IBIMA Publishing
Journal of Information Assurance & Cyber security
http://ibimapublishing.com/articles/JIACS/2018/415339/
Vol. 2018 (2018), Article ID 415339, 15 pages
DOI: 10.5171/2018. 415339

Research Article

Improving Malware Mitigation For Online Bitcoin Wallets

Sabreen MohammedSiraj Ahmadjee

Umm Al-Qura university, Makkah, Saudi Arabia

smahmadjee@uqu.edu.sa

Received date: 22 August 2017; Accepted date: 18 December 2017; Published date: 26 February 2018

Academic Editor: Ramlah Hussein

Abstract

Bitcoin cryptocurrency has risen in popularity. Therefore, it not only attracts users to use it, but also it attracts the malware developers to attack it. While online Bitcoin wallets allow users to conveniently store and manage their bitcoins, they are vulnerable to several attacks. There are several existing solutions that attempt to provide Bitcoin wallet security. We analysed them and compared them based on several criteria and determined that the existing solutions for securing online Bitcoin wallets used by individuals are vulnerable. We described the threat models for a Bitcoin two-factor wallet, analysed the risk, and presented the risk-treatment plan. We proposed a Bitcoin three-factor wallet that provides several security features on both the server and the client sides. This new proposed wallet is secured against some attacks. It takes a reasonable time to perform the transactions and is convenient to use.

Keywords: Bitcoin, Cryptocurrency, Online Wallet, Threshold Signatures.

Introduction

Bitcoin is a decentralised cryptocurrency, which means it does not need a central trusted financial network, such as a bank, to check the identity of the payer or the payee, or to verify and confirm the transactions. Bitcoin popularity has grown slowly in the years since it was introduced in 2009 (Narayanan et al., 2016). However, in 2013, major merchants started accepting bitcoins as an official payment for goods (Hill, 2014). Importantly, in 2014, bitcoin value reached a peak at around $1120 (blockchain.info, 2016). Users such as the online shops need to protect their bitcoins against attackers who attempt to steal them. In other words, they need to protect the wallets that store their bitcoins. They need to keep their private key safe because stealing a Bitcoin private key permits attackers to steal the money. Attackers can exploit the vulnerabilities of the wallet – where users store their private key - in order to steal the user's bitcoins.

Thieves can attack the wallets using several

Cite this Article as: Sabreen MohammedSiraj Ahmadjee (2018), "Improving Malware Mitigation for Online Bitcoin Wallets", Journal of Information Assurance & Cyber security, Vol. 2018 (2018), Article ID 415339, DOI: 10.5171/2018. 415339

types of attacks involving malware. Commonly, malware compromises bitcoins when they are stored on devices that frequently connect to the Internet in order to extract and transfer the private keys that used to accomplish the bitcoin transactions. Therefore, the attackers can spend the money. Indeed, malware attacks on Bitcoin wallets have increased significantly. Recently, researchers have reported discovering more than 140 forms of malware programs designed to exploit the weaknesses in Bitcoins wallets in order to steal their contents (Hill, 2014),(Goldfeder et al., 2015),(STEWART, 2014). Therefore, it is necessary to improve the existing methods of protecting and securing Bitcoin wallets to tackle malware attacks.

The aim of this research is to investigate and analyse the existing methods used for protecting and securing online Bitcoin wallets that are accessed over the Internet. This kind of wallet needs to be defended against malware attacks. This analysis is followed by suggested methods for effectively improving malware mitigation for online Bitcoin wallets.

Background

Elliptic Curve Digital Signature and Bitcoin

ECDSA algorithms

The Bitcoin system utilises a digital signature scheme, which is called Elliptic Curve Digital Signature (ECDSA). It works in the group $E(Z_p)$, which is a group of elliptic curve. ECDSA, like any digital signature scheme, consists of three algorithms: key generation, signing, and verification algorithms (Johnson et al., 2001). ECDSA explained in details in (Johnson et al., 2001) literature.

Wallet Security

Approaches for Wallet Security

Storing private keys in a single place will lead to a single point of failure which allows the attackers to compromise this storage place. Fortunately, different approaches to tackle this problem are presented in the following sections.

Multi-signature

This security approach is used to avoid a single point of failure. As the name suggests, to spend bitcoins, more than one signature is required to prove valid possession of the money. This means multiple secret keys, which should be stored in different places, are needed to generate the necessary signatures. A multi-signature transaction can be signed independently by each party in a non-interactive manner. This is one advantage of this approach over threshold signature because threshold signature requires more than one round of interaction (Gennaro et al., 2016).

However, it has pointed out several disadvantages of the multi- signature scheme (Gennaro et al., 2016). Firstly, there is a lack of confidentiality because multi-signature transactions are distinguishable in the public blockchain from regular transactions. Secondly, anonymity is harder to achieve. To provide better anonymity, a new change address that does not simply link to the input address should be utilised. Thirdly, the transaction's public keys and valid signatures M lead to increasing the transaction's size. Therefore, transaction fees will increase as they are based on transaction size. Lastly, a multi-signature transaction is not flexible because inserting multi-signature security into an already existing address is not easy because the syntax of the two kinds of addresses differ. Because of these drawbacks, a multi-signature scheme is not used for the proposed solution.

