An Identity-Based Blind Signature and Its Application for Privacy Preservation in Bitcoin

Yitao Chen[1], Qi Feng[2], Min Luo[2], Li Li[2], and Debiao He[2(✉)]

[1] Wuhan Maritime Communication Research Institute, Wuhan 430205, China
[2] School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China
{fengqi.whu,mluo,lli}@whu.edu.cn

Abstract. The privacy preservation in Bitcoin is increasingly important, partly due to its huge market capitalization and potential applications in distributed architectures. To protect the privacy of users in Bitcoin, a number of mechanisms have been proposed, where mixing service is a simple and frequently-used mechanism. The work, named Blindcoin, believes that an *unlinkable* blind signature scheme can help to guarantee the anonymity of users at the mixer side. Recently, Sarde and Banerjee presented an identity-based blind signature scheme. However, we found their scheme is vulnerable to a linkability attack. In this paper, we improve their scheme on this weakness and construct two *unlinkable* identity-based blind signature schemes, where one is in the standard setting and the other is in the proxy setting. Our approaches delinearize the two blinding factors so that malicious signer or proxy signer cannot find any helpful information from what she knows. The security, including unlinkability, of our schemes relies on the computational Diffie-Hellman assumption in the random oracle model as analyzed in this paper. We typically show that this is of great important to hide the relationship between message-signature pairs for the privacy-protecting in Bitcoin.

Keywords: Unlinkable blind signature · Privacy preservation · Bitcoin · Proxy blind signature · Identity-based cryptography

1 Introduction

The continued interest in Bitcoin is evident by its market capitalization, for example, it takes a market capitalization of $209,144,466,745 and has been topped the ranking of cryptocurrency since its publication in 2008[1]. However, Bitcoin provides a limited form of privacy preservation: static analysis attacks to de-anonymize an user are possible, even if she always creates pseudonyms when connecting to the Bitcoin system [4,20,25,28,30,31,35,44]. For example,

[1] See https://coinmarketcap.com/.

B. Chen and X. Huang (Eds.): AC3 2021, LNICST 386, pp. 10–24, 2021.
https://doi.org/10.1007/978-3-030-80851-8_2

Androulaki et al. [4] conducted an experiment in an university, where students uses Bitcoin as the daily transaction currency. By utilizing cluster analysis based on the transaction fingerprints, they finally profiled approximately 40% of the participants, even some of them apply a fresh address for each transaction.

Some discussions of the importance and cryptographic mechanisms for privacy preservation in Bitcoin can be found in [10,11,18,24,27,41]. Mixing service, the frequently-used mechanism for protecting privacy since it was proposed by Chaum [8], allows users to mix the input/output relationship of their transactions, within some anonymity set, so that cannot be linked to the correct origin and destination [1–3,5,12,13,26,37]. Although many such mixing services exist, Valenta and Rowan [37] argued the *anonymity means that the users (including sender and receiver of a transaction) should be the only entities that know the mapping from their input address to their output address*. It imposes tight constraint on these services, i.e., the mixer (who performs the mixing step) has no information about the mapping from a transaction's input to output address.

To alleviate the risk of deanonymization on the mixer side, Valenta and Rowan [37] implemented a blind signature within Mixcoin [5]. They conceptually defined a blind signature scheme with three procedures, i.e., blinding (hiding the original message together with a random "blinding factor"), signing (signing the blinded message) and unblinding (removing the "blinding factor" to get a valid signature on the actual message). The core security assumption that the blind signature works well in Valenta and Rowan's protocol is that *the blind signature can be publicly verified while the signer has no information about the connection between the pair of message and signature*.

Up to now, a number of identity-based blind signature schemes [16,19,21,23,42] and proxy blind signature schemes [34,38–40,43] have been proposed. All of them have stated to be unlinkable, i.e., the original signer and proxy signer (who is authorized to sign on behalf of the original signer) can use their private keys to generate a valid signature on the blinded message, and cannot discover which messages were signed by them after the unblinding phase. More recently, Sarde et al. [32] also proposed a new identity-based blind signature scheme from bilinear pairings. Unfortunately, we find that their schemes cannot guarantee unlinkability and we will present this weakness more clearly later.

Therefore, in this paper, we firstly present an attack on the unlinkability of the blind signature scheme by Sarde et al. [32], and construct two unlinkable identity-based blind signature schemes, where one is in the standard setting and the other is in the proxy setting. Our standard unlinkable blind signature scheme improves the scheme presented in [32] by delinearizing two blinding factors such that malicious signer cannot find any helpful information from what she knows. This is of great important to *hide the relationship between message-signature pairs* for the privacy-protecting in Bitcoin. Such an approach also works well in our proxy blind signature, which maintains unlinkability between the message-signature pair both on the original signer and proxy signer side. We theoretically analyze the security and performance of our proposed schemes. While our schemes are slightly slower compared to blind signature schemes presented

in [32], we demonstrate their great practical value by an example of potential application for privacy preservation in Bitcoin.

The rest of the paper is organized as follows. In the Sect. 2, we introduce some preliminaries. Section 3 reviews the blind signature scheme presented by Sarde et al. and Sect. 4 present a de-anonymizing attack on their scheme. Our improved unlinkable blind signature and unlinkable proxy blind signature schemes are described in (resp.) Sect. 5 and Sect. 6. Section 7 focuses on the security of our proposed schemes. In Sect. 8, we provide an theoretical evaluation of the proposed schemes. Finally, we give an example of potential application to privacy preservation in Bitcoin.

2 Preliminaries

In this section, we describe the relevant preliminaries required in the understanding of the proposed scheme.

