SPPBPS: A Secure Privacy-Preservation Bilinear Pairing Scheme for Bitcoin Cryptocurrency

Albert Kofi Kwansah Ansah\textsuperscript{1,3,*}, Li Lu\textsuperscript{1}, Daniel Adu-Gyamfi\textsuperscript{2,4}, Stephen Anokye\textsuperscript{1,3}, Chenfeng Tang\textsuperscript{1}

\textsuperscript{1}School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, 611731, China
\textsuperscript{2}School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu, 610054, China
\textsuperscript{3}Department of Computer Science and Engineering, University of Mines and Technology, Tarkwa TK237, Ghana
\textsuperscript{4}Department of Computer Science and Informatics, University of Energy and Natural Resources, Sunyani, P O Box 214 Ghana

\*afkansah@umat.edu.gh

Abstract. Despite the existing advances on research and education in cybersecurity and its applications, the field is still under discovery and new technologies are evolving. In recent years, Bitcoin as a cryptocurrency cyber application that records all financial transactions present on a blockchain where users can reach a secure and robust consensus on transactions emerges as the most popular peer-to-peer electronic payment system. Cryptocurrencies mostly rely on blockchain, as a secure decentralised append-only ledger to exchange digital currencies, and thereby attracting a billion-dollar economy. Bitcoin allows input transactions to link addresses and this can be used by an adversary to reveal real user identity. This obviously attracts privacy issues such as user’s information and identity leakage. Several existing techniques are found not satisfactory to fulfil practical and compatible anonymity requirements for users and their transaction. We present in this paper, a secure privacy-preservation scheme based on bilinear pairing and other cryptographic primitives. We proved that our proposed scheme is robust and can be implemented to preserve users and transaction privacy in the Bitcoin system by theoretical analysis and evaluation.

1. Introduction
Cryptocurrency is a decentralised peer-to-peer (P2P) virtual money exchange system based on cryptographic principles [1]. Cryptography, upon its several uses, can also be used to generate and distribute currency units [2] such as bitcoin (BTC) [3]. Nakamoto [3] is the first to describe Bitcoin as a decentralised peer-to-peer (P2P) electronic currency payment [5] which employs cryptographic primitives to guarantee and correct operations, in order to secure financial transactions with digital currency. Bitcoin cryptocurrency transactions are signed with the elliptic curve digital signature algorithm (ECDSA), where the parameters involved are from standards for efficient cryptography group (SECG), and secp256k1 [6].

Bitcoin is perceived to achieve anonymity [7], as it employs cryptography to control the generation of new coins. [8] The validation of the cryptocurrencies transactions are being controlled by
cryptocurrencies transactions process utilise distributed verification supplanting by a cryptographic proof-of-work (PoW) scheme \([8]\) in a blockchain. The purpose of verification is to confirm amounts and payer legitimacy whiles checking for double spending not to occur \([9]\). Once a transaction is validated and saved, it cannot be modified since any blocks involved are linked to their precursor that embeds in its cryptographic hash function. Varieties of mining techniques are employed into Cryptocurrencies for verification process based on system requirements. Blockchain \([10]\) acts similarly as a database to keep records of all the bitcoin transactions. Blockchain is made public for convenient access to all participating network nodes to enforce transparency \([11]\). However, an attack can be launched by an adversary to link addresses to reveal users’ real identity and once this is achieved, all transactions relating to the same user could be linked. This breaches users’ privacy.

A background study on Bitcoin privacy enabling approaches have been presented in \([7,12-13]\). To solve the leakage of real user identity to addresses \([14]\) presents a study on Bitcoin privacy and anonymity. Privacy of users in Bitcoin is said to be weak since Bitcoin only espouses pseudoanonymity and not anonymity. Bitcoin address reuse technique, web spidering, monitoring node IP addresses, tracking payments in the blockchain can be used to compromise Bitcoin users privacy. With these procedures available, user transactions can be linked to addresses by an adversary possibly by tracing the flow of the digital currency in the blockchain \([4]\). Privacy leakages are challenging and pricey to recover once an attack is launched \([15]\).

