Anonymous proof-of-asset transactions using designated blind signatures

Neetu Sharma, Rajeev Anand-Sahu, Vishal Saraswat, Joaquin Garcia-alfaro

Anonymous proof-of-asset transactions using designated blind signatures

N. Sharma[1], R. Anand-Sahu[2], V. Saraswat[3], J. Garcia-Alfaro[4]

[1] Pt. Ravishankar Shukla University, India
[2] University of Luxembourg, Luxembourg.
[3] Robert Bosch Engineering & Business Solutions Pvt. Ltd., India.
[4] Institut Polytechnique de Paris, Télécom SudParis, France.

Abstract. We propose a scheme to preserve the anonymity of users in proof-of-asset transactions. We assume bitcoin-like cryptocurrency systems in which a user must prove the strength of its assets (i.e., solvency), prior conducting further transactions. The traditional way of addressing such a problem is the use of blind signatures, i.e., a kind of digital signature whose properties satisfy the anonymity of the signer. Our work focuses on the use of a designated verifier signature scheme that limits to only a single authorized party (within a group of signature requesters) to verify the correctness of the transaction.

Keywords: Blind signature schemes, Anonymity, Designated verification, Cryptocurrencies, Identity-based Cryptography, Bilinear pairings.

1 Introduction

Blind signature schemes offer a practical way of handling privacy constraints in cryptocurrencry transactions [3]. A blind signature construction is essentially an interactive two-party protocol between the signer of a message and a group of signature requesters. The signer disguises the contents of the message, before signing it. This way, the signature requesters can verify the correctness of the operation, without learning anything about the message that has been signed. However, the validity of the signature can be verified by anyone within the group of requesters. There may be situations in which a particular signer wants to designate that just one entity in the group of receivers must able to verify the signature (and not the others). This is the objective of DVS (Designated Verifier Signature) schemes [9,13] addressed in this paper.

In the realm of blockchain cryptocurrencies (i.e., bitcoin-like digital cash schemes), the aforementioned situation may appear in the so-called proof-of-asset transactions, in which users must prove their solvency prior getting access to online services such as cryptocurrency exchange markets. In other words, situations in which users must prove that they control a given amount of assets (i.e., bitcoins) but without releasing the specific amount they owe. In addition, we assume situations in which the users want to uniquely designate who can verify those proof-of-asset transactions, e.g., to avoid that a leakage of the proof is used by other parties (i.e., advertisement services, gambling platforms, etc.).

We address the aforementioned challenges and present a designated verifier blind signature (DVS) construction using pairing-based cryptography. The security of our scheme relies on the hardness of the computational and the decisional bilinear Diffie-Hellman problem (cf. Section 3.1). We analyze the security of the approach and perform an efficiency comparison w.r.t. other existing similar approaches.

Paper Organization — Section 2 surveys related work. Section 3 provides some preliminaries. Section 4 presents our construction and discusses about the security and efficiency of our approach. Section 5 concludes the paper.

2 Related Work

Following seminal work by Chaum [3], Boldyreva [1] demonstrated and formalized the concept of blind signature schemes under the random oracle model and the computational Diffie-Hellman assumptions. Work by Chow et al. [5] proved, as well, *unlinkability* properties of blind signatures. Camenisch et al. [2] proposed new constructions without random oracle constraints, without achieving proofs against strong unforgeability. Liao et al. [11] provide new schemes under the hardness of strong Diffie-Hellman assumptions. Zhang and Kim [18], followed by Huang et al. [8], proposed identity-based blind signatures, achieving unlinkability. Zhang et al. [19] uncovered linkability attacks in [8] (signers being traced back under valid message-signature pairs). Pointcheval and Stern [12] settled fundamental security properties of blind signatures. Schröder et al. [14] offer fair guidelines for the security of blind signatures. They revisit the definition of unforgeability in [12] and propose a new unforgeability definition to avoid adversaries repeating a message for more than one signature.

The public verifiability of a signature is undesirable when a signature shares sensitive information between the signer and the verifier. To deal with this situation, the signer requires to sign the document for a fixed receiver with control on its verification. For this purpose, the idea of undeniable signature [4] was suggested by Chaum and Van Antwerpen. Desmedt and Yung reported in [7] some weaknesses in the aforementioned approach. Jakobsson et al. [9] proposed a non-interactive designated verifier proof which enables the signer to produce transfer-resistant signatures for a designated verifier. In other words, the verifier does not possess the capability to transfer the proof of origin of the signature to third parties. Jakobsson et al. [9] also suggested the necessity of keeping the anonymity of signers. A concrete construction satisfying such constraints (i.e., impossibility of transfer to third parties and signer anonymity) was provided by Saeednia et al. in [13]. Identity-based versions inspired by the previous approach were presented by Susilo et al. in [15], and later by Zhang and Wen in [20]. Limitations in [20] include the lack of proofs for unverifiability, non-transferability and strongness, and the possibility of a signer with direct access to the original message to blind and unblind messages and signatures, hence not fulfilling the standard definition settled in [3,12,14]. The construction presented in this paper addresses such shortcomings.

Bitcoin-like transaction anonymity has been addressed by Yi et al. proposing schemes achieving blindness and unforgeability [17]. More recently, Wang et al. in [16] has proposed the application of designated verifier blind signatures for bitcoin proof-of-asset transactions. When a vendor requires to an anonymous buyer to provide a proof of solvency prior enabling an online service (e.g., a certain amount of bitcoins), the buyer provides a proof about it in designated manner. Hence, only the specific vendor requesting solvency to the user can process the signature. The vendor cannot further use this proof with any other third party. Our new construction addresses the same problem, offering a more compact construction over pairings, improving the efficiency of the identity-based construction by Zhang and Wen in [20], and satisfying unverifiability, non-transferability and strongness properties (cf. Section 3 and citations thereof).

3 Preliminaries

3.1 Identity-Based Cryptography

A probabilistic polynomial time (PPT) algorithm is a probabilistic random algorithm that runs in time polynomial in the length of input. $y \xleftarrow{\$} A(x)$ denotes a randomized

algorithm $A(x)$ with input x and output y. For X being a set $v \xleftarrow{\$} X$ stands for a random selection of v from X. A function $f : N \to [0,1]$ is said to be negligible in n if for any polynomial p and for sufficiently large n, the relation $f(n) < 1/p(n)$ holds. For an element $g \in G$, where G is a set, we denote the group $G = \langle g \rangle$ if g generates or spans G.

Definition 1 (Bilinear Map). Let G_1 and G_2 be two cyclic groups with a prime order q, where G_1 is additive and G_2 is multiplicative. Let P be the generator of G_1. Then a map $e : G_1 \times G_1 \to G_2$ is said to be a cryptographic bilinear map if it fulfils the below conditions.

Bilinearity: For all integers $x, y \in \mathbb{Z}_q^*$, $e(xA, yA) = e(A, A)^{xy}$, or equivalently, for all $A, B, C \in G_1$, $e(A + B, C) = e(A, C)e(B, C)$ and $e(A, B + C) = e(A, B)e(A, C)$.
Non-Degeneracy: The points $A, B \in G_1$ with $e(A, B) \neq 1$. As G_1 and G_2 are prime ordered groups this property is equivalent to have $g := e(A, A) \neq 1$, or in other words $g := e(A, A)$ is a generator of G_2.
Computability: The map $e(A, B) \in G_2$ can be computes efficiently for all $A, B \in G_1$.

Definition 2 (Bilinear Map Parameter Generator). A bilinear map parameter generator $\mathfrak{B}$ is a PPT algorithm that takes as input security parameter λ and outputs a tuple

$$\langle q, e : G_1 \times G_1 \to G_2, P, g \rangle \leftarrow \mathfrak{B}(\lambda) \tag{1}$$

where q, G_1, G_2, e, P and g are as in Definition 1.

Definition 3 (Bilinear Diffie-Hellman Problem). Given a security parameter λ, let

$$\langle q, e, P, g \rangle \leftarrow \mathfrak{B}(\lambda)\,.$$

Let $BDH : G_1 \times G_1 \times G_1 \to G_2$ be a map defined by $BDH(X, Y, Z) = \omega$ where

$$X = xP, Y = yP, Z = zP \text{ and } \omega = e(P, P)^{xyz}\,.$$

The bilinear Diffie-Hellman problem (BDHP) is to evaluate $BDH(X, Y, Z)$ given $X, Y, Z \xleftarrow{\$} G_1$. (Without the knowledge of $x, y, z \in \mathbb{Z}_q$ — obtaining $x \in \mathbb{Z}_q$, given $P, X \in \mathbb{G}_1$ is solving the discrete logarithm problem (DLP)).

