Analysis of Man-In-The-Middle of Attack on Bitcoin Address

Abba Garba[1,3], Zhi Guan[2,3,*], Anran Li[1,3] and Zhong Chen[1,3]
[1]*Institute of Software, EECS, Peking University, China*
[2]*National Engineering Research Center for Software Engineering, Peking University, Beijing, China*
[3]*MoE Key Lab of High Confidence Software Technologies, Peking University, Beijing, China*

Keywords: Bitcoin, MITM Attack, Blockchain.

Abstract: In cryptocurrency systems such as Bitcoin, user use string-hashes from public keys, that look like random strings, to receive payments. Unfortunately, there is no authority to verify user identity. Normally a user cannot prove the address binds with her real identity. Technically, a victim could get a tampered address and pay coins to this tampered address. In this paper, we report on the large-scale of Bitcoin addresses, including secured and unsecured merchants websites, exchange platforms, online chat forums, social channels and blogs. We illustrate our data through a range of graphs based on transaction distribution. Our analysis consists of crawling many web pages related to cryptocurrency transactions. We scrap the web pages by persing 10,0045 bitcoin addresses related to merchants or individuals that receive bitcoin in their websites directly. We determine how many addresses are subject to Man-in-the-middle of attack in our analysis. We review some countermeasures from best practices of Bitcoin transactions.

1 INTRODUCTION

Bitcoin is a p2p electronics digital money whose value is not dependent on any financial institution, rather, it is based on the perception of the participants in the decentralized p2p network (Lischke and Fabian, 2016). In December, 2017 Bitcoin market capitalization soared to 240 billion dollars. Bitcoin is endorsed by users in a decentralized networked system, the overall consensus and guarantee for integrity of the system rely upon solving computational puzzles in a distributed replicated global ledger called blockchain (Bartoletti and Pompianu, 2014). The participants in the global ledger are cryptographically signed a secure list of transactions via consensus agreement among the nodes in the network (Nakamoto, 2008).

During the last few years bitcoin and other alternate cryptocurrencies have increasingly become popular mediums for exchanging assests over the internet. Companies from various industries around the world have started accepting Bitcoin as a means of payment. Some product vendors such as Dell and Lenovo accept bitcoin on their websites. Overstock and other service providers such as WordPress adopted Bitcoin as an optional payment method (Soska and Christin, 2015). Many organizations and individual placed their Bitcoin addresses on web pages related to forum posts, blogs, and social media for the purpose of receiving Bitcoin (Ateniese et al., 2014). For example `http://www.bitcoinate.org` receives bitcoin as donations, the web site is not secure by (CA). Many Product vendors and service providers around the world accept payment with bitcoin as listed on `https://steemit.com`.

To receive payment in Bitcoin, a payee anonymously publishes her address over the internet. After the payer transfers coins to payee's address, the payee redeems the transaction with her private key through p2p network protocol(Nakamoto, 2008).

Bitcoin as a system and its official implementation does not provide an integrated mechanism to check the authenticity of the addresses, user cannot identify the payment source (Fleder et al., 2015; Maesa et al., 2017).

However, little attention has been focused on the security of Bitcoin addresses placed randomly on the over the internet. This may allow a Man-in-the-Middle (MitM) attack(Callegati et al., 2009; Cheng et al., 2010; Cheng et al., 2010; Stricot-Tarboton et al., 2016) to tampers payee's address on the web pages.

*Corresponding author.
This work is supported by the National Key Research and Development Program of China NO.2018YFB0803601.

Garba, A., Guan, Z., Li, A. and Chen, Z.
Analysis of Man-In-The-Middle of Attack on Bitcoin Address.
DOI: 10.5220/0006864003880395
In *Proceedings of the 15th International Joint Conference on e-Business and Telecommunications (ICETE 2018) - Volume 2: SECRYPT*, pages 388-395
ISBN: 978-989-758-319-3

A Man in the middle (MitM): is the common type of attack used in communication between two parties over the internet. A third party *(attacker)* maliciously gains control of the communication channel in order to intercept, listen and change the content of the message without either party suspecting. Thus, MitM can occur in a various communication channels such message exchange between two parties over the internet using Bluetooth, NFC and Wifi access point (Conti et al., 2016).