Bitcoin has the opportunity to offer pseudoanonymity as a partial user’s information unlinkability. Bitcoin users are pseudonymously required to create new addresses for fresh transactions using their public keys \([14]\). Although there is no directory to maintain a log and other transactional related information in Bitcoin system, an adversary can associate offline data such as emails and shipping addresses with the online information to reveal private information about peers \([5,16]\). It is, however, imperative to anonymise user transactions to dodge tracking. Transactions can be obfuscated by employing cryptographic primitives. Use of blind signatures \([17]\) was mooted, albeit its resulting security protocol disparagingly relies on the existence of dependable anonymous communication channel. A generic secure multi-party computation or sorting as a building block has been used in almost all secure decentralised protocols proposed \([18-20]\) and thus lack efficiency.

Different level of privacy is achieved through one-time public/private key pairs, long-term public/private key pairs, public-key cryptosystems, pairing-based cryptosystems etc in recent Privacy-preserving schemes proposed \([15]\), public-key cryptography, and group signature scheme \([21]\). The authors present a privacy scheme and anonymity considerations based on bilinear pairing for Bitcoin cryptocurrency.

Chapter 2 sees preliminary of the problem. Chapter 3 seeks to introduce and discuss the proposed privacy-preserving scheme. The theoretical analysis and the evaluation of the scheme are tabulated in Chapter 4 and Chapter 5 presents the conclusion of this paper with future work direction.

2. Materials and Methods

2.1. Bilinear pairings and bilinearity

Let \(G_1, G_2\) and \(G_T\) be a finite cyclic groups of prime order \(p\), and let \(g_1, g_2\) be respectively generators of groups \(G_1\) and \(G_2\) \([8]\). A pairing of a non-degenerate bilinear map: \(e: G_1 \times G_2 \in G_T\). The map \(G_1 \times G_2\) \(\in G_T\) is said to be an admissible map if it satisfies the following three conditions:

1. Bilinearity: \(e(g_1^a, g_2^b) = e(g_1, g_2)^{ab}, \forall a, b \in \mathbb{Z}_p; \mathbb{Z}_p = \{0, 1, \ldots, p - 1\}\) is a Galois field \([8]\) of order \(p\).
2. Non-degeneracy: \(e(g_1, g_2) \neq 1\).
3. Efficiently computable of \(e(g_1, g_2)\) given \(g_1 \in G_1\) and \(g_2 \in G_2\).

Mapping of \(e\) is referred to as bilinearity. Modified Weil and Tate pairings on elliptic curves may be used to construct bilinear mapping \([22]\).
2.2. Design goals

Our design goal is to develop a secure privacy-preserving scheme for Bitcoin users and their transactions. We guarantee the above security requirements in our proposed scheme. We apply the system model and security requirements to design our proposed scheme. Proposed scheme should broadcast bitcoin transactions into the Bitcoin blockchain with the needed anonymity, making it hard for a dishonest node to link the transactions to users’ addresses with the view to revealing real users’ identity. Our system espouses cryptographic privacy techniques built on top of the Bitcoin system. For the privacy of the scheme, the authors used bilinear mapping based on modified Weil and Tate pairings and zero-knowledge proof protocol. Our proposed scheme achieves end-to-end data privacy and it is efficient and scalable.

2.3. Protocol setup

In our scheme, to anonymise a transaction message, the protocol runs as below. Users interfacing the scheme sign their transaction messages with a personal public key from a public key pair different from the default Bitcoin public-key pair generated at the user end of our scheme by the users. Users under no circumstance should leak this personal key pair. The secret key part is then used to prove knowledge of the transaction to get the transaction back from the scheme. The first execution of our scheme is as alluded above, supposedly as the user end of the scheme. The scheme can loosely be seen as having two ends as user and system ends. Our scheme group transactions of equal length and countersigns them with the same secret key. Inconsistent lengths are padded to same.