2.1 Tate Bilinear Pairings

Assume that $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ denote three cycle groups with the same order of prime q. There exists a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ with following properties:

- *Bilinearity*: For any elements $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$ and any integers $x, y \in \mathbb{Z}_q^*$, the equation $e(xP, yQ) = e(P, Q)^{xy}$ holds.
- *Non-degeneracy*: For some elements $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$, the inequation $e(P, Q) \neq \mathbf{1}_{\mathbb{G}_T}$ holds.
- *Computability*: Given two elements $P \in \mathbb{G}_1, Q \in \mathbb{G}_2$, there exist effective algorithms to compute $e(P, Q)$.

2.2 Computational Diffie-Hellman Assumption

Define $\mathbb{G}$ as a finite cycle group with the order of prime number $q = |\mathbb{G}|$ and generator of P. For unknown $x, y \in \mathbb{Z}_q$, the advantage to compute xyP from the tuple (P, xP, yP) for any probabilistic polynomial time (P.P.T) adversary $\mathcal{A}$ is negligible.

2.3 System Model

The architecture of this paper is shown in Fig. 1.

There are four (or five) types of participants with an (proxy) blind signature scheme: the key generation center, a user, a signer, a verifier, and a proxy signer just within proxy blind signature scheme.

- KGC: It is a trusted third party and its task is generating system parameters. Besides, it is also extract the private keys of the user, the signer and the proxy signer according to their identities.
- User: He/She is a client who intents to get signature on message m.

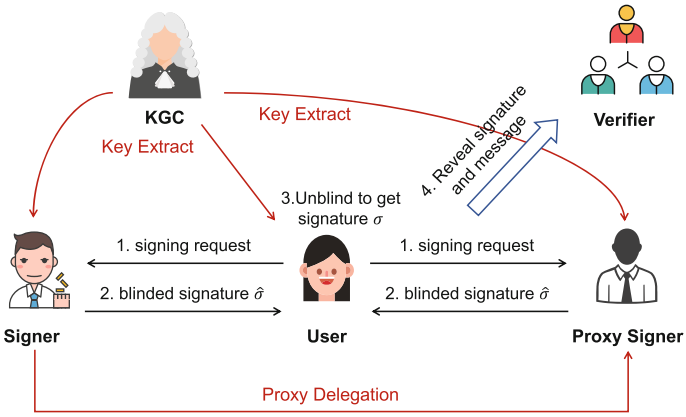

Fig. 1. The system model

- Signer: It is a server provider who gets his/her private key from the KGC and uses it to sign on blinded message provided from the User. After the signature process, he/she can not know any information about original message m.
- Proxy Signer: It is a proxy server provider being authorized to sign on behalf the Signer when the Signer is off-line. It is a specific character within the proxy blind signature scheme.
- Verifier: He/She can verify the signature on message m after the User publicly publishing the unblinded signature.

3 Review of Sarde et al.' Blind Signature Scheme

In [32], Sarde et al.' ID-based blind signature consists of the following algorithms:

- Setup: Let $\mathbb{G}_1$ denotes an additive group of generator P and order q and $\mathbb{G}_2$ denotes a multiplicative group of the same order, bilinear pairings e defines $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$, three cryptographic hash functions are given by $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $h_2 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q$, $h_3 : \{0,1\}^* \times \mathbb{G}_2 \rightarrow \mathbb{Z}_q$. KGC randomly samples master private key $s \leftarrow_R \mathbb{Z}_q^*$ and computes master public key by $P_{pub} = s \cdot P$. Finally, system parameters are publicly set as $\texttt{params} = \{\mathbb{G}_1, \mathbb{G}_2, e, q, P, P_{pub}, H_1, h_2, h_3\}$ and master private key s will be kept securely by KGC.
- Extract:
  1. On receiving signer's identity information ID, KGC computes $Q_{ID} = H_1(ID)$ as his or her public key.
  2. Output $S_{ID} = s \cdot Q_{ID}$ to the signer as private key.
- Blinding Phase:
  1. Signer randomly samples $r \leftarrow_R \mathbb{Z}_q^*$, computes and sends $R = r \cdot Q_{ID}$ to the user.

2. User samples two random values $k_1, k_2 \leftarrow_R \mathbb{Z}_q^*$, computes $u = h_2(R) \cdot k_1 \mod q$, $T = e(k_2 \cdot R + k_1 k_2 \cdot Q_{ID}, P_{pub})$ and $\hat{h} = h_3(m, T) + u \mod q$ and sends $\hat{h}$ to the signer.

– Signing Phase: Signer computes and returns the blinded signature $\hat{S} = (\hat{h} + r) \cdot S_{ID}$ to the user.
– Unblinding Phase:
  1. User unblinds the signature by $S = k_2 \cdot \hat{S}$, $h = \hat{h} - u \mod q$ and $d = k_2 \cdot (\hat{h} - k_1) \mod q$.
  2. Output the blind signature $\sigma = (S, h, d)$ of message m.
– Verify: On input a signature σ of message m, public key Q_{ID} and system parameters `params`, the verifier accepts the signature if and only if $h = h_3(m, e(S, P) \cdot e(Q_{ID}, P_{pub})^{-d})$ holds.

4 Attack on Sarde et al.' Blind Signature Scheme

In this section, we show that a curious signer in their blind signature scheme can link a signature with a signing requester:

1. Given a tuple of transcripts $\{r, R, S_{ID}, \hat{h}\}$ and candidate signature $\{S^*, h^*, d^*\}$ of message m^*, the signer firstly computes

$$k_1 = (\hat{h} - h^*) \cdot (h_2(R))^{-1}, k_2 = d^* \cdot (\hat{h} - k_1)^{-1}$$

2. Now the signer can check whether

$$e(k_2^{-1} \cdot S^*, P) = e(R + \hat{h} \cdot Q_{ID}, P_{pub})$$

to discover the signing requester.