In this paper we begin by examine how MitM attack is subject to tampering with Bitcoin addresses over the internet using HTTP/HTTPS. Background and related work of bitcoin address and other cryptocurrencies in Section 2; We present our system model in section 3; Section 4 we describe the methodology and analysis of Bitcoin addresses; Section 5 counter measures based on best practices in bitcoin transaction; finally conclusion and future work in section 6.

2 BACKGROUND AND RELATED WORK

In this section, we first introduce how Bitcoin addresses are constructed, we also look at several methods of sending and receiving bitcoin. We also describe several addresses of other digital currencies. Finally we explain related study.

2.1 Bitcoin

Transaction in Bitcoin is a cryptographically signed statement that transfer an exact sum of coins from a sender to receiver's address (Nakamoto, 2008). For a transaction, the Payer proves ownership of amount of Bitcoin using her private key, which already appeared in a ledger that moves amount of Bitcoin to her address. Payee use her key pair to receive amount of Bitcoin to her address. A Bitcoin address is constructed from public portion of a public/private keypair of ECDSA a name curve `secp256k1`. Client can sign transaction with her private key and anyone who knows the hex-hashes of the public key of `Bitcoin address` can verify that the signature is valid. One-way-cryptographic secure hash function is computed together with `RIPEMD` to generate bitcoin address. Given the elliptic curve domain parameter (p,a,b,G,n), the user's key pair is $(d,P=dG)$, the address is form by the application of:

$$hash = RIPEMD160(SHA256(ECPublickey))$$

$$checksum = SHA256(SHA256(prefix||hash))[0...3]$$

$$Address = Base58(prefix||hash||checksum)$$

As described above, Bitcoin address construction where the checksum is the first 4 bytes of the `double-SHA256` of the concatenated version and public key hash. The *Base*58 is a text encoding method. Although the address is the hash of the public key, it is used as the identity of a user in Bitcoin transaction. Bitcoin wallet allows you to create as multiple addresses for anonymous transaction. Example of bitcoin address can be represented as: `1A1zP1eP5QGefi2DMPTfTL5SLmv7DivfNa`

2.2 Types of Transaction Scripts in Bitcoin

There are various scripts used to manage transfer of asset from payer to the payee in the bitcoin network, the standard types of transaction script for sending and receiving payment in the bitcoin address are:

Pay-to-Public-Key-Hash (P2PKH): is a form of script that mostly used in bitcoin for making transactions,P2PKH scripts is computed by sending the public key and a digital signature made by the corresponding private key. The standard formats of P2PKH are:

`ScriptPubkey: OP_DUP OP_HASH160 <pubKeyHash>OP_EQUALVERIFY OP_CHECKSIG`
`Scriptsig:<sig><pubKey>`

Other scripts used in bitcoin transaction include the followings: Pay-to-Script-Hash(P2SH), Pay-to-MultiSig(P2MultSig), Pay-to-MultiSig(P2MultSig), Pay-to-Public Key(P2PK).

2.3 Other Digital Currency Addresses

Other cryptocurrencies used the same cryptographic algorithms or slightly different from bitcoin to construct address are describe below:

Litecoin: in litecoin address is generated based on the digital signatures and combination of 33 hexadecimal values; litecoin address always started with letter L. That is based on: *based*58*encoded* hashes of user public keys. In litecoin address formation is similar to bitcoin: ECDSA ← SHA256 ← RIPEMD160

Ripple: is a decentralized consensus payment system that allows people to make cross border transactions. (Armknecht et al., 2015). Ripple use same address generation as Bitcoin.