Threshold Signatures

The main difference between threshold scheme and multi-signature scheme is that the threshold signature scheme allows multiple parties to participate in signing a transaction without using multiple private keys (Narayanan et al., 2016). A single private key is split into shares that are distributed among the participants. Specifically, any subset of these secret shares can be used to reconstruct the private key, as long as the subset's size is greater than or equal to a specific threshold. Any subset that is smaller than this specific threshold will not reveal any information about the private key. A key property of the threshold scheme is that generating a signature does not require reconstructing the private key on any single machine (Narayanan et al., 2016).

The threshold signature scheme has several key advantages (Gennaro et al., 2016). This scheme preserves the user's confidentiality because it allows the participants to generate a single signature. Therefore, it will be invisible in the public blockchain that this kind of signature is used. Additionally, the threshold scheme increases anonymity because the change address provides the same benefit in the threshold scheme transaction as in a regular single key address. Therefore, the change address will be unlikable to the sending address when they are specified in sending transaction.

Also, threshold signatures schemes provide flexibility because it is possible to add threshold security scheme to an existing address because this scheme permits splitting up the key of the existing address as opposite to multi-signature. Also, the threshold signature scheme easily allows an arbitrary number of participants to be added because, unlike multi-signature transactions, they do not require a new address to be generated. Moreover, the threshold transaction is scalable because it maintains a constant transaction size even if the number of shares is increased; therefore, fees will not be increased.

Two-Party Signature

In order to apply 2-out-of-2 signature cases and use an optimal number of participants, (MacKenzie and Reiter, 2001) built a specialised scheme that is called "two-party signature scheme for DSA". They use a Pailler cryptosystem, which is a homomorphic cipher allowing one participant to do the computation using the ciphertexts of another participant's share secret without the ability to learn these share secrets. The final encrypted signature is decrypted by the homomorphic decryption key which is held by one party. This scheme is close to ideal for two-party signatures (Gennaro et al., 2016). (Mann and Loebenberger, 2015) apply this scheme to ECDSA and use it to their two-factor wallet. (Gennaro et al., 2016)'s appendix articulates their attempts to extend two-party signature schemes to multiple-party signature schemes. They found it quite inefficient when the large numbers of parties are involved, such as when employed in a company. This inefficiency arises because it requires $3M - 1$ rounds of interactions. Additionally, when the number of players increases, the computation time and the storage increase exponentially.

Two-Factor Wallet

(Mann and Loebenberger, 2015) and (Gennaro et al., 2016) implemented a two-factor wallet for individuals by applying (MacKenzie and Reiter, 2001)'s two-party signature scheme; as discussed, this scheme is optimal and nearly ideal for two-party signature. This kind of wallet requires control to be split over two personal devices in the user's possession, such as a desktop and a smartphone. The secret key shares are distributed between these devices.

Importantly, at no point during the communication is the private key reconstructed neither in the desktop nor on the smartphone. This security makes the attacker's job much harder and avoids the vulnerability of a single point of failure.

Furthermore, if the desktop is infected by the malware that can be able to change the transaction details, the user can notice that when the transaction information is sent to the mobile wallet. The user can then abort this transaction and prevent the attacker from stealing the bitcoins.

Clearly, the two-factor wallet improves the security of the wallet that is used by individuals at some extent. However, as discussed in the threat modelling section, an adversary who manages to compromise the first factor (the desktop) can bypass the second factor (the phone). This vulnerability means the two-factor wallet is not secure enough to protect the Bitcoin wallets used by individuals. Therefore, a solution that enhances the security of this kind of wallet is proposed in this paper, and it is described in the proposed solution chapter.

Threat Model

We performed risk assessment and identified effective countermeasures that help to avoid and mitigate the possible risks utilising the Microsoft threat modelling (J.D. Meier, 2003).

Two-Factor Wallet (Online wallet and Mobile wallet) Attack

Since this study focuses on online wallets, it will explain how the two-factor wallet that (Gennaro et al., 2016) and (Mann and Loebenberger, 2015) represent can be compromised if one applies it to an online wallet instead of a desktop wallet. One type of Bitcoin wallet is the hybrid online wallet where the wallet service provider does not have control over the private keys because they do not know them. The private key shares are generated, encrypted, and decrypted on the client side. Since this type of online wallet is to some extent similar to the desktop wallet where users have full control over their keys, it is more suitable to employ it in the two-factor wallet approach. The attacker can compromise the two factors: the device used to access the online wallet and participate in signing the transaction and the mobile wallet used to verify the transaction and participate in signing the transaction.

Method of Attack

An attacker can compromise the two devices and steal the two private key shares by performing the following phases.