This attack is workable because $\{r, R, S_{ID}, \hat{h}\}$ and $\{S^*, h^*, d^*\}$ of message m^* are all known to the signer and

$$\begin{aligned} e(k_2^{-1} \cdot S^*, P) &= e(k_2^{-1} \cdot (k_2^* \cdot (\hat{h}^* + r^*)) \cdot S_{ID}, P) \\ &= e((\hat{h}^* + r^*) \cdot S_{ID}, P)^{k_2^{-1} \cdot k_2^*} = e((\hat{h}^* + r^*) \cdot Q_{ID}, P_{pub})^{k_2^{-1} \cdot k_2^*} \end{aligned}$$

On the right side,

$$e(R + \hat{h} \cdot Q_{ID}, P_{pub}) = e((r + \hat{h}) \cdot Q_{ID}, P_{pub})$$

It is obliviously possible to discover the signing requester, because the probability of $k_2^{-1} \cdot k_2^* \cdot (\hat{h}^* + r^*) = r + \hat{h}$ is negligible on the conditions of 1) signer's random number $r^* \neq r$; 2) user's blinding factor $k_2 \neq k_2^*$; 3) the blinded message $\hat{h}^* \neq \hat{h}$ (defined jointly by signer's random number, user's blinding factor and one-way hash function).

Thus after the user publishes message-signature pair in public, a curious signer can link them to certain tuple that she keeps. Once we apply Sarde et al.' scheme for the privacy preservation in Bitcoin, a malicious mixer can trace the relationship between the transaction's sender and receiver without being authorized by the users. This attacks can work because the public u is multiplied by two integers. So it is easily factorizable if one integer is known. Our improvement is put the blind factor k_1 into the hash operation and guarantee security on the basis of one-way hash function.

5 Unlinkable ID-Based Blind Signature Scheme

In this section, we propose an improved version of Sarde et al.'s scheme, which satisfies untraceability besides the merits of original scheme.

- Setup: The proposed unlinkable proxy blind signature scheme are parameterized by $\texttt{params} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, P_1, P_2, P_{pub}, h_1, h_2\}$ where $\mathbb{G}_1$ and $\mathbb{G}_2$ are two group of the same order q, Tate bilinear pairing $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$ (an asymmetric pairing which is faster than the Weil bilinear pairing that was used in [32]), P_1 and P_2 denote the generators of $\mathbb{G}_1$ and $\mathbb{G}_2$, $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_1$, $h_2 : \mathbb{G}_1 \rightarrow \mathbb{Z}_q^*$ and $h_3 : \{0,1\}^* \times \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$ define three cryptographic hash functions. The master private key $s \leftarrow_R \mathbb{Z}_q^*$ is chosen by KGC and $P_{pub} = s \cdot P_2$ is the master public key. Finally, KGC publishes the system parameter $\texttt{params}$ and keeps s securely.
- Extract: On receiving signer's identity string ID, KGC computes $Q_{ID} = H_1(ID)$ as his or her public key, and returns the corresponding private key $S_{ID} = s \cdot Q_{ID}$ to the signer.
- Blinding Phase:
  1. Signer randomly samples $r \leftarrow_R \mathbb{Z}_q^*$, computes and sends $R = r \cdot Q_{ID}$ to the user.
  2. User samples two random values $k_1, k_2 \leftarrow_R \mathbb{Z}_q^*$, computes

$$u = h_2(k_1 \cdot R),$$

  $T = e(k_2 \cdot R + k_1 k_2 \cdot Q_{ID}, P_{pub})$, $\hat{h} = h_3(m, T) + u \mod q$ and sends $\hat{h}$ to the signer.
- Signing Phase: Signer computes and returns the blinded signature $\hat{S} = (\hat{h} + r) \cdot S_{ID}$ to the user.
- Unblinding Phase:
  1. User unblinds the signature by $S = k_2 \cdot \hat{S}$, $h = \hat{h} - u \mod q$, $d = k_2 \cdot (\hat{h} - k_1) \mod q$
  2. Output the blind signature $\sigma = (S, h, d)$ of message m.
- Verify: On input a signature σ of message m, public key Q_{ID} and system parameters $\texttt{params}$, the verifier accepts the signature if and only if $h = h_3(m, e(S, P) \cdot e(Q_{ID}, P_{pub})^{-d})$ holds.

Correctness. The correctness of our scheme can be verified following the prior work of Sarde et al.. Thus it is omitted here for simplicity.

6 Unlinkable ID-Based Proxy Blind Signature Scheme

Proxy signature is such a blind signature that a proxy signer is authorized to generate a blind signature on behalf of the original signer, and neither original signer nor proxy signer know the message. In this section, we further propose a unlinkable proxy blind signature scheme for convenience, economy and meet the

cooperation demands of the modern companies, for example, multiple company's managers are assigned to answer the mixing services in turns.

Our construction is also built on the Sarde et al.'s proxy blind signature scheme [32]. We can see that their scheme fails to satisfy the unlinkability, as for a curious proxy signer, the hash value h received from user is exactly identical to that parsed from signature. Here, we improve their version and fill in gap of unlinkability.

Our unlinkable proxy blind signature scheme consists of following five phases:

- Setup: The proposed unlinkable proxy blind signature scheme has similar definition around the system parameters $\texttt{params} = \{\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, q, P_1, P_2, g, P_{pub}, H_1, h_2\}$ as standard one, except the cryptographic hash function $h_2 : \{0,1\}^* \times \mathbb{G}_T \rightarrow \mathbb{Z}_q^*$ and element $g = e(P_1, P_2)$. Finally, KGC publishes $\texttt{params}$ and keeps s securely itself.
- Extract: On receiving original signer's identity ID_s and proxy signer's identity ID_p, KGC computes $Q_{ID_s} = H_1(ID_s)$ and $Q_{ID_p} = H_1(ID_p)$ as their corresponding public keys, and returns the private keys $S_{ID_s} = s \cdot Q_{ID_s}$ and $S_{ID_p} = s \cdot Q_{ID_p}$ to the original signer and proxy signer respectively.
- Proxy Delegation:
  1. *Proxy Generation*: The original signer randomly samples $\alpha \leftarrow_R \mathbb{Z}_q^*$, computes $V = g^\alpha$, $A = S_{ID_s} \cdot V + \alpha \cdot P_1$, $\gamma = h_2(\omega, e(A, P_2))$ and $U = \gamma \cdot S_{ID_s} + \alpha \cdot P_1$, where ω is the proxy warrant which consists the identity information of the original signer and the proxy signer, message type to be signed by the proxy signer, the delegation limits of authority, valid period of delegation and so on.
  2. *Proxy Delivery*: The original signer sends (ω, V, U) to a proxy signer and publishes the warrant-voucher pair (ω, V) in public.
  3. *Proxy Verification*: Upon receiving secret values from the original signer, the proxy signer accepts it if the following equations holds:

