Enhancing the Security of the Bitcoin Wallet Master Seed

♦ Recepción/ 27 junio 2019
♦ Aceptación/ 25 agosto 2019

Enhancing The Security Of The Bitcoin Wallet Master Seed

Mejorando La Seguridad De La Semilla Maestra Bitcoin Wallet

Farah Maath Jasim [a,*],
Ali Makki Sagheer [b],
Abdullah M. Awad [c]

[a,*] College of Computer Science and Information Technology,
University of Anbar, Ramadi, Iraq,
farahmaath86@gmail.com
[b] Al-Qalam University College
Kirkuk, Iraq,Prof.ali@alqalam.edu.iq
[c] College of Computer Science and Information Technology,
University of Anbar, Ramadi, Iraq
Abdullah.alawad@yahoo.com

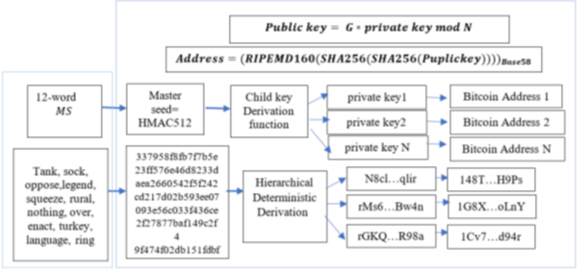
ABSTRACT/ Bitcoin cryptocurrency is a peer-to-peer electronic cash system. It is largely used for financial transactions on the Internet. Bitcoin has gained popularity due to its anonymity, privacy, and comparatively low transaction cost. Nevertheless, the use of Bitcoin as an emerging technology comes with challenges and various types of threats are associated with its use. However, the Bitcoin wallets security are one of the salient challenges. In a Bitcoin wallet, the user's funds are protected by Elliptic Curve Digital Signature Algorithm (ECDSA) private keys. The wallet generats these keys from a secure master seed of 512 random bits.Thus as a result, the Bitcoin wallet became a desirable target for attacks such as dictionary attacks. To avoid such attacks, the master seeds must be sufficiently long and have a very high level of entropy in which several currently used the Bitcoin wallets lack . In this paper, our aim is to enhance the security of the master seed of the Brain wallet achieving through introduction of additional entropy source, a Unicode model with no fixed ASCII code. Finally, the findings of this paper prove that the proposed encoding model ensured the resistance of the secure master seed generations against the dictionary attack. Keywords: Bitcoin wallet, Dictionary attack, ECDSA, Entropy,Unicode. **RESUMEN / la criptomoneda de Bitcoin es un sistema de efectivo electrónico de igual a igual. Se utiliza principalmente para transacciones financieras en Internet. Bitcoin ha ganado popularidad debido a su anonimato, privacidad y costo de transacción relativamente bajo. Sin embargo, el uso de Bitcoin como tecnología emergente presenta desafíos y varios tipos de amenazas están asociadas con su uso. Sin embargo, la seguridad de las billeteras Bitcoin es uno de los principales desafíos. En una billetera Bitcoin, los fondos del usuario están protegidos por claves privadas de Algoritmo de firma digital de curva elíptica (ECDSA). La billetera genera estas claves de una semilla maestra segura de 512 bits aleatorios, por lo que la billetera Bitcoin se convirtió en un objetivo deseable para ataques como ataques de diccionario. Para evitar tales ataques, las semillas maestras deben ser lo suficientemente largas y tener un nivel muy alto de entropía en el que carecen actualmente varias billeteras de Bitcoin. En este documento, nuestro objetivo es mejorar la seguridad de la semilla maestra de la billetera Brain logrando mediante la introducción de una fuente de entropía adicional, un modelo Unicode sin código ASCII fijo. Finalmente, los hallazgos de este documento prueban que el modelo de codificación propuesto garantiza la resistencia de las generaciones seguras de semillas maestras contra el ataque del diccionario. Palabras clave: billetera Bitcoin, ataque de diccionario, ECDSA, entropía, Unicode.**

Introduction

Recently, technology has yet changed the financial world with the proposition of electronic currency which became a more attractive system for banks, investors, and even developers [1]. In 2008, Satoshi Nakamoto introduced Bitcoin to the world. His peer-to-peer electronic cash system allows any two willing parties to transact directly to each other without the need for a trusted third party [2]. Bitcoin as an ecosystem is largely based on the concept of public key cryptography. Every Bitcoin user needs at least have one pair of cryptographic keys. A Bitcoin address is a 160-bit hash of the public portion of a public/private Elliptic Curve Digital Signature Algorithm (ECDSA) key pair[3]. The user can "sign" data with his\her own private key and anyone who knows his/her public key can verify that the signature is valid [4][5].