Phase One: Infection the devices

An attacker can follow any attack approach illustrated in Figure 1 to install malware on the user's devices. Importantly, to infect the second device, either the attacker can follow any approach represented by the attack tree which requires user interaction, or employ cross-platform infection as (Dmitrienko et al., 2014) did to compromise the second factor and steal the Transaction Authentication Number (TAN) of the online banking transaction. More details about cross-platform infection attacks are provided in (Dmitrienko et al., 2014). One form of sophisticated malware that can be installed on the user's device infects the user's browser. This malware utilises Man in the Browser MITB techniques, which are powerful and sophisticated attacks normally associated with Internet crime (Dougan and Curran, 2012). This kind of attack often targets online bank accounts and steals users' credentials, tamper with transactions and facilitate the theft of second-factor authentication and TANs from a second device. Therefore, MITB malware can also target online Bitcoin wallets.

Browsers allowing user to install any extension from arbitrary websites

(untrusted parties) can easily be infected by malware. The attacker can craft the malicious payload into an extension that runs with high privileges and then trick the user into installing the malware onto the device through any of the ways represented in Figure 1.

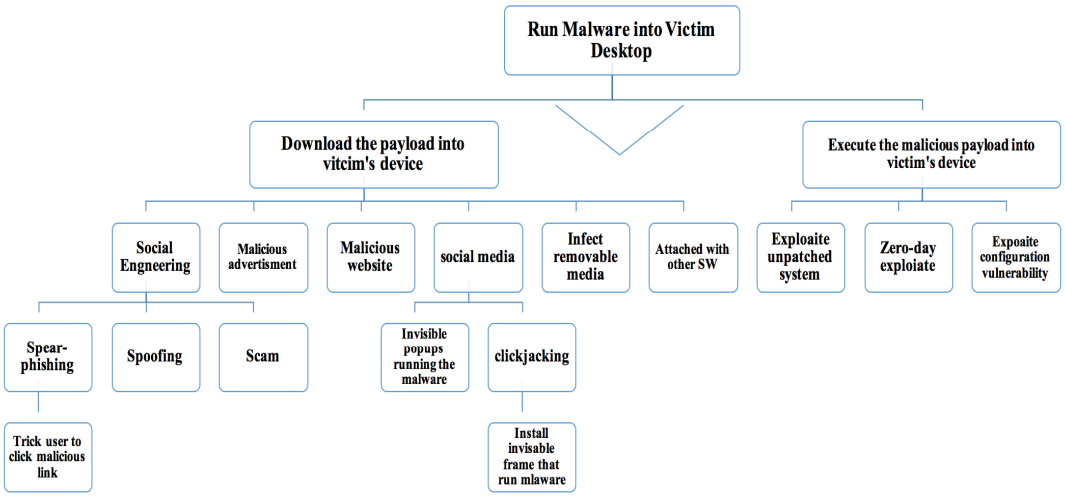

Figure 1: Attack tree with the goal of running the malware into victim's devices

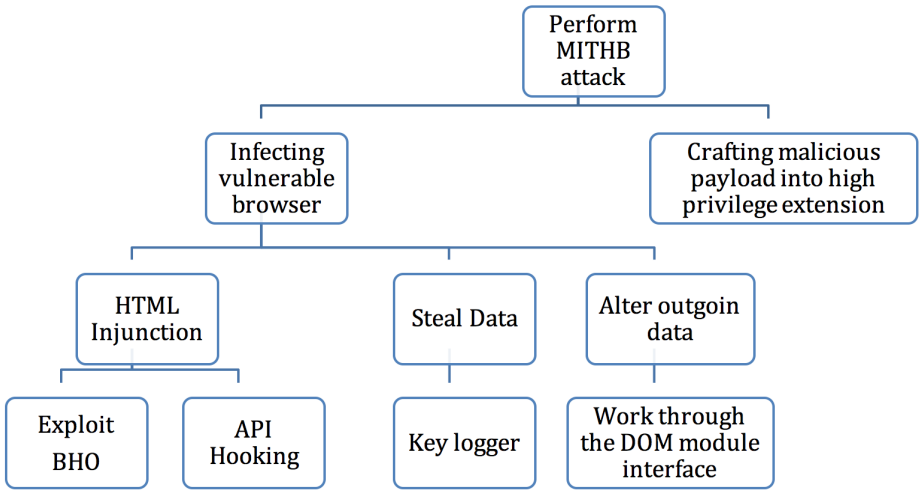

Figure 2: Attack tree with the goal of performing MITH browser attack

As (Barth et al., 2010) report in their paper, 88% of the Firefox extensions they analysed ran with full privileges, even though the extensions required less than these assigned privileges. Therefore, malicious extensions can infect the browser and perform an MITB attack. Figure 2 represents how an MITB attack can be performed.

After the malware is installed on the device, the malware will not launch until the browser is restarted. Then the malicious extension will be installed onto the browser's configuration. Next, the browser loads the extension which registers a handler for every page that the browser will load. As (Mário Almeida, 2011) mentioned, the MITB malware will monitor all of the user's activities. When any page is loaded by the browser, the malicious extension compares the URL against the list of sites that the attacker specified. If the URL matches one of the desired pages, the malware will change the page's content by adding an extra field in order to ask the user to verify the user's mobile phone number in order to send a link to an important application security update, which is in fact a malicious application. Once the user clicks submit, all the information including the phone number will be sent to the attacker in order to send the malicious message to the user to compromise the mobile device.