Dash: Dashcoin addresses format is similar to bitcoin.

Ethereum: An Ethereum address can be constructed based on the following formation:

$$hash \leftarrow Keccak256(pubkey) =$$
$$Address \leftarrow hash(Hash[13..32]) =$$

Monero: Monero's address use a combination of 95 alphanumeric characters. As Sarang (2017) highlighted, two key pairs are used to create Monero addresses: spend and view keys. In Monero *ed25519* keys is used to represent the pair of public keys, addresses are formed by hashed of the KECCAK-256. A public key pair and a spend key pair is added to the prefix of the network byte plus checksum to form a Monero public key address:

$$data \leftarrow network||pubkey_{spend}||pubkey_{view}$$
$$hash \leftarrow Keccak256(data)$$
$$Address \leftarrow Base58(data||hash[0...3])$$

Zcash: Unlike Bitcoin, Zcash uses two distinct types of public key addresses, shield and transparent addresses (Sasson et al., 2014).For example: T_addr is transparent transaction on the blockchain or Z_addr is hiding the transaction information to the public.

2.4 Authenticating Bitcoin Address

Bitcoin transactions are conducted in different ways; the secure ways to transfer asset are using QR codes or using Bitcoin ATM machines to convert bitcoin to fiat currency. Another way to make Bitcoin transactions are via exchange platforms. Also SMSGateway is considered to be an alternative medium for Bitcoin transactions especially for non-smart phone users. In Table 1, we classify several ways of authenticating Bitcoin transactions:

Different platforms vary in terms of registration requirements to verify users. In this case, most of the platforms use method like mobile phone (2FA) and finger-print patterns to authenticate users.

2.5 Related Work

Over the past few decades there has been a research efforts on MitM in both HTTP and HTTPS transactions (Callegati et al., 2009; Cheng et al., 2010; Stricot-Tarboton et al., 2016; Aviram et al., 2016)

In 2013 Andersen proposed a protocol to extend the bitcoin address with human readable message of the recipient, Andersen's proposal used PKI standard (X.59) to ensure security of communication using HTTPS between payer and payee against MitM (Andresen, 2013; Biryukov and Pustogarov, 2015). On the other hand, (Ateniese et al., 2014) proposed a certify Bitcoin address that allows clients to send and receive Bitcoin from certified trusted third party. However, this form of proposal had faced lots of challenges such as involving third party during the transaction (Moore and Christin, 2013).

On the analysis side, present a thorough analysis of online anonymous market place over a long period of time (Ron and Shamir, 2013; Fleder et al., 2015; Koshy et al., 2014; Miller et al., 2017; Kumar et al., 2017; Pedro Moreno-Sanchez* and Kate*, 2017). To our knowledge no one has conducted a thorough analysis of a large portion of Bitcoin addresses posted in various forums and merchant's website that accept Bitcoin directly.

3 MITM ATTACK

The main concern with respect to MitM security vulnerability is that most of the Bitcoin addresses posted randomly on forum posts, blogs, merchant's web sites, social media channels are not well protected.

3.1 Security Model

Consider the Bitcoin transaction between two parties in a less secure channel where *payer* might be a victim sends a sum of coins to *payee* whereby *payee* redeems the coins.

1. The victim publishes her address on her own website without HTTPS protection or without a certificate from a trusted authority.

2. The victim publishes her address on website or a forum post without HTTPS protection or without a certificate from a trusted CA.

3. The victim publishes her address on a HTTPS protected website or forum. We consider an attacker operating in the middle between *payer* and *payee* during the transaction. The payer initiates a payment by using the payees Bitcoin address to send payment. We assume communication protocol is vulnerable such that the attacker may gain advantage between *payer* and *payee* without their knowledge and replace the victim's address.

The attacker may be a *web service provider, an Internet Service Provider (ISP) or a malicious Wi-Fi access point*. The major challenge of the attacker is to identify a valuable address from its stored content or blockchain.