$$e(U, P_2) = e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(Q_{ID_s}, P_{pub})^V \cdot V)} \cdot V \tag{1}$$

- Proxy blind signature generation: When given the message m, user and proxy signer execute the following steps to generate a proxy blind signature:
  1. *Blinding*:
     (a) The proxy signer randomly samples $r \leftarrow_R \mathbb{Z}_q^*$, computes $R = g^r$, $K_p = U + R \cdot S_{ID_p}$ and sends R to the user via secure channel.
     (b) Now the user samples two random values $k_1, k_2 \leftarrow_R \mathbb{Z}_q^*$ and blind the message by $T = R \cdot g^{k_1} \cdot V^{k_2} \cdot e(Q_{ID_s}, P_{pub})^{\gamma k_2} \cdot e(Q_{ID_p}, P_{pub})^{Rk_2 + k_1 k_2}$, $\hat{h} = h_1(m, T) + k_1 \cdot k_2^{-1}$, and sends $\hat{h}$ to the signer.
  2. *Signing*: The proxy signer computes and returns the blinded signature $\hat{S} = \hat{h} \cdot K_p + r \cdot P_1$ to the user.
  3. *Unblinding*: Upon receiving $\hat{S}$, user unblinds it by computing $S = \hat{S} + k_1 \cdot P_1$, $h = \hat{h} - k_2$ and $d = R \cdot h - k_1 \cdot k_2$. The proxy blind signature on the message m is $\sigma = (S, h, d)$.

– Verify: On input a signature σ of message m, public keys Q_{ID_s}, Q_{ID_p}, proxy warrant-voucher pair (ω, V), the verifier firstly recovers $\gamma = h_2(\omega, e(Q_{ID_s}, P_{pub})^V \cdot V)$, then accepts the signature if

$$h = h_1(m, e(S, P_2) \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h}) \quad (2)$$

holds.

Correctness

The correctness of Eq. (1), i.e., the proxy verification can be proved as follows:

$$\begin{aligned} e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(Q_{ID_s}, P_{pub})^V \cdot V)} \cdot V &= e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(S_{ID_s} \cdot V, P_2) \cdot V)} \cdot V \\ &= e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(S_{ID_s} \cdot V, P_2) \cdot g^{\alpha})} \cdot g^{\alpha} \\ &= e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(S_{ID_s} \cdot V + \alpha \cdot P_1, P_2))} \cdot g^{\alpha} \\ &= e(Q_{ID_s}, P_{pub})^{h_2(\omega, e(A, P_2))} \cdot g^{\alpha} = e(Q_{ID_s}, P_{pub})^{\gamma} \cdot g^{\alpha} \\ &= e(\gamma \cdot S_{ID_s} + \alpha \cdot P_1, P_2) = e(U, P_2) \end{aligned}$$

Furthermore, the correctness of Eq. (2), i.e., the proxy blind signature verification can be proved as follows.

$$\begin{aligned} & e(S, P_2) \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(\hat{S} + k_1 \cdot P_1, P_2) \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(\hat{h} \cdot K_p + r \cdot P_1 + k_1 \cdot P_1, P_2) \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(U + R \cdot S_{ID_p}, P_1)^{h+k_2} \cdot g^{r} \cdot g^{k_1} \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(\gamma \cdot S_{ID_s} + \alpha \cdot P_1 + R \cdot S_{ID_p}, P_2)^{h+k_2} \cdot R \cdot g^{k_1} \cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \\ &\cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(S_{ID_s}, P_2)^{\gamma \cdot (h+k_2)} \cdot g^{\alpha \cdot (h+k_2)} \cdot e(S_{ID_p}, P_2)^{R \cdot (h+k_2)} \cdot R \cdot g^{k_1} \\ &\cdot e(Q_{ID_s}, P_{pub})^{-h\gamma} \cdot e(Q_{ID_p}, P_{pub})^{-d} \cdot V^{-h} \\ &= e(Q_{ID_s}, P_{pub})^{\gamma \cdot k_2} \cdot V^{k_2} \cdot e(Q_{ID_p}, P_{pub})^{R \cdot (h+k_2) - d} \cdot R \cdot g^{k_1} \\ &= T \end{aligned}$$

7 Security Analysis

In this section, we will show that the proposed improved schemes meet the security requirements, especially the *unlinkability* that plays a critical role for the privacy preservation in Bitcoin.