Definition 4 (BDHP Parameter Generator). A BDHP parameter generator $\mathfrak{C}$ is a PPT algorithm that takes as input security parameter λ and outputs a tuple

$$\langle q, e : G_1 \times G_1 \to G_2, P, g, X, Y, Z \rangle \leftarrow \mathfrak{C}(\lambda) \tag{2}$$

where q, G_1, G_2, e, P, g, X, Y and Z are as in Definition 3.

Definition 5 (Bilinear Diffie-Hellman Assumption). Given a security parameter λ, let

$$\langle q, e, P, g, X, Y, Z \rangle \leftarrow \mathfrak{C}(\lambda)\,.$$

. The bilinear Diffie-Hellman assumption (BDHA) states that for any PPT algorithm $\mathcal{A}$ which attempts to solve BDHP, its advantage $\mathbf{Adv}_{\mathfrak{C}}(\lambda)$, defined as

$$\mathbf{Pr}[\mathcal{A}(q, e, P, g, X, Y, Z) = BDH(X, Y, Z)]\,,$$

is negligible in λ.

Definition 6 (Decisional BDHP). Given a security parameter λ, let

$$\langle q, e, P, g, X, Y, Z\rangle \leftarrow \mathfrak{C}(\lambda)\,.$$

Let $\omega \xleftarrow{\$} G_2$. The decisional bilinear Diffie-Hellman problem (DBDHP) is to decide if

$$\omega = BDH(X, Y, Z)\,.$$

That is, if $X = xP, Y = yP, Z = zP$, for some $x, y, z \in \mathbb{Z}_q$, then the DBDHP is to decide if

$$\omega = e(P, P)^{xyz}\,.$$

(Without the knowledge of $x, y, z \in \mathbb{Z}_q$ — obtaining $x \in \mathbb{Z}_q$, given $P, X \in \mathbb{G}_1$ is solving the discrete logarithm problem (DLP)).

Definition 7 (DBDHP Parameter Generator). A DBDHP parameter generator $\mathfrak{D}$ is a PPT algorithm that takes as input security parameter λ and outputs a tuple

$$\langle q, e : G_1 \times G_1 \to G_2, P, g, X, Y, Z, \omega\rangle \leftarrow \mathfrak{D}(\lambda) \tag{3}$$

where q, G_1, G_2, e, P, g, X, Y, Z and ω are as in Definition 6.

Definition 8 (Decisional BDHA). Given a security parameter λ, let

$$\langle q, e, P, g, X, Y, Z, \omega\rangle \leftarrow \mathfrak{D}(\lambda)\,.$$

The bilinear Diffie-Hellman assumption (DBDHA) states that, for any PPT algorithm $\mathcal{A}$ which attempts to solve DBDHP, its advantage $\mathbf{Adv}_{\mathfrak{D}}(\lambda)$, defined as

$$\begin{bmatrix}\mathbf{Pr}[adva(q, e, P, g, X, Y, Z, \omega) = 1] - \\ \mathbf{Pr}[\mathcal{A}(q, e, P, g, X, Y, Z, BDH(X, Y, Z)) = 1]\end{bmatrix}, \tag{4}$$

is negligible in λ.

3.2 Identity-based Strong Designated Verifier Blind Signatures

In this section, we provide formal definitions related with the construction of identity-based strong designated verifier blind signature (hereinafter, ID-SDVBS) schemes [20]. In such schemes, a signer with identity ID_S intends to send a signed message to a designated verifier with identity ID_V such that no one other than the designated verifier can verify the signature. The scheme consists of the five algorithms described next:

1. *params* $\leftarrow \mathsf{Setup}(\lambda)$: Executed by the Private Key Generator (PKG), taking a security parameter λ as input and producing, as output, the master secret s and the public parameters (*params*) of the system. The remaining algorithms listed below receive all the values of *params* as implicit inputs.
2. $(Q_{ID}, S_{ID}) \leftarrow \mathsf{Key\ Extract}(ID)$: The PKG takes as input an identity ID and produces, as output, a public and private key pair $(Q_{\mathsf{ID}}, S_{\mathsf{ID}})$.
3. $\sigma \leftarrow \mathsf{DVBSig}(S_{\mathsf{ID}_S}, Q_{\mathsf{ID}_V}, m)$: Signer and user run this interactive process. Inputs include the signer's public and secret key $(Q_{\mathsf{ID}_S}, S_{\mathsf{ID}_S})$, the designated verifier's public key Q_{ID_V} and a message m. Signer and user stop this process in polynomial time, producing either a signature σ of m, or false (in case an error happens).

4. $b \leftarrow \mathsf{DVBVer}(S_{\mathsf{ID}_V}, Q_{\mathsf{ID}_S}, m, \sigma)$: Run by the verifier, taking as inputs S_{ID_V} (secret key of the verifier), Q_{ID_S} (public key of the signer), a message m and a signature σ. It returns a bit b which is 1 if the signature is valid (otherwise, it returns 0 if the signature is invalid).
5. $\widehat{\sigma} \leftarrow \mathsf{DVBSim}(Q_{\mathsf{ID}_S}, S_{\mathsf{ID}_V}, m)$: Run by the verifier, it takes as inputs S_{ID_V} (verifier's secret key), Q_{ID_S} and Q_{ID_V} (public keys of the signer and the designated verifier), and a message m. It generates a signature $\widehat{\sigma}$ as output.

Next, we provide definitions about the properties we aim to satisfy.

Definition 9 (Correctness). If the signature σ on a message m is correctly computed by a signer ID_S, then the designated verifier ID_V must be able to verify the correctness of the message-signature pair (m, σ). That is,

$$\mathbf{Pr}\Big[1 \leftarrow \mathsf{DVBVer}\Big(\begin{matrix} S_{\mathsf{ID}_V}, Q_{\mathsf{ID}_S}, m, \\ \mathsf{DVBSig}(Q_{\mathsf{ID}_V}, S_{\mathsf{ID}_S}, m)\end{matrix}\Big)\Big] = 1\,.$$

Definition 10 (Unforgeability). An ID-SDVBS scheme is said to be strong existential unforgeable against adaptive chosen message and adaptive chosen identities attack if for any security parameter λ, no probabilistic polynomial time adversary $\mathcal{A}(\lambda, t, \varepsilon, q_{H_1}, q_{H_2}, q_E, q_S, q_V)$, which runs in time t, has a non-negligible advantage

$$\begin{aligned}\varepsilon &:= \mathbf{Adv}^{\text{SEUF-CID2-CMA2}}_{\text{ID-SDVBS},\mathcal{A}}(\lambda) \\ &:= \mathbf{Pr}[1 \leftarrow \mathsf{DVBVer}(S_{\mathsf{ID}_{V^*}}, Q_{\mathsf{ID}_{S^*}}, m^*, \sigma^*)]\end{aligned}$$

against the challenger $\mathcal{B}$ in the following game:

1. *Setup*: The challenger $\mathcal{B}$ generates the systems public parameter *params* for security parameter λ.
2. *Query Phase*: – The adversary $\mathcal{A}$ may request upto q_{H_1} hash queries on its adaptively chosen identities and upto and q_{H_2} hash queries on its adaptively chosen messages and obtain responses from $\mathcal{B}$ acting as a random oracle.
– $\mathcal{A}$ may request upto q_E key extraction queries on its adaptively chosen identities and obtain the corresponding private keys.
– $\mathcal{A}$ may request upto q_S signature queries on its adaptively chosen messages and adaptively chosen identities for the signer and the designated verifier and obtain a valid strong designated verifier signature.
– $\mathcal{A}$ may request upto q_V verification queries on signatures on its adaptively chosen messages m and adaptively chosen identities for the signer and the designated verifier and obtain the verification result 1 if it is valid and 0 if invalid.
3. *Output*: Finally, $\mathcal{A}$ outputs a (message, signature) pair (m^*, σ^*) for identities ID^*_S of the signer and ID^*_V of the designated verifier such that:
– $\mathcal{A}$ has never submitted ID^*_S or ID^*_V during the key extraction queries.
– σ^* was never given as a response to a signature query on the message m^* with the signer's identity ID^*_S, and the designated verifier's identity ID^*_V;
– σ^* is a valid signature on the message m^* from a signer with identity ID^*_S for a designated verifier with identity ID^*_V.