Consider these two categories of attackers:

1. An attacker close to the *payee*, such that when the payee publishes her address through the attacker, the attacker will modify the target's address to attacker's address.

2. An attacker close to the *payer*, the payer retrieves the payee's address from the attacker, and the

Table 1: Bitcoin address authentication via platforms.

Transaction Categories	Example	Veri. ID	Possible attacks
Exchange Platforms	Mt. Gox,Huobi, Silk road, Karen and bitsquare.	Yes	DDoS, Double spend, Ponzi scheme, Txt Malleability attacks
Merchant's websites	Overstock, Dell , dish, cheapair.	Yes	MITM
Unprotected websites	Chat forums, Blogs, Social media	No	MITM, DDoS, Phishing attacks, Phone-porting attacks.
SMSGateway	37coin, Coinpip, Coinapault's	Yes	DDoS, IMSI attacks
Email	Coinbase, Blockchain	Yes	DDoS, Phone-porting attack s
Bitcoin ATMs	Robocoin,Genesiscoin, Lambassu,General bytes, Coinsource	Yes	Phishing attack
QR code, Bluetooth/ NFC	Bitcoin wallets, Airbitz, Box tip	No	MITM, Relay Attacks

attacker easily modifies the address with her own address.

In this case, how can an attacker identify high value addresses? Initially the attacker can easily filter potential Bitcoin addresses from web content and then use its inner checksum to verify that it is a Bitcoin address. The major computation for an attacker is the *REGEX filtering* and the double *SHA-256 checksum* generation. Once obtained an address, the attacker can check the Bitcoin blockchain to see if it is a persistent address with high receiving coins. After that, the attacker can replace the address with her own. Ironically, attacker can use *one-time address* to keep privacy.

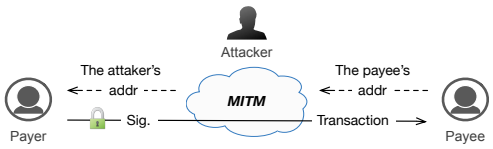

Figure 1: Indicate the Man-in-the-middle of attack: Temper with addresses from both victims.

It is very easy for an attacker to generate a lot of addresses *on-the-fly* with some hierarchical address generation methods (Wallets).

In this case, MitM is very powerful as the nature of HTTP connection data can be transferred in the form of plain text. Attacker may easily replace Bitcoin addresses placed randomly on the web pages using HTTP. The attacker can then simply modify the content of the web page that contain Bitcoin addresses and replace them with an address under her control. On the other hand, HTTPS could suffer the same attacks if the certificate is not valid because its security guarantees ties on validity of the certificate.

4 ANALYSES OF BITCOIN ADDRESSES

We combine different methodologies to analyze bitcoin addresses collected from websites and global ledgers. Websites consist of crawling many web pages related to cryptocurrency transactions and Bitcoin blockchain were accessed through Blockchainifo. We used this information to classify addresses based on the transaction distribution. We show how REGEX filtering can be used to obtain the Bitcoin addresses in the network traffics. We also analyze how many proportions of addresses use HTTP/HTTPS during transaction. Finally we examine number of active and non-active addresses.

4.1 Methodology

First Step: Scraping the Websites
We designed and developed a few lines of code that allow us to crawl many web pages to extract Bitcoin address transactions and validate using `REGEX`, we then scrap web pages to get the relevant data we needed.

Second Step: Parsing the Websites
Different websites needed to be parsed to extract information associated with each bitcoin address. Considering the number of addresses collected was large (10,045), a great deal of manual work were needed

to collect and record each address using (*Googledoc*) with its associated transactions such as: Bitcoin received, number of transaction inputs. In this case, we first identified addresses scraped from the website and then we categorized the addresses in to five: Donations, Crowd funds, Ransoms, Merchants and Others.

