E-payment Security Analysis In Depth

Adam Ali.Zare Hudaib
(Certified Ethical Hacker) *adamhudaib@gmail.com*
"Two Mas" ltd Krs414007
Network Security Research
Amman, Jordan. Poland

Abstract

Technology is the basis of our lives. The growth of the Internet has made it an ease for consumers to find items for purchase, but no longer is cash a viable way for payment. This increase in e-commerce has driven the need to create an online payment system. Unfortunately there are a lot of flaws and internet frauds that people are facing.

In this research we will review different payment protocols and security methods that are being used to run online payment systems. We will survey some of the popular systems that are being used today, with a deeper focus on the PayPal system, NFC and bitcoins. In addition, we will also discuss the weaknesses in the systems that can compromise the customer's trust.

Keywords: E-payment, Security Protocol, Bitcoin, NFC.

1. INTRODUCTION

Progress in web technologies has led to rapid growth of hybrid web applications that combine the Application Programming Interfaces (APIs) of multiple web services (e.g., search APIs, map APIs, payment APIs, etc.) into integrated services like personal financial data aggregations and online shopping websites. The pervasiveness of these applications, however, brings in new security concerns. The web programming paradigm is already under threat from malicious web clients that exploit logic flaws caused by improper distribution of the application functionality between the client and the server (e.g., relying on client logic to validate user privileges). The program logic of a hybrid web application is further complicated by the need to securely coordinate different web services that it integrates: failing to do so leaves the door wide open for attackers to violate security invariants by inducing inconsistencies among these services.

Intuitively, logic bugs related to multiple web services can be much more difficult to avoid than those in traditional single-service web applications – it is analogous to real-life experiences that when multiple parties discuss a subject by making individual one-on-one phone calls, it is generally difficult for each party to comprehend the whole picture. So in this research we will review different payment protocols and security methods that are being used to run online payment systems.

2. E-PAYMENT SECURITY ANALYSIS IN DEPTH

There are a lot of payment systems that are being used today. But every system has its pros and cons. We will focus on this further.

2.1 Payment Systems

There is payment by instruction type of systems, when a payer basically orders the bank to move a sum of money from her account into a payee's account. Examples in this category are credit and debit cards as well as many forms of cheques. The moment at which the money is actually moved from the payer's account into the payee's account depends on the system, but at all times banks and credit card companies will try to prevent discrepancies between accounts. The central

security aspect in these systems is to ensure that only legitimate account holders are able to issue payment instructions. Of course, digital signatures are the solution for doing this over a large, open network such as the Internet. Since digital signatures only make sense if there is an infrastructure for certifying public keys, a lot of effort is devoted to just this. See, for instance, the SET (Secure Electronic Transaction) proposal, a joint effort by MasterCard, VISA, and other influential partners, which specifies a hierarchy of certification authorities on top of the payment protocols [1].

Prepaid systems are conceptually close to electronic equivalents of cash. Telephone cards, smart card based systems, as well as e-cash fall into this category. The user's account is debited as soon as the card or device is reloaded with electronic cash. During payments the electronic cash is released again, and only then the payee's account will be credited. In the meantime the issuer keeps a float corresponding to the outstanding cash. The central security aspect in this type of system is to ensure that cards or representations of cash cannot be forged. When forgery happens, the float will ultimately be insufficient to credit all of the payees' accounts for received payments. Of course, it should also be ensured that only legitimate account holders can reload cash from their accounts. However, this security aspect is now limited to the infrequent withdrawal protocol, and is no part anymore of the more frequent payment protocol.

Although the payment protocol is functionally a protocol between two parties (payer and payee) many payment systems require that the payee contacts a third party (e.g., the bank or the credit-card company acting as an acquirer) before accepting a payment. If that is the case, the system is called an on-line payment system; the communication between a payee and its acquirer may be using any communication medium (not necessarily the Internet). If such a contact with a third party is not required during the payment protocol, the system is called off-line. In an off-line system payees are required to contact their acquirer on a regular basis for clearing all received payments.