Phase Two: Stealing the Wallet File

Because hybrid wallets generate, encrypt, and decrypt private keys on the client side, more specifically in the client's browser and typically in JavaScript (Eskandari et al., 2015), MITB malware can lie-in-wait either for the private key share to be generated or after decrypting it in order to sign the transaction. MITB malware can also easily discover the password used to protect this private key share by stealing the password once the user enters it and then sending it to a malicious server. By doing this, an attacker can obtain the first share key. In addition to the first private share, the attacker also needs to steal the mobile wallet's private share in order to steal the user's funds. After successfully infecting the mobile device as mentioned in the previous section, the attacker can steal the private key share used in the mobile device by the same method.

Determine Threats

Table 1 summarises the threat modelling process.

Table 1 Summary of the threat modelling process

Threat Description	Attacker steals the bitcoins from the two-factor wallet by stealing private key shares
Threat target	User's browser and mobile device
Risk ranking	High
Attack techniques	MITB malware
Countermeasures	1. Use three secret key shares. 2. Use the Token and the fraud detection mechanism. 3. Check the Token and use the fraud detection mechanism in every request. 4. Abort the transaction and log it if fraud is detected. 5. Use the isolation mechanism and the extensions that run with least privilege.

Proposed Solution

Three-Factor Wallet

The main countermeasure proposed here to avoid the described risks is using a three-factor wallet instead of two-factor wallet; a proposed three-factor wallet will be more secure for the wallet used by individuals. To utilise this wallet, it is necessary to have three private key shares to sign any transaction. One on the user's desktop, the second on the user's mobile phone, and the third on an online server. Therefore, if attackers manage to steal both of the secret shares as described in the threat model, they cannot spend the bitcoins in the user's wallet because they also need the third share to sign any transaction. Involving the server in the signing process makes the transaction very difficult to compromise as the hacker must also attack the server or trick it into signing the transaction. Therefore, this new proposed online Bitcoin three-factor wallet is more secure than the existing two-factor wallet.

In practice, it is assumed that the three participants are communicating over a secure channel. As previously mentioned, the hybrid online wallet is suitable for use in the two-factor online wallet. Therefore, it will also be suitable to be used in the three-factor wallet. The service provider can be a trusted dealer that constructs the key, generates the shares, and distributes the shares to the parties. The server should then completely delete all records of the desktop and the mobile phone's key shares. Of course, the user needs to trust the server to execute that task. Once each one of the three participants has the key share, they execute the following steps to sign the transaction:

- User authenticates its desktop and mobile phone to the server.
- User initiates the request (Bitcoin transaction) from the desktop and sends it to the server.
- Server checks the authorisation of the request by checking the Token and the user's behaviour as would be described in the security analysis section.
- Server sends the transaction to the mobile wallet in order to allow the user to confirm it.
- When both the user (using the mobile phone) and the server confirm the transaction, then the three parties start the three-party threshold scheme in order to sign the transaction.
- The three parties then exchange the required messages for the three-party threshold scheme over a secure connection in order to generate a signature.
- Server embeds the complete signature into the transaction.
- Server broadcasts the correctly signed transaction onto the Bitcoin network.

Figure 3 represents a sequence diagram showing how the three parties communicate in order to sign the transaction using the three-party signature protocol.

Three-Factor Scheme

The three-factor wallet can be implemented by applying the scheme that (Gennaro et al., 2016) was represented in their appendix, which is an extended version of Mackenzie and Reiter's two-party signature scheme used in two-factor wallet. The three-factor wallet requires a 3-out-of-3 threshold signature to be applied. Consequently, it is suitable to use the multi-party scheme to implement the three-factor wallet.

Key Generation and Setup

The extension of the two-party signature protocol requires $3M - 2$ rounds. As a result,

the 3-out-of-3 signature scheme is accomplished in seven rounds.

Before the parties start the protocol to generate the final encrypted signature, they first have to generate their share keys. Therefore, each participant has to perform the following steps:

- Each participant P_i has to choose a random value $x_i \in Z_q$ and then compute $y_i = G^{x_i}$ which is assumed to be public. Hence, the secret key will be $x = \prod_1^i x_i \ mod\ q$ and the public key will be y = $G^x \ in\ \mathcal{G}$.
- All participants share a secret key D for the additively homomorphic encryption scheme E. The secret key is shared in a 3-out-of-3 scheme among the three parties. Therefore, given a large integer N, one can compute $\alpha = E(a)\ and\ \beta = E(b)$, where $a, b \in Z_N$. Using these values, it is possible to do $+_E$ which is an efficient computable operation that can be done over the ciphertext space as follows:

$$\alpha +_E \beta = E(a + b\ mod\ N)$$

Also one can compute:

$$x \times_E \alpha = E(x\ a\ mod\ N)$$ where x is an integer

The message space and ciphertext space are denoted as M_E and C_E, respectively.