Table 2: The table describe the addresses crawled from different source based on the category.

Description	No. Of Addresses
Donations	3,282
Merchants	2,667
Crowd funds	1,326
Others	2,108
Ransoms	662
Total Addresses	**10,045**

Table 2: Indicated a statistics we looked at based on the transaction distribution in our analysis result, realized that 33% with 3,282 addresses from donations, as compared to merchants 27% with 2,667 addresses. For others category %13 collected which resulted to 1,326 addresses as compare to crowd funds which resulted to 20% with total addresses of 2,108, while ransoms category indicated that 7% with total addresses of 662. Our results strongly shown that large number of bitcoin addresses were subject to man-in-the-middle-of-attack.

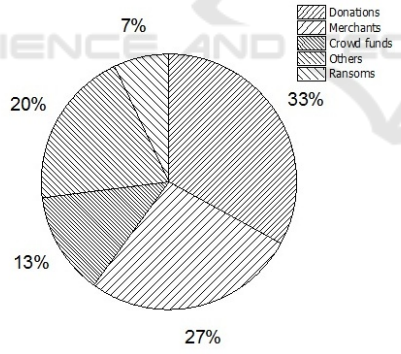

Figure 2: Shows percentage of Bitcoin addresses by category.

Third Step: Filtering Addresses in Network Traffics

Bitcoin and other alternate currencies use different format to represent address, we use `REGEX` filtering to extract the potential addresses from a web content. Furthermore in recent survey reveal that the average percentage of text on web pages accounted to 26.88% as compare to early 90s (Cocciolo, 2015).

Table 3. Indicates example of `REGEX` formats for the bitcoin and other alternate currencies mentioned in this paper.

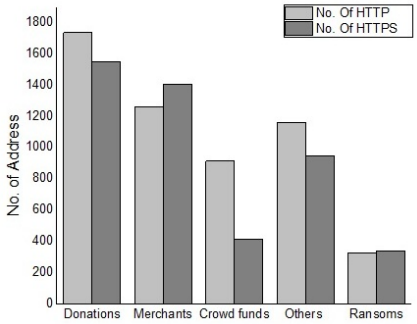

Figure 3: Shows transaction of BTC in HTTP/HTTPS.

4.2 Taxonomy of Bitcoin Addresses

We then present summary of the analysis of bitcoin addresses based on transaction distributions.

After analyzing the addresses, we realized that there were Total of 845,370.19BTC associated with 10,045 addresses collected in addition to that 749,894.00 addresses were involved in the input transactions with each corresponding addresses.

As depicted in table 4. There was variance between the following corresponding categories: Donation addresses received 278,972.161BTC, with input transactions of 247,465 addresses, resulted in 33% of the total receiving bitcoin, followed by merchant addresses receiving 228,249.95BTC, with 202,491.00 associated input addresses involved during the transactions. Crowd funds received 109,898.13BTC with 97,486.00 addresses involved; while others category received 169,074.04BTC with 149,978 corresponding addresses. For simplicity we considered remaining addresses as ”Others” because it accounts for many addresses collected over the internet. Ransoms has a lowest receiving addresses of 59,175.100BTC with total number of transactions of 52,492.

4.3 Transaction of Bitcoin on HTTP/HTTPS

We collected bitcoin addresses from various websites both with HTTP/HTTPS. We constructed a graph to represent the activities associated with each address such as balance and number of transaction inputs.

We then presented our data as depicted in Figure 5, by analyzing users activities associated with their addresses. We also categorized percentage of transaction addresses using HTTPS/HTTP as shown in table 5.

We examine that, our results shown in Table 5. 54% of the total addresses we collected conducted transactions over the internet without authentication, while only 46% have proper security using HTTPS.

Table 3: This table shows an example of regular expression bitcoin and alternate digital currencies.

