Arab J Sci Eng (2015) 40:1069–1080
DOI 10.1007/s13369-015-1568-2

RESEARCH ARTICLE - COMPUTER ENGINEERING AND COMPUTER SCIENCE

Provably Secure and Pairing-Based Strong Designated Verifier Signature Scheme with Message Recovery

SK Hafizul Islam · G. P. Biswas

Received: 1 April 2014 / Accepted: 19 January 2015 / Published online: 8 February 2015

Abstract In this paper, an efficient and secure strong designated verifier signature with message recovery scheme is presented using elliptic curve and bilinear pairing. In our scheme, the signer implants a message on the signature and sends it without message to the verifier, who then extracts the original message and validates the message-signature pair. However, an outsider is unable to verify the message-signature pair since the verifier's private key is strictly required for verification. Our scheme has been designed to achieve confidentiality, integrity, authentication and non-repudiation of message transmitted through hostile networks. Our scheme is secure against adaptive chosen message attack in the random oracle model under the intractability assumption of Co-Bilinear Diffie–Hellman problem. Besides, our scheme is computation and communication efficient than other schemes, and hence, it may be useful in many small message applications and also for the resource-constrained environments.

Keywords Elliptic curve cryptography · Designated verifier · Message recovery · Co-Bilinear Diffie–Hellman assumption · Provable security

S. H. Islam (✉)
Department of Computer Science and Information Systems, Birla Institute of Technology and Science, Pilani 333031, Rajasthan, India
e-mail: hafi786@gmail.com; hafizul.ism@gmail.com; hafizul@pilani.bits-pilani.ac.in

G. P. Biswas
Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826004, Jharkhand, India
e-mail: gpbiswas@gmail.com

1 Introduction

In 1996, Jakobsson et al. [1] firstly formulated the strong designated verifier signature (SDVS) scheme. In SDVS scheme, a signer recognized with the identity ID_A calculates a signature for the verifier recognized with the identity ID_B, who can only verify the authenticity, but he cannot prove to an outsider recognized with the identity ID_C that the signer ID_A was the actual signer since the verifier ID_B has the capability to produce another valid SDVS intended for him, which is indistinguishable from the signature computed by the signer ID_A. The fact is that the outsider ID_C cannot verify the signature since the private key of the verifier ID_B is strictly involved in the message-signature verification process. With the elliptic curve cryptography (ECC) [2–4] and bilinear pairing [5,6], several identity-based SDVS schemes [1,7–17] are studied widely.

1.1 Related Works and their Problems

In 2007, Lee and Chang [18] implemented a SDVS scheme with message recovery facility, called SDVSMR scheme. However, they have not defined any formal security model and formal security analysis. In 2004, Saeednia et al. [19] proposed a SDVS scheme without any formal security analysis. Unfortunately, Lee and Chang [20] analyzed that Saeednia et al.'s scheme [19] has some security problem, i.e., the signature can also be verified by the signer. It means that if the private key of the signer revealed to an adversary, then he can verify the signature using signer's private key. Then, they devised an enhanced SDVS scheme without any formal security analysis. Inspired from the Lee and Chang's scheme [18], in 2010, Yang and Liao [21] proposed a new strong designated verifier signature with message recovery (SDVSMR) scheme without any formal security analysis. In 2013, Shim

[22] designed an SDVS scheme in the standard model based on the bilinear pairing and the security assumption given by Lysyanskaya et al. [23]. However, it has no message recovery facility. Unfortunately, Kang et al. [24] showed that both the schemes [18] and [20] are vulnerable to the delegatability attack.

In 2004, Susilo et al. [25] presented an identity-based SDVS (ID-SDVS) scheme with identity-based cryptosystem (IBC) [26] and elliptic curve bilinear pairing. Zhang and Mao [7] designed a new pairing-based ID-SDVS scheme, but Kang et al. [8] analyzed that the scheme in [7] is insecure against the *strongness* property of SDVS scheme since an outsider can eavesdrop an old signature and obtain some information that is to be used for the verification of subsequent signatures. Kang et al. [8] devised an improved scheme without formal security analysis. Lee et al. [9] demonstrated that Kang et al.'s scheme [8] is *universally forgeable* and Kumar et al.'s scheme [10] violates the *strongness* property. In 2009, Kang et al. [11] proposed another ID-SDVS scheme with low costs from computation and communication aspect and analyzed its formal security. However, Du and Wen [12] proved that the Kang et al.'s scheme [11] is *universally forgeable* and violates the *strongness* property. In 2009, Yang et al. [13] presented an efficient and provably secure ID-SDVS scheme based on bilinear computational Diffie–Hellman (BCDH) assumption. Sun et al. [14] constructed a provably secure ID-SDVS scheme using bilinear pairing. In 2011, Huang et al. [15] presented a security model for ID-SDVS scheme that is shown to be stronger than previous models and subsequently proposed a new provably secure ID-SDVS scheme in their security model.