A basic requirement of a payment protocol is that it allows a payee to receive payments from any payer. A payment can be seen as some sort of authentication of the payer towards the payee (to show that the payment is authentic). Authentication can be based on secret key cryptography or on public key cryptography. In the latter case, the payee only needs to have a public key available in order to verify incoming payments. Although the costs of equipping smart cards with crypto co-processors are expected to become marginal, it is important to note that the property of public verifiability can be obtained using simple smart cards only, provided one applies a method of what we call signature transport. In such a system, signatures are created by the issuer only, and later endorsed by the payer during the payment protocol, depending on a challenge from the payee. The trick is to achieve that sufficiently many payments can be made between successive reloads, which requires optimal use of the limited amount of EEPROM available on simple smart cards. The added advantage is that the secret key for creating signatures is only used by the issuer. In case authentication is based on secret key (symmetric) cryptography, how- ever, the payer and payee must have a shared secret key available in order to complete a payment. A straightforward solution is to give all users the same secret key, but this is generally considered insecure, as this would mean that breaking a single smart card (i.e., extracting its secret key) will suffice to break the complete system. The standard solution is therefore to break the symmetry between payers and payees by equipping the merchants with a highly secure tamper-proof box called a SAM that contains a master key. The payers' keys are derived from this master key in a process called diversification by applying a cryptographic hash (e.g., SHA-1) to the concatenation of the master key and the payer's card number. The idea is that the SAM is more difficult to break than a smart card, and also that it is possible to routinely check (as part of the maintenance) if the SAMs have not been tampered with. In the EMV standard (developed by Europay, MasterCard, and Visa) a first step is made toward including public key authentication. To prevent frauds in which cards with fake card numbers are introduced, each card carries a fixed RSA certificate that shows the validity of the card number. At the start of each payment, the certificate can be verified against the public key stored in the POS terminal. The remainder of the payment protocol again relies on a secret master key stored in the SAM of the POS terminal.

Checkout based on using electronic payment system consists of some typical steps. For example, if using PayPal, it starts when the button "Check out with PayPal" on page of the merchant website is clicked. Then user is directed to page on PayPal, where he can click the "Pay Now" button to pay. Then, the shopper's browser is redirected back to the merchant's website to finish the order, which usually does not require the shopper's actions. Finally, the shopper gets the confirmation page. The checkout process is arranged in this way to ensure that all three parties – the shopper, the e-payment system, and the merchant, stay consistent despite their different locations across the Internet.

Dynamic web are invoked through HTTP requests: the client sends an HTTP request through a URL with a list of arguments and receives an HTTP response (often a web page) dynamically constructed by the server as the outcome of the call.

These responses serve as the building blocks for the workflows of various checkout solutions offered by different payment systems service providers (Amazon, PayPal, and Google). Some of the solutions, such as PayPal Standard and Amazon Simple Pay, are entirely based upon HTML, while the others, like PayPal Express and Checkout By Amazon, implement SOAP and NVP APIs. Also e-payment systems websites communicate exclusively over HTTPS to guarantee end-to-end security [2].

2.2 Securing Checkout Processes

The main security goal of a checkout system is to maintain the following payment-completion invariant: Merchant M changes the status of an item I to "paid" with regard to a purchase being made by Shopper S if and only if:

1. M owns I;
2. a payment is guaranteed to be transferred from an account of S to that of M in the e-payment system;
3. the payment is for the purchase of I, and it is valid for only one piece of I;
4. the amount of this payment is equal to the price of I [3].