Bitcoin system consists of three main components : the Wallet, Blockchain, and Miner. Each of which has a specific role that contributes to perform the Bitcoin transactions. A Bitcoin wallet is responsible for holding the Blockchain copies, whereas the Blockchain acts as a bank ledger and it is available on the network with a private key. A Bitcoin miner is responsible for preventing the Bitcoin double spending and provide security for the transaction processes over the network figure [6][7][8]. The wallet is an interface between the user and the digital currency. As the matter of the fact, the Bitcoin wallet is a key pool or container for private keys. Technically, the wallets store the private keys required to spend bitcoins as simple database or in a structured files. Moreover, it saves a combination of balances from all Bitcoin address listed in the wallet which are considered the history of wallet transactions [5] [9].

Besides private keys, the Brain wallet introduces Mnemonic words to the Bitcoin [8]. Mnemonic Code Technique was first introduced in BIP39 protocol to establish a deterministic bijection between a wallet seed and a group of words that are easy to be remembered by wallet's user. Instead of dealing with a serial of binary or hexadecimal representations, wallets users are more likely to remember these understandable words used to securely generate and recover wallets master seed ,hence, private key[10][11]. Figure (1) shows an example of the relationship between the main seed which is the master seed and the address.



Public key = G • private key mod N
Address = (RIPEMD160(SHA256(SHA256(Puplickey))))Base58
12-word MS
Master seed= HMAC512
Child key Derivation function
private key1
private key2
private key N
Bitcoin Address 1
Bitcoin Address 2
Bitcoin Address N
Tank, sock, oppose,legend, squeeze, rural, nothing, over, enact, turkey, language, ring
337958f8fb7f7b5e 23ff576e46d8233d aea2660542f5f242 cd217d02b593ee07 093e56c033f436ce 2f27877baf149c2f 4 9f474f02db151fdbf
Hierarchical Deterministic Derivation
N8cl…qlir
rMs6…Bw4n
rGKQ…R98a
148T…H9Ps
1G8X…oLnY
1Cv7…d94r

Figure 1. Master Seed and Private Keys Relationship[11]

This paper focuses on a challenging problem arose in the domain to the security issues surrounding Bitcoin wallet.One salient issue with wallet security is the generated source of the master seed. In 2015, Brain wallet became the subject of several academic studies with much concentration given to newly introduced features [12][13][14]. The source of the master seed in Brain wallet is a Mnemonic Massphrase (MS) derived from fixed dictinary with fixed ASCII code. The key point with such technique is using fixed dictionaries with fixed ASCII code. However, to avoid dictionary or brute-force attacks, the master seed must contain at least one character with non-fixed ASCII and therefore ensuring high entropy[15]. Thus, this paper aims to improve the security of the master seed through the use of non-fixed ASCII Unicode model. In addition, a new dictionary is used as a proof of concept of the proposed model. The rest of the paper is organized as follows. Section 2 reviews related papers and researches on the security Bitcoin wallet. Section 3 presents BIP39. Section 4 discusses the proposed work and its implementation. Section 5 and Section6, present results and discussion of findings, conclusions, and future work.

Related Work

Recently, ample of academic research have been dedicated toward improving the Bitcoin wallet security. The suggestions were focusing on analyzing and studying the wallets performance to present the security weaknesses or on providing a security enhancement.