7.1 Analysis of Blind Signature

A blind signature scheme should meet three security properties: blindness, unforgeability, and unlinkability [7,9,15,17,33]. Now we examine the security of our scheme described in Sect. 5 according to the property:

- *Blindness*: To protect the privacy of signed message, the signer should not know the content of message when she signs. In our scheme, user securely samples two random values k_1, k_2 and calculates T and u to randomize the message m using one-way hash function. We can see that, at this point, the signer has no information about T and u, thus she cannot obtain the content of message.
- *Unforgeability*: To guarantee authenticity and non-repudiation of signature, no one, except the signer, can produce a valid blind signature without permission. The unforgeability of our scheme is lying on signer's private key $S_{ID} = s \cdot Q_{ID}$ (calculated by KGC's master private key s) and random value r. It is easy to prove that our scheme can be reduced to CDH assumption in the random oracle model using Forking Lemma [29]. Thus, no one can forge a valid signature without private key.
- *Unlinkability*: To provide a fully preservation of message's privacy, the signer should not trace the connection between revealed signature and the blinded message she signed before. In our scheme, before the message m and its signature $\sigma = \{S, h, d\}$ are published, the user will break the linear relationship of blinding factors k_1, k_2 among S and d, which means that the signer cannot find any helpful information from the tuple of $\{r, Q_{ID}, S_{ID}, \hat{h}\}$ she knows. Thus, it is hard for signer to link the blinded signature $\{\hat{S}, \hat{h}\}$ with the public message-signature pair.

7.2 Analysis of Proxy Blind Signature

A proxy blind signature scheme should meet seven security properties: distinguishable, identifiablity, prevention of misuse, non-repudiation, unforgeability, verifiability and unlinkability [6,14,22,36]. Here, we analyze that our scheme in Sect. 6 satisfies these properties:

- *Distinguishability*: To maintain clear boundaries of responsibility, the proxy blind signature generated by proxy signer should be distinguishable from the normal one by original signer. In our scheme, the original signer's private key is S_{ID_s} calculated from ID_s while the proxy signer's private key is S_{ID_p} calculated from different ID_p. Furthermore, the warrant ω, who consists the detail proxy information, is one of the components of the proxy key W and finally embedded into the proxy blind signature. Thus, one can distinguish the proxy blind signature from a normal one easily by verifying the validity of the proxy blind signature.
- *Identifiablity*: For publicly verifiable accountability, the proxy signer, original signer and their agency relationships should be efficiently identified. Using the proxy warrant-voucher pair (ω, V) and both signer's identities, one can verify the validity of blind signature $\sigma = \{S, h, d\}$. However, original signer's identity ID_s and proxy signer's identity ID_p appear in different location of the verification equation (2), it will be unacceptable if any one of them is mismatched. Thus, the proxy signer can be efficiently identified from the proxy signature.

- *Prevention of misuse*: To protect the interests of the original signer, any misuse of proxy key pair deviated from producing proxy signature could be detected publicly. In our improved scheme, the original signer issues $U = h_2(\omega, e(A, P)) \cdot S_{ID_s} + \alpha \cdot P$ during the proxy generation, where the warrant ω consists the detail proxy information, such as identities of both parties, message type to be signed by the proxy signer, the delegation limits of authority, valid period of delegation and so on. Based on the security of original signer's private key S_{ID_s} and random value α, the proxy signer cannot sign any messages deviated from the warrant ω.
- *Non-repudiation*: The proxy blind signature is a proof of both proxy signer and original, therefore, a proxy blind scheme should guarantee neither of them can later deny their signatures. During the blinding phase of our scheme, the proxy key K_p is created by original signer's private key S_{ID_s} and proxy signer's private key S_{ID_p}. Cooperating with the identifiablity analyzed before, we can say that neither of them can sign in place of the other party nor both of them can deny having signed the message.
- *Unforgeability*: To guarantee authenticity of signature, no one, except the proxy signer, can produce a valid proxy blind signature without permission. The unforgeability of our scheme is lying on original signer's private key $S_{ID_s} = s \cdot Q_{ID_s}$, proxy signer's private key $S_{ID_p} = s \cdot Q_{ID_p}$ (both are calculated by KGC's master private key s) and random values α, r. Similarly, based on CDH assumption, it is easy to prove that an P.P.T adversary (even a original signer or signature receiver) cannot forge in our improved scheme without private key.
- *Verifiability*: The proxy blind signature should be verified by anyone. Our improved blind signature can satisfy verifiability as the verifier can check the validity of the proxy blind signature using Eq. (2). The correctness of it is already shown by Eq. (1).
- *Unlinkability*: To protect the privacy of message, the signer (*including proxy signer and original signer*) should not trace the connection between revealed signature and the blinded message she signed before. In the proposed scheme, on one side, the original signer or a verifier has no information about random factor r and blinding factors k_1, k_2, so it's difficult for them to find the relationship between the blinded signature $\{\hat{h}, \hat{S}\}$ and proxy blind signature $\sigma = \{S, h, d\}$. On the other side, given all the signature transcript $\{\hat{h}_i, R_i, \hat{S}_i\}$, it is still unable for proxy signer to link a published proxy blind signature $\sigma = \{S, h, d\}$ with one she signed before because she still has no helpful information of blinding factors k_1, k_2 (including $e(Q_{ID_s}, P_{pub})^{k_2}$, V^{k_s}, $e(Q_{ID_p}, P_{pub})^{k_1 \cdot k_2}$, $e(Q_{ID_p}, P_{pub})^{k_2}$, $e(P, P)^{k_1}$ and so on). Thus, our improved proxy blind signature scheme can achieve perfectly unlinkability.

8 Performance Analysis and Comparison

To show the practicality of our protocols proposed in Sect. 5 and 6, we analyze their performance and compare them with Sarde et al.' schemes [32]. The notations used in this section are as follows:

- T_{bp}: the execution time of the bilinear pairing.
- T_{sm}: the execution time of scalar multiplication over group $\mathbb{G}_1$.
- T_{gm}: the execution time of multiplication of two elements over group $\mathbb{G}_T$ (and $\mathbb{G}_2$ in Sarde et al.).
- T_{ga}: the execution time of addition of two elements over group $\mathbb{G}_1$.
- T_{inv}: the execution time of inversion of an integer under modulo q.
- T_{im}: the execution time of integer multiplication modulo q.
- T_h: the execution time of hashing.