Definition 11 (Unverifiability). An ID-SDVBS scheme is said to be existential designated unverifiabile against adaptive chosen message and adaptive chosen identities attack if for any security parameter λ, no probabilistic polynomial time adversary $\mathcal{A}(\lambda,$

$t, \varepsilon, q_{H_1}, q_{H_2}, q_E, q_S, q_V)$ which runs in time t has a non-negligible advantage

$$\begin{aligned}\varepsilon &:= \mathbf{Adv}^{\text{EDV-CID2-CMA2}}_{\text{ID-SDVBS},\mathcal{A}}(\lambda) \\ &:= \big|\mathbf{Pr}[\mathcal{A}(Q_{\mathsf{ID}_{S^*}}, Q_{\mathsf{ID}_{V^*}}, m^*, \sigma^*) = 1] - \\ &\mathbf{Pr}[\mathcal{A}(Q_{\mathsf{ID}_{S^*}}, Q_{\mathsf{ID}_{V^*}}, m^*, \mathsf{DVBSig}(S_{\mathsf{ID}_{S^*}}, Q_{\mathsf{ID}_{V^*}}, m^*)) = 1]\big|\end{aligned}$$

against the challenger $\mathcal{B}$'s response σ^* in the following game:

1. *Setup*: Challenger $\mathcal{B}$ generates the system public parameters *params* from λ.
2. *Query Phase 1*: – Adversary $\mathcal{A}$ may request up to q_{H_1} hash queries on its adaptively chosen identities; and up to q_{H_2} hash queries on its adaptively chosen messages. $\mathcal{A}$ may obtain responses from $\mathcal{B}$, acting as a random oracle.
– $\mathcal{A}$ may request upto q_E key extraction queries on its adaptively chosen identities and obtain the corresponding private keys.
– $\mathcal{A}$ may request upto q_S signature queries on its adaptively chosen messages and adaptively chosen identities for the signer and the designated verifier and obtain a valid strong designated verifier signature.
– $\mathcal{A}$ may request upto q_V verification queries on signatures on its adaptively chosen messages m and adaptively chosen identities for the signer and the designated verifier and obtain the verification result 1 if it is valid and 0 if invalid.
3. *Challenge*: At some point, $\mathcal{A}$ outputs a message m^* and identitie ID^*_S of the signer and ID^*_V of the designated verifier on which it wishes to be challenged such that $\mathcal{A}$ has never submitted ID^*_S or ID^*_V during the key extraction queries. The challenger $\mathcal{B}$ responds with a "signature" σ^* and challenges $\mathcal{A}$ to verify if it is valid or not.
4. *Query Phase 2*: $\mathcal{A}$ continues its queries as in Query Phase 1 with an additional restriction that now it cannot submit a verification query on σ^*.
5. *Output*: $\mathcal{A}$ outputs a bit b^* which is 1 if the signature is valid and 0 if invalid.

Definition 12 (Non-transferability). An ID-SDVBS scheme is said to achieve non-transferability if the signature generated by the signer is computationally indistinguishable from that generated by the designated verifier, that is,

$$\sigma \leftarrow \mathsf{DVBSig}(Q_{\mathsf{ID}_V}, S_{\mathsf{ID}_S}, m) \;\approx\; \widehat{\sigma} \leftarrow \mathsf{DVBSim}(Q_{\mathsf{ID}_S}, S_{\mathsf{ID}_V}, m).$$

Definition 13 (Strongness). An ID-SDVBS scheme is said to be strong designated if given $\sigma \leftarrow \mathsf{DVBSig}(S_{\mathsf{ID}_S}, Q_{\mathsf{ID}_V}, m)$, anyone, say V^*, other than the designated verifier V can produce identically distributed transcripts that are indistinguishable from those of σ from someone, say S^*, except the signer S. That is,

$$\sigma \leftarrow \mathsf{DVBSig}(Q_{\mathsf{ID}_V}, S_{\mathsf{ID}_S}, m) \;\approx\; \widehat{\sigma} \leftarrow \mathsf{DVBSim}(Q_{\mathsf{ID}_{S^*}}, S_{\mathsf{ID}_{V^*}}, m).$$

Definition 14 (Blindness). An ID-SDVBS scheme must ensure the fact that the signer knows nothing about the message she signs. In other way, after producing signatures on different messages, the signer cannot relate that which message corresponds to which signature. More precisely, if signer has made a list of certain messages, and the requests for signatures have been placed by the user by picking messages randomly from that list, then after looking all the signatures together, the signer cannot make a list of corresponding (messages, signature) pairs. Our security model for the blindness is motivated by the models considered in [10,18]. Let $\mathcal{A}$ be a probabilistic polynomial time adversary/algorithm which has control over the malicious signer. Let $\mathcal{U}_1$ and $\mathcal{U}_2$ be two honest users who interact with the signer in the following attack game.

1. $\mathcal{A}$ is provided responses to it's *Key extraction queries* $(Q_{ID}, S_{ID}) \leftarrow \mathsf{Key\ Extract}(ID)$ as in the security game for unforgeability.

2. $\mathcal{A}$ outputs messages m_0, m_1.
3. $b \in \{0,1\}$ is defined to be a bit. User $\mathcal{U}_1$ and $\mathcal{U}_2$ randomly selects a bit b and pick m and m' as their random input taps. Here m is corresponding to bit b and so as m' corresponding to $(b-1)$.
4. $\mathcal{A}$ communicates with users $\mathcal{U}_1$ and $\mathcal{U}_2$ in the random order during the signature issuing protocol.
5. If the user $\mathcal{U}_1$ does not fail and posses a signature σ_m and also user $\mathcal{U}_2$ does not fail and posses signature $\sigma_{m'}$, then $\mathcal{A}$ is provided these additional information σ_m and $\sigma_{m'}$, which are outputs essentially based on the bit b and $(b-1)$, the actual value of the bit depends upon the value of b of-course. The game does not abort, and continue, even if either of the users fails but the other does not, and the corresponding output is forwarded to $\mathcal{A}$ in that case.
6. Finally, $\mathcal{A}$ outputs a bit $b' \in \{0,1\}$ which is 1 if $b' = b$.

We define advantage of the adversary $\mathcal{A}$ in the above game, by following

$$Adv_{\mathcal{A}}^{Blind} = (2 \times Pr[b' = b]) - 1$$

It is straightforward that $\mathcal{A}$ can always output a true bit with probability $\frac{1}{2}$. But, in this case the advantage is clearly 0. A signature is said to satisfying blindness, if there is no probabilistic polynomial-time algorithm/adversary $\mathcal{A}$ who wins the above game with non-negligible advantage.

4 Proposed Construction

Find below the algorithms of our construction (Setup, Key Extract, Designated Blind Signature DVBSig, Designated Verification DVBVer and Transcript Simulation DVBSim):

– **Setup** – In this algorithm, on input security parameter λ PKG outputs the master private key $s \in \mathbb{Z}_q^*$ and the public parameters

$$params = (1^\lambda, G_1, G_2, q, e, H_1, H_2, P, P_{pub}),$$

where G_1 is an additive cyclic group of prime order q with generator P, G_2 is a multiplicative cyclic group of prime order q, and $H_1 : \{0,1\}^* \to G_1$, $H_2 : \{0,1\}^* \times G_1 \to \mathbb{Z}_q^*$ are two cryptographic secure collision resistant hash functions, and $P_{pub} = sP \in G_1$ is system's public key, $e : G_1 \times G_1 \to G_2$ is a bilinear map (cf. Section 3.1).

– **Key Extract** – On input identity $ID_i \in \{0,1\}^*$, the PKG computes public key as $Q_{ID_i} = H_1(ID_i) \in G_1$ and private key as $S_{ID_i} = sQ_{ID_i} \in G_1$, for this identity.

– **DVBSig** – Signs message $m \in \{0,1\}^*$ (it can be verified by a designated verifier V).

1. Signer S selects a random $r \xleftarrow{\$} \mathbb{Z}_q^*$ and calculates:
   - $U = rQ_{\mathsf{ID}_S} \in G_1$;
2. As commitment, the signer sends the calculated value U to the user.
3. *Blinding Phase*: In this algorithm, user selects $x, y \in_R Z_q^*$ as blinding factors. Then calculates
   - $U' = xU + xyQ_{\mathsf{ID}_S}$;
   - $h = H_2(m, U') \in \mathbb{Z}_q^*$;
   - $h_1 = x^{-1}h + y$;
4. User sends this calculated value h_1 to the signer.
5. *Signing Phase*: After receiving h_1, the signer calculates
   - $V = (r + h_1)S_{\mathsf{ID}_S} \in G_1$; and sends back the user the value V.