Cryptocurrency	Regular Expression	Crypto-Algorithms
Bitcoin	[13][a-km-zA-HJ-NP-Z1-9]{25,34}	ECDSA, SHA-256
Litecoin	[LM3][a-km-zA-HJ-NP-Z1-9]{26,33}	Scrypt, SHA-256
Ripple	r[0-9a-zA-Z]{33}	ECDSA, SHA-256
Dash	X[1-9A-HJ-NP-Za-km-z]{33}	ECDSA, SHA-256
Ethereum	(0x)1[0-9a-fA-F]{40}	ECDSA, KECCAK-256
Monero	4[0-9AB][1-9A-HJ-NP-Za-km-z]{93}	EdDSA, KECCAK-256
Zcash	([t][a-km-zA-HJ-NP-Z1-9]{34})|([z][a-km-zA-HJ-NP-Z1-9]{95})	zk-SNARK, SHA-256

Table 4: Taxonomy of Bitcoin Address.

Description	No. Of Address	No. Of TxT	BTC Received
Donations	3282	247,465	278,972.161BTC
Merchants	2,667	202,491.00	228,249.95BTC
Crowd funds	1,326	97,486.00	109,898.13BTC
Others	2,108	149,978	169,074.04BTC
Ransoms	662	52,492	59,175.100BTC
Total Addresses	**10,045**	**749,894.00**	**845,370.19BTC**

Table 5: Shows Pct. of addresses use HTTP and HTTPS.

Addresses without CA		Addresses with CA	
No. Of addresses	5,393	No. Of Addresses	4,652
Pct.%	48%	Pct.%	52%

Our findings indicated that lots of addresses were vulnerable to *tampering address* using MitM.

Active and non-active addresses: We examined number of active and non-active addresses in table 6 to present our results. As depicted in the table 6 indicated that nearly 1,660 of the addresses collected were dormant which accounted to only 17% out of the total sample of the addresses we collected in this paper. This indicated that about 83% of the addresses in our analysis are active receiving transactions.

Table 6: Indicate addresses that are active and non active.

	Addresses	
	Non Active	Active
Donations	382	2900
Merchants	763	1904
Crowd funds	259	1067
Others	192	1916
Ransoms	64	598
	1660	**8385**
	17%	**83%**

4.4 Regular Expression Performance Evaluation

As shown in the Table 7 we used bitcoin to demonstrate regular expression to extract bitcoin address from the web page. For example btccharity.html is a bitcoin donation page and the ordinary.html is a normal web page without bitcoin address. We run a few lines of Python code to extract bitcoin address. We used MacBook Pro with processor: 2.5GHz, Intel Core i7, Memory: 16GB, and single thread program for our experiment. We found out that the script could deal with more than 40 MB data per second. Table 8 depicted the summary of our experiment.

5 COUNTER MEASURES

5.1 Vanity Address

A vanity address is an address with some meaningful sub-string in the address. For example, the Bitcoin address of a valid key pair (d, P) is: `1anaLysis Vj8ALj6mfBsbifRoD4miY36v`. The vanity address is produced through brute-force searching of elliptic curve key pairs that can produce an address with a pre-defined pattern. The Bitcoin project provides a command line tool, `vanitygen` for the generation of vanity addresses.

Let's assume *payee* has a public identity. This identity is also available to the *payer*. This is a reasonable assumption because the payer should always know who she intends to pay. The idea is the *payee*

Table 7: Shows performance evaluation of Regular filtering of bitcoin address in the network traffic.

Files		**Run Times**				**Speed(MB/s)**	
Name	Size(KB)	Rounds	RGX (ms)	Validate(ms)	TT(ms)	RGX	TT
btccharity.html	8.9	1	0.212	1.217	1.429	40.997	6.082
ordinary.html	90	1	1.763	1.374	3.137	49.853	28.017
btccharity.html	8.9	100	20.394	139.270	159.664	42.617	5.443
ordinary.html	90	100	176.611	133.890	310.501	49.765	28.306
btccharity.html	8.9	10000	1965.440	12647.865	14613.305	44.221	5.947
ordinary.html	90	10000	18040.518	13574.551	31615.069	48.718	27.800

will generate a vanity address and use her identity as the pattern. When the *payer* gets the address, she will then check if the address is a vanity address with the corresponding identity.