a. The scheme takes valid Bitcoin transactions
b. Setup(n) runs to generate public parameters. Input: (n), output: \( g^k = (g_1, g_2, G, g_T, e, H); H: \{0, 1\}^* \rightarrow G \).
c. KeyGen\( (g^k) \) runs to generate unique public-key pair. Output: \((pk_{PE}, sk_{PE}) = (g_1^n, u); u \in_R Z_p^*, g_1^n \in G\).
d. Sender signs a created transaction with the unique public key. Input: \((btm, sk_{PE})\), output: \(\sigma\).
e. Transactions \(\{btm_{\sigma_i}\}_{i=1}^n\) belonging to various users are collected by the scheme.
f. Scheme checks transactions of same lengths and sub-grouped them.
g. Scheme runs KeyGen\( (g^k) \) \(n\) times to generate a set of public-key pairs. Output:
   \[\{sp\}_{1 \leq i \leq n} = \{(sp_{pk_1}, sp_{sk_1}), \ldots, (sp_{pk_n}, sp_{sk_n})\}\]
   \((sp_{pk_i}, sp_{sk_i}) = (g_1^{a_i}, a_i); a_i \in_R Z_p^*\) and \(g_1, g_1^{a_i} \in G\).
h. Scheme randomly select public-key pair from \((g)\) public-key pair \(\{sp\}\).
i. Sets of compared transactions \(G_{\neq 1} = \{btm_{\sigma_i}\}_{i=1}^n\) are countersigned CSign\() with secret key from \((h)\).
   Input: \((btm, sp_{sk_i})\), output: \(\sigma_c \in G\). A trapdoor \(ps_{pki}\) is embedded for signature verification by recipient.
j. Sender executes a commitment protocol (Pederson commitment) [23] to receive transaction back.
k. The anonymised transaction is thus broadcast into the Bitcoin network and consequently stored on the blockchain.

Algorithm: SPPBPS

1. \(btm\) //valid Bitcoin transaction
2. \(g^k = (g_1, g_2, G, g_T, e, H) \leftarrow \) Setup\((1^n)\)
3. \(u \leftarrow Z_p^*\)
4. \((g_1^n, u) = (pk_{PE}, sk_{PE}) \leftarrow \) keyGen\( (g^k)\)
5. \(h \leftarrow H(bit_{tm})\)
6. \(\sigma \leftarrow [h + ts1]^u\)
7. \(\{\text{btm}_i\}_{i=1}^{m} \rightarrow \text{SPPBPS}\)
8. \(\max(|\{\text{btm}_i\}_{i=1}^{m}|) \rightarrow \text{SPPBPS}\)
9. if \(|\text{btm}| = x_{\text{max}}\):
   \(G_1 = \{\text{btm}_i\}_{\text{len}=x_{\text{max}}}\)
   else if \(|\text{btm}| = y_{\text{max}}-1\):
   \(G_2 = \{\text{btm}_i\}_{\text{len}=y_{\text{max}}-1}\)
   else if \(|\text{btm}| = z_{\text{max}}-2\):
   \(G_3 = \{\text{btm}_i\}_{\text{len}=z_{\text{max}}-2}\)
else:
   \(G_n = \{\text{btm}\}_{\text{other len}}\)
10. \(a_i \leftarrow Z_p^*\)
11. \(\{\text{sp}_{\text{phi}}, \text{sp}_{\text{ski}}\} \leftarrow \text{KeyGen}(g^k)^m\)
12. \(\{g^d_1, a_i\} \leftarrow \{\text{sp}_{\text{phi}}, \text{sp}_{\text{ski}}\}\)
13. \(h' \leftarrow H(\text{btm}_n)\)
14. \(c_{\text{us}} \leftarrow [h' + ts2]^a_1\)
15. \(p_1 \leftarrow [1, p - 1]\)
16. \(g_1 \leftarrow g^k\)
17. \(c_1 \leftarrow \text{commit}_1(\text{btm}, p_1)\)
18. \(c_2 \leftarrow \text{SPPBPS}\)
19. \(g_2 \leftarrow G_T(\text{SPPBPS})\)
20. \(p_2 \leftarrow [1, p - 1]\)
21. \(c_2 \leftarrow \text{Commit}_2(\text{btm}', p_2)\)
22. \(\text{SPPBPS} \leftarrow c_2\)
23. if \(c_2 = c_1:\)
   "accept commitment"
   Return \([\text{btm}]_{\text{anon}}\)
else:
   "reject commitment"
   Return \(\perp\)

Remark: \(\leftarrow, \leftarrow \) are user executions, otherwise \(\text{SPPBPS}\) executions.

2.4. Privacy analysis and evaluation
In this section, the authors provide a theoretical discussion to analysis and evaluate the proposed privacy scheme. Our scheme uses a symmetric pairing and it is a bilinear map where \(G_1 = G_2 = G\). We construct our scheme with a security and privacy parameter \(n\) with setup (Setup) function to generate the bilinear pairing group. Key generation (KeyGen) function is then run to generate the key pairs.