In 2015, Visser presented and built a Random Number Generator (RNG) prototype application to provide an addition source of entropy to the master seed of Bitcoin wallet. The RNG prototype generates the entropy pool which collects the random data bit from device sensors, then computes SHA256 to the collecting data, to be an input for XSalsa20 steam cipher algorithm. By using device sensor data, it is needed to take into account the limited amount of resources compared to a desktop system. Due to this limitation, the proposed solution may be considered costly. Moreover, it is not practical for other wallets devices[16].

In 2016, Vasek et al. published a cybercrime analysis of brain wallets using 300 billion passwords collected from various password lists sources, such as Urban Dictionary, English Wikipedia, Two Words, WikiQuotes, Phrases, and Password Dumps (LinkedIn, MySpace, RockYou, etc). During the period of test, over 884 brain wallets were compromised containing over 100K$ worth of bitcoin. The general conclusion given by the researchers is that weaker password choices were the main reason behind the successfulness of the attack[17].

In 2017, Gentilal designed and implemented Trust zone hardware wallet to provide more security and trust environment to store the private keys. The wallet encrypts all sensitive information by a random key to be more resilient to the dictionary and side-with channel attacks. Nevertheless, the encryption method depended on the stored key. Hence there is no chance to restore the master seed and any information in the situations of losing the wallet device. However, this paper does not concentrate on the master seed security[18].

L. Van Der Horst et al. forensically examined two popular Bitcoin clients wallets (Bitcoin Core and Electrum). Researches were able to recover data such as metadata, installation data, time stamps and usage indications using commercial forensic software for Android forensics. In one instance, the private key was also recovered. However, the proposed approach could only be used to obtain human-readable data[19].

Mann and Loebenberger demonstrated how one can introduce two-factor authentication to a Bitcoin to enhance security. However, the approach does not focus on protecting the password or data contained in the Bitcoin wallet itself from other attacks[20].

In 2019, Volety et al. examined and studied the security of two publicly available Bitcoin wallet (Multi-bit HD and Electrum). It was carried through implementing a software package to perform a dictionary attack to crack the master seed of both selected wallets. Moreover, steps in the proposed system were: (1) Extract the dictionary; (2) Create the combinations for a dictionary file and (3) Check for the correct MS. Thus ,they succeeded to restore and crack several user wallets, which were able to identify more than one combination of a given 12 words (Seed) successfully in a short time,. However, the findings of this research are only limited to these two Bitcoin wallets[21].

BIP39

BIP39 standard enhances the security of Bitcoin Barin wallet by using MS constructed from randomly selected words from a language dictionary, which could be English, French, Chines, Japanese, or Spanish. For example, a 2048 words language can generate 2048^12 MS of the length of 12 words. MS is used to generate master seed of bitcoin wallet. Using a MS enhances the entropy of the generated master seed and can be used to reconstruct the wallet during the events of wallet loss or corruption[22]. The steps of constructing a MS are summarized as follows[9]:

1. Create a random sequence (entropy) of 128 to 256 bits.
2. Create Check sum (CS) of the random sequence by taking the first bits of its SHA256hash, where$CS = ENT/32$. Table1 describes the relationship between the initial entropy lengths (ENT), CS and MS number word.
3. Add CS to the end of the random sequence.
4. Divide the sequence into sections of 11 bits, using those to index a dictionary of 2048 pre-defined words.
5. Produce 12-24 words representingMS.

Table 1: The Relationship between The Initial (ENT) and CS

ENT	CS	ENT+CS	MS word Sequences
128	4	132	12
160	5	165	15
192	6	198	18
224	7	231	21
256	8	264	24

Following BIP39, MS is encoded to ASCII byte by using Unicode Transform Function (UTF8). Then MS is encrypted by Hash Massage authentication (HMAC512) with 2048 iterations with key stretching added for the generation of the master seed. In terms of hashes, which is equivalent to adding an extra 11 bits of security to the seed (2048=2^11) [23].

Dictionary attacks are full brute force approaches. However, instead of trying every possible key space in a defined solution space, a 'dictionary' file is used, this file is called a Rainbow table. The dictionary file is a plaintext file contains a combination of all MS sequences. By its nature, Dictionaries files are UTF8-encoded file in size and usually are located in a server storage [6][24].

