Paper I: Vulnerabilities in Online Banks

Vulnerabilities in Online Banks

Thomas Tjøstheim and Vebjørn Moen

Abstract

This paper describes some attacks on online banks that authenticate each customer through the use of a unique user identifier and a Personal Identification Number (PIN). Many user identifiers contain structure which make them easy to generate on a computer. Given a generated set of identifiers it is possible to do a brute-force attack on the PINs. A general attack model is described and some example attacks against a Scandinavian online bank are discussed.

1 Introduction

Online banks have thrived with the explosive growth and availability of the Internet. A wide variety of services are offered to the customers. Paying a bill, checking the account balance, or applying for a loan can now be comfortably done from one's own home or office. However, the new possibilities introduced with Internet banking have also resulted in new security challenges. It is difficult to create both user friendly and secure Internet banking solutions. Can customers really trust that an attacker will not be able to break into their accounts?

Online banks claim that they are secure as they have many security features like firewalls, Public Key Infrastructure (PKI), Intrusion Detection Systems (IDSs), money auditing systems, and Secure Socket Layer (SSL). The average customer seems to be satisfied with the level of security in Internet banks. However, security is complex and not some magic potion that you add to your system to make it secure. Security should be considered from the start of the system development phase. A careful analysis of the environment is necessary to determine the needed security services and to determine how to implement these services correctly. Analysis of a security protocol is very difficult, due to the many ways an attacker can take advantage of the protocol environment.

In this paper we show that online banks authenticating each customer through an N-digit PIN in combination with a structured user identifier are vulnerable to both brute force and Denial of Service (DoS) attacks. There are at least three online banks in Norway that use or have used this form of customer authentication. However, the authors have decided not to explicitly name any banks, as this has been requested from one of the banks, and the fact that some of the banks are still vulnerable to the attacks described in this paper.

The rest of this paper is organized as follows: Section 2 presents the general attack model, Section 3 describes structure and generation strategies when Social Security Numbers (SSNs) and account numbers are used as unique identifiers, Section 4 discusses some example attacks on a real Internet bank in Scandinavia, and Section 5 concludes the paper.

2 Attack model

A common way of authenticating customers in online banks is to require a unique identifier for each customer together with a secret that only the customer knows. This section describes a general attack model against online banks where each customer has a unique user ID and an N-digit PIN as their secret. The PIN can be either static or dynamic. A static PIN stays the same while a dynamic PIN is changed for each login; it can for example be generated by a PIN calculator.

A customer gains access to an account only by entering a valid user ID and PIN combination. Typically, there will be a limit on how many times (often three or five) a wrong PIN can be entered for a given customer's user ID. The objective of this limit is to prevent a brute-force attack against the customer's PIN. A customer will temporarily lose access to the account if the limit is exceeded and must contact the bank in order to receive a new PIN.

2.1 Brute-force attack

Figure 1 depicts the attack model for the brute-force attack. The generated set of user IDs will contain a subset of the user IDs belonging to the customers of an online bank (we will see examples of how to generate such sets in Section 3). A program logs into the online bank's web pages and automatically enters different user ID and PIN pairs. One can observe that it does not matter if the PIN is dynamic or not. The only difference is that if a PIN is dynamic the attacker would have to attack the account at the moment a valid PIN and user ID combination is found.

Running the attack from only one host is not realistic, as this most likely will be detected by the bank's IDS. A distributed attack could be done by for example spreading a virus that contains the brute-force program in addition to having a control program that gets feedback from the zombie machines. Furthermore, it is possible to avoid the IDS by spreading the attack over several days in order to hide the number of tried logins in the anticipated natural traffic from legitimate users accessing the bank.

The probability of accessing at least one account is

$$\begin{aligned} P(\text{At least one}) &= 1 - P(\text{none}) \\ &= 1 - (1 - P_{success})^Y \end{aligned} \tag{1}$$

where Y is the number of online bank customers in the generated set of user IDs. The anticipated number of cracks is

$$\mu = Y \times P_{success}. \tag{2}$$

The probability that an attacker cracks a random customer's account is

$$P_{success} = \frac{X}{10^N}. \tag{3}$$

Here, X is the maximum number of allowed wrong entries for a PIN, and N is the number of digits in the PIN.