2.4.1. Analysis. It is pretty evident that the privacy of Bitcoin users are significantly exposed to threat in the blockchain with its public nature. Privacy is still achievable and maintainable to a certain level in bitcoin transaction processing chain by applying appropriate cryptographic primitives to anonymise transactions. In SPPBPS, the privacy-preserving is carried out in the said layer as seen in Figure 1. Users anonymised transactions are then added to the Bitcoin blockchain. The scheme makes use of ECDSA for signature authenticity. Anonymous transaction verification of users is performed by the verifier. Anonymised transactions are sent back to the anonymous users using ZKP protocol through Pederson commitment scheme before broadcasting into the blockchain. Users use their transaction private-key to sign their transactions and are required to generate as many public key pairs as desired. This is in line with the design and operations of Bitcoin, therefore, our scheme is compatible with the Bitcoin system. An adversary trying to have unauthorized access to transactions cannot identify individual transactions due to implementation of a Pederson commitment protocol. This lauds our privacy scheme and,
therefore, privacy-preserving requirements are achieved for user transactions. Enhancing user privacy further, a new key pair is required to be generated for each transaction. This keeps the transactions from being linked to particular users. To successfully link users to their transactions, an adversary must have ample knowledge of the transaction to break the commitment by solving the DL hard problem. The formal function definitions for the proposed privacy scheme are as follows: We have \( btm \) as a user valid Bitcoin transaction. The scheme implements zero-knowledge proof through the Pedersen commitment scheme (PCS) [23-24]. Two settings loosely called user procedures and system procedures are carried out in the scheme. In the user procedures, users would generate unique public-key pairs of which the private key is used to claim anonymised transaction back to himself for broadcasting into Bitcoin blockchain using PCS.

1) Setup\((1^n)\): on input \( n \) being a security parameter, setup outputs public parameters \( g^k = (p, g_1, g_2, G, G_T, e, H) \); \( e \) GxG \( \rightarrow \) G\( T \) is a bilinearity and \( H: \{0, 1\}^* \rightarrow \) G is a one-way collision resistant hash function.

2) KeyGen\((g^k)\): for each user, outputs unique public-key pairs: \( (pk_E, sk_E) = (g_1^u, u) ; \) \( u \in \mathbb{Z}_p^* \), \( g_1^u \in G \), signature \( \sigma = \{H(btm) \parallel ts\}_i \in G \). btm is transaction message and ts is time-stamp to resist replay attack.

KeyGen\((g^k)\) on system procedures faction is run multiple times to output set of unique system public-key pairs and randomly select one to sign transactions collected together as a group to produce the desirable anonymity. A trapdoor which is the public key of the private key pair to be used to countersign transactions is embedded for the recipient.

a) Setup\((1^n)\): outputs \( g^k = (g_1, g_2, G, G_T, e, H) \). and \( H: \{0, 1\}^* \rightarrow G \) a one-way permutation hash function.

b) KeyGen\((g^k)\): run \( n \) times. Output set of public-key pairs \( \{sp\}_{\forall i s/n} = \{(sp_{pk1}, sp_{ski}), \ldots , (sp_{pkn}, sp_{skn})\} \). Scheme selects randomly a pair \( (sp_{pki}, sp_{ski}) = (g_2^a_i, a_i), a_i \in \mathbb{Z}_p^* \) and \( g_2, g_2^a_i \in G \). Counter signature \( \sigma_c = \{H(btm) \parallel ts\}_i \in G \), ts is current timestamp.

![Figure 1. Privacy framework](image)

2.4.2. Evaluation. Threat model. Assuming an adversary launches an attack on linking input and output addresses to reveal users’ real identity to ascertain user’s transaction behaviour, by tethering real names to transactions and bitcoin addresses in the blockchain. Historical data passed from the blockchain can also be analysed by an adversary to lunch an offline attack [24-26]. We consider \( btm \) as coming from users’ wallet. Bitcoin wallet properties can be used in an adversal fashion by an dishonest node to create Bitcoin addresses and transactions. Eve can still be unable to link any transaction to a particular user even if she plays a dishonest node role and participate in our scheme because each user signs his message with his own personal public key. Users then claim their transactions back with their corresponding private key known. Again, all transactions go into the blockchain would have the same counter signature signed with the same secret key in our setting. Hence, our scheme achieves needed
anonymity. User privacy model. Our proposed scheme uses pairing-based cryptographic schemes of bilinear map specially called symmetric pairing \( (G_1 = G_2 = G) \) [8]. It is constructed with \( n \), a security parameter and runs setup() to output \( g^k \) and KeyGen function to generate users’ ‘personal’ public-key pairs and \( \sigma \).