The Proposed Model

To ensure the proper implementation of the proposed encoding model, The structure of BIP39 was analyzed, which focus on how the MS is encoding and what are the major encoding model that is producing the

MS_byte. From this point on, it is needed to decide which is the most critical part that should be switched to a more secure part .In the proposed model, there is an extra layer of security, which was software implemented by adding the Unicode function of the encoding model. The research setup consists of three phases as follows, and illustrated in Figure (2):

- Phase 1: Preparing the environment and programming the BIP39 software, for creating mnemonics passphrase and installing other existed dictionaries, which are used later to build the proposed software.
- Phase 2: The proposed wordlist creates, then adds the proposed encoding model to a software package.
- Phase 3: Executing the proposed software to produce a new MS to generate the secret master seed.

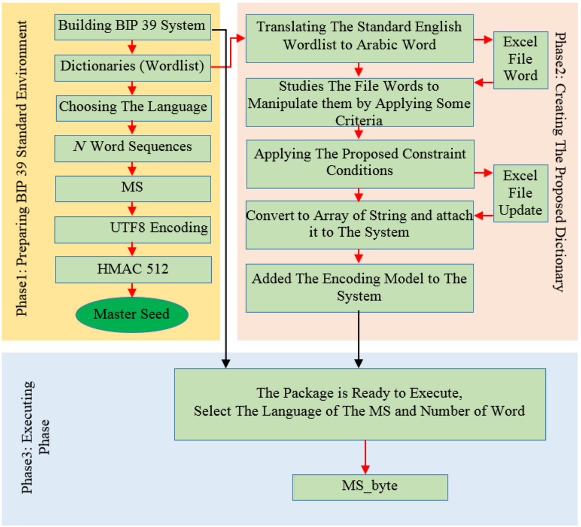

Phase1: Preparing BIP 39 Standard Environment
Building BIP 39 System
Dictionaries (Wordlist)
Choosing The Language
N Word Sequences
MS
UTF8 Encoding
HMAC 512
Master Seed
Translating The Standard English Wordlist to Arabic Word
Studies The File Words to Manipulate them by Applying Some Criteria
Applying The Proposed Constraint Conditions
Convert to Array of String and attach it to The System
Added The Encoding Model to The System
Excel File Word
Excel File Update
Phase2: Creating The Proposed Dictionary
Phase3: Executing Phase
The Package is Ready to Execute, Select The Language of The MS and Number of Word
MS_byte

Figure 2. The Proposed Platform Configuration

Recently, Bitcoin cryptocurrency users have increased in the Middle East[25][26]. Thus, this motivates us to find it necessary to add Arabic dictionary (wordlist) .which is representing a proactive idea of Arab users. Due to the main characteristics of the Arabic language such as its alphabet consists of 28 characters larger than English characters, the script is written from right to left and has no equivalent of capital letters. Furthermore, The Arabic word could be consisting of completely connected letters such as محمد, علي, نبيل, or a single word may consist of more than one components like فرح, احمد, محمود. Except for Hamza (ء), they have different shapes based on its position in the word or sub-word. Each letter may take one more of these four positions (isolated; initial; middle; final) [27]. As a result, adding such wordlist is richer and more complex than the English language. Due to its Semitic nature, several words and verbs can be constructed from one basic root. Thus, it could be difficult to crack it by the brute force attacks.

BIP 39 supports few languages and does not support many others such as the Arabic language. Technically speaking, building a new dictionary must satisfy several criteria such as smart selection of common words .and meaning-similar words should be avoided. After that, wordlist must be sorted for a more efficient lookup process [21]. In this work, the below steps were followed using Microsoft Excel software to create a 2048-word Arabic dictionary:

A) The English wordlist found in BIP39 was taken and translated into Arabic.
B) Non-compatible words with above criteria were eliminated by applying several constraint conditions, such as:
   1. Several English words have the same meaning in Arabic, i.e. they are translated to one Arabic word.
   2. Some English words are translated into a phrase in Arabic.
   3. Exclusion of prepositions and time conditions words.
   4. Time of all used verbs is present tense.
   5. Converting uncommon verbs for their derived noun to be easier to memorize and remember
   6. Replace the less common and more difficult words with more common words.
   7. English months' names are changed to their Arabic equivalent.
   8. Use of plural anomaly instead of normal plurals
   9. No pronouns, missing verbs, and questions tools.

