How Smart Is Your Phone For Banking

Author: Anna Karlsson, (annka673@student.liu.se) and Emil Helg (emihe386@student.liu.se)
Supervisor: Nahid Shahmehri, (nahid.shahmehri@liu.se)
Linkopings university, Sweden

Abstract—This report is a study of the security provided by the mobile apps that the banks in sweden have developed, in comparison to the security they provide to the banks websites via web browsers. This report came to the conclusion that while the security implementation of the banks may be somewhat old and contain some known vulnerabilities, for the average user these security vulnerabilities would not be of much consequence.

1 Introduction

This project is done as a part of the course TDDD17 Information Security, held during the spring term of 2014 at IDA (Department of Computer and Information Science), Linkping University.

1.1 Purpose

There has been a very fast development on applications for smartphones in the past years and several banks offer apps for their costumers, to use some of their services directly on our phones. The purpose of our project is to analyze the security of Swedish bank apps to see how well these apps deal with information regarding transactions. We have also noticed that the authorization process on the apps is simpler than on the different banks websites and we want to know how this affects the security. We also want to have a somewhat understanding of what the banks do when these systems get hijacked.

1.2 Problems/questions

We are going to look at some of the banks in Sweden and the services they provide, while trying to get an understanding of how security is handled by them.

- How does division on the server differ for access of information between the app and the website.
- What does the banks do, so that the information the user can access signed in on their webpage is not available on the app?
- Are there different solutions for different banks?
- Are the banks security solutions open design or secret?
- Which cryptographic methods do the banks use?
- What do the banks do to maintain confidentiality and integrity of data in their systems? Which method(s) are they using?

2 Background

This section contain a brief explanation of key concepts used. But also briefly describe the practical and theoretical methods we will make use of.

2.1 Key concepts

2.1.1 Wireshark

Wireshark is an analyzer tool for network protocol, which makes it possible to do a deeper inspection of the packets being sent over a network, such as which protocols are being used and what information are they carrying. This is done by firstly live capture of the traffic and then analyzing it offline. [1]

2.1.2 Cryptography

Designing the systems for communication over insecure channels and the problems that is related to that is called the study of cryptography. So basically you can say that it is different ways for sending plaintext message by encrypting the message so it becomes a ciphertext, which is unreadable without decryption, and then send it. When the receiver then gets the ciphertext, it gets decrypt into the original plaintext message. We are going to discuss some cryptographic protocols in the result of the report and these are explained below. [2]

Symmetric-key cryptography
Techniques using a symmetric cryptographic key means that the same key is used for encryption and decryption of the message when sending and receiving information. This means that the key must be a secret for everyone except the sender and receiver. So when using a symmetric key this includes first sending the symmetric key in a secure way so no one else gets it. For this, there exists different key exchange protocols.
Symmetric keys can be implemented, either as block cipher or stream cipher. Block cipher encrypts a block of the message at a time and stream ciphers encrypts characters one by one. [2]

Asymmetric-key-cryptography
Asymmetric cryptography uses a public key for encryption and a private key for decryption. This means that if someone lets call him Bob wants to send a message to Alice, meant for only Alice to read, Bob uses Alice's public key to encrypt the message, and

when Alice receives the message in ciphertext, she uses her private key to decrypt it. Public-key systems are also used in the opposite direction for digital signatures which means that the reciever can verify that the person sending the message is the one it claims to be. [3]

RC4

Is a stream cipher symmetric encryption technique, which was designed in 1987 by Ron Rivest. It is used in many applications and protocols such as WEP and TLS. RC4 is very fast to encrypt data and is therefore good for large data. [4] But RC4 has some weaknesses and in some implementations can lead to insecure systems such as in WEP, there has also been speculations about weaknesses in the TLS implementation as well and last year Microsoft recommended to disable RC4 in systems where it is possible. [5]

AES

AES stands for Advanced Encryption Standard and is a symmetric block cipher designed by Joan Daemen and Vincent Rijmen. AES can be done with different sizes of the blocks such as 128, 192 or 256 bits. AES is also a fast encryption and is supposed to replace RC4 and is standardized by NIST (National Institute of Standards and Technology). Since AES is fast it is good for implementation in small devices and smart cards. [6]