6. *Unblinding Phase*: Then, the user calculates
   - $V' = xV$
   - $\sigma = e(V', Q_{\mathsf{ID}_\mathcal{V}})$. $(U', \sigma) \in G_1 \times G_2$ is the strong designated verifier blind signature on the message m.

– **DVBVer** – After receiving signature (U', σ) on a message m, a verifier computes $h = H_2(m, U') \in \mathbb{Z}_q^*$ and accepts the signature if $\sigma = e(U' + hQ_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}})$.

– **DVBSim** – The designated verifier V can produce the same signature $\widehat{\sigma}$ intended for itself, by performing this algorithm: chooses an integer $\widehat{r}, \widehat{x}, \widehat{y} \stackrel{\$}{\leftarrow} \mathbb{Z}_q^*$ and computes:
   - $\widehat{U} = \widehat{r}Q_{\mathsf{ID}_\mathcal{S}} \in G_1$;
   - $\widehat{U'} = \widehat{x}U + \widehat{x}\widehat{y}Q_{\mathsf{ID}_\mathcal{S}} \in G_1$;
   - $\widehat{h} = H_2(m, \widehat{U'}) \in \mathbb{Z}_q^*$;
   - $\widehat{h_1} = \widehat{x}^{-1}\widehat{h} + \widehat{y}$;
   - $\widehat{V} = (\widehat{r} + \widehat{h_1})Q_{\mathsf{ID}_\mathcal{S}} \in G_1$;
   - $\widehat{V'} = \widehat{x}\widehat{V}$; and
   - $\widehat{\sigma} = e(\widehat{V'}, S_{\mathsf{ID}_\mathcal{V}})$.

4.1 Security Analysis

The verification of our proposed scheme is as follows. If the generated signature (U', σ) on a message m from a signer with identity $\mathsf{ID}_\mathcal{S}$ for a designated verifier with identity $\mathsf{ID}_\mathcal{V}$, then the proposed scheme follows:

$$\begin{aligned}
e(U' + hQ_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) &= e(xU + xyQ_{\mathsf{ID}_\mathcal{S}} + hQ_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) \\
&= e(xU + xyQ_{\mathsf{ID}_\mathcal{S}} + hQ_{\mathsf{ID}_\mathcal{S}}, sQ_{\mathsf{ID}_\mathcal{V}}) \\
&= e(xrQ_{\mathsf{ID}_\mathcal{S}} + xyQ_{\mathsf{ID}_\mathcal{S}} + hQ_{\mathsf{ID}_\mathcal{S}}, sQ_{\mathsf{ID}_\mathcal{V}}) \\
&= e(x(rQ_{\mathsf{ID}_\mathcal{S}} + yQ_{\mathsf{ID}_\mathcal{S}} + x^{-1}hQ_{\mathsf{ID}_\mathcal{S}}), sQ_{\mathsf{ID}_\mathcal{V}}) \\
&= e(x(rS_{\mathsf{ID}_\mathcal{S}} + yS_{\mathsf{ID}_\mathcal{S}} + x^{-1}hS_{\mathsf{ID}_\mathcal{S}}), Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(x(r + (x^{-1}h + y))S_{\mathsf{ID}_\mathcal{S}}, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(x(r + h_1)S_{\mathsf{ID}_\mathcal{S}}, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(xV, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(V', Q_{\mathsf{ID}_\mathcal{V}}) \\
&= \sigma \ .
\end{aligned}$$

Next, we discuss the achievement of the following security properties: (1) unforgeability, (2) unverifiability, (3) non-transferability, (4) strongness and (5) blindness.

Theorem 1. *(Unforgeability) Given a security parameter λ, if there exists a PPT adversary $\mathcal{A}(\lambda, t, \varepsilon, q_{H_1}, q_{H_2}, q_E, q_S, q_V)$ which breaks the unforgeability of the proposed ID-SDVBS scheme in time t with success probability ε, then there exists a PPT adversary $\mathcal{B}(\lambda, t', \varepsilon')$ which solves BDHP with success probability at least*

$$\begin{aligned}
\varepsilon' \geq \Big(1 - \frac{1}{q^2}\Big)\Big(1 - \frac{2}{q_{H_1}}\Big)^{q_E + q_V} \\
\Big(1 - \frac{2}{q_{H_1}(q_{H_1} - 1)}\Big)^{q_S}\Big(\frac{2}{q_{H_1}(q_{H_1} - 1)}\Big)\varepsilon
\end{aligned}$$

in time at most

$$t' \leq (q_{H_1} + q_E + 3q_S + q_V)S_{G_1} + (q_S + q_V)P_e + q_S O_{G_1} + O_{G_2} + S_{G_2} + t$$

where S_{G_1} (resp. S_{G_2}) is the time taken for one scalar multiplication in G_1 (resp. G_2), O_{G_1} (resp. O_{G_2}) is the time taken for one group operation in G_1 (resp. G_2), and P_e is the time taken for one pairing computation.

Proof of Theorem 1: Let for a security parameter λ, $\mathcal{B}$ is challenged to solve the BDHP for

$$\langle q, e, G_1, G_2, P, aP, bP, cP \rangle$$

where G_1 is an additive cyclic group of prime order q with generator P, G_2 is a multiplicative cyclic group of prime order q with generator $e(P, P)$, and $e : G_1 \times G_1 \to G_2$ is a cryptographic bilinear map (cf. Section 3.1). $a, b, c \xleftarrow{\$} \mathbb{Z}_q^*$ are unknown to $\mathcal{B}$. The goal of $\mathcal{B}$ is to solve BDHP by computing $e(P, P)^{abc} \in G_2$ using $\mathcal{A}$, the adversary who claims to forge our proposed ID-SDVBS scheme. $\mathcal{B}$ simulates the security game for unfogeability with $\mathcal{A}$ as follows.

Setup: $\mathcal{B}$ generates the systems public parameter

$$params = \langle q, e : G_1 \times G_1 \to G_2, P, P_{pub} := cP, H_1, H_2 \rangle$$

for security parameter λ where the hash functions H_1 and H_2 behave as random oracles and responds to $\mathcal{A}$'s queries as below.
H_1-queries: To respond to the H_1 queries, $\mathcal{B}$ maintains a list

$$L_{H_1} = \{(\mathsf{ID}_i \in \{0,1\}^*, r_i \in \mathbb{Z}_q^*, R_i \in G_1)_{i=1}^{q_{H_1}}\}$$

which is initially empty. $\mathcal{B}$ randomly chooses two indices $\alpha, \beta \in [1, q_{H_1}]$ and sets $i = 0$. When $\mathcal{A}$ makes an H_1-query for an identity $\mathsf{ID} \in \{0,1\}^*$ $\mathcal{B}$ proceeds as follows.

1. If the query ID already appears in L_{H_1} in some tuple $(\mathsf{ID}_i, r_i, R_i)$ then $\mathcal{B}$ responds to $\mathcal{A}$ with $H_1(\mathsf{ID}) = R_i \in G_1$;
2. otherwise $\mathcal{B}$ sets $i = i + 1$ and
   - if $i = \alpha$, $\mathcal{B}$ sets $r_i = \perp$ and $R_i = aP$;
   - if $i = \beta$, $\mathcal{B}$ sets $r_i = \perp$ and $R_i = bP$;
   - if $i \neq \alpha, \beta$, $\mathcal{B}$ chooses $r_i \xleftarrow{\$} \mathbb{Z}_q^*$ and sets $R_i = r_i P$;
3. Finally, $\mathcal{B}$ adds the tuple $(\mathsf{ID}_i := \mathsf{ID}, r_i, R_i)$ to L_{H_1} and responds to $\mathcal{A}$ with $H_1(\mathsf{ID}) = R_i$.

H_2-queries: For response of H_2 queries, $\mathcal{B}$ maintains a list

$$L_{H_2} = \{((m, U') \in \{0,1\}^* \times G_1, h \in \mathbb{Z}_q^*)\}$$

initially which is empty. When $\mathcal{A}$ queries the oracle H_2 at (m, U'), $\mathcal{B}$ responds as follows.