Sabreen MohammedSiraj Ahmadjee (2018), Journal of Information Assurance & Cybersecurity, DOI: 10.5171/2018. 415339

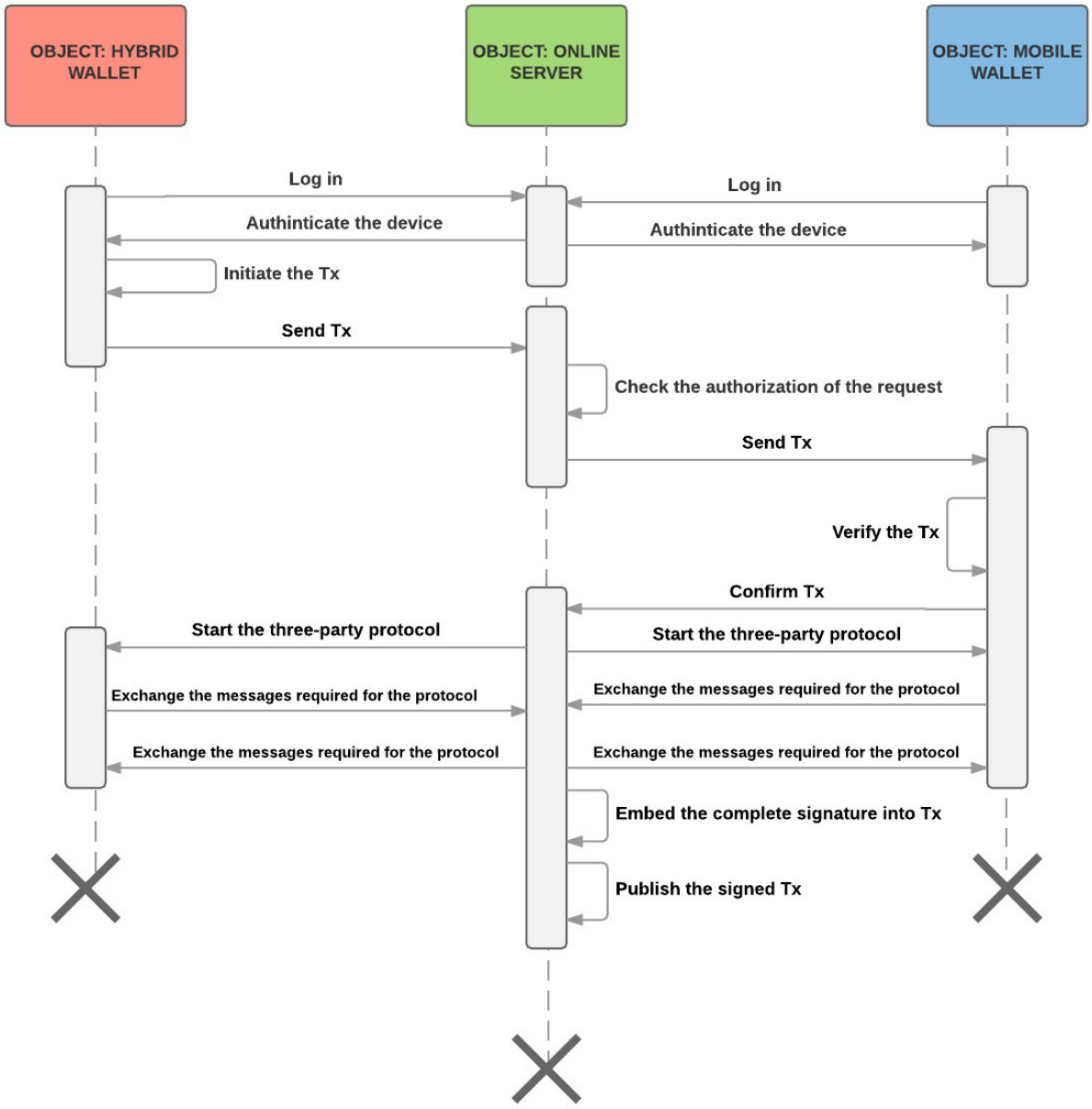

Figure 3: Three-factor wallet sequence diagram

Sabreen MohammedSiraj Ahmadjee (2018), Journal of Information Assurance & Cybersecurity, DOI: 10.5171/2018. 415339

Signature Generation

To implement the 3-out-of-3 signature scheme, the multi-party protocol will proceed for seven rounds. Every participant accepts some inputs, applies some computations, and passes the output of the computation to another participant. This protocol can be implemented by the desktop, the online server (service provider), and the mobile in order to provide a secure three-factor wallet as follows:

Round 1 (Desktop):

- Selects $k_1 \overset{R}{\leftarrow} Z_q$
- $z_1 = k_1^{-1} \, mod \, q$
- $\alpha_1 = E(z_1)$
- $\beta_1 = E(x_1 z_1 \, mod \, q)$
- $\alpha_1^{'} = \beta_1^{'} = \perp$
- Send these values to the second part (online server) < M, $\alpha_1, \beta_1, \alpha_1^{'}, \beta_1^{'}$>

Round 2 (Online server):

- Selects $k_2 \overset{R}{\leftarrow} Z_q$
- $z_2 = k_2^{-1} \, mod \, q$
- $\alpha_2 = \alpha_1 \otimes z_2$
- $\beta_2 = E(x_2 z_2 \, mod \, q) \otimes \beta_1$
- $\alpha_2^{'} = E(z_2)$
- $\beta_2^{'} = E(x_2 z_2 \, mod \, q)$
- Send these values to the third party (smartphone) $< M, \alpha_1, \beta_1, \alpha_2, \beta_2, \alpha_1^{'}, \beta_1^{'}, \alpha_2^{'}, \beta_2^{'} >$