The proposed work uses the Unicode function to ensure the conversion of words to a unique string of bytes as there is no fixed ASCII code [28].Table (2) shows some examples of shaping Unicode number for some Arabic letters. The below algorithm summarizes the implementation of passphrase encoding; depending on the selected language, the correct transformation function is called.

```
The Algorithm of Encoding Procedure
Input:- $MS$, selected language of $MS$ ($Lang$).
Output:-  Array of bytes
Step1: Start
Step2: read  MS
Step3: Lang= select the language of $MS$
        If  $Lang = "Arabic"$  then
                $MS_{byte} = Encoding \cdot Unicode \cdot Getbytes \cdot (MS)$
        $Else\ MS\_byte = Encoding \cdot UTF8 \cdot GetBytes(MS)$
Step4: return ($MS\_byte$).
Step5: End.
```

Table 2: Unicode Arabic letters in Non-ASCII code Unicode

ENT	Isolated	Initial	Middle	Final	Unicode
Hamza	ء				336
Alif	أ			ـا	356
Ayn	ع	عـ	ـعـ	ـع	576
kaf	ك	كـ	ـكـ	ـك	676
Taa1	ت	تـ	ـتـ	ـت	426
Taa2	ة			ـة	416
Wow1	و			ـو	726
Wow2	ؤ			ـؤ	366

Result and Discussion

The finding of this paper focuses on two matters: the first matter discusses the effect of increasing the byte size of the security of the master seed; the second matter is measuring the entropy of the language which has used in the proposed system. Moreover, the entropy results are compared with the entropy of one of widely used language which supported by the BIP39, to determine validity and sufficiency of the proposed dictionary as a source of master seed.

Byte Size Impact

The byte size is calculated for each sentence after the encoding process. Due to the random behavior of BIP39, It is difficult to calculate optimal size bytes for each Ms sequence of the algorithm. The minimum (Min) and the maximum (Max) byte size of all BIP39 sequences have been generated using UTF8 and the proposed Unicode through running several tests scenarios reaching to 1000 tries, see Table(3). Those experiments were conducted to examine if the proposed system improved the performance of the algorithm by increasing the size of the MS byte or not.

Table 3: MS Byte size

n-word	English passphrase		Proposed Arabic passphrase	
	Min	Max	Min	Max
12	72	80	112	132
15	90	102	140	162
18	109	126	170	194
21	122	145	198	226
24	143	161	232	266

Table (3) shows the approximate results of minimum and maximum byte size rate to the experiments MS set. As it evident when observing byte size values in the testing table, the proposed method indicated in the last two columns achieves a more size byte than the other UTF8 encodings which are the first two columns.

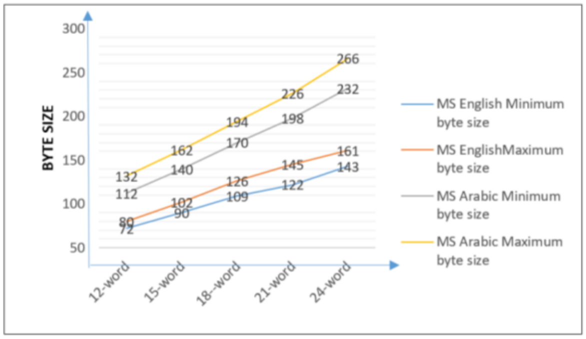


Figure 3. Ms Size Byte for the Standard English dictionary and the proposed Arabic dictionary

To further understand the obtain results, Figure (3) up, shows the five sequence options of BIP39 represented in Colum of n-word (12, 15, 18, 21, and 24). The maximum byte size of the 24-word of the MS English is located through the period of the minimum of (15-18) words of the MS Arabic. In other word, the BIP39 needs to generate at least 40- words of the MS English to reach the maximum of the 24-words of the MS Arabic. Those results contrast with the idea of a brain wallet, those words are too many to memorize them in the usual way. This led to improving the performance of the algorithm by increasing the size of the main source. Since the byte size of the entropy source must be large [29]. Therefore, using Arabic as a source to generate master seed in BIP39 is proven to better than English.

