a MitM proxy. Also, the applet must require end-to-end encryption when communicating with both the infrastructure and the merchant. To achieve these goals the applet can require HTTPS when connecting to the infrastructure and the merchant, and explicitly check that the authenticated server is the correct one. Input validation [19], such as a whitelisting approach, can be useful to avoid hostile communication points. We leave it to the BankID community to evaluate the feasibility of this approach.

In the long-term, the BankID community should evaluate the implications of moving to a traditional PKI where the clients possess their own private-public key pairs. The move would improve the strength of the authentication, and yield a simpler design. Also, several of the problems identified in [18] could be reduced or solved. Such a change comes with a cost. However, a national security infrastructure must fulfill minimum security requirements, including resistance to MitM attacks, and offer strong authentication. Many countries around the world have already put to use, or are contemplating national identity systems based on PKI and cryptographic smartcards. Hence, there are many experiences around the world that should be taken into consideration if the BankID community decides to offer a traditional PKI.

8 Related Work

A series of three articles analyze Norwegian banking systems [17, 16, 18]. The first paper shows that some Norwegian banks were vulnerable to a combined brute-force and distributed denial-of-service attack in 2003 and 2004. The authors go on to discuss the effects of the banks’ security-through-secrecy policy, concluding that it prohibits learning and causes the same mistakes to be repeated. The second paper elaborates on the problems with a bank’s non-disclosure policy in a Norwegian court case. The third paper contains a risk analysis of BankID from the customer’s point of view. It concludes that users of BankID are exposed to a number of significant risks that should be mitigated. Our attack builds on the above-mentioned article series and zooms in on weaknesses touched upon in the risk analysis of BankID [18]. In particular, our work further testify to the inefficacy of the banks' security-through-secrecy policy.

According to an IBM white paper [14], the two-factor authentication schemes widely adopted by online banks are insufficient in protecting against combined phishing and MitM attacks. An adversary first sends a phishing e-mail to the banking customer with a link to a proxy controlled by the attacker. If the victim takes the bait, the adversary plays an active role in the log-in process by relaying user input to the bank and the bank's responses back to the user. Upon completing authentication, the attacker can seize the session or mix fraudulent transactions with the users legitimate transactions.

Our attack uses elements of a combined phishing and MitM attack, but goes further by using the bank's own software, the BankID applet, to gain the victim's trust. The BankID attack starts as a phishing attack with a phishing e-mail to the bank customer, but continues with loading the unmodified and digitally signed BankID applet instead of a fake applet. By doing so, an adversary abuses a crucial point of trust in BankID. The unmodifiable applet, formerly a disadvantage to an attacker, becomes an advantage in terms of gaining the trust of banking customers. After finishing the applet log-in procedure, the attack procedes as in the combined phishing and MitM attack, where the attacker must trick the victim into supplying his OTP and fixed password.

In [2], Anderson argues that a false threat model was accepted, due to the lack of feedback on why British retail banking systems failed. In doing so, the financial industry developed increasingly complex systems to protect against cryptanalysis and technical attacks, when it would have been wiser to focus on implementation and managerial failures. Analyses of banking systems published after Anderson's initial paper underscore the observation that systems fail not because of inadequate cryptographic primitives, but rather design flaws and implementation errors [6, 3].

9 Conclusion

The security of the Norwegian banking industry's new PKI solution, BankID, was repeatedly broken in '07. A MitM attack enabled attackers to access customer accounts in two online banks. Our attack used techniques well-known to cyber criminals and was based solely on public information. An exploit was demonstrated for FSAN and a group of security professionals to highlight the severity of the problem.

BankID's untraditional design hinders the system from providing a high level of security. The decision to store customers' private-public key pairs in a central location has resulted in a weak authentication protocol. A redesign of BankID is called for if the system is to offer the intended degree of security.

Our exploit underscores the importance of independent evaluation of national systems. BankID's design flaw contradicts advice given by security experts, and should have been detected and resolved long before the system was put into production. An unbiased scrutinization of the infrastructure and its documentation by leading security analysts would most likely have identified the problem.

The BankID community needs to improve their risk management processes. Today, the system owners fail to identify and quickly resolve problems. This was demonstrated to us by the nine months it took the banks to address our initial exploit. The subsequent version rollback attack further testifies to the inefficacy of BankID's current risk management processes.

As BankID is now gaining serious momentum in Norway—and is pushed by the BankID community to become the main identity system in Norway—both government and citizens need a better perception of the true level of security. In light of our attacks and the findings in [18], a thorough analysis of BankID is called for. This could increase the trustworthiness of BankID in the long run. At the time of writing the gap is too big between the actual level of security, and how the BankID community describes their system (see `www.bankid.no`.)

9.1 Final Remark

We would like to emphasize that only BankID accounts belonging to members of the NoWires Research Group were used to develop and demonstrate the MitM attack. No accounts belonging to others were involved in any way during our work with this paper.