The client can generate a key pair (d,P) and send the public key P to a vanity address generation service provider, the service provider can enumerate all the addresses from public key $P+iG$, where G is the generator and increment i from 1 until when $i=k$ a vanity address is found. The service provider send k to the client, the corresponding key pair with vanity address is $(d+k,P+kG)$. This method means the client can generate a vanity address with longer prefix identity through outsourcing the computation to the cloud.

Difficulty of Generating Vanity Address: The difficulty of generating specific vanity address is calculated based on the bitcoin address generation process. For example:

$$\frac{Number\ of\ possible\ address}{Number\ of\ address\ with\ vanity prefix}$$

In this case, the accurate formula for generating vanity address is complex. Therefore we find that if we want to generate a prefix with '1b' we need only 22 times attempts and a prefix with '1bi' it requires 1330 times attempts. With longer prefix, we can approximately compute the difficulty with the following formula:

$$diff_n = diff_{n-1} \cdot (58 \pm \Delta), n >= 3, 0 \leq \Delta \leq 0.1 \quad (1)$$

In our experiment, the speed to generate address is about 75 Million keys/s using the nVidia GeForce GTX 1070 Ti. With the above formula, we can generate a 6 length prefix (for examples '1bitcoi') will take about 200 seconds.

5.2 Alternative to Vanity Address

a) Outsource Vanity address:
The security of vanity address against MitM attack is based on the following hypothesis:

First, the victim's identity can not be guessed previously by the attacker, so the attacker has to do on-the-fly computation to generate a vanity address with the same prefix identity. So if the victim is a specific attack target or the victim's identity is included in a pre-defined identity set, than the attacker can precompute all the victims' addresses.

Second, the valid period of the vanity address should be shorter than attacker's address generation time. Normally an attack as a router only has very limited time for vanity address generation before the user notice the network delay. But for an attacker as a web server owner, the attacker have enough time to modify the address on the pages. This indicates that vanity address can not be used to protect long term addresses such as a fixed address for donation. The payer need to finish the transaction before the web server owner has the time to modify the address. And the payee should not re-use the same identity.

b) Blockchain Domain service:

Consider blockchain name services such as: Blockstack, Ethereum domain services and Peernames.

c) Anti-tampering address mechanisms:

Bitcoin and its original implementation was not built-on with mechanisms to check the integrity of public address during transaction, client may check the address before sending the payment. Verification mechanism helps client to verify whether the address is related to the receiver.

e) Integrating HTTPS with X.509 during payment: Another solution is to create random payment ID address using existing framework such (X.509) certificate of authentication. Always consider transaction via HTTPS instead of of HTTP because it is difficult for MitM to tamper with transaction conducted on website that has a trusted certificate of authentication using (X.509) as compare to HTTP.

6 CONCLUSIONS AND FUTURE WORK

We have analyzed a large portion of bitcoin addresses placed on web pages randomly. In our analysis, we have demonstrated that this creates significant security challenges. Particularly, we showed that MitM attacks may tamper with a victim's address posted on web site that are not well secured. Alternative digital currencies following Bitcoin may also face the same security challenges. In summary, this form of attack can happen not only with bitcoin addresses but with any unauthenticated information. Our counter measures will provide sufficient guidelines to users who posted their bitcoin addresses on web pages randomly. Future research should consider the potential effects of MitM and bitcoin and alternate cryptocurrencies transactions on HTTP/HTTPS.

REFERENCES

Andresen, G. (2013). Payment protocol.

Armknecht, F., Karame, G. O., Mandal, A., Youssef, F., and Zenner, E. (2015). *Ripple: Overview and Outlook*, pages 163–180. Springer International Publishing, Cham.