Setup\((1^n)\): outputs symmetric bilinear group \( g^k = (g_1, g_2, G, G_T, e, H) \); e: \( G \times G \to G_T \) as the bilinear map and one-way hash function given by \( H: \{0,1\}^* \to G \). User selects unique ‘personal’ public-key pairs denoted by PE and represented as \((pk_{PE}, sk_{PE})\). User must not leak or associate her PE with public-key pair of Bitcoin addresses. The user will use her personal public-key pair for commitment and proof of knowledge. Personal public-key pairs \((pk_{PE}, sk_{PE}) = (g^k_1, u)\) is generated on the user end of our privacy-preservation layer. Sign(\( btm, sk_{PE} \)): \( btm \in G_T \). To sign \( btm \), under the secret key \( sk_{PE} = u \). Compute \( \sigma = [ H(btm) \parallel ts1]^u \), where TS1 is current time stamp to resist a potential replay attack [8]. This is done in the SPPBPS layer.

Verify\((btm, \sigma, pk_{PE})\): Given \( pk_{PE} \in G, btm \in G_T \) and \( \sigma \in G \), compute \( h = H(btm) \in G \) and accept if: \( e(\sigma, g_1) = e(H(btm), pk_{PE}) = e(h, pk_{PE}) \). Correctness: \( e(\sigma, g_1) = e(H(btm) \parallel ts1)^u, g_1^1 = e(h, g_1^1) \).

KeyGen\((g^k)\): outputs unique key pairs: \((s_{pk}, s_{sk}) = (g^a_q, a) \) for a \( \in R \); and \( g^a_q \in G \).

CSign\((btm, ps_{sk})\): CSign is a counter signature. To counter sign \( btm \), using the secret key \( ps_{ski} = a_i \). Compute \( \sigma_c = \sigma^{a_i} = [ H(btms) \parallel ts2]^a_i \). It is remarked that, this is carried out in the SPPBPS layer.

CSigVer\((\sigma_c, btm, ps_{pk})\): Given \( ps_{pki}, btm_\sigma \) and \( \sigma_c \in G \), compute \( h' = H(btm_\sigma) \) and accept or reject if: \( e(\sigma_c, g_2) = e(H(btms), ps_{pki}) = e(h', ps_{pki}) \). Correctness: \( e(\sigma_c, g_2) = e((H(btm_\sigma) \parallel TS2)^a_i, g_2) = e(h', g_2^a_c) \).

On linking input and output addresses to reveal users identities, the authors assume a dishonest node as an adversary. The authors state that if it users with \( \nu \) bitcoins belonging to the following input addresses \( U_i = (l_1, l_2, ..., l_\nu) \) and wish to send same to the following output addresses \( O_i = (O_1, O_2, ..., O_\nu) \) so that (i) users in \( U_i \) receive \( \nu \) bitcoins back on their output address \( O_i \), and (2) it is hard to link output addresses thus, only users in \( U_i \) know that outputs \( O_i \) belong to them. However, inputs and output addresses can be mapped by a dishonest node to reveal real users identities. Therefore, since mapping is still possible, it is augest to anonymise the linkage and our scheme sought to achieve that. \( \sigma_c \) protects privacy of users and their transactions. An adversary could be motivated within a space of time to acquire addresses or transactions of \( \nu \) users; \( U = \{u_1, u_2, ..., u_\nu\} \) with \( n_u \) addresses; \( A = \{a_1, a_2, ..., a_n\} \). A dishonest node may belong to the Bitcoin network. This is enough motivation for them to link users and their addresses. A commitment to \( btm \in G_T \) is by computing \( c_1 = Commit_1(btm, p_{i_1}) = g_1^{btm}g_2^{p_{i_1}} \in G_T \subseteq Z_p^*, p_1 \in R \). To prove \( c_1 \) is a commitment to \( btm \), prover opens the commitment by revealing \( btm \) and \( p_1 \). The prover will be unable to find two values \( btm' \) and \( p_2 \), with \( btm' \neq btm \) such that \( Commit_2(btm', p_2) = c_1 \) because it could not solve the DLP in \( G_T \), hence this commitment is computationally binding.