References

1. Adams, C., Lloyd, S.: Understanding PKI—Concepts, Standards, and Deployment Considerations, 2nd edn. Addison-Wesley (2003)
2. Anderson, R.: Why cryptosystems fail. In: ACM 1st Conference on Computer and Communication Security. Fairfax, VA, USA (1993)
3. Anderson, R., Bond, M., Clulow, J., Skorobogatov, S.: Cryptographic processors—a survey. Technical Report 641, University of Cambridge (2005). URL `http://www.cl.cam.ac.uk/~mkb23/research/Survey.pdf`
4. Andrews, M., Whittaker, J.A.: How to Break Web Software—Functional and Security Testing of Web Applications and Web Services. Addison-Wesley (2006)
5. BankID: Hva gjør kunden ved mistanke om at noe er galt? (2007). URL `http://www.bankid.no/index.db2?id=4066`. Last checked March 2008 (in Norwegian)
6. Berkman, O., Ostrovsky, O.M.: The unbearable lightness of pin cracking. In: Financial Cryptography and Data Security (FC). Lowlands, Scarborough, Trinidad/Tobago (2007). URL `http://www.arx.com/documents/The_Unbearable_Lightness_of_PIN_Cracking.pdf`
7. Biancuzzi, F.: Disclosure Survey (2006). URL `http://www.securityfocus.com/columnists/415`. Last checked March 2008
8. Christey, S., Wysopal, C.: Responsible vulnerability disclosure process (2002). URL `http://www.whitehats.ca/main/about_us/policies/draft-christey-wysopal-vuln-disclosure-00.txt`. Last checked March 2008

9. Cranor, L.F., Garfinkel, S. (eds.): Security and Usability—Designing Secure Systems That People Can Use. O'Reilly (2005)
10. Espelid, Y., Netland, L.H., Klingsheim, A.N., Hole, K.J.: A proof of concept attack against norwegian internet banking systems. In: Proc. Financial Cryptography and Data Security (2008)
11. Gartner: Gartner study finds significant increase in e-mail phishing attacks (2004). URL `http://www.gartner.com/press_releases/asset_71087_11.html`. Last checked March 2008
12. Gjøsteen, K., Hole, K.J.: Nei, ennå ikke trygg. Aftenposten (29. Nov, 2007). URL `http://www.aftenposten.no/meninger/debatt/article2126133.ece`. Last checked March 2008 (in Norwegian)
13. Gühring, P.: Concepts against man-in-the-browser attacks (2006). URL `http://www2.futureware.at/svn/sourcerer/CAcert/SecureClient.pdf`. Last checked March 2008
14. Gundel, T.: Phishing and internet banking security (2005). URL `ftp://ftp.software.ibm.com/software/tivoli/whitepapers/Phishing_and_Internet_Banking_Security.pdf`
15. Hoglund, G., McGraw, G.: Exploiting Software—How to Break Code. Addison-Wesley (2004)
16. Hole, K.J., Moen, V., Klingsheim, A.N., Tande, K.M.: Lessons from the Norwegian ATM system. IEEE Security & Privacy **5**(6), 25–31 (2007)
17. Hole, K.J., Moen, V., Tjøstheim, T.: Case study: Online banking security. IEEE Security & Privacy **4**(2), 14–20 (2006)
18. Hole, K.J., Tjøstheim, T., Moen, V., Netland, L., Espelid, Y., Klingsheim, A.N.: Next generation internet banking in Norway. Tech. Rep. 371, Institute of Informatics, University of Bergen (2008). Available at: `http://www.ii.uib.no/publikasjoner/texrap/pdf/2008-371.pdf`
19. Huseby, S.H.: Innocent Code. Wiley (2004)
20. Kent, S.T., Millett, L.I. (eds.): IDs—Not That Easy: Questions About Nationwide Identity Systems. The National Academies Press (2002)
21. Kent, S.T., Millett, L.I. (eds.): Who Goes There? Authentication Through the Lens of Privacy. The National Academies Press (2003)
22. Marsh, S., Dibben, M.R.: Trust, untrust, distrust and mistrust—an exploration of the dark(er) side. In: iTrust 2005, *LNCS*, vol. 3477, pp. 17–33. Springer (2005)
23. Schneier, B.: Two-factor authentication: too little, too late. Communications of the ACM **48**(4), 136 (2005)
24. Sun Microsystems, Inc.: Applets. URL `http://java.sun.com/applets/`. Last checked March 2008
25. The Norwegian Banks' Payment and Clearing Centre: BankID FOI white paper (Release 2.0.0) (2006). (in Norwegian)
26. Viega, J., McGraw, G.: Building Secure Software—How to Avoid Security Problems the Right Way. Addison-Wesley (2002)