USH for Anonymous and Unlinkable Coin Spending for BFDChain

Pu, Duan
Befund Foundation Ltd.

Abstract— In this paper, we propose Unlinkable Secret Handshake (USH), which is based on Tate Paring cryptography, as a new protocol to provide anonymous and privacy-preserving coin spending for BFDChain.

I. Introduction

As one of the most important security aspects, how to achieve unlinkability and anonymity is critical to the success of any blockchain systems because of its decentralized nature. We designed an unlinkable secret handshake (USH) protocol for BFDChain, which provides more anonymous and privacy-preserving coin spending.

The rest of this paper is organized as follows: In Section II, we present the motivation to design and use USH for BFDChain. Then we provide a summary of threats to privacy and anonymity to on-chain currencies. We then provide detailed technical discussion of USH on BFDChain and its security analysis in Section IV and Section V, respectively. Section VI concludes this paper.

II. MOTIVATION

One significant advantage to use on-chain currency is that user's real-world identity is not disclosed when sending or receiving payment, i.e. cryptocurrency payment is sent to or received by user's blockchain address that is a hashed value of a single-time used elliptic curve point. However, as payment is usually for real-world offline transaction, payer may need to disclose his email address or shipping address to complete the offline transaction or prove his eligibility for the transaction. For example, Alice wants to purchase alcohol from Bob. She needs to disclose her real-world identity document to prove that she is over 21 years old before getting legally qualified for the alcohol purchase. This causes two potential risks: (1) Bob links Alice's real-world identity to her cryptocurrency identity/address (2) if Alice wants to purchase alcohol in the future, she will need to disclose the same identity again to others which increase the risk to link up all her previous transactions.

To solve this issue, we propose USH to achieve anonymity and unlinkability for user's credential like the identity mentioned in the above example. We assume that before the coin spending, both payer and payee have already obtained credentials from a central authority that cooperates with BFDChain (for exmaple, DMV already built a sidechain on BFDChain). Credential assignment can be built as a service or sub-chain attached to BFDchain. The credential is to prove that the owner of the credential is a member of a group, i.e., eligible for certain purchase. USH protocol guarantees that (1) Bob cannot link Alice's real-world identity to her BFDchain ID/address after proving that she is member of the group (2) Alice's credential is unlinkable, i.e., future same kind of transactions won't disclose her previous transaction. Furthermore, even the central authority cannot know that it was Alice who made the purchase from the public ledger of BFDchain.

It is also well known that on-chain cryptocurrency like BFDchain is built on elliptic curve cryptograph (ECC), i.e., payer/payee's addresses in coin wallet are hashed values of elliptic curve point for signing and verifying by elliptic curve digital signature algorithm (ECDSA). USH is also designed on the same ECC based cryptosystem, so the new unlinkability and anonymity for coin spending can be easily implemented on BFDchain.

III. PRIVACY ISSUES IN CRYPTOCURRENCY

There are many security concern and attacks for on-

chain currencies, e.g., doubling spending [1] and mining pool attacks [2] for Bitcoin. This paper focuses on the privacy and anonymity issues for cryptocurrencies. They achieve anonymity by keeping public keys anonymous, i.e., the public can see payment is sent from payer to payee, but cannot link the transaction to anyone's real identity. The reason is that both payer and payee use their hashed elliptic curve point as the address to send and receive payment. To further achieve the unlinkability, it is advised to use a new key pair for each transaction so that no multiple transactions can be linked to the same user. However, the anonymity and unlinkability provided by the current on-chain currency are vulnerable through different attacks like address reuse, blockchain and public address analysis [3][4], etc. Also, adversary can make use of public web crawlers the correlates social networks with cryptocurrency address like Bitcoin [5]. In [6] the authors mention that an adversary can associate the offline data like emails and shipping addresses with the online information, then eventually get the private information about the user. In [7] the authors show that cryptocurrency transactions can be linked to the user cookies and then to the user's real identity and the user's purchase history is revealed. Research [8][9] also shows that the IP address in cryptocurrency P2P network might be the vulnerability that compromises unlinkability.