2.5 Modelling the scheme
Scheme was inspired by PCS, which runs three algorithms: Gen(), Commit() and verify(). The scheme runs committer \( C \), holding secret \( btm \in G_T \) to commit to and then receiver \( R \) which agrees on \( g^k = (g_1, g_2, G, G_T, e, H) \); \( g \) is generator and \( p \) is the order of \( G_T \). The scheme is defined as follows:

Commitment phase:
R samples \( g_2 \in \mathbb{R} \), and then sends to C. C also samples \( p_2 \in \mathbb{R} \), and then computes commitment \( c_2 = g_2^p \cdot g_1^{b_{tm}} \) and then sends to R.

Verification phase:
The pair \( (b_{tm}', p_2) \) is sent from C to R. R checks if \( c_2 \) matches \( c_1 \), and accepts, otherwise, return \( \bot \). Here, only a legitimate user can prove the commitment by solving DLP to reveal the contents of \( b_{tm} \), therefore, privacy is preserved in this setting. ZKP protocol is implemented for users to prove a statement to the verifier without revealing anything about it. Prover P then confirms verifier V that it has knowledge of \( b_{tm} \) and \( p_1 \) of public \( c_1 = \text{Commit}_1(b_{tm}, p_1) \). P randomly chooses \( b_{tm}' \in \mathbb{R} \), and compute recommitment value \( r' \in \mathbb{G}_T \), and sends \( c_2 = \text{Commit}_2(b_{tm}', p_2) \) to verifier V. Verifier V sends a challenge \( e \in \mathbb{R} \) to prove P. Prover P sends \( u = b_{tm}' + e \cdot b_{tm} (\text{mod} \ p) \) and \( v = p_2 + e \cdot p_1 (\text{mod} \ p) \) to verifier V. Verifier V accepts if \( \text{Commit}(u, v) = c_2 \cdot c_1^e \).

Protocol hides \( b_{tm} \) information theoretically. Even given \( c_2, e, u, \) and \( v \), a dishonest node or user with infinite computation power is unable to learn anything about \( b_{tm} \). For any \( b_{tm}' \), there exist message \( b_{tm}'' \), such that \( u = b_{tm}' + e \cdot b_{tm} = b_{tm}'' + e \cdot b_{tm} (\text{mod} \ p) \). Thus \( b_{tm}'' = b_{tm} + (b_{tm} - b_{tm}') \) (mod \( p \).

The authors prove that only a legitimate user can reveal the contents of the original message using the secure private key. This could be verified from the equation as follows:

From \( \text{CSignVer}(\sigma_c, b_{tm}, p_{PK}) \), Given \( p_{PK} \), a \( b_{tm} \) and counter signature \( \sigma_c \), \( h' = H(b_{tm}) \) accept if:

\[
e(\sigma_c, g_1) = e(H(b_{tm}), p_{PK}) = e(h, p_{PK})
\]

Thus, proving that: \( e(h', p_{PK}) = e(H(b_{tm}), g_1^e) \) (1) knowing \( h' \) and \( p_{PK} \). \( b_{tm} \) would not be revealed using the general Bitcoin wallet public-key pair in (1).

3. Conclusion
In this paper, a secure privacy-preserving scheme based on bilinear pairing was presented. The scheme sought to augment users’ privacy and achieves anonymity for users and their transactions. We focus on unlinkability and untraceability of users and their transactions in the blockchain ecosystem where bitcoin financial transactions occur and as a secure decentralised append-only ledger to exchange digital currencies. The authors proved that our proposed scheme is robust and can be implemented to preserve users and transaction privacy in the Bitcoin system by theoretical analysis and evaluation. The scheme is proven secure against an adversary to associate offline data to launch any successful attack. We consider the implementation stage in our future work. The scheme shall be enhanced to cater for other privacy settings in the blockchain as a whole.