The Entropy Estimation

There were also some important differences in the entropy of languages. Entropy is a measure of the uncertainty in a random variable. The Shannon entropy is used to which quantifies the expected value of the information contained in a string. The entropy $H\,(m)$ of any string can be calculated as: $H(m) = \sum_{i=1}^{n} p(\,m_i\,)\, log_b\, \frac{1}{p(m_i)}$ Whereas m is the string, p is the probability [30].The optimal entropy of the English (4.70) whereas the Arabic entropy was (4.81)[31].

Table 3: Entropy of results for MS set

n-word	The entropy of English *MS*		The entropy of Arabic *MS*	
	Minimum byte size	Maximum byte size	Minimum byte size	Maximum byte size
12	3.88	3.94	4.29	4.36
15	4.23	4.24	4.37	4.24
18	4.07	4.24	4.41	4.22
21	4.14	4.05	4.44	4.35
24	4.24	4.08	4.47	4.46

The results of Table (4) below indicates that all values are within the optimal entropy values period for both languages. Moreover, it is noted that the entropy of the MS Arabic has exceeded to the entropy of the MS English.

Discussion

The results demonstrate two observations; the first observation is that any attack is made more difficult by lengthy keys (MS). This has been achieved by increasing the byte size of the data, which requires the attacker to provide more server storage to create the dictionary combination files. Moreover, the attacks need to build an additional encoding model to generate the source of the master seed. Consequently, adding the proposed encoding model is considered as an extra level to create MS. So, this leads to increasing the cost and run time for such an attack. The second observation, the Unicode function in the proposed model produces a unique string of bytes with non-fixed ASCII. In contrast, using even just one such non-ASCII character in any password can make brute force and dictionary attacks infeasible. Last, the proposed encoding model to BIP39 standard is software and hardware independent. Meaning, the proposed model and the new wordlist can be adopted and implemented in any Bitcoin wallet.

Comparing with The Related Work

The proposed model is simply a software modification to BIP39. Which means no extra hardware needs to increase the entropy of MS. On contrast, Visser [16] and Gentilal [18] which were their works depended on the hardware to provide an additional entropy requirement, which is may increase the original wallet cost, this aspect is undesirable in this domain.

Conclusion and Future work

The security of the Bitcoin wallet is one of the subjects which received high attention in academic research during the past five years. This paper studied how to increase resistance of Bitcoin wallet against dictionary attacks through increasing master seed entropy. The introduced encoding model dramatically increased the entropy of generating master seed when compared with normally BIP39 generated seed. The inclusion of non-fixed ASCII encoded characters by the proposed encoding model as an extra source of entropy renders dictionary attacks to be immensely hard to achieve, since by definition it requires building huge rainbow tables which are both space and time consuming process.

References

[1] E. P. E. Deepika and E. R. Kaur, "Cryptocurrency: Trends, Perspectives and Challenges."

[2] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[3] A. M. Sagheer, "Elliptic curves cryptographic techniques," in *2012 6th International Conference on Signal Processing and Communication Systems*, 2012, pp. 1–7.

[4] D. Wang, "Secure Implementation of ECDSA Signature in Bitcoin," *Inf. Secur. Univ. Coll. Longon*, 2014.

[5] A. Narayanan, J. Bonneau, E. Felten, A. Miller, and S. Goldfeder, *Bitcoin and cryptocurrency technologies: a comprehensive introduction*. Princeton University Press, 2016.

[6] C. Barski and C. Wilmer, *Bitcoin for the Befuddled*. No starch press, 2014.

[7] R. Caetano, *Learning Bitcoin*. Packt Publishing Ltd, 2015.

[8] D. L. K. Chuen, *Handbook of digital currency: Bitcoin, innovation, financial*

instruments, and big data. Academic Press, 2015.

[9] C. Mann and D. Loebenberger, "Two-factor authentication for the Bitcoin protocol," Int. J. Inf. Secur., vol. 16, no. 2, pp. 213–226, 2017.