To defend again these attacks, in [10] the authors propose ZeroCoin, which applies zero-knowledge authentication protocol to Bitcoin to provide anonymity. An extension of ZeroCoin is proposed by [11] as ZeroCash, which uses an improved version of zero-knowledge proof. Also, the second generation of cryptocurrencies like Ethereum [12] are designed with better consideration for security and privacy issues.

IV. USH FOR UNLINKABLE AND ANONYMOUS COIN SPENDING ON BFDCHAIN

The motivation to design and use of USH for BFDchain is to provide privacy-preserving and anonymous coin spending for both online and offline interactions for participants. We design USH based on Tate Paring cryptography that is also built on top on elliptic curve so that our protocol can be implemented on BFDChain easily.

4.1 Mathematical Background

Definition 1: Bilinear Pairing

A pairing is a bilinear map e: $G_1 \times G_1 \rightarrow G_2$ if, for any P, Q $\in G_1$ and any a, b $\in Z^*_q$ we have $e(a \cdot P, b \cdot Q) = e(a \cdot P, Q)^b = e(P, b \cdot Q)^a = e(P, Q)^{a \cdot b}$ and $e(P,Q) = e(Q, P)$

To provide an efficient computation of the bilinear map, we choose G_1, G_2 and e as a set of points on an elliptic curve, a multiplicative cyclic group over integers and Tate pairing, respectively. ECC has a special discrete logarithm problem, Elliptic Curve Discrete Logarithm Problem (ECDLP), defined as the basis of elliptic curve cryptography (ECC). Based on the parameters chosen above, another assumption we need is the *BDH Assumption* described as follows:

Definition 2: Bilinear Diffie-Hellman (BDH) Assumption

Given P, $a \cdot P$, $b \cdot P$, $c \cdot P$ for random a, b, c $\in Z^*_q$ and P $\in G_1$, it is not possible to compute $e(P, P)^{a \cdot b \cdot c}$ with a non-negligible probability, i.e., it is hard to compute $e(P, P)^{a \cdot b \cdot c}$

In the follows we present the detail of the use of the homomorphic randomization in USH.

Definition 3: Homomorphic Randomization Function in SH

Suppose user U joins a group g_i by obtaining a credential $g_i \cdot H_1(ID_U)$, where $g_i \in G_2$ represents the group secret and $H1(ID_U) \in G1$ represents U's assigned pseudonym. U selects a large random integer $s_U \in G_2$, and uses the addition operation defined in G_1 to compute $s_U \cdot g_i \cdot H_1(ID_U)$ (s_U times of addition of $g_i \cdot H_1(ID_U)$).

Formal Definition of Secret Handshake: Alice has a pair consisting of a pseudonym and a credential, *ID_A* and *C_A*, assigned from a central authority. Bob has his *ID_B* and *C_B* assigned from a central authority. After the execution of an SH protocol, Alice and Bob know that *C_A* and *C_B* indicate common same group membership if SH succeeds; otherwise, Alice (Bob) cannot know anything about *C_B*(*C_A*). In addition, Alice (Bob) cannot know who she (he) interacted with.

4.2 USH for BFDChain

We will use the Fig.1 to illustrate how USH works on top of BFDChain.

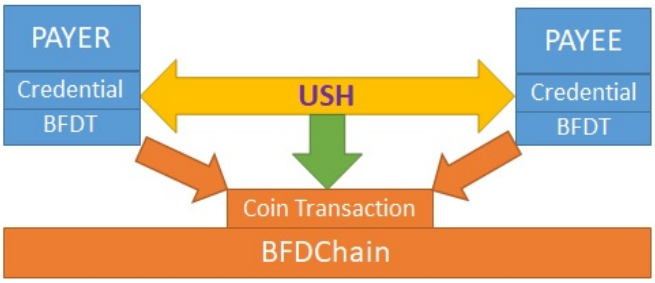

Fig.1 USH for BFDChain

In this section, we present the detail of USH on BFDChain. We assume that payer and payee have already registered in a central authority (CA) and obtained credential that can be used to authenticate their membership to a secret group. CA collaborates with BFDChain so that the credential assignment to payer and payee is a service on the chain, a sub-chain attached to the main chain. Then the payer and payee apply the unlinkable and anonymous secret handshake protocol, USH, to authenticate that the payer is eligible for the transaction/purchase. The payer can also verify the payee's eligibility if required. If authentication succeeds, the payer and payee continue for their transaction with BFDT. Otherwise, either party can cancel the transaction.