1. If the query (m, U') already appears in the H_2-list in the tuple (m, U', h) then $\mathcal{B}$ respond with $H_2(m, U') = h \in \mathbb{Z}_q^*$.
2. Otherwise $\mathcal{B}$ picks a random $h \in \mathbb{Z}_q^*$ and adds the tuple (m, U', h) to the H_2-list and responds to $\mathcal{A}$ with $H_2(m, U') = h$.

Key extraction queries: When $\mathcal{A}$ makes a private key query on identity ID, $\mathcal{B}$ proceeds as follows.

1. Runs the above algorithm for responding to H_1-query for identity ID and obtains $H_1(\mathsf{ID}) = R_i$.

2. If $i = \alpha$ or β, $\mathcal{B}$ reports failure and halts.
3. If $i \neq \alpha, \beta$, $\mathcal{B}$ responds to $\mathcal{A}$ with the private key $S_{\mathsf{ID}} := r_i P_{pub}$ on the identity ID.

It can be verified that the provided private key $S_{\mathsf{ID}} = r_i P_{pub}$ is a valid private key for the user with identity $\mathsf{ID}_i := \mathsf{ID}$ since

$$r_i P_{pub} = r_i cP = cr_i P = cH_1(\mathsf{ID})\,.$$

Note that $\mathcal{B}$ aborts the security game during a key extraction query with probability $\frac{2}{q_{H_1}}$.

Signature queries: To respond to the signature queries, $\mathcal{B}$ maintains a list

$$L_S = \{(m_\ell \in \{0,1\}^*, \mathsf{ID}_{\mathcal{S}\ell} \in \{0,1\}^*, \mathsf{ID}_{\mathcal{V}\ell} \in \{0,1\}^*, \\ x_\ell \in \mathbb{Z}_q^*, U'_\ell \in G_1, \sigma_\ell \in G_2)_{\ell=1}^{q_S}\}$$

which is initially empty with $\ell = 0$. When $\mathcal{A}$ queries the signature on a message m from a signer with identity $\mathsf{ID}_{\mathcal{S}}$ for a designated verifier with identity $\mathsf{ID}_{\mathcal{V}}$, $\mathcal{B}$ proceeds as follows.

1. If the query $(m, \mathsf{ID}_{\mathcal{S}}, \mathsf{ID}_{\mathcal{V}})$ already appears in L_S in some tuple $(m_\ell, \mathsf{ID}_{\mathcal{S}\ell}, \mathsf{ID}_{\mathcal{V}\ell}, x_\ell, U'_\ell, \sigma_\ell)$ then $\mathcal{B}$ responds to $\mathcal{A}$ with the signature (U'_ℓ, σ_ℓ).
2. Otherwise $\mathcal{B}$ sets $\ell = \ell + 1$ and for responding to H_1-query for identities $\mathsf{ID}_{\mathcal{S}}$ and $\mathsf{ID}_{\mathcal{V}}$ runs the above algorithm and obtains $H_1(\mathsf{ID}_{\mathcal{S}}) = R_i$ and $H_1(\mathsf{ID}_{\mathcal{V}}) = R_j$.
3. If $\{i, j\} = \{\alpha, \beta\}$, $\mathcal{B}$ reports failure and halts.
4. If $i \neq \alpha, \beta$, then $\mathcal{B}$ computes the private key for $\mathsf{ID}_{\mathcal{S}}$, $S_{\mathsf{ID}_{\mathcal{S}}} = r_i P_{pub}$ and proceeds as follows.
   - randomly chooses $x_\ell \in \mathbb{Z}_q^*$;
   - sets $U'_\ell = x_\ell P \in G_1$;
   - runs the H_2-query algorithm to obtain $h_\ell = H_2(m, U'_\ell) \in \mathbb{Z}_q^*$;
   - sets $V'_\ell = x_\ell P_{pub} + h_\ell S_{\mathsf{ID}_{\mathcal{S}}} \in G_1$;
   - computes $\sigma_\ell = e(V'_\ell, Q_{\mathsf{ID}_j} = R_j)$.
5. Otherwise if $j \neq \alpha, \beta$, then $\mathcal{B}$ computes the private key $\mathsf{ID}_{\mathcal{V}}$, $S_{\mathsf{ID}_{\mathcal{V}}} = r_j P_{pub}$ and proceeds as follows.
   - randomly chooses $x_\ell \in \mathbb{Z}_q^*$;
   - sets $U'_\ell = x_\ell P \in G_1$;
   - runs the H_2-query algorithm to obtain $h_\ell = H_2(m, U'_\ell) \in \mathbb{Z}_q^*$;
   - sets $V'_\ell = x_\ell P_{pub} + h_\ell S_{\mathsf{ID}_{\mathcal{V}}} \in G_1$;
   - computes $\sigma_\ell = e(V'_\ell, Q_{\mathsf{ID}_i} = R_i)$.
6. Finally, $\mathcal{B}$ adds the tuple $(m_\ell, \mathsf{ID}_{\mathcal{S}\ell}, \mathsf{ID}_{\mathcal{V}\ell}, x_\ell, U'_\ell, \sigma_\ell)$ to L_S and responds to $\mathcal{A}$ with the signature (U'_ℓ, σ_ℓ).

Note that $\mathcal{B}$ aborts the security game during a signature query with probability $\frac{2}{q_{H_1}(q_{H_1}-1)}$.

Verification queries: When $\mathcal{A}$ makes a verification query on the signature (U', σ) on a message m from a signer with identity $\mathsf{ID}_{\mathcal{S}}$ for a designated verifier with identity $\mathsf{ID}_{\mathcal{V}}$, $\mathcal{B}$ proceeds as follows.

1. $\mathcal{B}$ runs the above algorithm for responding to H_1-query for identities $\mathsf{ID}_{\mathcal{S}}$ and $\mathsf{ID}_{\mathcal{V}}$ and obtains $H_1(\mathsf{ID}_{\mathcal{S}}) = R_i$ and $H_1(\mathsf{ID}_{\mathcal{V}}) = R_j$.
2. If $j \in \{\alpha, \beta\}$, $\mathcal{B}$ reports failure and halts.
3. If $j \neq \alpha, \beta$, then $\mathcal{B}$ computes $\mathsf{ID}_{\mathcal{V}}$'s private key, $S_{\mathsf{ID}_{\mathcal{V}}} = r_j P_{pub}$, and proceeds as in the verification of the proposed scheme and responds to $\mathcal{A}$ accordingly.

Note that $\mathcal{B}$ aborts the security game during a verification query with probability $\frac{2}{q_{H_1}}$.

Output: After $\mathcal{A}$ has made its queries, it finally outputs a valid signature (U'^*, σ^*) on a message m^* from a signer with identity $\mathsf{ID}^*_{\mathcal{S}}$ for a designated verifier with identity $\mathsf{ID}^*_{\mathcal{V}}$ with a non-negligible probability ε such that:

– $\mathcal{A}$ has never submitted $\mathsf{ID}_{\mathcal{S}}^*$ or $\mathsf{ID}_{\mathcal{V}}^*$ during the key extraction queries;
– $(U^{'*}, \sigma^*)$ was never given as a response to a signature query on the message m^* with the signer's identity $\mathsf{ID}_{\mathcal{S}}^*$, and the designated verifier's identity $\mathsf{ID}_{\mathcal{V}}^*$; and
– $\sigma^* = e(U^{'*} + h^* Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})$.

If $\mathcal{A}$ did not make H_1-query for the identities $\mathsf{ID}_{\mathcal{S}}^*$ and $\mathsf{ID}_{\mathcal{V}}^*$, then the probability that verification equality holds is less than $1/q^2$. Thus, with probability greater than $1 - 1/q^2$, both the public keys were computed using H_1-oracle and there exist indices $i, j \in [1, q_{H_1}]$ such that $\mathsf{ID}_{\mathcal{S}}^* = \mathsf{ID}_i$ and $\mathsf{ID}_{\mathcal{V}}^* = \mathsf{ID}_j$. If $\{i, j\} \neq \{\alpha, \beta\}$, then $\mathcal{B}$ reports failure and terminates.