This invariant, though intuitive, implies a set of intertwined binding relations that should be respected in every step of the transaction. These bindings unequivocally link the merchant to a piece of the item being sold, the price of the item to the payment the merchant receives, and the payment for this specific purchase to the shopper. Complexity in preserving the invariant. To achieve this security goal, a checkout system is expected to preserve the aforementioned invariant throughout a transaction. This turns out to be nontrivial, particularly in the presence of two web services. Specifically, the challenges in keeping both servers in consistent states include, but are not limited to, the following:

1) Confusion in coordination. Given their incomplete views of a transaction, the merchant and the payment system need to work together to preserve the invariant. This, however, is often hindered by the partial knowledge each party has about the other: the code of their systems is often off-limits to each other; the payment system typically provides nothing but vague descriptions of its operations. As a result, misunderstanding often arises on the security assurance either party offers. For example, a merchant may assume that every notification of a payment completion from the payment system must be about one of his transactions, but the payment system may not have this guarantee and may expect a merchant to verify it by itself.

2) Diversity in the adversary's roles. The merchant and the payment system expose their APIs to the public, which enables the adversary to play more diverse roles than just the shopper, and thus to gain a deeper involvement in the checkout process than he

could in a more traditional client-server interaction. The shopper can directly invoke a merchant's APIs, which mimics the behavior of the payment system; the shopper can also mimic a merchant to register with the payment system a callback API.

3) Parallel and concurrent services. Both the merchant website and the payment system need to serve many customers, and a shopper can concurrently invoke multiple purchase transactions. This further complicates the trilateral interactions, opening avenues for cross-transaction attacks.

4) Authentication and data integrity. Compared with the two-party web applications, authentication in a payment system-based checkout system involves three parties and is thus more difficult in avoiding authentication and data integrity breaches.

2.3 Technologies Used for Online Payment Security

There are a few different protocols that are used for online security today. The most common security mechanism is SSL. Some of the others include TLS, and SET.

Secure Sockets Layer, more commonly known as SSL, is a protocol that is used to maintain client and server authentication. A site is easily identified as using SSL if it has the small yellow padlock at the bottom of the browser.

In SSL, communication between the server and the client is encrypted using their certificates. This encryption creates virtual information that is not hackable by others. The steps of how SSL works is shown in the following diagram:

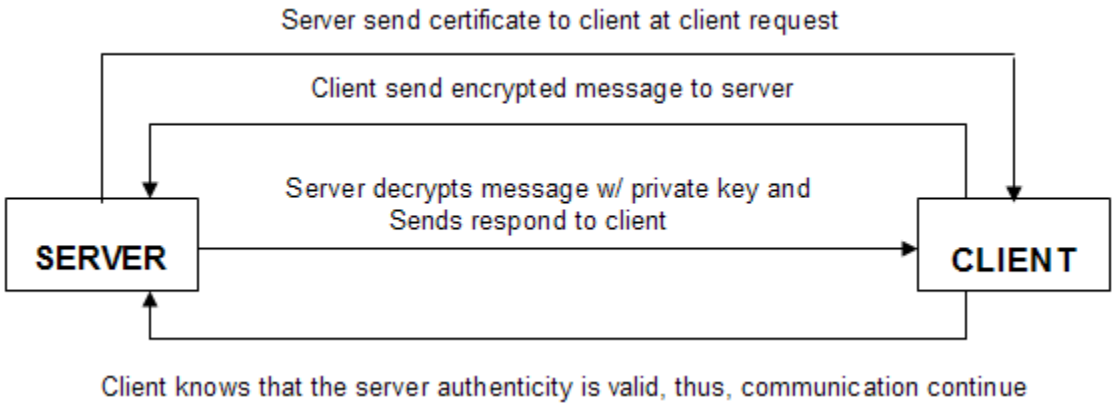

FIGURE 1: SSL Algorithm.

SSL Version 1, a test version, was quickly replaced by SSL version 2, which was the first version, released to the public and was shipped with the Netscape Navigator browser. Today version 2 is still supported despite having some security problems. Later, Microsoft came out with its own version of SSL called PCT. SSL Version 3 is a complete redesign of SSL and fixes the problems found in previous versions as well as having additional features [4].