Table 1. Comparison of computational cost of blind signature

Phases	Sarde et al. [32]	This paper
Extract	$T_{sm} + T_h$	$T_{sm} + T_h$
Blinding	$T_{bp} + 3 \times T_{sm} + 2 \times T_{im} + 2 \times T_h$	$T_{bp} + 4 \times T_{sm} + T_{im} + 2 \times T_h$
Signing	T_{sm}	T_{sm}
Unblinding	$T_{sm} + T_{im}$	$T_{sm} + T_{im}$
Verification	$2 \times T_{bp} + T_{sm} + T_{inv} + T_h$	$2 \times T_{bp} + T_{sm} + T_{inv} + T_h$

The comparison of algebraic operations required for different phases are summarized in Table 1 and Table 2. We mark that the computational cost for unlinkability of blind signature is T_{sm} in the binding phase. The computational cost for unlinkability of proxy blind signature is $T_{bp} + 2 \times T_{sm} + 2 \times T_{im}$ in the blinding phase and T_{im} in the unblinding phase of proxy blind signature generation. However, our proposed schemes still benefit from the higher-performance of Type-3 bilinear pairing, which has been optimized many years and can be executed faster than the symmetrical pairing used by Sarde et al.

9 Application for Privacy Preservation in Bitcoin

We review the Blindcoin project [37] in this section, which utilizes a blind signature scheme to hide the mapping between a user's input and output addresses from mix.

According to the transaction architecture (i.e., UTXO or unspent transaction outputs) of Bitcoin, it is linkable between senders and receivers within a transaction, therefore, by analyzing the public content (e.g., analytical attack), one can infer some privacy information. Some analysis attacks have succeed to extract users' identities [4,20,25,28,30,31,35,44]. The simplest solution to mitigate this attack is to obfuscate the transaction's relationships with the help of mixer. Senders wrap transactions with the output addresses of mixer, and then mixer wrap other irrelevant transactions with the output addresses of receivers. When a massive of transactions engage in this mixing task, the relationships

Table 2. Comparison of computational cost of proxy blind signature

Phases		**Sarde et al. [32]**	**This paper**
Extract		$2 \times (T_{sm} + T_h)$	$2 \times (T_{sm} + T_h)$
Proxy delegation	**Proxy generation**	$T_{bp}+4\times T_{sm}+T_{gm}+T_{ga}+T_h$	$T_{bp}+4\times T_{sm}+T_{gm}+T_{ga}+T_h$
	Proxy verification	$3 \times T_{bp} + T_{sm} + 3 \times T_{gm} + T_h$	$3 \times T_{bp} + T_{sm} + 3 \times T_{gm} + T_h$
Proxy blind signature generation	**Blinding**	$3 \times T_{bp} + 4 \times T_{sm} + 4 \times T_{gm} + T_{ga} + T_h$	$4 \times T_{bp} + 6 \times T_{sm} + 4 \times T_{gm} + T_{ga} + 2 \times T_{im} + T_h$
	Signing	$2 \times T_{sm}$	$2 \times T_{sm}$
	Unblinding	$2 \times T_{sm} + 2 \times T_{ga}$	$2 \times T_{sm} + 2 \times T_{ga} + T_{im}$
Verification		$4 \times T_{bp} + 4 \times T_{gm} + 4 \times T_{sm} + 3 \times T_{inv} + 2 \times T_h$	$4 \times T_{bp} + 4 \times T_{gm} + 4 \times T_{sm} + 3 \times T_{inv} + 2 \times T_h$

between each transaction's origin and destination are hided well. Blindcoin, presented by Valenta et al., followed the mixing mechanism and combine the blind signature scheme to hide the user's privacy at the mixer side. There are three kind of participants in Blindcoin, i.e., sender $\mathcal{S}$, mixer $\mathcal{M}$ and receiver $\mathcal{R}$. Sender $\mathcal{S}$ anonymously transfers a mount of Bitcoin to receiver $\mathcal{R}$ with the assistance of mixer $\mathcal{M}$ by executing following steps:

- Setup: The mixer $\mathcal{M}$ publishes the mix parameters $\mathsf{mixparams}$ into the public ledger including the expiry date t_1, t_2, t_3, t_4, the value v of Bitcoin put into a transaction, the mixing fee ρ, block chunk ω and so on.
- Sender Submits Offer: The sender $\mathcal{S}$ sends its offer to $\mathcal{M}$ including $\mathsf{mixparams}$ and blinded token $[T = \{\mathsf{addr}_{\mathsf{out}}, n\}]_{\mathsf{Blind}}$. This token T contains the output address $\mathsf{addr}_{\mathsf{out}}$ and secure random number n, and being masked using the blinding phase of blind signature scheme.
- Mixer Answers Partial Warranty: If $\mathcal{M}$ accepts this offer, then it wrap partial warranty using blinded token T, an escrow address $\mathsf{addr}_{\mathsf{esc}}$ for the sender to pay to, and $\mathsf{mixparams}$. This partial warranty will be signed using signing algorithm of blind signature scheme, i.e., forming $\mathsf{PriW} = \{[T]_{\mathsf{Blind}}, \mathsf{addr}_{\mathsf{esc}}, \mathsf{mixparams}\}_{\mathsf{sig}}$.
- Sender Pays: Sender $\mathcal{S}$ then transfer v amount of Bitcoin from any input address $\mathsf{addr}_{\mathsf{in}}$ to $\mathsf{addr}_{\mathsf{esc}}$ by time t_1.
- Mixer Completes Warranty: Once the sender pays the funds, mixer $\mathcal{M}$ must complete the warranty by signing the blinded token, i.e., forming $\mathsf{PubW} = \{[T]_{\mathsf{Blind}}\}_{\mathsf{sig}}$ and publishes it to the public ledger by time t_2. This public information allows any third party to verify that the sender did indeed transfer their funds to the escrow address on time and the mixer has completed the warranty by time t_2.
- Sender Anonymously Unblinds Output Addresses: After seeing PubW in the public ledger, $\mathcal{S}$ can apply unblinding phase to recover the signed token,

formed as $\{T = \{\mathsf{addr}_{\mathsf{out}}, n\}\}_{\mathsf{sig}}$. The sender (anonymously connects with another identity $\mathcal{S}'$) posts the signed token to the public log by time t_3.