Cryptographic Hash Functions

Hash functions are functions that make a bit string with fixed-sized from a block of data. For crypthographic hash functions this bit string is the cryptographic hash value of this function and should be made so that a change to the origin data return a different hash value and the reciever can thereby detect that modification of the data has been made. The functions that is the standard today has different probabilities of collisions (you want a low possibility of collision for a secure hash function). In this report we are dealing with MD5 and SHA where there has been detected collisions in MD5 but in SHA-1 which is the most common used SHA version there is only theoretical collisions and no one has manage to make it in practice. [6]

2.1.3 Certificate

Certificates are electronic documents containing a digital signature and a public key. Certificates are used for authentication, in which they are used to verify that the user are who he claims to be. There are so called PKI (Public Key Infrastructure) where a CA (Certificate Authority) signs the certificate. This is mostly used with servers sending their certificate to the clients that connect to them. This allows the clients to back-track the certificate chain to a CA, which proves the identity of the server. The most used standard is X.509. [3]

2.1.4 TLS/SSL

TLS (Transport Layer Security) and SSL (Secure Sockets Layer) are cryptographic protocols that are used for secure communications over the Internet and are using asymmetric cryptography to exchange symmetrics keys. This asymmetric cryptography is used since TLS/SSL uses X.509 certificate. So sites using TLS/SSL must have a certificate by a CA for implementation. There are three types of TLS, they are: 1.0, 1.1 and 1.2 from 1999, 2006 resp. 2008. [3], [7], [8], [9]

2.1.5 BankID

BankID is Swedens leading electronic identification and is based on the technical standard PKI. The development of BankID has been made by a number of large banks and is used by companies, authorities and in public. The creators of BankID include Danske Bank, Ikano Bank, Lnsfrskringar Bank, Nordea, SEB, Skandiabanken, Sparbanken Syd, Sparbanken resund, Svenska Handelsbanken, Swedbank and landsbanken. In 2011 BankID launched a secure identification for mobile phones and tablets called Mobile Bank ID. [10]

2.2 Theoretical Methods

The theoretical standpoint of this paper, will be based on research articles we have found related to mobile bank applications and their security, the banks user policy stated on their websites and a interview with a staff member on one of the banks.

2.3 Practical Methods

The practical part of this report, will be to analyse the packets being sent from the mobile apps and web browsers to the banks different servers. Our way of doing this will be to use wireshark to ”sniff” the packets coming from our smartphone to see what information can be inferred, for example, if we can see which protocol is used, which encryption is used and overall what information we can read from the data being sent. We will also use wireshark to compare how data and use of protocols might differ when using the app on the phone to access the bank from when accessing the banks website via a web browser.

3 RESULTS

This section contains the results from the banks different policies, the interview and the data we could inferre from the systems being tested.

3.1 Theoretical results

3.1.1 What does the banks do in case of hijacking

When we read the terms of agreement a user has to accept to use the banks services online, it does not look so promising. Since most of the banks we have read about basically writes that they dont take any responsibility of the information when using their site. However when talking to the banks, they say that a user get the same insurance as if he lost his credit card and someone else was using it. But in worst case it can lead to a court case and the user may not get his lost money back, which would be the main concern for an average user. But since it nowadays is a very hard competition between the banks about the customers and neither of them want to look bad in the press, this means that in most cases this would not be such a big of an issue and the banks will compensate the loss.

The interview on the other hand gave not that much. While promising at first we are still waiting, close to 3 weeks later, for the interview form we sent in.

3.2 Practical results

To ease the way to read the packets being sent to and from our smartphones, we set up an access point on one of our computers. The idea was that this would allow us an easy way to read the packets being sent, since we could then listen to the interface on our computer. To further more speedup the setup we used the program connectify, which allow a fast and easy setup of a sufficient wireless network using the computer with the program as an access point. [11]

A problem with this setup was that connectify transferred the packets from the wireless interface to the wired and vice versa, before wireshark was allowed to read the packets. To solve this we listened to the wired interface for the packets being sent, and the wireless interface for the packets received.

To filter the packets, we found that all the securely sent and received packets was sent using an SSL connection of some kind, this allowed us to filter the packets regarding only this to decrease the packets we had substantially.