The purpose of SSL is to provide a means to allow secure communication between two parties. However, one party must have a certificate trusted by the other in order to help prevent man in the middle attacks. SSL also supports authentication, encryption and key exchange.

SSL uses a handshake protocol. Suppose a client wants to make a purchase from a website server, but this server does not know anything about the client.

The first step is for the client to send a message to the server. After the server receives the message, it acknowledges it by sending the client a message in return. The server also sends the client its certificate and asks for the client's certificate. The client sends its certificate, a client key exchange message, and a certificate verification message. Both the client and server send change cipher spec messages and then send finished messages to end the handshake [5].

A website implements SSL by using HTTPS, which stands for Hypertext Transfer Protocol over Secure Socket Layer. This web protocol was developed by Netscape to encrypt and decrypt page requests as well as the pages that are returned by the web server. HTTPS uses port 443 instead of port 80, which is used for HTTP.

SSL uses a key size of 40-bits for the RC4 stream encryption algorithm. This is considered a sufficient degree of encryption for commercial exchange. Both HTTPS and SSL support the use of X.509 digital certificates from the server. This way, the user can authenticate the sender if needed [6].

One of SSL's strengths is its ability to help prevent some common attacks. SSL is strong against the brute force attack because it uses 128 bits. The dictionary attack which tends to be more efficient than a brute force attack is where an attack tries every word in a dictionary as a possible password for an encrypted message. This attack is also avoidable because SSL has very large key spaces. The replay attack which reruns messages that were sent earlier is prevented since SSL uses 128-bit nonce value to indicate a unique connection. And as mentioned earlier, the Man-In-the-Middle Attack is prevented by using signed certificates to authenticate the server's public key.

Despite the fact that SSL has the ability to prevent some common attacks, it still has some weaknesses. One of the weaknesses found in SSL is the brute force attack against weak ciphers. This weakness was forced by the US export on Netscape. This weakness still remains one of the most obvious weaknesses of the SSL protocol and it has broken many times [7].

Another weakness in SSL is the renegotiation of the master key. It is known that after a connection has been established, the same master key gets used all the way through the connection. This could be a serious security flaw if SSL are layered underneath a long running connection. One possible solution for this flaw is to force renegotiation of the master key at different times. This way, the difficulty and the cost of the any brute force attack will be multiplied by the number of times that the master key has changed [8].

The Transaction Layer Security protocol, commonly known as TLS, is based on SSL and will soon become its successor. TLS has some changes in its MAC, has clearer and more precise specifications, cleaner handling because of not having a client certificate, and more flexibility.

Secure Electronic Transaction, SET, provides a way for the client's credit card number to be sent to authorizing banks. However, there was not enough market acceptance of SET to make it commonplace.

2.4 Example of Using SSL by PayPal

With today's technology, the Internet has become the most popular place for people who want to buy goods and services. In order for people to do such kind of trading, they need a safe online payment system that they can trust. PayPal is one of the world's largest online payment systems. All you need is an account with PayPal and you will be ready to send and receive payments online securely.

The technologies used by PayPal consist of the main security mechanisms that most sites would employ. PayPal uses HTTPS and SSL to encrypt the data stream when a user establishes a session with the PayPal site. It is unknown what security mechanisms are used to protect their databases containing information about their customers.

According to the PayPal website, PayPal encrypts information sent to their website using SSL. It uses an encryption key that is 128-bits long, which is currently the most secure level being used today. Before proceeding, the server checks whether or not the user's browser uses SSL 3.0 or higher.

PayPal also uses an electronic firewall to protect its data from the Internet. Their servers are behind the firewall and not directly connected to the Internet in order to protect private information from unauthorized computers [9].

It is unclear what other forms of technology PayPal uses to ensure security, because of their avoidance to divulge too much information to the public as to put them at a security risk to attackers.