- Mixer Pays to Output Address: Once the signed output address being published in public ledger by time t_3, $\mathcal{M}$ computes a beacon function with the inputs of t_3, n and block chunk ω for each token $T = \{\mathsf{addr}_{\mathsf{out}}, n\}$. The chunk destined for that output address will be kept by $\mathcal{M}$ if $\mathsf{Beacon}(t_3, \omega, n) \leq \rho$; else, $\mathcal{M}$ pays v amount of Bitcoin to all unblinded output addresses before time t_4.
- One Party Cheats: If $\mathcal{M}$ fails to pay a chunk to each of the output addresses that are passed the beacon function by time t_4, then $\mathcal{S}$ can publish the partial warranty $\mathsf{PriW} = \{[T]_{\mathsf{Blind}}, \mathsf{addr}_{\mathsf{esc}}, \mathsf{mixparams}\}_{\mathsf{sig}}$, and all the public information (e.g., transaction $\mathsf{tx}(v, \mathsf{addr}_{\mathsf{in}}, \mathsf{addr}_{\mathsf{esc}})$ presented by time t_1, the signed token $\{T = \{\mathsf{addr}_{\mathsf{out}}, n\}\}_{\mathsf{sig}}$) to incriminate $\mathcal{M}$. Every verifier can check the public log and blockchain to see if both parties followed the protocol honestly.

Discussion and Conclusion. Based on the *unlinkability* of blind signature scheme, $\mathcal{M}$ can check the validity of output address but cannot link the output address $\mathsf{addr}_{\mathsf{out}}$ to the corresponding input address $\mathsf{addr}_{\mathsf{in}}$. This is why unlinkability is a core feature to guarantee privacy of Bitcoin even a mixer (who knows many information) is curious or malicious. Therefore, our proposed protocols are extremely valuable for the privacy preservation in Bitcoin.

Acknowledgments. We would like to thank the anonymous reviewers. The work was supported by the National Natural Science Foundation of China (Nos. 61972294, 61932016), the Special Project on Science and Technology Program of Hubei Province (No. 2020AEA013), the Natural Science Foundation of Hubei Province (No. 2020CFA052) and the Wuhan Municipal Science and Technology Project (No. 2020010601012187).

References

1. Bitcoin fog. http://bitcoinfog.com. Accessed 2020
2. Bitmixer. https://bitcointalk.org/index.php?topic=415396.160. Accessed 2020
3. Onionbc. http://6fgd4togcynxyclb.onion/. Accessed 2020
4. Androulaki, E., Karame, G.O., Roeschlin, M., Scherer, T., Capkun, S.: Evaluating user privacy in bitcoin. In: Sadeghi, A.-R. (ed.) FC 2013. LNCS, vol. 7859, pp. 34–51. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39884-1_4
5. Bonneau, J., Narayanan, A., Miller, A., Clark, J., Kroll, J.A., Felten, E.W.: Mixcoin: anonymity for bitcoin with accountable mixes. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 486–504. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_31
6. Chande, M.K., Lee, C.C., Li, C.T.: Cryptanalysis and improvement of a ECDLP based proxy blind signature scheme. J. Discrete Math. Sci. Cryptogr. **21**(1), 23–34 (2018)
7. Chaum, D.: Blind signature system. In: Chaum, D. (eds.) Advances in Cryptology, pp. 153–153. Springer, Boston (1984). https://doi.org/10.1007/978-1-4684-4730-9_14

8. Chaum, D.L.: Untraceable electronic mail, return addresses, and digital pseudonyms. Commun. ACM **24**(2), 84–90 (1981)
9. Fan, C.I., Chen, W.K., Yeh, Y.S.: Randomization enhanced Chaum's blind signature scheme. Comput. Commun. **23**(17), 1677–1680 (2000)
10. Feng, Q., He, D., Zeadally, S., Khan, M.K., Kumar, N.: A survey on privacy protection in blockchain system. J. Network Comput. Appl. **126**, 45–58 (2019)
11. Genkin, D., Papadopoulos, D., Papamanthou, C.: Privacy in decentralized cryptocurrencies. Commun. ACM **61**(6), 78–88 (2018)
12. Heilman, E., Alshenibr, L., Baldimtsi, F., Scafuro, A., Goldberg, S.: TumbleBit: an untrusted bitcoin-compatible anonymous payment hub. In: Network and Distributed System Security Symposium (2017)
13. Heilman, E., Baldimtsi, F., Goldberg, S.: Blindly signed contracts: anonymous on-blockchain and off-blockchain bitcoin transactions. In: Clark, J., Meiklejohn, S., Ryan, P.Y.A., Wallach, D., Brenner, M., Rohloff, K. (eds.) FC 2016. LNCS, vol. 9604, pp. 43–60. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53357-4_4
14. Hu, L., Zheng, K., Hu, Z., Yang, Y.: A secure proxy blind signature scheme based on ECDLP. In: 2009 International Conference on Multimedia Information Networking and Security, vol. 1, pp. 454–457. IEEE (2009)
15. Hwang, M.S., Lee, C.C., Lai, Y.C.: An untraceable blind signature scheme. IEICE Trans. Fundam. Electron. Commun. Comput. Sci. **86**(7), 1902–1906 (2003)
16. James, S., Gowri, T., Babu, G., Reddy, P.V.: Identity-based blind signature scheme with message recovery. Int. J. Electri. Comput. Eng. (2088–8708) **7**(5) (2017)
17. Juang, W.S., Lei, C.L.: Partially blind threshold signatures based on discrete logarithm. Comput. Commun. **22**(1), 73–86 (1999)
18. Khalilov, M.C.K., Levi, A.: A survey on anonymity and privacy in bitcoin-like digital cash systems. IEEE Commun. Surv. Tutor. **20**(3), 2543–2585 (2018)
19. Kong, W., Shen, J., Vijayakumar, P., Cho, Y., Chang, V.: A practical group blind signature scheme for privacy protection in smart grid. J. Parallel Distrib. Comput. **136**, 29–39 (2020)
20. Koshy, P., Koshy, D., McDaniel, P.: An analysis of anonymity in bitcoin using P2P network traffic. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 469–485. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_30
21. Kumar, M., Katti, C.P., Saxena, P.C.: A secure anonymous e-voting system using identity-based blind signature scheme. In: Shyamasundar, R.K., Singh, V., Vaidya, J. (eds.) ICISS 2017. LNCS, vol. 10717, pp. 29–49. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-72598-7_3
22. Lal, S., Awasthi, A.K.: Proxy blind signature scheme. J. Inf. Sci. Eng. Cryptology ePrint Archive, Report 72 (2003)
23. Li, J., Zhang, Y., Yang, S.: Cryptanalysis of new proxy blind signature scheme with warrant. In: International Conference of Computational Methods in Sciences and Engineering (ICCMSE 2005) (2005)
24. Li, X., Jiang, P., Chen, T., Luo, X., Wen, Q.: A survey on the security of blockchain systems. Future Gener. Comput. Syst. **107**, 841–853 (2020)
25. Liao, K., Zhao, Z., Doupé, A., Ahn, G.J.: Behind closed doors: measurement and analysis of Cryptolocker ransoms in bitcoin. In: 2016 APWG Symposium on Electronic Crime Research (eCrime), pp. 1–13. IEEE (2016)
26. Maxwell, G.: CoinSwap: transaction graph disjoint trustless trading, October 2013