These are the noticeable results we got from reading the information found in the packets sent and received, which comes from the server hello packet (or an extension of this) since it defines the encryption

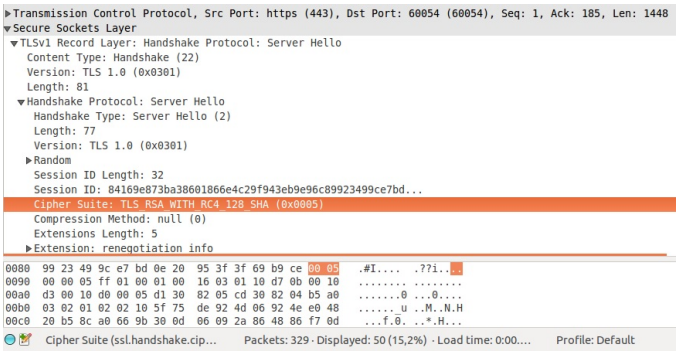

Fig. 1. A server hello packet

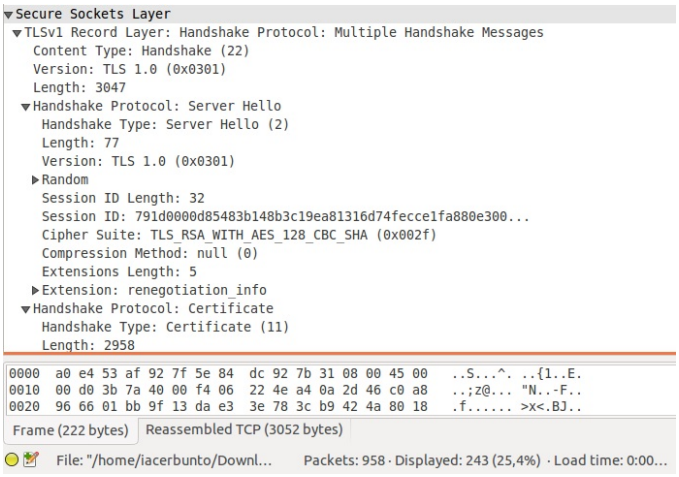

Fig. 2. A server hello packet

algorithms, hash function and more, that will be used throughout the session.

- 100% of the mobile apps uses the TLS v 1.0 protocol
- 75% of the web sites uses TLS v 1.2 while 25% uses TLS v 1.1
- 75% of the mobile apps and web sites uses SHA while 25% uses MD5
- 100% of the mobile apps and web sites uses RSA certificates
- 75% of the mobile apps and web sites uses RC4 while 25% uses AES
- 12.5% of the web sites uses ECDHE for key agreement/exchange and 82.5% uses RSA
- 50% of the banks share the first 24 bit ip addresses on the server which we connected to using either the mobile app or the web browser

We got these results while using Firefox v.28, Google Chrome v.34.0.1847.131, Android 4.2.2 and Android 4.4.2.

At the figures (Fig. 1 and 2) we can see two of the different server hellos sent by different banks. Fig. 1 shows a TLS connection which have been decided to use RC4 for encryption and decryption, and using SHA as hash function. While Fig. 2 shows a TLS connection

which have been decided to use AES for encryption and decryption, and also using SHA as hash function

3.3 Evaluation and Comparison

The method we applied to find out what kind of secure communications the banks make use of seems appropriate. What would be better though, would be if more banks web sites and mobile apps would have been tested. The results from the packet sniffing may also be somewhat biased, since factors such as web browser and operating system on the smartphone, will or can influence the result. In this case we believe these factors to not have influenced the result, based on what kind of library the web browsers and operating system have implemented. Also this report in no way checks the real strength of the implementations of the protocols used. We base the strength in security only on the protocols used.