[10] Z. Zheng, C. Zhao, H. Fan, and X. Wang, "A Key Backup Scheme Based on Bitcoin."

[11] M. D. Rai, M. P. Shetty, G. L. Alva, A. Hegde, and M. Shiran, "WALLET FOR BITCOIN CRYPTOCURRENCY," 2018.

[12] S. Eskandari, J. Clark, D. Barrera, and E. Stobert, "A first look at the usability of bitcoin key management," *arXiv Prepr. arXiv1802.04351*, 2018.

[13] M. Conti, C. Lal, S. Ruj, and others, "A Survey on Security and Privacy Issues of Bitcoin," *arXiv Prepr. arXiv1706.00916*, 2017.

[14] F. Tschorsch and B. Scheuermann, "Bitcoin and beyond: A technical survey on decentralized digital currencies," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 3, pp. 2084–2123, 2016.

[15] K. D. Zeilenga, "SASLprep: Stringprep profile for user names and passwords," 2005.

[16] B. C. M. B. Visser, "Additional source of entropy as a service in the Android user-space," 2015.

[17] M. Vasek, J. Bonneau, R. Castellucci, C. Keith, and T. Moore, "The Bitcoin Brain Drain: Examining the Use and Abuse of Bitcoin Brain Wallets," in *International Conference on Financial Cryptography and Data Security*, 2016, pp. 609–618.

[18] M. Gentilal, P. Martins, and L. Sousa, "TrustZone-backed bitcoin wallet," in *Proceedings of the Fourth Workshop on Cryptography and Security in Computing Systems*, 2017, pp. 25–28.

[19] L. Van Der Horst, K.-K. R. Choo, and N.-A. Le-Khac, "Process Memory Investigation of the Bitcoin Clients Electrum and Bitcoin Core," *IEEE Access*, vol. 5, pp. 22385–22398, 2017.

[20] C. Mann and D. Loebenberger, "Two-factor authentication for the Bitcoin protocol," *Int. J. Inf. Secur.*, vol. 16, no. 2, pp. 213–226, 2017.

[21] T. Volety, S. Saini, T. McGhin, C. Z. Liu, and K.-K. R. Choo, "Cracking Bitcoin wallets: I want what you have in the wallets," *Futur. Gener. Comput. Syst.*, vol. 91, pp. 136–143, 2019.

[22] A. Yeow, "BIP39," *URL https//github.com/bitcoin/bips/blob/master/bip-0039.mediawiki/(Accessed 6 Feb. 2018)*, 2015.

[23] R. V. Fritsche and others, "Recommendations for implementing a Bitcoin wallet using smart card," 2018.

[24] D. Richardson and others, "Information security: an investigation into password habits," 2015.

[25] A. Yeow, "Global Bitcoin Nodes Distribution," *URL https//bitnodes. earn. com/(accessed 08 Novemb. 2018)*, 2015.

[26] G. Hileman and M. Rauchs, "Global cryptocurrency benchmarking study," *Cambridge Cent. Altern. Financ.*, vol. 33, 2017.

[27] R. A. Alotaibi and L. A. Elrefaei, "Arabic Text Watermarking: A Review," *arXiv Prepr. arXiv1508.01485*, 2015.

[28] J. M. Aliprand, "The unicode standard," *Libr. Resour. Tech. Serv.*, vol. 44, no. 3, pp. 160–167, 2011.

[29] E. Barker, W. Barker, W. Burr, W. Polk, and M. Smid, "Recommendation for key management part 1: General (revision 3)," *NIST Spec. Publ.*, vol. 800, no. 57, pp. 1–147, 2012.

[30] C. E. Shannon, A. D. Wyner, and N. J. A. Sloane, *Claude E. Shannon: Collected Papers*. John Wiley & Sons, 1993.

[31] P. He, Z. Chen, S. He, and M. R. Lyu, "Characterizing the natural language descriptions in software logging statements," in *Proceedings of the 33rd ACM/IEEE International Conference on Automated Software Engineering*, 2018, pp. 178–189.