Solution to BDHP: Otherwise, as in the forking lemma, $\mathcal{B}$ repeats the game with the same random tape for x_ℓ but with different choices of a random set for H_2-queries to obtain another forgery (U^*, σ') on the message m^* with h' such that $h^* \neq h'$ and $\sigma^* \neq \sigma'$. Then,

$$\begin{aligned}
\frac{\sigma^*}{\sigma'} &= \frac{e(U^{'*} + h^* Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})}{e(U^{'*} + h' Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})} \\
&= \frac{e(h^* Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})}{e(h' Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})} \\
&= \frac{e(Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})^{h^*}}{e(Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})^{h'}} \qquad (5) \\
&= e(Q_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}})^{(h^* - h')} \\
&= e(aP, bcP)^{(h^* - h')} \\
&= (e(P, P)^{abc})^{(h^* - h')} . \qquad (6)
\end{aligned}$$

Let $(h^* - h')^{-1} \mod q = \hat{h}$. Then, from the above equation, $\mathcal{B}$ solves the BDHP by computing

$$e(P, P)^{abc} = (\sigma^* / \sigma')^{\hat{h}} \qquad (7)$$

Probability calculation: If $\mathcal{B}$ does not abort during the simulation then $\mathcal{A}$'s view is identical to its view in the real attack. The responses to H_1-queries and H_2-queries are as in the real attack, since each response is uniformly and independently distributed in G_1 and $\mathbb{Z}_q^*$ respectively. The key extraction, signature and verification queries are answered as in the real attack.

The probability that $\mathcal{B}$ does not abort during the simulation is

$$\Big(1 - \frac{2}{q_{H_1}}\Big)^{q_E + q_V} \Big(1 - \frac{2}{q_{H_1}(q_{H_1} - 1)}\Big)^{q_S} . \qquad (8)$$

The probability that $\mathcal{A}$ did H_1-query for the identities $\mathsf{ID}_{\mathcal{S}}^*$ and $\mathsf{ID}_{\mathcal{V}}^*$ and that $\{\mathsf{ID}_{\mathcal{S}}^*, \mathsf{ID}_{\mathcal{V}}^*\} = \{\mathsf{ID}_\alpha, \mathsf{ID}_\beta\}$ is

$$\Big(1 - \frac{1}{q^2}\Big)\Big(\frac{2}{q_{H_1}(q_{H_1} - 1)}\Big) . \qquad (9)$$

Clearly $\mathcal{B}$'s advantage ε' for solving the BDHP, that is, the total probability that $\mathcal{B}$ succeeds to solve BDHP, is the product of $\mathcal{A}$'s advantage ε of forging the proposed

ID-SDVBS and the above two probabilities. Hence

$$\varepsilon' \geq \Big(1-\frac{1}{q^2}\Big)\Big(1-\frac{2}{q_{H_1}}\Big)^{q_E+q_V} \Big(1-\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)^{q_S}\Big(\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)\varepsilon\,.$$

Time calculation: It can be observed that running time of the algorithm $\mathcal{B}$ is same as that of $\mathcal{A}$ plus time taken to respond to the hash queries, key extraction queries, signature queries and verification queries, $q_{H_1}+q_{H_2}+q_E+q_S+q_V$. Hence the maximum running time required by $\mathcal{B}$ to solve the BDHP is

$$\begin{aligned} t' \leq & (q_{H_1}+q_E+3q_S+q_V)S_{G_1}+(q_S+q_V)P_e \\ & + q_S O_{G_1} + O_{G_2} + S_{G_2} + t \end{aligned}$$

as $\mathcal{B}$ requires to compute one scalar multiplication in G_1 to respond to H_1 hash query, one scalar multiplication in G_1 to respond to key extraction query, three scalar multiplications in G_1 to respond to signature query, one scalar multiplication in G_1 to respond to verification query; one pairing computation to respond to signature query, one pairing computation to respond to verification query, one group operation in G_1 to respond to signature query, and, one group operation in G_2 and one scalar multiplication in G_2 to output a solution of BDHP. □

Theorem 2. *(Unverifiability) Given a security parameter λ, if there exists a PPT adversary $\mathcal{A}(\lambda, t, \varepsilon, q_{H_1}, q_{H_2}, q_E, q_S, q_V)$ which breaks the designated unverifiability of the proposed ID-SDVBS scheme in time t with success probability ε, then there exists a PPT adversary $\mathcal{B}(\lambda, t', \varepsilon')$ which solves DBDHP with success probability at least*

$$\varepsilon' \geq \Big(1-\frac{1}{q^2}\Big)\Big(1-\frac{2}{q_{H_1}}\Big)^{q_E+q_V} \Big(1-\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)^{q_S}\Big(\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)\varepsilon$$

in time at most

$$\begin{aligned} t' \leq & (q_{H_1}+q_E+3q_S+q_V)S_{G_1}+(q_S+q_V)P_e \\ & + q_S O_{G_1} + S_{G_1} + S_{G_2} + P_e + t \end{aligned}$$

where S_{G_1} (resp. S_{G_2}) is the time taken for one scalar multiplication in G_1 (resp. G_2), O_{G_1} (resp. O_{G_2}) is the time taken for one group operation in G_1 (resp. G_2), and P_e is the time taken for one pairing computation.

Proof of Theorem 2: Let for a security parameter λ, $\mathcal{B}$ is challenged to solve the DBDHP for

$$\langle q, e : G_1 \times G_1 \to G_2, P, aP, bP, cP, \omega \rangle$$

where G_1 is an additive cyclic group of prime order q with generator P, G_2 is a multiplicative cyclic group of prime order q with generator $e(P,P)$, and $e : G_1 \times G_1 \to G_2$ is a cryptographic bilinear map as described in Section **??**and $\omega \xleftarrow{\$} G_2$. $a, b, c \xleftarrow{\$} \mathbb{Z}_q^*$ are unknown to $\mathcal{B}$. The goal of $\mathcal{B}$ is to solve DBDHP by verifying if $e(P,P)^{abc} = \omega$ using $\mathcal{A}$, the adversary who claims to forge our proposed ID-SDVBS scheme.

$\mathcal{B}$ simulates the security game for strongness with $\mathcal{A}$ by doing the Setup and by responding the H_1-queries, H_2-queries, Key extraction queries, Signature queries and Verification queries as in the security game for unforgeability.

Output: After $\mathcal{A}$ has made its queries, it finally outputs a message m^*, an identity $\mathsf{ID}_{\mathcal{S}}^*$ of a signer and an identity $\mathsf{ID}_{\mathcal{V}}^*$ of a designated verifier on which it wishes to be challenged.

If $\mathcal{A}$ did not make H_1-query for the identities $\mathsf{ID}_{\mathcal{S}}^*$ and $\mathsf{ID}_{\mathcal{V}}^*$, then the probability that verification equality holds is less than $1/q^2$. Thus, with probability greater than $1 - 1/q^2$, both the public keys were computed using H_1-oracle and there exist indices $i, j \in [1, q_{H_1}]$ such that $\mathsf{ID}_{\mathcal{S}}^* = \mathsf{ID}_i$ and $\mathsf{ID}_{\mathcal{V}}^* = \mathsf{ID}_j$. If $\{i, j\} \neq \{\alpha, \beta\}$, then $\mathcal{B}$ reports failure and terminates.

Solution to DBDHP: Otherwise, $\mathcal{B}$

– chooses a random $r \xleftarrow{\$} \mathbb{Z}_q^*$;
– sets $U' = rP$;
– sets $h = H_2(m^*, U')$;
– sets $\sigma = e(bP, cP)^r \omega^h$;

and challenges $\mathcal{A}$ to verify the validity of the signature (U, σ). Then, the verification holds if and only if each of the following holds

$$
\begin{aligned}
& & \sigma &= e(U' + hQ_{\mathsf{ID}_{\mathcal{S}}}, S_{\mathsf{ID}_{\mathcal{V}}}) \\
\iff & & \sigma &= e(rP + haP, bP_{pub}) \\
\iff & & \sigma &= e(rP, bP_{pub}) e(haP, bP_{pub}) \\
\iff & & \sigma &= e(P, bP_{pub})^r e(aP, bP_{pub})^h \\
\iff & & \sigma &= e(bP, P_{pub})^r e(aP, bP_{pub})^h \\
\iff & & \sigma &= e(bP, cP)^r e(aP, bcP)^h \\
\iff & & \sigma &= e(bP, cP)^r (e(P, P)^{abc})^h \\
\iff & e(bP, cP)^r \omega^h &&= e(bP, cP)^r (e(P, P)^{abc})^h \\
\iff & \omega^h &&= (e(P, P)^{abc})^h \\
\iff & \omega &&= e(P, P)^{abc}
\end{aligned}
$$

Then, from the above equation, $\mathcal{B}$ solves the DBDHP by simply returning the response of $\mathcal{A}$ to the strongness challenge.