Round 3 (Smartphone):

- Selects $k_3 \overset{R}{\leftarrow} Z_q$
- $z_3 = k_3^{-1} \, mod \, q$
- $R_3 = G^{k_3} \, in \, \mathcal{G}$
- Sends $< R_3 >$ to the online server

Round 4 (Online Server):

- $R_2 = R_3^{k_3} \, in \, \mathcal{G}$
- Sends $< R_3, R_2 >$ to the desktop

Round 5 (Desktop):

- $R_1 = R_2^{k_1}\ in\ G$
- $\prod 1 \leftarrow znp$: $\exists \eta_1, \eta_2 \in [-q^3, q^3]$ such that:
  - ${R_1}^{\eta_1} = R_2$ and $G^{\eta_2/\eta_1} = y_1$
  - $D(\alpha_1) = \eta_1\ and\ D(\beta_1) = \eta_2$
- Sends $< R_1, \prod 1 >$ to the online server

Round 6 (Online Server):

- Verifies $\prod 1$
- $\prod 2 \leftarrow znp$: $\exists \eta_1, \eta_2 \in [-q^3, q^3]$ such that:
  - ${R_2}^{\eta_1} = R_3$ and $G^{\eta_2/\eta_1} = y_2$
  - $D(\alpha_2) = \eta_1 D(\alpha_1)\ and\ D(\beta_2) = \eta_2 D(\beta_1)$
  - $D(\alpha_1^{'}) = \eta_1\ and\ D(\beta_1^{'}) = \eta_2$
- Sends $< R_1, R_2, \prod 1, \prod 2 >$ to the smartphone

Round 7 (Smartphone):

- Verifies $\prod 1, \prod 2$
- $c \overset{R}{\leftarrow} Z_{q^8}$
- $m = H(M)$
- $r = H^{'}(R1) \in Z_q$
- $\mu^{'} = E(z_3)$
- $\mu = [(mz_3\ mod\ q) \otimes \alpha_2] \otimes [(rx_3 z_3 \otimes \beta_2] \otimes E(cq)$
- $\prod 3 \leftarrow znp\ \eta_1, \eta_2 \in [-q^3, q^3]$ such that:
  - ${R_3}^{\eta_1} = G$ and $G^{\eta_2/\eta_1} = y_3$
  - $D(\mu) = m\eta_1 D(\alpha_2) + r\eta_2 D(\beta_2)$
  - $D(\mu^{'}) = \eta_1$
- Sends $< \mu, \mu^{'}, \prod 3 >$ to the online server and $< \mu, \mu^{'}, \prod 2, \prod 3 >$ to the user's PC

Online Server:

- Verifies $\prod 3$

Desktop:

- Verifies $\prod 2, \prod 3$

Finally:

All the three parties call the distributed decryption protocol for D in order to get $s = DEC(\mu) mod\ q$

then the parties output the signature for $M\ \sigma = (r, s)$.

Security Analysis

The three-factor wallet prevents attackers who manage to steal the first two key shares from spending the bitcoins in the victim's wallet because they need the third share which resides on the online server.

When the user initiates the Bitcoin transaction and sends it to the online server, the online server has to check the authorisation of this request to verify that the user is legitimate before starting the threshold signature with the user. This check can be done by adding a special encrypted Token pattern comprised of a timestamp value, nonce, user's IP address, and MAC address. The user can determine the trusted devices. Therefore, the server will hold a whitelist of the public IP and the MAC addresses for the devices from which the user can perform requests. In fact, the server uses the IP Geolocation which links the IP address to an Internet-connected real-world geographical location of the desktop and the mobile devices; the IP address is linked to a country, region, and Internet Service provider (ISP) (Bendale and Kumar, 2014).

After successful authentication and before any transactions, the server generates this unique Token by utilising a unique key which exists only on the server. The Token is sent back to the client and inserted into a hidden field. When the user attempts to perform the transaction, the server receives the request. It then reads and decrypts the Token value using the same key that was utilised to create it. If the server is unable to correctly decrypt the Token, that means there is an intrusion attempt. Then the server should block the transaction, log it, and notify the user. If the server decrypts the Token, it will validate the values that the token was composed with to verify their validity. Firstly, the server compares the IP and MAC addresses against the current addresses that submitted the request. If they match, the server checks the timestamp by comparing it against the server's current timestamp in order to prevent the reply attacks. Finally, the server confirms that the nonce submitted with the request matches the nonce existing in the Token. If the nonce is valid, the request continues; otherwise, the transaction should be blocked. In fact, the user is allowed to determine only two trusted devices, IP addresses, and MAC addresses. If the user needs to change the devices, he must make a substitution.