Given a generated user ID, an attacker must first consider whether it is valid and belongs to an online bank user, or not. An attacker would ideally like to maximize the amount of customer user IDs in the initially generated set. This would give the most effective and *silent* attack.

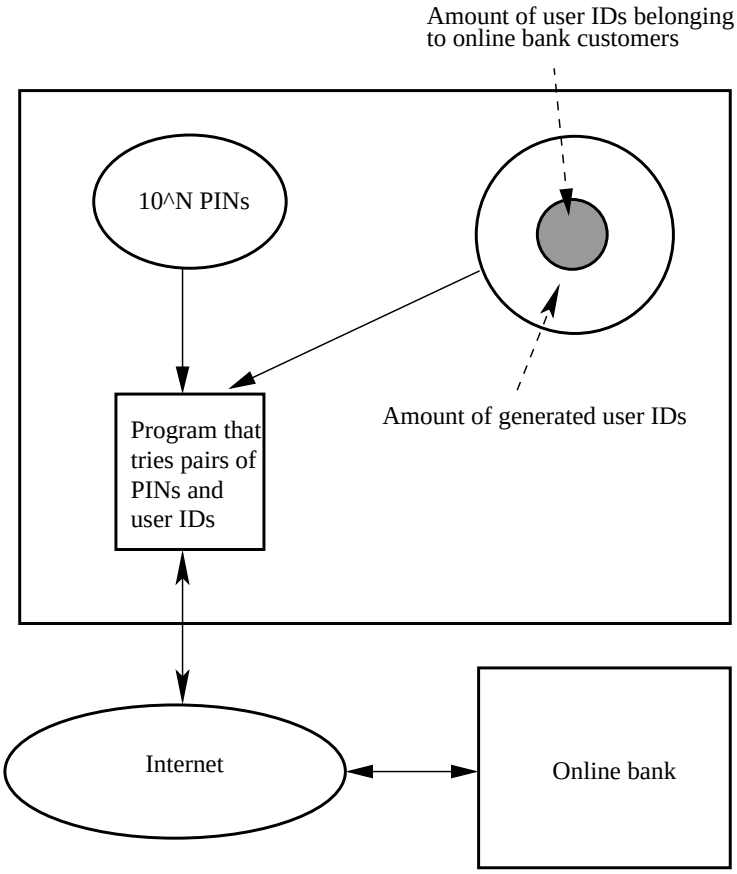

Figure 1: Attack model

2.2 DoS attack

If an attacker can acquire *many* user IDs, there is also a possibility of doing a distributed DoS attack against the bank's customers. The login scheme of online banks simplifies an application layer based DoS attack. An attacker can temporarily shut down access to accounts by entering X incorrect PINs for each valid user ID.

2.3 Combined DoS and brute-force attack

The probability in (3) assumes a combined DoS and brute-force attack. It can be discussed if the attacker is better off guessing $X-1$ times since customers are not denied access this way, and it might take longer before the bank's IDS detects the attack. However, if the attacker controls a network of "zombie" machines, it is very difficult to both identify the attacker and stop the attack from all the machines in the controlled network. The chaos created by the combined attack could also be an advantage for the attacker. For more information on how to execute a distributed DoS attack please consult [1].

3 Generating user IDs

This section describes two real cases of user IDs being used in online banks. We will study structure and generation strategies for Norwegian SSNs and account numbers. The arguments that apply to the Norwegian user IDs are similar for other countries. In particular, we have verified that the same generation strategies, with minor modifications, can be applied to both Swedish and Danish user IDs.

3.1 Norwegian SSNs

Norwegian SSNs are not confidential. Given a reasonable documented need, the SSNs can be requested from a national register. Many public institutions like hospitals, banks, and tax authorities have legal access to people's SSNs, but it can be difficult for private persons to argue the need for many SSNs. Therefore it might be better for an attacker to generate a set of SSNs. Most SSNs have a specific structure which make them easy to generate.

3.1.1 Structure of Norwegian SSNs

The Norwegian SSN [2] consists of 11 digits: $x_1x_2x_3x_4x_5x_6i_1i_2i_3c_1c_2$.