Our main idea is to use the homomorphic randomization function in the Protocol Randomization phase to randomize user's assigned pseudonyms to provide reusability of credentials. More specifically, we let a user generate a secret random number in execution of the secret handshake. The user multiplies the random number to an elliptic curve point that represents the user's pseudonym. The random number minimizes the correlation among authentication messages even they are produced by reuse of the same credential. First we present the parameters that are required in USH as (q, G_1, G_2, e, H_1, H_2). Here q is a large prime number, G_1 denotes an additive cyclic group of prime order q, G_2 denotes a multiplicative cyclic group of the same order q, H_1 is a collision-free hash function that maps a string with arbitrary length to an element in G_1, H_2 is a collision-free hash function that maps a string with arbitrary length to a string with fixed length, and e denotes a bilinear map. G_1 and G_2 are selected in such a way that *Discrete Logarithm Problem (DLP)* is assumed to be hard in both of them.

Our proposed USH has three phases: *Protocol Initialization*, *Group Secret Mapping and Authentication Computing*. To be consistent with other cryptographic protocol, we use Alice to represent payer and Bob for payee. In the follows we present the detail of the three phases.

Phase 1: Protocol Initialization

The CA determines the pairing parameters $(q, G_1, G_2, e, H_1, H_2)$ and group secrets $[g_1, ..., g_n]$, where $g_i \in G_2$. The CA publishes the pairing parameters while keeping the group secrets in private. Alice requests the CA to join the group with group secret $g_A \in [g_1, ..., g_n]$. The CA verifies Alice's qualification to decide whether Alice can join the group. If yes, the CA grants the group membership to Alice by issuing her a credential $g_A \cdot H_1(ID_A) \in G_1$, where $g_A \in G_2$. The credential is a secret of Alice to prove her membership in group g_A to another user in the same group. Alice cannot deduce g_A from $g_A \cdot H_1(ID_A)$ and $H_1(ID_A)$ by the assumption that DLP is hard in G_1. It is important for preventing forgery of credentials. Users Alice and Bob use their credentials, $K_A = g_A \cdot H_1(ID_A)$ and $K_B = g_B \cdot H_1(ID_B)$, to generate authentication messages to one another. Alice randomly generates two large random numbers, n_{A1} and s_{A1}. Alice computes $W_{A1} = s_{A1} \cdot H_1(ID_A)$, as described in the homomorphic randomization function above. That said, s_{A1} is used to minimize the correlations of authentication messages produced by the same credential. Since using a credential multiple times will not create messages that can link to the user identity, our credential is reusable. n_{A1} prevents replay attacks. Bob also randomly generates two large numbers, n_{B1} and s_{B1}, for the same purpose. Detailed interactions of group secret mapping phase and authentication computing phase are described as follows:

Phase 2 and Phase 3: Group Secret Mapping and Authentication Computing

(a) *Alice* $\rightarrow$ *Bob:* n_{A1}, $W_{A1} = s_{A1} \cdot H_1(ID_A)$

(b) *Bob: Compute* $V_{B,A} = H_2(U_{B,A} \,||\, n_{A1} \,||\, n_{B1} \,||\, 0)$, $U_{B,A} = e(W_{A1}, s_{B1} \cdot K_B)$. *Here Bob implements the secret mapping function to map his own credential* $K_B = g_B \cdot H_1(ID_B)$ *from an element in* G_1 *to an element in* G_2 *through bilinear map e.*

(c) *Bob* $\rightarrow$ *Alice:* n_{B1}, $W_{B1} = s_{B1} \cdot H_1(ID_B)$, $V_{B,A}$

(d) *Alice: Compute* $V'_{B,A} = H_2(U'_{B,A} \| n_{A1} \| n_{B1} \| 0)$, $U'_{B,A} = e(W_{B1}, s_{A1} \cdot K_A)$. *If* $V_{B,A} = V'_{B,A}$, *then Alice knows Bob belongs to the same group, i.e.,* $g_A = g_B$. *Otherwise, Bob belongs to a different group, i.e.,* $g_A \neq g_B$ *or Bob belongs to no group. Here Alice implements the result computing function to find out whether* $H_2(U'_{B,A} \| n_{A1} \| n_{B1} \| 0) = H_2(U_{B,A} \| n_{A1} \| n_{B1} \| 0)$.