Although this token eliminates some attacks as will be discussed in the evaluation chapter, it cannot prevent a powerful attacker who manages to compromise all of the information in the user's possession. Therefore, it is necessary to add an extra layer of security to provide a defence in-depth to the three-factor user's wallet. This can be accomplished by adding fraud detection software that monitors the user's behaviour on the server side. This technique captures and then analyses all of the user's web traffic from login until session completion. Machine learning techniques such as Neural Network and Random Forests can be used to detect the fraud. (Wei et al., 2013) present a framework that has proved its efficiency and accuracy by being tested in a major Australian bank. Furthermore, this framework also reduces the false positive and false negative rates; more information about this framework can be found in (Wei et al., 2013)'s paper. Entrust has reported that one of the most effective solutions against MITB is a fraud detector that monitors user behaviour (Entrust, 2014). Therefore, online Bitcoin wallets can also utilise these kinds of fraud detection applications to detect abnormal behaviour by attackers attempting to use their own devices or the victim's device.

The attacker who attempts to perform the transaction from the user's devices by altering the transaction's form values can be prevented also by running the browser's extensions in a sandbox that isolates them from other extensions, and most importantly, from the browser's internals, web pages, and the operating system resources. SandFOX is an example of a client-side browser policies that creates a sandbox environment for the Firefox browser (Saini et al., 2015).

In addition to applying an appropriate sandbox to the browser, it is important that a user who operates an online Bitcoin wallet never installs any extensions from arbitrary web sites because they might already contain malware or run with full browser privileges. Users should install extensions only from trusted stores.

Evaluation

Effectiveness against possible attacks

The proposed three-factor wallet that we proposed provides several features that help to secure it against possible attacks. The three-party threshold signature that the wallet employs helps the wallet to not become a single point of failure. Therefore, the attacker who attempts to compromise this wallet needs to attack the three different places where the three secret key shares exist. Additionally, the encrypted Token that the server generates when the user connects to it eliminates the attacker's ability to steal the user's session and make unauthorised

requests. In other words, it prevents cross side request forgery (CSRF) attacks (Blatz, 2007).

The user can only specify two devices to perform the transaction, and the devices can be changed only if the user provides second-factor authentication and verifies the change using email. If the server detects login from a device or location that the server does not recognize, it will deny the access and send an email to the user detailing the attempted login. The server asks the user to verify this change not only by the email but also by entering the second-factor authentication. Therefore, attackers attempting to replace the user's addresses with their own addresses need to obtain the second factor and also compromise the user's email. Moreover, attackers cannot easily substitute the user's email by their email to receive the verification email because they need to confirm this change using the user's second email that the user entered at the time of registration. Additionally, the fraud detection application that monitors the user behaviour that the server utilises can prevent any attempted attacks.

An attacker might manage to perform a transaction with the user's devices by changing the outgoing data such as the destination public address or the amount of bitcoins that the user attempts to send. MITB malware such as URLzone performs such attacks (Dougan and Curran, 2012). Indeed, this malware aims to steal money from the victim's bank account by modifying the form fields that the user submits to the server, such as the amount of money and the receiver's account. The server processes the modified transaction normally and cannot detect the changes. The same attack can be achieved in the online Bitcoin wallet. However, in the proposed three-factor wallet, the server sends the transaction to the mobile wallet to verify it before the three parties start signing the transaction. Therefore, the server will send the modified version to the user's mobile wallet, and the user can detect the change.

One might argue that the mobile phone can also be compromised. In this case, attackers would need to perform a sophisticated attack which requires replacing the modified transaction the server sent to the user's mobile phone, presenting the original transaction that the user requested, and then signing the attacker's transaction. This attack is more complicated than those performed by URLzone malware. Since these malwares normally run in the browser's extension, running the browser's extensions in a sandbox is important because it will prevent the malware from being able to access the browser's internals or modify any transactions.

Notably, the three-factor online wallet runs with multiple independent defences on both the server and user sides to provide a defence in depth. Therefore, if one of the defences is hacked, the others remain viable.

Availability

In the three-factor wallet, all three shares should be available to sign any transaction. Users only need to worry about the two shares that they obtain and should keep them safe, secure, and available for signing. The third share is stored on the server; therefore, the server is responsible for keeping it secure and safe. Users should trust and depend on the server to provide its share at the time of signing. However, the server might be unavailable at the very moment when the user needs to sign the transaction because it might go down for several reasons. For instance, the server may receive thousands of requests while it does not have the capacity and resources to handle them. Therefore, a load balancer should be added to distribute the requests across a number of servers (Cardellini et al., 1999) that should also have an additional secret key share in order to participate in signing the transaction. Moreover, a denial of service attack (Neumann, 2000) could occur which makes the server unavailable at a time when the user needs to sing a transaction. Therefore,

precautions, countermeasures, and mitigations against any trouble or attack should be placed on the server's side, so that the server is available on demand.

Usability

The proposed three-factor wallet is similar to the two-factor wallet in terms of usability; users only need to initiate the transaction from the first device and confirm it from the second device. Therefore, the three-factor wallet does not add extra overhead to the user. In fact, with online banking, users usually have to use their phone to receive the SMS TAN and then enter it to complete the transaction. Therefore, the usability of three-factor wallet is reasonable.