$x_1x_2x_3x_4x_5x_6$ is the birth date of the individual on the form *ddmmyy*.

$i_1i_2i_3$ is called the individual number and is used to separate people born on the same date. The national register distributes SSNs in the order they receive birth messages. They start with the highest available valid individual number for that day and proceeds downwards for each new birth message. The individual number is based on the century the person is born in, as shown in Table 1. It is also possible to distinguish boys from

girls by looking at i_3, which is odd for boys and even for girls.

c_1c_2 are control digits that are calculated as weighted sums of the first 9 and 10 digits, respectively.

$$\begin{aligned} c_1 &= 11 - (3x_1 + 7x_2 + 6x_3 + x_4 + 8x_5 + 9x_6 + 4i_1 + 5i_2 + 2i_3 \pmod{11}) \\ c_2 &= 11 - (5x_1 + 4x_2 + 3x_3 + 2x_4 + 7x_5 + 6x_6 + 5i_1 + 4i_2 + 3i_3 + 2c_1 \pmod{11}) \end{aligned}$$

If either c_1 or c_2 is calculated to be $10 \pmod{11}$ the SSN is discarded, and if c_1 or c_2 is equal to 11 then it is set equal to 0. Let's assume that c_1 and c_2 are approximately independent, then an SSN will be discarded with the following probability:[1]

$$\begin{aligned} p(c_1 \cup c_2) &= p(c_1) + p(c_2) - p(c_1 \cap c_2) \\ &= \frac{1}{11} + \frac{1}{11} - (\frac{1}{11})^2 = \frac{21}{121}. \end{aligned}$$

Individual number	Year in birth date	Born
500–749	>54	1855–1899
000–499		1900–1999
500–999	<55	2000–2054

Table 1: Correspondence between individual number and birth date

3.1.2 SSN generation strategies

How can an attacker maximize the ratio of customer SSNs in the initially generated set? Four different strategies will be discussed.

Strategy 1: The simplest strategy is to generate SSNs in such a way that all of the online bank's customers are covered. If for example the online bank only has customers in the age group 18–100, one could generate all possible SSNs for this group. With this scenario, all customers are born in the 20th century and are therefore given individual numbers in the range 000–499. Hence, for each day in the year we get 500 possible SSNs, but an estimate of 21/121 will be invalid numbers. The number of possible SSNs for people that are 18–100 is $500 \times 365 \times 83 \times (100/121) \approx 12.5$ million. Let Z denote the total number of customers in an online bank. The proportion of customer SSNs would then be

$$R_{customers} = \frac{Z}{12.5 \text{ million}}. \tag{4}$$

The drawback, from the attacker's point of view, is of course the huge number of SSNs that has to be checked. A large portion of the SSNs will neither belong to *real* people nor to online bank users. There is also a high probability for the bank's IDS detecting the attack because of the big workload.

[1] To control the assumption of stochastic independence, a computer program was written that generated all the possible SSNs for the 20th century and counted the number of discarded SSNs. The results from the computer program gave the probability estimate 21/121 down to the 5th decimal.

Strategy 2: A strategy for increasing the concentration of customer SSNs is to focus on a specific age group that has a high percentage of online bank customers. For example, 34 % of customers in pure online banks are males in the age group 26–35 [3]. The number of generated SSNs for this particular group would be $250 \times 365 \times 10 \times (100/121) \approx 754,132$. The ratio of customer SSNs would then be

$$R_{customers} = \frac{Z \times 0.34}{754,132}. \tag{5}$$

Strategy 3: However, one still has to generate a lot of SSNs belonging to non-existing people. This problem can be avoided by taking advantage of the chronological assignment of SSNs to newborn and immigrants. Instead of generating all valid SSNs for one day, it is possible to use population statistics to reduce the amount of generated SSNs. As an example, one can look at the period corresponding to males in the age group 26–35. There is an average of about 33,343 SSNs assigned per year for this group [4]. This gives an average of $33,343/365 \approx 91.4$ people per day. Let S be the number of assigned SSNs for a particular day. This number will of course vary from day to day. To get an idea of how S varies one can make the simplifying, but only approximately true assumption, that each day in the year is equally probable for the assignment of an SSN. This gives a binomial probability distribution with $n = 33,343$ (the average for a year) and the probability $p = 1/365$ (ignoring leap years) that a person is assigned an SSN for a particular day. The standard deviation for a random variable V having a binomial distribution is