Popular companies such as Microsoft and PayPal have been attacked by hackers from all around the world because of some security flaws in their systems. Despite the fact that PayPal is one of the world's largest online payment systems, it has some security flaws and weak points.

Because of PayPal's heavy reliance on SSL as a means to achieve security, many of their weaknesses are therefore the weaknesses of SSL.

One of the attacks on PayPal was done by few experienced hackers from Russia who discovered a serious security flaw in the "address confirmation process" of PayPal's members' accounts. The hacking process was exposed to everyone on the internet in Russian language. It has been confirmed that PayPal had some technical difficulties in fixing the problem mentioned above, which means that a lot of PayPal accounts with confirmed address could be hacked into [10].

Email scams have been the source of security problems for PayPal. Many users have received emails appearing to be from PayPal urging them to click on a link and log in to their accounts.

This scam is dangerous because hackers obtain passwords by using false e-mail messages. Third parties hack PayPal accounts, using the passwords obtained, and then login to the accounts to steal money. They send false e-mails to PayPal users leading them to think that the e-mails were sent by PayPal when they are not [11]

Another email scam shows up directly in the email inbox and does not require the user to click on any links. This email scam is a bit more advanced as it shows a form directly in the email. The email asks the user to give information in order to confirm the information in PayPal's database; not doing so may risk cancellation of their account.

The problem with this email is not only is it not from PayPal, but it leads the naïve user to a website that appears to be an authentic PayPal website. The act of pretending to be somebody else's website is referred to as web spoofing.

In the email scam can provide links to not a PayPal site. Although the link shown is a PayPal site, after clicking on it, the address for the link appears to be different. This may go unnoticed to naïve users because the images and content are very similar to the real PayPal's website. The problem with this is when the user logs in to their account to prevent their account from being cancelled. Since this website is not authentic, the user submits their username and password to an unknown third party. This third party is then able to store their username and password into a database to cause damage to these users' accounts.

The use of client certificates would help stop web spoofing. However, client side certificates are hardly used, so the practicality of this is not very great. PayPal should also protect the images that are used on their website such that it cannot be saved or used by the public. This would help prevent attackers from stealing the images to create pages that mirror PayPal in its appearance.

PayPal's security relies heavily on user passwords. Although they limit the number of attempts to login, the limit seems pretty high. PayPal limits login attempts to ten accesses before locking the account [12]. While the chances of attackers using brute force to break into an account is rather slim, the opportunities are greater than other websites in which a user is limited to three attempts. This is a very significant security issue. If an attacker is able to access a user's account, he can wreak havoc by stealing money from the user.

An initial solution to this problem is to decrease the limit of login attempts. In addition, PayPal should have another layer of protection after entering a correct password, such as a question that requires a correct answer similar to what is done for password retrieval.

Other solutions that would help with online payment security involve the user to be alert at all times. One possible way to prevent thieves from stealing your password is to never trust any e-mail coming from PayPal and never click on a link that would take you to PayPal directly from an e-mail. So instead of clicking on the link it is better to go to the website by typing the website's URL. Also, it is a good idea to make sure the URL entered contains https://.

It should be remained that according to a report by a team of researchers from Indiana University and Microsoft Corp. major e-payment systems, e-retailers and e-commerce platforms—including Google Checkout, PayPal, Amazon Payments, Buy.com, JR.com, nopCommerce and Interspire—have payment system security software flaws that can be exploited to confirm payment to an illegitimate web site or to receive products for free or at reduced prices. The team's research so far has only touched on what it calls "the simplest trilateral interactions" among payment services providers, e-commerce platforms and online retailers.

XiaoFeng Wang, however, says payment service providers should take a more effective and proactive role in ensuring their systems are properly deployed. Also the researcher adds that payment services providers should also give merchant software developers tools to analyze their service integration. In the research it was also shown how someone could manipulate the e-mail payment notification system within Amazon Payments to make a payment on one e-commerce site result in a payment confirmation to a different site.