(e) *Alice* $\rightarrow$*Bob:* $V_{A,B} = H_2(U'_{B,A} \| n_{A1} \| n_{B1} \| 1)$

(f) *Bob: Compute* $V'_{A,B} = H_2(U_{B,A} \| n_{A1} \| n_{B1} \| 1)$. *If* $V_{A,B} = V'_{A,B}$, *Bob knows that Alice belongs to the same group. Otherwise, Alice belongs to a different group, i.e.,* $g_A \neq g_B$ *or Alice belongs to no group.*

The protocol succeeds when $V_{B,A} = V'_{B,A}$ and $V_{A,B} = V'_{A,B}$ in steps (d) and (f). Based on the BDH assumption, it succeeds if, and only if, $g_A = g_B$. Otherwise, if it fails, Alice and Bob only know $g_A \neq g_B$. Users other than Alice and Bob cannot know whether $g_A = g_B$ or not, because they cannot compute $V'_{B,A}$ and $V'_{A,B}$ without K_A and K_B. A sketch of proof for $V_{B,A} = V'_{B,A}$ is shown in (1). The rest of proof for $V_{A,B} = V'_{A,B}$ can be derived similarly.

$$
\begin{aligned}
V_{B,A} &= H_2(U_{B,A} \| n_{A1} \| n_{B1} \| 0) = H_2(e(W_{A1}, s_{B1} \cdot K_B) \| n_{A1} \| n_{B1} \| 0) \\
&= H_2(e(s_{A1} \cdot H_1(ID_A), s_{B1} \cdot g_B \cdot H_1(ID_B)) \| n_{A1} \| n_{B1} \| 0) \\
&= H_2(e(s_{A1} \cdot g_B \cdot H_1(ID_A), s_{B1} \cdot H_1(ID_B)) \| n_{A1} \| n_{B1} \| 0) \\
&= H_2(e(s_{B1} \cdot H_1(ID_B), s_{A1} \cdot g_B \cdot H_1(ID_A)) \| n_{A1} \| n_{B1} \| 0) \\
&= H_2(e(W_{B1}, s_{A1} \cdot K_A) \| n_{A1} \| n_{B1} \| 0) \quad \text{//iff, } g_A = g_B \\
&= H_2(U'_{B,A} \| n_{A1} \| n_{B1} \| 0) = V'_{B,A} \qquad (1)
\end{aligned}
$$

The secret handshake protocol guarantees that 1) Bob cannot link Alice's real-world identity to her BFDChain address after authenticating her credential. 2) Alice's credential is unlinkable. In future the same kind of transactions won't be linked to her previous transaction. Furthermore, even the CA cannot know that it was Alice who made the purchase with Bob given the public ledger of BFDChain.

V. SECURITY ANALYSIS

In the proposed USH protocol we consider any passive attack that aims at compromising confidentiality of sensitive information by analyzing transmitted messages, e.g., dictionary attack to transmitted messages. We consider both outside adversaries and inside adversaries as follows.

Outside adversary: a malicious party that is an outsider of an authentication process. An outside adversary does not participate in the authentication process and knows nothing about the transmitted messages.

Inside adversary: a malicious party that participates in an authentication process. The adversarial insider may have some matched group secrets with the targeted victim and try to discover other unmatched group secrets the victim has.

Based on the adversary model and attack model introduced above, we claim that our privacy-preserving correlation technique provides the following main security properties: unlinkability with reusable credential and group membership authenticity. In Theorem 1 we first prove that USH that uses homomorphic randomization function provides unlinkability with reusable credential.

Theorem 1. USH holds Unlinkability on Homomorphic Randomization

A privacy-preserving authentication protocol that uses homomorphic randomization function in protocol initialization phase guarantees unlinkability with reusable credential.