4 Related work

4.1 Personal banking apps leak info through phone

In January this year Ariel Sanches published his result of an research he made on the information that banking app leaks information through the users phones. Sanches performed the research on home banking apps of the most influential banks in the world and he made static analysis and black box testing using iPhone/iPad devices. In the result he showed a number of weaknesses in the home banking apps such that, as many as 90% of them doesnt have SSL-links and no jailbreak detection. Half of them were also vulnerable to XSS (Cross site scripting). Sanches final conclusion were:
As this research shows, financial industries should increase the security standards they use for their mobile home banking solutions. [12]

5 Conclusions

Based on our findings both theoretical and practical, we have concluded that some of the systems being used are somewhat old, and may suffer from some security vulnerabilities. Since many of the banks uses RC4 which is an old protocol for encryption/decryption and suffer from known vulnerabilities, and especially when not properly implemented as can be seen in WEP, we regard this as somewhat of a blunder. We would suggest that the banks improve what encryption/decryption protocol they use with the TLS protocol.

That many of the banks also uses TLS 1.0 when android seems to allow use of TLS 1.2 is something we also are sceptical about. It may be that the banks believe that this impelemntation increases the amount of smartphones that are able to make use of the application, but we believe that this may not be the best practise in this regard.

Also that MD5 is used as hash function in some cases, when MD5 is known to contain many vulnerabilities which have been proven both theoretically and practically, does not really build any trust that this services should be regarded as secure. We are not completely sure which version of SHA the rest of the banks use, but we suspect that they are using SHA-1, which are known to have collisions, and should therefore be upgraded to SHA-2.

The sharing of the first 24 bit for the servers on some of the banks, could imply that these banks may have servers running which both the mobile app and the web server connects to. But this has to be proven empirically, since the banks may just have the servers behind the same router, but providing different services. Also while only the first 16 bit is the same for the rest banks, this in no way proves that they may not have servers which both the web server and mobile app connects to. But since the servers we connected to had separate ip addresses we are under the impression that the banks have different servers handling the mobile app and web browser connections.

While the banks approach is very forthcoming to the customers, when technical data about the system is asked about, they was not as forthcoming about providing the information. This implies that the banks dont follow open design, but instead uses security by obscurity.

While we may sound very sceptical about the services provided. For the average user the systems in combination with the safety-net the banks have for users, makes the experience for the user when a problem arises quite nice in regard that most if not all the lost money, which would be the main concern for the user, will be replaced by the bank.

REFERENCES

[1] (2014) The wireshark website. [Online]. Available: http://www.wireshark.org/about.html

[2] W. Trappe and L. Washington, *Introduction to Cryptography with Coding Theory*. Pearson Prentice Hall. [Online]. Available: http://books.google.se/books?id=BP96MG-uZDwC

[3] M. Bishop, *Computer Security: Art and Science*. Addison-Wesley, 2003. [Online]. Available: http://books.google.se/books?id=pfdBiJNfWdMC

[4] P. Prasithsangaree and P. Krishnamurthy, "Analysis of energy consumption of rc4 and aes algorithms in wireless lans." in *GLOBECOM*. IEEE, 2003. [Online]. Available: http://dblp.uni-trier.de/db/conf/globecom/globecom2003.htmlPrasithsangareeK03

[5] J. Leyden. (2003, 14th of November) Microsoft, cisco: Rc4 encryption considered harmful, avoid at all costss. [Online]. Available: http://www.theregister.co.uk/2013/11/14/ms_moves_off_rc4/

[6] C. Paar and J. Pelzl, *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer, 2010. [Online]. Available: http://books.google.se/books?id=N_e4NAEACAAJ

[7] T. Dierks. (1999, January) RFC 2246 - The TLS Protocol, Version 1.0. [Online]. Available: https://www.ietf.org/rfc/rfc2246.txt

[8] ——. (2006, April) Rfc 4346 - the transport layer security (tls) protocol, version 1.1. [Online]. Available: http://www.ietf.org/rfc/rfc4346.txt

[9] ——. (2008, August) Rfc 5246 - the transport layer security (tls) protocol, version 1.2. [Online]. Available: http://tools.ietf.org/html/rfc5246

[10] (2014) Bankid website. [Online]. Available: https://www.bankid.com/en/What-is-BankID/

[11] (2014) Connectify webpage. [Online]. Available: http://www.connectify.me

[12] A. Sanchez. (2014, 8th of January) Personal banking apps leak info through phone. [Online]. Available: http://blog.ioactive.com/2014/01/personal-banking-apps-leak-info-through.html