$$Sd(V) = \sqrt{np(1-p)}. \tag{6}$$

From (6) we have $Sd(S) \approx 9.5$. To get an estimate of how S varies for each day one can construct an interval with ± 3 standard deviations. The probability that S lies in the interval is

$$\begin{aligned} &P(91.4 - 3Sd(X) \leq S \leq 91.4 + 3Sd(X)) \\ &\approx P(62 \leq S \leq 120) = 0.9974, \end{aligned}$$

since a a binomial distribution can be approximated with a normal distribution.

If the attacker generates 120 SSNs for each day, then:

$$P(120 \leq S \leq 120 + m)$$

is the probability that the attacker looses between 0 and m SSNs for that day. This probability is limited to ≤ 0.0013 since ± 3 standard deviations are used. The number of generated SSNs would be $120 \times 365 \times 10 = 438,000$. We can assume that almost all of the online bank's customers in the age group 26–35 are covered. An approximated ratio of customer SSNs is then:

$$R_{customers} = \frac{Z \times 0.34}{438,000}. \tag{7}$$

SSNs that do not belong to *real* persons can be minimized if the attacker for example generates 62 SSNs for each day. This way the attacker will loose some SSNs belonging to real people, but will with probability $P(S \leq 62 - m)$ which is ≤ 0.0013 generate between 0 and m too many SSNs for a particular day.

The number of generated SSNs would then be $62 \times 365 \times 10 = 226,300$. This will correspond to approximately generating 0.68 (226,300/333,430) of the total number of SSNs for the age group 26–35. If we assume a uniform distribution of online customers among the assigned SSNs, an estimate of the customer ratio is then:

$$R_{customers} = \frac{Z \times 0.34 \times 0.68}{226,300}. \tag{8}$$

Strategy 4: Another possibility is to filter out the SSNs belonging to online bank customers by trying to exploit response information from the bank's web pages. Two filtering examples are shown in Section 4.2.

3.2 Norwegian account numbers

An account number is a unique identifier for a customer's account, and can be generated in much the same way as an SSN. This is not hard when the structure is known.

3.2.1 Structure of Norwegian account numbers

A Norwegian account number [5] consists of 11 digits: $b_1b_2b_3b_4a_1a_2a_3a_4a_5a_6c_1$

$b_1b_2b_3b_4$ indicates which bank the account belongs to. Each bank has a set of serial numbers that identify the particular bank.

a_1a_2 is the type of account, e.g. a salary account or a high interest account. There is no standard for which numbers have to be used, each bank can define its own system.

$a_3a_4a_5a_6$ are digits that uniquely identify a customer's account. When a new account is created the smallest available 4-digit number is chosen.

c_1 is a control digit that is calculated as a weighted sum of the first 10 digits: $c_1 = 11 - (5b_1 + 4b_2 + 3b_3 + 2b_4 + 7a_1 + 6a_2 + 5a_3 + 4a_4 + 3a_5 + 2a_6 \pmod{11})$

However, if c_1 is calculated to be $10 \pmod{11}$, the account number is discarded.

3.2.2 Account number generation strategies

The strategies for generating account numbers are easier than for SSNs. An attacker can find out which serial and account type numbers a particular bank uses. Given one of the bank's serial numbers and an account type, an attacker can generate the next four digits $a_3a_4a_5a_6$ in such a way that it gives a valid account number. Note that there are only $10,000 \times 10/11 \approx 9,090$ valid combinations.

An attacker can also take advantage of the fact that the smallest available account number is always chosen. It is also likely that an attacker can filter out valid account numbers by guessing incorrectly X times for a given account number and observing the response. Given this, it is possible to generate an interval of account numbers that the attacker knows belongs to customers.

4 Example attacks on a real Internet bank

Let Bank B denote the Norwegian branch of a Scandinavian bank that specializes in online banking services. The security solution was changed in 2004. In this section we will look at some theoretical example attacks on the Norwegian bank B, both before and after the change of security solution.