Ateniese, G., Faonio, A., Magri, B., and De Medeiros, B. (2014). Certified bitcoins. In *International Conference on Applied Cryptography and Network Security*, pages 80–96. Springer.

Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J. A., Dukhovni, V., et al. (2016). Drown: Breaking tls using sslv2. In *USENIX Security Symposium*, pages 689–706.

Bartoletti, M. and Pompianu, L. (2014). An analysis of bitcoin op return metadata. https://arxiv.org/pdf/1702.01024.pdf.

Biryukov, A. and Pustogarov, I. (2015). Bitcoin over tor isn't a good idea. In *Security and Privacy (SP), 2015 IEEE Symposium on*, pages 122–134. IEEE.

Callegati, F., Cerroni, W., and Ramilli, M. (2009). Man-in-the-middle attack to the https protocol. *IEEE Security & Privacy*, 7(1):78–81.

Cheng, K., Gao, M., and Guo, R. (2010). Analysis and research on https hijacking attacks. In *Networks Security Wireless Communications and Trusted Computing (NSWCTC), 2010 Second International Conference on*, volume 2, pages 223–226. IEEE.

Cocciolo, A. (2015). The rise and fall of text on the web: a quantitative study of web archives. *Information Research: An International Electronic Journal*, 20(3):n3.

Conti, M., Dragoni, N., and Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE Communications Surveys & Tutorials*, 18(3):2027–2051.

Fleder, M., Kester, M. S., and Pillai, S. (2015). Bitcoin transaction graph analysis. *arXiv preprint arXiv:1502.01657*.

Koshy, P., Koshy, D., and McDaniel, P. (2014). An analysis of anonymity in bitcoin using p2p network traffic. In *International Conference on Financial Cryptography and Data Security*, pages 469–485. Springer.

Kumar, A., Fischer, C., Tople, S., and Saxena, P. (2017). A traceability analysis of moneros blockchain. In *European Symposium on Research in Computer Security*, pages 153–173. Springer.

Lischke, M. and Fabian, B. (2016). Analyzing the bitcoin network: The first four years. *Future Internet*, 8(1):7.

Maesa, D. D. F., Marino, A., and Ricci, L. (2017). Data-driven analysis of bitcoin properties: exploiting the users graph. *International Journal of Data Science and Analytics*, pages 1–18.

Miller, A., Möser, M., Lee, K., and Narayanan, A. (2017). An empirical analysis of linkability in the monero blockchain. *arXiv preprint arXiv:1704.04299*.

Moore, T. and Christin, N. (2013). Beware the middleman: Empirical analysis of bitcoin-exchange risk. In *International Conference on Financial Cryptography and Data Security*, pages 25–33. Springer.

Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

Pedro Moreno-Sanchez*, M. B. Z. and Kate*, A. (2017). Listening to whispers of ripple: Linking wallets and deanonymizing transactions in the ripple network. *Proceedings on Privacy Enhancing Technologies ; 2016 (4):436453*.

Ron, D. and Shamir, A. (2013). Quantitative analysis of the full bitcoin transaction graph. In *International Conference on Financial Cryptography and Data Security*, pages 6–24. Springer.

Sasson, E. B., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E., and Virza, M. (2014). Zerocash: Decentralized anonymous payments from bitcoin. In *Security and Privacy (SP), 2014 IEEE Symposium on*, pages 459–474. IEEE.

Soska, K. and Christin, N. (2015). Measuring the longitudinal evolution of the online anonymous marketplace ecosystem. In *USENIX Security Symposium*, pages 33–48.

Stricot-Tarboton, S., Chaisiri, S., and Ko, R. K. (2016). Taxonomy of man-in-the-middle attacks on https. In *Trustcom/BigDataSE/I? SPA, 2016 IEEE*, pages 527–534. IEEE.