27. Meiklejohn, S., Orlandi, C.: Privacy-enhancing overlays in bitcoin. In: Brenner, M., Christin, N., Johnson, B., Rohloff, K. (eds.) FC 2015. LNCS, vol. 8976, pp. 127–141. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48051-9_10
28. Ober, M., Katzenbeisser, S., Hamacher, K.: Structure and anonymity of the bitcoin transaction graph. Future Internet **5**(2), 237–250 (2013)
29. Pointcheval, D., Stern, J.: Security proofs for signature schemes. In: Maurer, U. (ed.) EUROCRYPT 1996. LNCS, vol. 1070, pp. 387–398. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-68339-9_33
30. Reid, F., Harrigan, M.: An analysis of anonymity in the bitcoin system. In: Altshuler, Y., Elovici, Y., Cremers, A., Aharony, N., Pentland, A. (eds.) Security and Privacy in Social Networks, pp. 197–223. Springer, New York (2013). https://doi.org/10.1007/978-1-4614-4139-7_10
31. Reynolds, P., Irwin, A.S.: Tracking digital footprints: anonymity within the bitcoin system. J. Money Laundering Control (2017)
32. Sarde, P., Banerjee, A.: A secure ID-based blind and proxy blind signature scheme from bilinear pairings. J. Appl. Secur. Res. **12**(2), 276–286 (2017)
33. Shao, Z.: Improved user efficient blind signatures. Electron. Lett. **36**(16), 1372–1374 (2000)
34. Shaobin, W., Fan, H., Guohua, C.: Secure efficient proxy blind signature schemes based DLP. In: Seventh IEEE International Conference on E-Commerce Technology (CEC 2005), pp. 452–455. IEEE (2005)
35. Spagnuolo, M., Maggi, F., Zanero, S.: BitIodine: extracting intelligence from the bitcoin network. In: Christin, N., Safavi-Naini, R. (eds.) FC 2014. LNCS, vol. 8437, pp. 457–468. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-662-45472-5_29
36. Tan, Z., Liu, Z., Tang, C.: Digital proxy blind signature schemes based on DLP and ECDLP. MM Res. Preprints **21**(7), 212–217 (2002)
37. Valenta, L., Rowan, B.: Blindcoin: blinded, accountable mixes for bitcoin. In: Brenner, M., Christin, N., Johnson, B., Rohloff, K. (eds.) FC 2015. LNCS, vol. 8976, pp. 112–126. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48051-9_9
38. Verma, G.K., Singh, B.: Efficient message recovery proxy blind signature scheme from pairings. Trans. Emerg. Telecommun. Technol. **28**(11), e3167 (2017)
39. Verma, G.K., Singh, B., Singh, H.: Provably secure certificate-based proxy blind signature scheme from pairings. Inf. Sci. **468**, 1–13 (2018)
40. Xue, Q., Cao, Z.: A new proxy blind signature scheme with warrant. In: IEEE Conference on Cybernetics and Intelligent Systems, 2004. vol. 2, pp. 1386–1391. IEEE (2004)
41. Zheng, Z., Xie, S., Dai, H.N., Chen, X., Wang, H.: Blockchain challenges and opportunities: a survey. Int. J. Web Grid Serv. **14**(4), 352–375 (2018)
42. Zhu, H., Tan, Y.a., Zhang, X., Zhu, L., Zhang, C., Zheng, J.: A round-optimal lattice-based blind signature scheme for cloud services. Future Gener. Comput. Syst. **73**, 106–114 (2017)
43. Zhu, H., Tan, Y.a., Zhu, L., Zhang, Q., Li, Y.: An efficient identity-based proxy blind signature for semioffline services. Wireless Commun. Mobile Comput. **2018** (2018)
44. Zola, F., Eguimendia, M., Bruse, J.L., Urrutia, R.O.: Cascading machine learning to attack bitcoin anonymity. In: 2019 IEEE International Conference on Blockchain (Blockchain), pp. 10–17. IEEE (2019)