Proof: Suppose user k requests to join a group with group secret $g_k \in G_2$. The CA grants the group membership to k by issuing a credential $g_k \cdot H_1(ID_K) \in G_1$, where ID_k represents k's assigned pseudonym associated with g_k. k cannot deduce g_k from $g_k \cdot H_1(ID_k)$. There are two adversaries, t_1 and t_2, who want to know whether they are interacting with same party and which group this party belongs to. Assume that t_1 is not a member in group g_k and t_2 is. t_2 may communicate with other legitimate owners of group secret g_k, corrupt some valid parties and obtain their secrets. Here we use U^k to denote the set of users who own the group secret g_k. Now we define a *Linkability Detection Game* as follows.

Step 1: The adversaries t_1 *and* t_2 *communicate with k based on their own choices. From* t_1*'s viewpoint, he is interacting with party* k_1*; from* t_2*'s viewpoint, he is interacting with party* k_2.

Step 2: T_2 *selects other parties* U^c *and corrupt them.*

Step 3: After the executions of USH protocol t_1 *and* t_2 *want to find out whether* $k_1 = k_2$. *If yes, they know that they are interacting with a same party and* t_2 *knows that* k_2 *has group secret* g_k.

We say that t_1 and t_2 win the Linkability Detection Game if they find out that $k_1 = k_2$ and k_2 has group secret g_k.
Now we define the following probabilities:

$G_L = Pr[t_1 \text{ and } t_2 \text{ win Linkability Detection Game}] - 0.5$

When t_2 does not compromise any valid owner of U^k, the above probability becomes:

$G_{L|(U^c \cap U^k)=\varnothing} = Pr[t_1 \text{ and } t_2 \text{ win Linkability Detection Game} \mid (U^c \cap U^k)=\varnothing] - 0.5$

Here we say that our protocol holds unlinkability with reusable credential if $G_{A|(U^c \cap U^k)=\varnothing}$ is negligible for any adversaries t_1 *and* t_2. In our protocol k generates random numbers s_{k1} and s_{k2} to manipulate his assigned pseudonym ID_k and get elliptic curve points $W_{k1} = s_{k1} \cdot H_1(ID_k)$ and $W_{k2} = s_{k2} \cdot H_1(ID_k)$ as his randomized identities for t_1 and t_2, respectively. Here W_{k1} and W_{k2} are generated to map k's pseudonyms to random ECC points through the homomorphic randomization function as random oracles. By the assumptions that DLP is hard in G_1, no polynomial-time adversary cannot deduce s_{k1} and s_{k2} from $W_{k1} = s_{k1} \cdot H_1(ID_k)$ and $W_{k1} = s_{k1} \cdot H_1(ID_k)$ and thus cannot know W_{k1} and W_{k2} are generated based on the same pseudonym ID_k. That said, their guess about whether $k_1 = k_2$ is no better than a random guess. Our protocol holds unlinkability with reusable credential.

Theorem 2. USH holds Group Membership Authenticity

In our protocol, the group membership authenticity means that an adversary cannot convince another party that it owns the group memberships as this party has by interacting with him. In other words, the adversary cannot impersonate owners of some targeted group secrets. Because of the hardness of ECDLP and BDH assumption, our protocol provides group membership authenticity that any polynomial-time adversary only has negligible probability of cheating as a valid owner of some group membership without corrupting another valid owner of the targeted group secret. Now we define the *Group Membership Owner Impersonation game*.

Proof: suppose there is an adversary A who aims at impersonating the owner of a group secret s. A may communicate with legitimate owners of the targeted s in network G, corrupt some valid users and obtain their secrets. Here we use U^T to denote the set of users who own the targeted s. A picks a target user $u^T \in U^T$, and wants to convince u^T that A is also an owner of the targeted s, i.e. $A \in U^T$. We define the *Group Membership Owner Impersonation Game* for a randomized, polynomial-time adversary A as follows.
Step 1: The adversary A communicates with owners of the targeted s based on its own choice. A may compromise certain user $U^C \subseteq U$ and obtain their secrets, where U^C denotes the set of compromised users and U denotes the whole user set in the network G.
Step 2: A selects a target user $u^T \notin U^C$ and $u^T \in U^T$, where users in U^T own the targeted s.
Step 3: A wants to convince u^T that A owns the s, i.e. $A \in U^T$.