2.5 Bitcoin

Bitcoin is a peer-to-peer payment network and digital currency based on an open source protocol, which makes use of a public transaction log. Bitcoin was introduced in 2009 by pseudonymous developer "Satoshi Nakamoto". It is called a cryptocurrency because it uses public-key cryptography. When paying with bitcoin, no actual monetary exchange takes place between buyer and seller. Instead, the buyer requests an update to a public transaction log, the blockchain. This master list of all transactions shows who owns what bitcoins currently and in the past and is maintained by a decentralized network that verifies and timestamps payments using a proof-of-work system. The operators of this network, known as "miners", are rewarded with transaction fees and newly minted bitcoins [13].

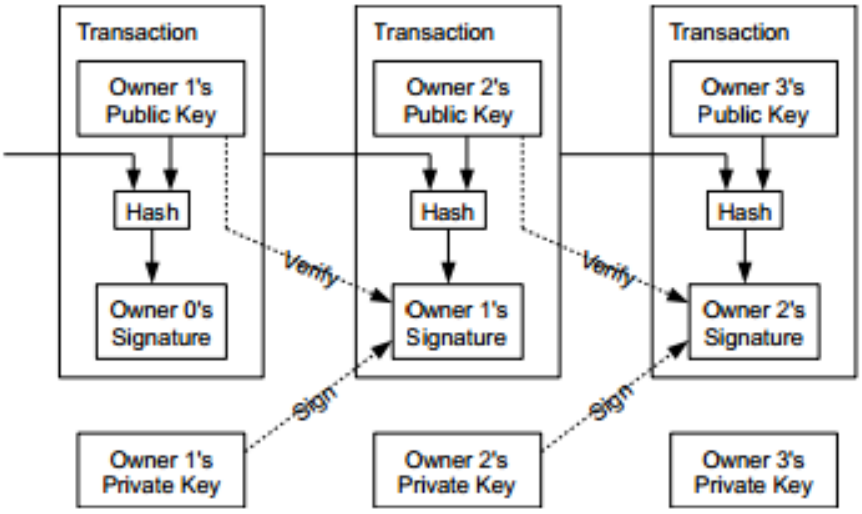

FIGURE 2: Bitcoin Payment Processing.

In order to make a payment, a user requests an update to the master transaction list, the blockchain, and the transaction is validated by the network. Although transactions can be validated instantly, it takes bitcoin miners approximately 10 minutes to record the payment within the blockchain and confirm it was not spent twice. In addition, transactions that pay a fee may be processed more quickly [14].

Bitcoin payment processing fees are optional and generally substantially lower than those of credit cards or money transfers. Currently, doing the work of payment processing is rewarded with newly created bitcoins. But this reward is halved every few years eventually phasing out all together when the total number of bitcoins have been released. Once the Bitcoin ceiling of 21 million units is reached, payment processing will only be incentivized with transaction fees.

Bitcoin functions using public-key cryptography, in which a user generates a pair of cryptographic keys: one public and one private. Only the private key can decode information encrypted with the public key; therefore the keys' owner can distribute the public key openly without fear that anyone will be able to use it to gain access to the encrypted information. An example public wallet (owned by the FBI) demonstrates its structure, a string of 34 numbers and letters (the private key, however, must be kept secret and secure) [15]. The public key can be used as an "address" to which other users can send bitcoins. Anyone wishing to use Bitcoin can create one or more Bitcoin addresses, which are collected and tracked in "wallets". Anyone can send bitcoins to the public address provided by the owner of the wallet, while the private key must be entered by the wallet owner to send bitcoins. Securing and protecting the private key is the essence of wallet security. If the private key for an address is not kept secret, the bitcoins may be stolen; theft has been documented on numerous occasions, and the practical day-to-day security of Bitcoin wallets remains an on-going concern.