4.1 Bank B before 2004

A customer in bank B is supposedly authenticated by having a valid SSN, a $N = 4$ digit PIN, and a *personal* certificate. A new certificate must be downloaded for each new host used to connect to the bank. Before 2004 a customer downloaded a new certificate by entering a valid PIN and SSN pair. With this scenario an attacker could try to brute force the PIN by attempting to download a new certificate. An attacker had ($X = 3$) tries at guessing the correct PIN.

4.2 Brute-force attack

How does the brute-force attack described in Section 2.1 apply to bank B? Given the first strategy in Section 3.1.2, all SSNs for people in the age group 18–100 are generated. It is realistic to assume that nearly all of B's customers are covered in this SSN generation. In Norway, bank B had more than $Y = 220,000$ customers in 2003. From (1), the following probability can then be obtained

$$P(\text{At least one crack}) = 1 - (1 - \frac{3}{10^4})^{220,000} \approx 1,$$

and from (2), the anticipated number of cracks are

$$\mu = 220,000 \times \frac{3}{10^4} = 66.$$

The ratio of customer SSNs is from (4) $220,000/12.5$ million ≈ 0.018.

The second strategy was to only generate SSNs belonging to males in the age group 26–35. The expected number of B's customers in this age group is $Y = 220,000 \times 0.34 = 74,800$. This gives the following probability:

$$P(\text{At least one crack}) = 1 - (1 - \frac{3}{10^4})^{74,800} \approx 1.$$

The anticipated number of cracks when checking all SSNs one time is

$$\mu = 74,800 \times \frac{3}{10^4} \approx 22.$$

From (5), the ratio of customer SSNs is $74,800/754,132 \approx 0.099$.

The third strategy was to exploit the chronological ordering of SSNs combined with the use of population statistics. We apply the strategy to both the age group of 26–35 and 18–100. Table 2 shows some different results when the average number of SSNs ± 3 standard deviations is generated per day for the two groups. In particular, the table lists the total number of SSNs generated and the approximated number of B's customers covered.

In Section 3.1.2 a binomial probability distribution was assumed, and the two cases of generating 120 SSNs and 62 SSNs each day for the group 26-35 were

Strategy 3						
Variation	**SSNs per day**	**Age**	**Total SSNs**	**≈ B SSNs**	**Ratio SSNs**	**Cracks**
1	148	18–100	4,483,660	220,000	0.049	66
2	84	18–100	2,544,780	160,180	0.063	48
3	120	26–35	438,000	74,800	0.171	22
4	62	26–35	226,300	50,864	0.225	15

Table 2: Overview of results for strategy 3, when the average number of SSNs ±3 standard deviations is generated each day for the two age groups

considered. In the first case one can expect to almost cover all of the 74,800 customers in B and obtain about the same probabilities as when we generated all the valid SSNs for the same age group. The ratio (7) of customer SSNs would be $74,800/438,000 \approx 0.171$. With 62 SSNs generated each day the following estimate of the ratio of B SSNs is obtained from (8) to be $50,864/226,300 \approx 0.225$.

The birth statistics [4] for men and women in the age group 18–100 show that there is a total of 3,495,131 people (SSNs) and this gives an average of $3{,}495{,}131/83 \times 365 \approx 115.4$ per day. The standard deviation is calculated from (6) to be ≈ 10.7.

From Table 2 we see that the anticipated number of accounts cracked is dependent on the number of SSNs generated. There are different attack variations depending on how the bank will react to the attack. For instance, the most effective attack would be to try and verify the 226,300 SSNs generated with variation 4 in Table 2. Depending on how B would react to the first attack, the same attack could be repeated with about 15 anticipated cracks each time. On the other hand, if the attacker only gets one chance at verifying the SSNs and the number of SSNs does not matter, then variation 1 in Table 2 yields the best attack.

The fourth strategy was to try to filter out the SSNs belonging to online bank customers. Two different approaches were discovered for the case of bank B:

1. When a valid SSN is combined with a wrong PIN the following error message is returned: “You have entered the wrong SSN or PIN." After three incorrectly entered PINs, and only if this is an SSN belonging to a customer, will the bank respond that the customer has been denied access to the bank. This means that an attacker can guess three times for each SSN, and not only find valid PIN and SSN combinations, but also filter out which of the SSNs belong to B’s customers.