Probability calculation: If $\mathcal{B}$ does not abort during the simulation then $\mathcal{A}$'s view is identical to its view in the real attack. The responses to H_1-queries and H_2-queries are as in the real attack, since each response is uniformly and independently distributed in G_1 and $\mathbb{Z}_q^*$ respectively. The key extraction, signature and verification queries are answered as in the real attack.

The probability that $\mathcal{B}$ does not abort during the simulation is

$$\Big(1 - \frac{2}{q_{H_1}}\Big)^{q_E + q_V} \Big(1 - \frac{2}{q_{H_1}(q_{H_1} - 1)}\Big)^{q_S}. \tag{10}$$

The probability that $\mathcal{A}$ did H_1-query for the identities $\mathsf{ID}_{\mathcal{S}}^*$ and $\mathsf{ID}_{\mathcal{V}}^*$ and that $\mathsf{ID}_{\mathcal{S}}^* = \mathsf{ID}_\alpha$ and $\mathsf{ID}_{\mathcal{V}}^* = \mathsf{ID}_\beta$ is

$$\Big(1 - \frac{1}{q^2}\Big)\Big(\frac{2}{q_{H_1}(q_{H_1} - 1)}\Big). \tag{11}$$

Clearly $\mathcal{B}$'s advantage ε' for solving the DBDHP, that is, the total probability that $\mathcal{B}$ succeeds to solve DBDHP, is the product of $\mathcal{A}$'s advantage ε of breaking the strongness of the proposed ID-SDVBS and the above two probabilities. Hence

$$\varepsilon' \geq \Big(1-\frac{1}{q^2}\Big)\Big(1-\frac{2}{q_{H_1}}\Big)^{q_E+q_V} \Big(1-\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)^{q_S}\Big(\frac{2}{q_{H_1}(q_{H_1}-1)}\Big)\varepsilon\,.$$

Time calculation: It can be observed that running time of the algorithm $\mathcal{B}$ is same as that of $\mathcal{A}$ plus time taken to respond to the hash queries, key extraction queries, signature queries and verification queries, that is, $q_{H_1}+q_{H_2}+q_E+q_S+q_V$. Hence the maximum running time required by $\mathcal{B}$ to solve the BDHP is

$$\begin{aligned} t' \leq & (q_{H_1}+q_E+3q_S+q_V)S_{G_1}+(q_S+q_V)P_e \\ & + q_S O_{G_1} + S_{G_1} + S_{G_2} + P_e + t \end{aligned}$$

as $\mathcal{B}$ requires to compute one scalar multiplication in G_1 to respond to H_1 hash query, one scalar multiplication in G_1 to respond to key extraction query, three scalar multiplications in G_1 to respond to signature query, one scalar multiplication in G_1 to respond to verification query; one pairing computation to respond to signature query, one pairing computation to respond to verification query, one group operation in G_1 to respond to signature query, and, one scalar multiplication in G_1, one scalar multiplication in G_2 and one pairing computation to output a solution of DBDHP. □

Theorem 3. *(Non-transferability) Let σ be the signature generated by the signer. Then, σ is computationally indistinguishable, i.e.,*

$$\sigma \leftarrow \mathit{DVBSig}(Q_{\mathsf{ID}_V}, S_{\mathsf{ID}_S}, m) \;\approx\; \widehat{\sigma} \leftarrow \mathit{DVBSim}(Q_{\mathsf{ID}_S}, S_{\mathsf{ID}_V}, m).$$

Proof of Theorem 3: As defined in Section 3.2, the property *non-transferability* holds when the signatures generated by the signer is indistinguishable from the one generated by the designated verifier. To show this property for our proposed scheme, below we show that: two signatures (U', σ) and $(\widehat{U'}, \widehat{\sigma})$, generated on message m by the signer and the designated verifier respectively, are indistinguishable. It can be observed that the indistinguishability holds immediately as the two distributions:

$$\begin{array}{lll} U = rQ_{\mathsf{ID}_S}; & & \widehat{U} = \widehat{r}Q_{\mathsf{ID}_S}; \\ U' = xU + xyQ_{\mathsf{ID}_S}; & & \widehat{U'} = \widehat{x}U + xyQ_{\mathsf{ID}_S}; \\ h = H_2(m, U'); & & \widehat{h} = H_2(m, \widehat{U'}); \\ h_1 = x^{-1}h + y; & \text{and} & \widehat{h_1} = \widehat{x}^{-1}h + \widehat{y}; \\ V = (r+h_1)S_{ID_S}; & & \widehat{V} = (\widehat{r}+\widehat{h_1})Q_{\mathsf{ID}_S}; \\ V' = xV; & & \widehat{V'} = \widehat{x}V; \\ \sigma = e(V', Q_{ID_V}); & & \widehat{\sigma} = e(\widehat{V'}, S_{\mathsf{ID}_V}); \end{array}$$

are identical. □

Theorem 4. *(Strongness) Let σ be the signature generated by a signer S. Let V be the designated verifier, such that $\sigma \leftarrow \mathit{DVBSig}(S_{\mathsf{ID}_S}, Q_{\mathsf{ID}_V}, m)$. Then, only V can produce identically distributed transcripts that are indistinguishable from those of σ from someone, say S^*, except the signer S. That is,*

$$\sigma \leftarrow \mathit{DVBSig}(Q_{\mathsf{ID}_V}, S_{\mathsf{ID}_S}, m) \;\approx\; \widehat{\sigma} \leftarrow \mathit{DVBSim}(Q_{\mathsf{ID}_{S^*}}, S_{\mathsf{ID}_{V^*}}, m).$$

Proof of Theorem 4: By using the description below, it can be evidenced that our ID-SDBVS scheme achieves *strongness* as defined in Section 3.2. For this purpose we essentially show that for a signature $\sigma \leftarrow \mathsf{DVBSig}(Q_{\mathsf{ID}_\mathcal{V}}, S_{\mathsf{ID}_\mathcal{S}}, m)$ generated by the signer $\mathcal{S}$ for the designated verifier $\mathcal{V}$, the designated verifier $\mathcal{V}^*$ (other than $\mathcal{V}$) can generate a signature $\sigma \leftarrow \mathsf{DVBSim}(Q_{\mathsf{ID}^*_\mathcal{S}}, S_{\mathsf{ID}^*_\mathcal{V}}, m)$, as a signature generated by the signer $\mathcal{S}^*$ (other than $\mathcal{S}$) using the transcript simulation, where $Q_{\mathsf{ID}^*_\mathcal{S}}$ and $S_{\mathsf{ID}^*_\mathcal{V}}$ are defined as in the following, since

$$\begin{aligned}
\sigma &= e(xrS_{\mathsf{ID}_\mathcal{S}} + xh_1 S_{\mathsf{ID}_\mathcal{S}}, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(xrS_{\mathsf{ID}_\mathcal{S}} + xh_1 S_{\mathsf{ID}_\mathcal{S}}, mQ_{\mathsf{ID}^*_\mathcal{V}}) && \text{where } Q_{\mathsf{ID}_\mathcal{V}} = mQ_{\mathsf{ID}^*_\mathcal{V}} \\
&= e(xrmS_{\mathsf{ID}_\mathcal{S}} + xh_1 mS_{\mathsf{ID}_\mathcal{S}}, Q_{\mathsf{ID}^*_\mathcal{V}}) \\
&= e(xrS_{\mathsf{ID}^*_\mathcal{S}} + xh_1 S_{\mathsf{ID}^*_\mathcal{S}}, Q_{\mathsf{ID}^*_\mathcal{V}}) && \text{where } S_{\mathsf{ID}^*_\mathcal{S}} = mS_{\mathsf{ID}_\mathcal{S}}. \\
&= e(xrQ_{\mathsf{ID}^*_\mathcal{S}} + xh_1 Q_{\mathsf{ID}^*_\mathcal{S}}, S_{\mathsf{ID}^*_\mathcal{V}})
\end{aligned}$$

□

Theorem 5. *(Blindness) Let $x, y \in_R Z_q^*$ denote a random selection of blinding factors. Let $\mathcal{B}$ simulate the security game for blindness with $\mathcal{A}$. Let $\mathcal{A}$ be the probabilistic polynomial-time algorithm and has (Q_{ID}, S_{ID}) from the key extraction queries as in the security game for unforgeability. Then, our proposed signature scheme satisfies the blindness property established in Definition 14.*

Proof of Theorem 5: We follow the technique from [18] to prove Theorem 5. If two signatures (U_i', σ_i') and (U_j', σ_j') generated by users $\mathcal{U}_i(U_i, h_{1_i}, V_i)$ and $\mathcal{V}_i(U_j, h_{1_j}, V_j)$for $(i, j \in \{0, 1\})$ are provided to adversary $\mathcal{A}$ (who is in control over the signer, but not over the users), the adversary cannot draw the true bit b in a correct order, corresponding to the received signatures, where (U_i, h_{1_i}, V_i) and (U_j, h_{1_j}, V_j) are essentially the values exchanged between the users and the signer during the interactive signature protocol.