Time Efficiency

The proposed threshold signature scheme suggested for use with the three-factor wallet is the previously mentioned multi-party threshold signature of (Gennaro et al., 2016). They evaluated the performance of this protocol and reported that when three parties participate in the protocol, the execution time was approximately 6 seconds, comparable to online banking where the user also needs to wait several seconds to receive the SMS. Therefore, applying the multi-party threshold signature scheme for 3-out-of-3 threshold cases is efficient according to (Gennaro et al., 2016)'s evaluation.

Conclusion

This article found that employing three-factor for a Bitcoin wallet is more secure than the two-factor which was found to be in a higher risk. A three-factor was explained using 3-out-of-3 signature protocol. Three-factor wallet employed 3-out-of-3 signature protocol in which the secret shares are distributed not only between the user's devices but also on an online server. This method making the attempts to compromise Bitcoin wallet is more difficult as the online server needs to be compromised as well.

Three-factor wallet and its proposed mechanisms should be considered in future research. The implementation of this method should be tested using "use cases" with realistic scenarios by using several techniques to verify that this solution fulfils the security controls that are claimed. The evaluation of this proposal is required to validate its effectiveness and identify potential improvements.

References

1. BARTH, A., FELT, A. P., SAXENA, P. & BOODMAN, A. Protecting Browsers from Extension Vulnerabilities. NDSS, 2010. Citeseer.

2. BENDALE, J. & KUMAR, J. R. 2014. Review of different IP geolocation methods and concepts. *International Journal of Computer Science and Information Technologies,* 5**,** 436-440.

3. BLATZ, J. 2007. CSRF: Attack and Defense. *McAfee® Foundstone® Professional Services, White Paper*.

4. BLOCKCHAIN.INFO. 2016. *Bitcoin News_ Live Bitcoin Exchange Rate* [Online]. Available: https://blockchain.info/markets/ [Accessed June 2016].

5. CARDELLINI, V., COLAJANNI, M. & YU, P. S. 1999. Dynamic load balancing on web-server systems. *IEEE Internet computing,* 3**,** 28-39.

6. DMITRIENKO, A., LIEBCHEN, C., ROSSOW, C. & SADEGHI, A.-R. On the (in) security of mobile two-factor authentication. International Conference on Financial Cryptography and Data Security, 2014. Springer, 365-383.

7. DOUGAN, T. & CURRAN, K. 2012. Man in the browser attacks. *International Journal of Ambient*

Computing and Intelligence (IJACI), 4**,** 29-39.

8. ENTRUST 2014. Defeating Man-in-the-Browser Malware.

9. ESKANDARI, S., CLARK, J., BARRERA, D. & STOBERT, E. A first look at the usability of bitcoin key management. 2015. Workshop on Usable Security (USEC.

10. GENNARO, R., GOLDFEDER, S. & NARAYANAN, A. Threshold-optimal DSA/ECDSA signatures and an application to Bitcoin wallet security. International Conference on Applied Cryptography and Network Security, 2016. Springer, 156-174.

11. GOLDFEDER, S., GENNARO, R., KALODNER, H., BONNEAU, J., KROLL, J. A., FELTEN, E. W. & NARAYANAN, A. 2015. Securing Bitcoin wallets via a new DSA/ECDSA threshold signature scheme. Accessed 2015-06-09.

12. HILL, A. 2014. Bitcoin: Is Cryptocurrency Viable?

13. J.D. MEIER, A. M., MICHAEL DUNNER, SRINATH VASIREDDY, RAY ESCAMILLA, ANANDHA MURUKAN. 2003. *Chapter 3: Threat Modeling* [Online]. Available: https://msdn.microsoft.com/en-us/library/ff648644.aspx [Accessed 20 Aug 2016].

14. JOHNSON, D., MENEZES, A. & VANSTONE, S. 2001. The elliptic curve digital signature algorithm (ECDSA). *International journal of information security,* 1**,** 36-63.

15. MACKENZIE, P. & REITER, M. K. Two-party generation of DSA signatures. Annual International Cryptology Conference, 2001. Springer, 137-154.

16. MANN, C. & LOEBENBERGER, D. Two-factor authentication for the Bitcoin protocol. International Workshop on Security and Trust Management, 2015. Springer, 155-171.

17. MÁRIO ALMEIDA, U. B., ARAS TARAHAN 2011. Man-in-the-Browser Attacks.

18. NARAYANAN, A., BONNEAU, J., FELTEN, E., MILLER, A. & GOLDFEDER, S. 2016. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press.

19. NEUMANN, P. G. 2000. Denial-of-service attacks. *Communications of the ACM,* 43**,** 136-136.

20. SAINI, A., GAUR, M. S., LAXMI, V. & NANDA, P. sandFOX: secure sandboxed and isolated environment for firefox browser. Proceedings of the 8th International Conference on Security of Information and Networks, 2015. ACM, 20-27.

21. STEWART, J. 2014. *Cryptocurrency-Stealing Malware Landscape* [Online]. Available: https://www.secureworks.com/research/cryptocurrency-stealing-malware-landscape [Accessed 20 July 2016].

22. WEI, W., LI, J., CAO, L., OU, Y. & CHEN, J. 2013. Effective detection of sophisticated online banking fraud on extremely imbalanced data. *World Wide Web,* 16**,** 449-475.