2. It is also possible for an attacker to filter an SSN by trying to register a new customer. When registering, bank B only verifies the correspondence between the SSN and the name. Fake email, phone numbers and so on can be entered. Only when entering an SSN that belongs to a customer will a specific error report be sent: “There was en error with registration. Please contact. . . " Otherwise the person with this SSN will be registered as a new customer. A disadvantage is that an attacker will register a *large* amount of new customers and this will probably be detected. The

advantage of this method compared to the first is that an attacker can filter the SSNs before executing the brute-force attack.

4.3 Bank B in 2004

The certificate downloading scheme in bank B was altered in 2004. In addition to entering a valid PIN and SSN combination, a customer must also enter a valid *one-time password* that is sent either to the customers' mobile phone or mailbox. If the password is sent as an SMS it has 15 minutes validity, while a password in the regular mail is valid for 14 days.

The brute-force attack described in Section 4.1 is still possible. An attacker that finds valid PIN and SSN combinations can decide how the one-time password shall be delivered. If the password is delivered by SMS, the attacker could try to *sniff* the password with an interceptor [6]. However, it is much easier and cheaper for the attacker to get the password from the mailbox. Given a valid SSN it was possible to find the matching name and the address. This could for instance be done by using a Norwegian pension fund web site [7]. This site authenticates members using only SSNs. When a member logs in, the site displays the name and address corresponding to the SSN. If the attacker chooses password delivery by mail, he decides when the password shall be delivered, and will have a good indication of when to steal the mail.

5 Conclusions

This paper shows that online banks authenticating customers through the use of a PIN in combination with an SSN or an account number are vulnerable to both brute-force and DoS attacks at the application level. The degree of exposure to brute-force attacks depends on the number of digits in the PIN. Whether the PIN is static or is changed for each login makes no difference in this case.

In Section 4 it was shown that bank B is vulnerable to a brute-force attack as the PIN only has 4 digits. The PKI solution in bank B is of limited value, since a new certificate and corresponding private key can be downloaded given a valid PIN and SSN combination and the one-time password.

An easy countermeasure against the attacks described in this paper would be to use user IDs that do not contain any structure, so that they would be difficult to generate automatically. The reason the banks have not done this, might be that it simplifies customer handling to use SSNs or account numbers as unique customer identifiers. Another suggestion is a fully functional PKI solution that require customers to meet in person with the Registration Authorities (RAs) when opening an account. This would enable stronger user authentication and possibly solve many of the vulnerabilities in online banks.

Much of the security in online banks relies on IDSs and money auditing systems. The problem with this approach is that it deals more with detection than attack prevention. A combined brute-force and DoS attack is hard to protect against with the current security schemes. The online banks have little other protection than temporarily closing down service for customers. The potential damage to the bank's reputation and the loss in revenue could be substantial.

References

[1] J. Mirkovic, S. Dietrich, D. Dittrich, and P. Reiher, *Internet Denial of Service, Attack and Defense Mechanisms.* Pearson Education, 2005.

[2] E. Selmer, "Personnummerering i Norge: Litt anvendt tallteori og psykologi," *Nordisk matematisk tidskrift*, 1964, (in Norwegian).

[3] M. Hjorth, "En kartlegging av nettb@nkkunders holdninger," Master's thesis, University of Bergen, 2002, (in Norwegian).

[4] Statistics Norway, "Statistikkbanken," last visited: August 29th 2005. [Online]. Available: http://statbank.ssb.no/statistikkbanken/

[5] European Committee for Banking Standards, "Norway - domestic account number," 2003. [Online]. Available: http://www.ecbs.org/Download/TR201/norway.pdf

[6] Spylife.com, "Cellular telephone interceptor," last visited: August 29th 2005. [Online]. Available: http://www.spylife.com/digital_interceptor.html

[7] S. Pensjonskasse, "Elektronisk lånesøknad," last visited: August 30th 2005. [Online]. Available: http://www.pensjonskassa.no/elaan/Elaan