It is sufficient to show that there exist two random factors (X', y') that maps (U_i, h_{1_i}, V_i) to (U_j', σ_j') for each $i, j \in \{0, 1\}$ (where $\sigma_j' = e(V_j', Q_{\mathsf{ID}_\mathcal{V}})$ and $X' \in G_1$). We define $X' = V_i - V_j'$ (where $V_j' = r' S_{\mathsf{ID}_\mathcal{S}} + h_{1_j}' S_{\mathsf{ID}_\mathcal{S}}; r' \in_R Z_q^*$) and $y' = -h_{1_i} - (-h_{1_j}')$, since:

$$\begin{aligned}
\sigma &= e(V_i, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X' + V_j', Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X' + r' S_{\mathsf{ID}_\mathcal{S}} + h_{1_j}' S_{\mathsf{ID}_\mathcal{S}}, Q_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X', Q_{\mathsf{ID}_\mathcal{V}}) e(r' Q_{\mathsf{ID}_\mathcal{S}} + h_{1_j}' Q_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X', Q_{\mathsf{ID}_\mathcal{V}}) e(U_j' + h_{1_j}' Q_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X', Q_{\mathsf{ID}_\mathcal{V}}) e(U_j', S_{\mathsf{ID}_\mathcal{V}}) e((y' + h_{1_i}) Q_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) \\
&= e(X', Q_{\mathsf{ID}_\mathcal{V}}) e(U_j' + (h_{1_i} + y') Q_{\mathsf{ID}_\mathcal{S}}, S_{\mathsf{ID}_\mathcal{V}}) .
\end{aligned}$$

Hence, we can claim that there will always exist random values, i.e., the blinding factors, which hold the same relation as in the signature issuing protocol. □

4.2 Performance Estimation

Inspired by the performance analysis discussed by Debiao et al. in [6], we discuss next the expected computation time for the generation and verification of signatures using our approach.

We assume the same pairing used by Debiao et al., i.e., a Tate pairing, which is capable of achieving an equivalent of 1024-bit RSA security. It is defined over the supersingular elliptic curve $E = F_p : y^2 = x^3 + x$ with embedding degree 2 was used, where q is a 160-bit Solinas prime $q = 2^{159} + 2^{17} + 1$ and p a 512-bit prime satisfying $p + 1 = 12qr$. Accordingly, operation times are assumed as follows: 6.38 ms for each scalar multiplication; 5.31 ms for each exponentiation in G_2; 3.04 ms for each map-to-point hash execution; and 20.04 ms for each pairing computation. Other operations, such as the cost of an inverse operation over Z_q^*, are omitted in our analysis, since it takes less than 0.03 ms. Likewise, the operation time of performing one general hash function is also omitted, since it is expected to take less than 0.001 ms, hence negligible compared to the time taken by aforementioned (most costly) operations (cf. [6] and citations thereof for further details).

A careful analysis of our approach shows that each signature generation would require five scalar multiplications (i.e., 6.38 ms each), one map-to-point hash execution (i.e., 3.04 ms), and one pairing computation (i.e., 20.04 ms). In other words, our approach would require about 54.98 ms per signature generation. In terms of signature verification, our approach would require one scalar multiplication, one map-to-point hash execution and one pairing computation. Hence, leading to about 29.46 ms per signature verification. If we conduct now the same analysis to the closest approach in the literature, i.e., the identity-based construction by Zhang and Wen in [20], we would obtain about 67.74 ms per signature generation (i.e., five scalar multiplications, one map-to-point hash execution and one pairing computation) and 89.58 ms per signature verification (i.e., one scalar multiplications, one map-to-point hash execution and four pairing computations). Hence, and by using the performance analysis in [6], our construction offers higher efficiency while addressing the limitations in [20] (i.e., lack of *Blinding* and *Unblinding* procedures in their signature protocol, as well as lack of unverifiability, non-transferability and strongness properties).

5 Conclusion

We have presented a designated verifier signature scheme to enable anonymity in proof-of-asset transactions. It allows cryptocurrency users to prove their solvency in a privacy-friendly manner, while designating a single authorized party (from a group of signature requesters) to be able to verify the correctness of the transaction. The approach uses pairing-based cryptography. More precisely, an adaptive approach using an identity-based setting. The security of our construction has been proved using the hardness assumption of the decisional and computational bilinear Diffie-Hellman problem. We have also presented an early estimation of the computation cost of our approach, in terms of signature generation and signature verification. The estimation shows that the computational cost and operation time of the new scheme is significantly more efficient that previous efforts in the literature, while addressing the previous limitations.

References

1. A. Boldyreva. Efficient threshold signature, multisignature and blind signature schemes based on the gap-diffie-hellman-group signature scheme. *IACR ePrints*, 2002:118, 2002.
2. J. Camenisch, M. Koprowski, and B. Warinschi. Efficient blind signatures without random oracles. In *ICSCN*, pages 134–148. Springer, 2004.
3. D. Chaum. Blind signatures for untraceable payments. In *Advances in cryptology*, pages 199–203. Springer, 1983.

4. D. Chaum and H. Van Antwerpen. Undeniable signatures. In *Conference on the Theory and Application of Cryptology*, pages 212–216. Springer, 1989.
5. S. S. Chow, L. C. Hui, S.-M. Yiu, and K. Chow. Two improved partially blind signature schemes from bilinear pairings. In *ACISP*, pages 316–328. Springer, 2005.
6. H. Debiao, C. Jianhua, and H. Jin. An id-based proxy signature schemes without bilinear pairings. *Annals of Telecommunications*, 66(11-12):657–662, 2011.
7. Y. Desmedt and M. Yung. Weaknesses of undeniable signature schemes. In *Theory and Application of of Cryptographic Techniques*, pages 205–220. Springer, 1991.
8. Z. Huang, K. Chen, and Y. Wang. Efficient identity-based signatures and blind signatures. In *ICCNS*, pages 120–133. Springer, 2005.
9. M. Jakobsson, K. Sako, and R. Impagliazzo. Designated verifier proofs and their applications. In *International Conference on the Theory and Applications of Cryptographic Techniques*, pages 143–154. Springer, 1996.
10. A. Juels, M. Luby, and R. Ostrovsky. Security of blind digital signatures. *Advances in Cryptology?CRYPTO'97*, pages 150–164, 1997.
11. J. Liao, Y. Qi, P. Huang, and M. Rong. Pairing-based provable blind signature scheme without random oracles. In *ICCIS*, pages 161–166. Springer, 2005.
12. D. Pointcheval and J. Stern. Security arguments for digital signatures and blind signatures. *Journal of cryptology*, 13(3):361–396, 2000.
13. S. Saeednia, S. Kremer, and O. Markowitch. An efficient strong designated verifier signature scheme. In *ICISC*, pages 40–54. Springer, 2003.
14. D. Schröder and D. Unruh. Security of blind signatures revisited. In *International Workshop on Public Key Cryptography*, pages 662–679. Springer, 2012.
15. W. Susilo, F. Zhang, and Y. Mu. Identity-based strong designated verifier signature schemes. In *Australasian Conference on Information Security and Privacy*, pages 313–324. Springer, 2004.
16. H. Wang, D. He, and Y. Ji. Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography. *Future Generation Computer Systems*, 107:854–862, 2020.
17. X. Yi and K.-Y. Lam. A new blind ecdsa scheme for bitcoin transaction anonymity. In *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, pages 613–620, 2019.
18. F. Zhang and K. Kim. Id-based blind signature and ring signature from pairings. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 533–547. Springer, 2002.
19. J. Zhang, T. Wei, J. Zhang, and W. Zou. Linkability of a blind signature scheme and its improved scheme. In *International Conference on Computational Science and Its Applications*, pages 262–270. Springer, 2006.
20. N. Zhang and Q. Wen. Provably Secure Blind ID-Based Strong Designated Verifier Signature Scheme. In *CHINACOM'07*, pages 323–327. IEEE, 2007.