References
[1] Vijayan A and Thomas T 2014 Anonymity Unlinkability and Unobservability in Mobile Ad Hoc Networks, IEEE International Conference on Communication and Signal Processing pp 1880-1884
[2] Farell R 2015 An analysis of the cryptocurrency industry Online available from http://repository.upenn.edu
[3] Duffield E and Hagan K 2014 Darkcoin, Peer-to-peer cryptocurrency with anonymous blockchain transactions and an improved proof-of-work system, Technical report. http://www.darkcoin.io/downloads/DarkcoinWhitepaper.pdf
[4] Nakamoto S 2008 Bitcoin: A Peer-to-Peer Electronic Cash System. http://bitcoin.org/bitcoin
[5] Kwansah Ansah A K, Adu-Gyamfi D and Anoky S 2019 Privacy Preservation of Users in P2P E-Payment System IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT) pp 1-8
[6] Daniel R and Brown L 2010 Standards for Efficient Cryptography 2 (SEC 2): Recommended Elliptic Curve Domain Parameters Standards for Efficient Cryptography 2 (Sec 2):37
[7] Bonneau J, Miller A, Clark J, Narayanan A, Kroll J A and Felten E W 2015 SoK: Research Perspectives and Challenges for Bitcoin and Cryptocurrencies IEEE Symposium on Security & Privacy pp 104-121
[8] Ara A, Al-Rodhaan M, Tian Y and Al-Dhelaan A 2017 A Secure Privacy-Preserving Data Aggregation Scheme Based on Bilinear ElGamal Cryptosystem for Remote Health Monitoring Systems IEEE Access pp 12601–12617
[9] Anonymous Mining 2014 https://en.bitcoin.it/wiki/Mining
[10] Ahram T, Sargolzaei A, Sargolzaei S, Daniels J and Amaba B 2017 Blockchain Technology Innovations IEEE Technology and Engineering Management Conference (TEMSCON)
[11] Fauzi P, Meiklejohn S, Mercer R C Orlandi 2018 Quisquis: A New Design for Anonymous Cryptocurrencies pp 1-36
[12] Tschorsch F and Scheuermann B 2016 Bitcoin and Beyond: A Technical Survey on Decentralized Digital Currencies IEEE Communications Survey & Tutorials (Vol. 18 No. 3 Third Quarter)
[13] Bergstra J and de Leeuw K M M 2013 Bitcoin and Beyond: Exclusively Informational Money http://arXiv.org
[14] Bonneau J, Narayanan A, Miller A, Clark J, Kroll J A and Felten E W 2014 Mixcoin: Anonymity for Bitcoin with Accountable Mixes (in Financial Cryptography and Data Security vol 8437) (Springer) pp 486-504
[15] Mauro C, Sandeep K E, Chhagan L and Sushmita R 2017 A Survey on Security and Privacy Issues of Bitcoin. http://arxiv.org/abs/1706.00916
[16] Green M and Miers I 2017 Bolt: Anonymous Payment Channels for Decentralized Currencies Proceedings of the ACM SIGSAC Conference on Computer and Communications Security 45(1) pp 599–613
[17] Chaum D 1983 Blind signatures for untraceable payments (in CRYPTO) 198 (Springer)
[18] Yang E Z 2012 Secure multiparty Bitcoin anonymization Blog posting http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization/
[19] Ben-Sasson E, Chiesa A, Garman C, Green M, Miers I, Tromer E and Virza M 2014 ZeroCash: Decentralized Anonymous Payments from Bitcoin In IEEE Symposium on Security and Privacy pp 459-474
[20] Ibrahim M H 2017 SecureCoin. A Robust Secure and Efficient Protocol for Anonymous Bitcoin Ecosystem in International Journal of Network Security (Vol 19 No.2) pp 295-312
[21] Maas M 2004 Pairing-based cryptography (Thesis) Dept. Mathe. comput. Tech. Univ. (Eindhoven, Eindhoven, The Netherlands)
[22] Camenisch J and Lysyanskaya A 2004 Signature Schemes and Anonymous Credentials from Bilinear Maps in Advances in Cryptology – CRYPTO vol 3152 (Springer) pp 56-72
[23] Pedersen T P 1992 Non-interactive and information-theoretic secure verifiable secret sharing In CRYPTO ’91 (LNCS vol 576) (Springer Verlag) pp 129–140
[24] Bellare M and Goldreich O 1992 On Defining Proofs of Knowledge Proceedings of Crypto ’92 (LNCS vol 740) (Springer Verlag) pp 390–420
[25] Meiklejohn S, Pomarole M, Jordan G, Levchenko K, McCoy D, Voelker G M and Savage S 2013 A fistful of bitcoins: Characterizing payments among men with no names In Proc. of the Conference on Internet Measurement Conference (IMC’13 ACM) pp 127–140
[26] Reid F and Harrigan M 2013 An analysis of anonymity in the bitcoin system In Security and Privacy in Social Networks (Springer) pp 197–223