Wallets allow a user to complete transactions between addresses by requesting an update to the blockchain, the public transaction log. Wallets come in a variety of forms: apps for mobile devices and computers, hardware devices, and paper tokens. When making a purchase with a mobile device, the use of QR codes to simplify transactions is ubiquitous.

2.6 NFC

Near field communication (NFC) is a set of standards for smartphones and similar devices to establish radio communication with each other by touching them together or bringing them into proximity, usually no more than a few inches.

Near Field Communication is used mostly in paying for purchases made in physical stores or transportation services. A consumer using a special mobile phone equipped with a smartcard waves his/her phone near a reader module. Most transactions do not require authentication, but some require authentication using PIN, before transaction is completed. The payment could be deducted from a pre-paid account or charged to a mobile or bank account directly [16].

Mobile payment method via NFC faces significant challenges for wide and fast adoption, due to lack of supporting infrastructure, complex ecosystem of stakeholders, and standards. Some phone manufacturers and banks, however, are enthusiastic.

Although the communication range of NFC is limited to a few centimeters, NFC alone does not ensure secure communications. In 2006, Ernst Haselsteiner and Klemens Breitfuß described different possible types of attacks, and detail how to leverage NFC's resistance to man-in-the-middle attacks to establish a specific key. Unfortunately, as this technique is not part of the ISO standard, NFC offers no protection against eavesdropping and can be vulnerable to data modifications. Applications may use higher-layer cryptographic protocols (e.g., SSL) to establish a secure channel.

NFC attack examples:
The RF signal for the wireless data transfer can be picked up with antennas. An attacker can typically eavesdrop within 10m and 1m for active devices and passive devices, respectively. With the use of a patch loop antenna it is possible to place a receiver close to the target and disguise it. This is much like ATM skimming in that it needs to be near the location however in this case no contact with the device or reader is required [17].

It is easy to destroy data by using a jammer. There is no way currently to prevent such an attack. However, if NFC devices check the RF field while they are sending, it is possible to detect attacks.

It is much more difficult to modify data in such a way that it appears to be valid to users. To modify transmitted data, an intruder has to deal with the single bits of the RF signal. The feasibility of this attack, (i.e., if it is possible to change the value of a bit from 0 to 1 or the other way around), is amongst others subject to the strength of the amplitude modulation. Transmitting Manchester-encoded data with a modulation ratio of 10% permits a modification attack on all bits. Because NFC devices usually include ISO/IEC 14443 protocols, the relay attacks described are also feasible on NFC. For this attack the adversary has to forward the request of the reader to the victim and relay back its answer to the reader in real time, in order to carry out a task pretending to be the owner of the victim's smart card [18].

Losing the NFC RFID card or the mobile phone will open access to any finder and act as a single-factor authenticating entity. Mobile phones protected by a PIN code acts as a single authenticating factor.

Lawfully opened access to a secure NFC function or data is protected by time-out closing after a period of inactivity. Attacks may happen despite provisions to shut down access to NFC after the bearer has become inactive. Additional features to cover such an attack scenario dynamically shall make use of a second wireless authentication factor that remains with the bearer in case of the lost NFC communicator. Relevant approaches are described as an electronic leash or its equivalent, a wireless key.

Conclusions
We presented our analysis for E-payment, as an example of security challenges in third-party service integration. We found serious logic flaws in leading merchant applications, popular online stores and payment providers (i.e., PayPal). We discussed the weaknesses in the systems that can compromise the customer's trust.

Web applications increasingly integrate third-party services. The integration introduces new security challenges due to the complexity for an application to coordinate its internal states with those of the component services and the web client across the Internet.

Online payments through are relatively safe because they use SSL technology which is the safest mechanism being used today or another secure methods (for example, using public-key cryptography). But the problem is the SSL protocol is not flawless, and users who see the yellow padlock at the bottom of the browser may get a false sense of security. Also there are always some flaws in security methods.