We say that A wins the Group Membership Owner Impersonation Game if it convinces u^T that it is an owner of s.
Now we define the following probabilities:

$G_A = Pr[A \text{ wins Group Membership Owner Impersonation Game}]$

When A does not compromise any valid owner of s, the above probability becomes:

$G'_{A|(U^C \cap U^T)=\varnothing} = Pr[A \text{ wins Group Membership Owner Impersonation Game} \mid (U^C \cap U^T)=\varnothing]$

Here we say that our protocol holds *Group Membership Authenticity* if $G'_{A|(U^C \cap U^T)=\varnothing}$ is negligible for any adversary A. Similar to Theorem 1, we can further prove that USH holds group membership authenticity. Detail of the proof is omitted here since it is out of the scope of this paper.

Theorem 1 proves that payer only needs to obtain the credential from the central authority once, then he can process this credential with homomorphic randomization and use it any times for authentication without being linked to his real identity or his original credential. Theorem 2 proves that no one can impersonate the owner of the assigned credential.

VI. CONCLUSION

In this paper, we proposed USH to achieve privacy-preserving and anonymous coin spending for BFDChain. We discussed the motivation and provided the formal security analysis of the proposed protocol. It will be critical for the success of any on-chain currency since it allows participants to bring

their real-world membership to the cryptocurrency for their coin spending with unlinkability and anonymity.

REFERENCES

[1] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in Proceedings of the 2012 ACM Conference on Computer and Communications Security, ser. CCS '12. New York, NY, USA: ACM, 2012, pp. 906–917.

[2] N. T. Courtois and L. Bahack, "On subversive miner strategies and block withholding attack in bitcoin digital currency," CoRR, vol. abs/1402.1718, 2014.

[3] D. Ron and A. Shamir, "Quantitative analysis of the full bitcoin transaction graph," in Financial Cryptography and Data Security: 17th International Conference, FC 2013. Springer Berlin Heidelberg, 2013, pp. 6–24.

[4] E. Androulaki, G. O. Karame, M. Roeschlin, T. Scherer, and S. Capkun,"Evaluating user privacy in bitcoin," in Financial Cryptography and Data Security: 17th International Conference, FC 2013. Springer Berlin Heidelberg, 2013, pp. 34–51.

[5] M. Fleder, M. S. Kester, and S. Pillai, "Bitcoin transaction graph analysis," CoRR, vol. abs/1502.01657, 2015.

[6] P. Koshy, D. Koshy, and P. McDaniel, "An analysis of anonymity in bitcoin using p2p network traffic," in Financial Cryptography and Data Security: 18th International Conference, FC 2014,. Springer Berlin Heidelberg, 2014, pp. 469–485.

[7] S. Goldfeder, H. A. Kalodner, D. Reisman, and A. Narayanan, "When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies," CoRR, 2017.

[8] A. Biryukov, D. Khovratovich, and I. Pustogarov, "Deanonymisation of clients in bitcoin p2p network," in Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, ser.CCS '14. ACM, 2014, pp. 15–29.

[9] A. Biryukov and I. Pustogarov, "Bitcoin over tor isn't a good idea," in 2015 IEEE Symposium on Security and Privacy, May 2015, pp. 122–134.

[10] I. Miers, C. Garman, M. Green, and A. D. Rubin, "Zerocoin: Anonymous distributed e-cash from bitcoin," in 2013 IEEE Symposium on Security and Privacy, May 2013, pp. 397–411.

[11] E. B. Sasson, A. Chiesa, C. Garman, M. Green, I. Miers, E. Tromer, and M. Virza, "Zerocash: Decentralized anonymous payments from bitcoin," in 2014 IEEE Symposium on Security and Privacy, May 2014, pp. 459–474.

[12] Ethereum: A secure decentralized generaliaed transaction ledger EIP-150 revision", Gavin Wood