But in reality, the security of online payment also depends on the customer himself. He should gain knowledge in how to use the internet so that he can be more aware of email scams and website URLs that may not be from payment system website. For example for PayPal users, the lack of knowledge and common sense appears to have caused more problems than insecurity. However, there probably is no best way to be fully secured other than to just avoid online purchases altogether.

We believe that our study takes some steps in the security problem space that web applications have brought. In future work we are considering the security challenges that come with web service integrations in other scenarios, e.g., social networks and web authentication services, cancel, return flows. Fundamentally, we believe that the emergence of this new web programming paradigm demands new research efforts on ensuring the security quality of the systems it produces.

3. REFERENCES

[1] S. Murdoch and R. Anderson. “Verified by Visa and MasterCard SecureCode: or, How Not to Design Authentication”. Financial Cryptography and Data Security, Jan. 2010, pp. 42-45.

[2] “PayPal. PayPal - Data Security and Encryption”. Internet: http://www.paypal.com/cgi-bin/webscr?cmd=p/gen/security-outside [Dec. 10, 2013].

[3] 3. Rui Wang, Shuo Chen, XiaoFeng Wang, Shaz Qadeer. ”How to Shop for Free Online Security Analysis of Cashier-as-a-Service Based Web Stores”. Internet: http://research.microsoft.com/pubs/145858/caas-oakland-final.pdf [Dec. 1, 2013].

[4] 4. “The Secure Sockets Layer Protocol”. Internet: http://www.cs.bris.ac.uk/~bradley/publish/SSLP/chapter4.html [Nov. 22, 2013].

[5] 5. “SSL: Intercepted today, decrypted tomorrow”. Netcraft, pp. 10-12, May 25, 2013.

[6] 6. “SSL/TLS in Detail“. Microsoft TechNet, July 31, 2003.

[7] 7. “Description of the Secure Sockets Layer (SSL) Handshake“. Internet: http://www.support.microsoft.com [Dec. 1, 2013].

[8] 8. “Secure electronic transaction”. Internet: http://en.wikipedia.org/wiki/Secure_Electronic_Transaction [Dec. 12, 2013].

[9] 9. “The Secret PayPal Hack Method – 100% Guaranteed!”. Hack Expert, Nov. 11, 2003. Internet: http://www.astronomysight.com/_message/00000465.htm [Dec. 12, 2013].

[10] 10. “PayPal Email Scam – Web Site Version”. Internet: http://www.fightidentitytheft.com/paypal_scam.html [Nov. 25, 2013].

[11] 11. “PayPal Email Scam – Email Form”. Internet: http://www.fightidentitytheft.com/paypal_scam_email_form.html [Nov. 25, 2013].

[12] 12. “SearchSecurity.com”. Internet: http://searchsecurity.techtarget.com/sDefinition/0%2C%2Csid14_gci214006%2C00.html [Dec. 12, 2013].

[13] 13. “Bitcoin”. Internet: http://en.wikipedia.org/wiki/Bitcoin.html [Dec. 10, 2013].

[14] 14. “Bitcoins”. Internet: http://www.weusecoins.com/en/ [Dec. 8, 2013].

[15] 15. Alex Hern. “Bitcoin me: How to make your own digital currency”. Internet: http://www.theguardian.com/technology/2014/jan/07/bitcoin-me-how-to-make-your-own-digital-currency [Dec. 5, 2013].

[16] 16. “Near field communication”. Internet: http://en.wikipedia.org/wiki/Near_field_communication [Dec. 10, 2013].

[17] 17. Mike Clark. “Inside Secure adds sales agents”. Internet: http://www.nfcworld.com/2012/12/05/321436/inside-secure-adds-sales-agents, Dec. 5, 2012 [Dec. 10, 2013].

[18] 18. “NFC and Contactless Technologies”. Internet: http://nfc-forum.org/what-is-nfc/about-the-technology/ [Dec. 1, 2013].