Based on the security of the discrete logarithm problem (DLP), in 1994, Nyberg and Rueppel [16] proposed the idea of digital signature with message recovery (DSMR) scheme. However, only few DSMR schemes have been constructed in the literature. Tseng and Hwang [17] proposed a DSMR scheme and its variant based on elliptic curve discrete logarithm problem (ECDLP). In 2004, Shao [27] showed that the schemes proposed in [17] are vulnerable to *insider forgery attack*, and does not satisfy the *forward security* and *non-repudiation* properties, and subsequently proposed an improved scheme to overcome these weaknesses. In 2005, Zhang et al. [28] presented the first ID-based digital signature with partial message recovery (ID-DSPMR) scheme in the random oracle model. However, Tso et al. [29] pointed out that, in some undesirable situation, a correctly generated signature may be misjudged and rejected, and in such cases, the message cannot be recovered correctly. To cope this weakness, Tso et al. [29] proposed an ID-DSPMR scheme with reduced computational cost and the length of the signature as well than others. In 2007, Li and Chen [30] also proposed an efficient ID-DSPMR based on bilinear pairing and analyzed its formal security under the q-Strong Diffie–Hellman (q-SDH) assumption. Kalkan et al. [31] proposed the generalized concept of ID-based ElGamal signature with partial message recovery scheme.

1.2 Motivations and Contributions

As discussed earlier, the DSMR schemes give opportunity to recover the original digital message from the signature, and hence, the message does not need to be transmitted separately. However, an outsider may recover the message and verify the exactness of message-signature pair without verifier's secret key. Therefore, the message confidentiality is violated in DSMR scheme. In order to manage this problem, we combine the ideas of SDVS and DSMR schemes and then designed an efficient and provably secure strong designated verifier signature with message recovery (SDVSMR) scheme with elliptic curve and bilinear pairing. In the proposed scheme, only the designated verifier recovers the message and validates the message-signature pair. However, any outsider has no such ability, because the verifier's private key is strictly required in the message-signature validation process. In our scheme, the signer is allowed to send the signature without message, and thus, it can save both the communication bandwidth and computation cost. In the random oracle model, our scheme is provably secure against the adaptive chosen message attack with the intractability of Co-Bilinear Diffie–Hellman (Co-BDH) problem. The computation and communication cost analysis showed that our scheme is more efficient than others. Our scheme is appropriate in the area of small message applications and the environments where the computing ability and communication bandwidth are limited.

1.3 Roadmap of the Paper

We structured the paper in the following ways. In Sect. 2, we presented some mathematical preliminaries. The attack model of SDVSMR scheme in the random oracle model is discussed in Sect. 3, and various security properties of SDVSMR scheme are studied in Sect. 4. Section 5 describes our scheme. The provable security analysis of the proposed scheme is discussed in Sect. 6, and Sect. 7 deals with the comparative results of our scheme with existing schemes. In Sect. 8, we made some concluding remarks.

2 Mathematical Preliminaries

The descriptions of some preliminaries needed in our signature scheme are given here.

2.1 Elliptic Curve Cryptography

Recently, the elliptic curve cryptography (ECC) [2,3] has accepted as an efficient tool in public key cryptography (PKC) due to the computation, communication and security strengths. For example, it offers same level of security at reduced key sizes than other PKCs. Below is the brief explanation of ECC.

Let F_q be a prime field with order $q = p^n$, where p is a large prime number and the group $E(F_q)$ consisting of points from a supersingular elliptic curve, which is given below, over F_q.

$$y^2 \bmod q = (x^3 + ax + b) \bmod q \quad (1)$$

where $x, y, a, b \in F_q$ and $(4a^3 + 27b^2) \bmod q \neq 0$. Assume that the point $P(x, y)$ on the Eq. (1), the point $Q(x, -y)$ is called the negative of P, i.e., $Q = -P$. Let $P(x_1, y_1)$ and $Q(x_2, y_2)(P \neq Q)$ be two points on (1); if $P = Q$, then the line (i.e., tangent at P) joining the points P and Q intersects the curve (1) at $-R(x_3, -y_3)$ and the reflection of it with respect to x-axis is the point $R(x_3, y_3)$, i.e., $P + Q = R$. The set $E(F_q)$ including the point O, called "point at infinity" or "zero point," makes an additive elliptic curve cyclic group G_q, i.e., $G_q = \{(x, y) : x, y \in F_q \text{ and } (x, y) \in E(F_q)\} \cup \{O\}$ of prime order p. The scalar point multiplication on G_q is defined as $kP = P + P + \cdots + P$ (k times). A generator point $P \in G_q$ has order n if $nP = O$, where n is the smallest positive integer.

The order of the elliptic curve $E(F_q)$ defined over F_q denoted as $\mathcal{O}(E(F_q))$ that satisfies the following relation $q + 1 - 2\sqrt{q} \leq \mathcal{O}(E(F_q)) \leq q + 1 + 2\sqrt{q}$, where the interval $[q+1-2\sqrt{q}, q+1+2\sqrt{q}]$ is called the *Hasse* interval [4]. For the group $E(F_q)$ defined over F_q, $\mathcal{O}(E(F_q)) = q + 1 \cdot t$, where $|t| \leq \sqrt{q}$ and t is called *trace* of the group $E(F_q)$ over F_q. Since $2\sqrt{q}$ is small relative to q, we have $\mathcal{O}(E(F_q)) \approx q$. In the next subsection, we discussed the types of elliptic curve used for bilinear pairing [4].

2.2 Bilinear Pairing

Let $(G_q, +)$ be a cyclic group of elliptic curve points computed with the generator P and $(G_m, \cdot)$ be another group with order the same prime order p, where $p \geq 2^k$ and k is security parameter. The mapping $\hat{e} : G_q \times G_q \rightarrow G_m$ is called an admissible bilinear paring if it satisfies the properties described below [5,6]:

- **Bilinearity:** For all $P, Q, R \in G_q$, we have $\hat{e}(P + Q, R) = \hat{e}(P, R)\hat{e}(Q, R)$ and $\hat{e}(P, Q + R) = \hat{e}(P, Q)$ $\hat{e}(P, R)$. Therefore, for $a, b \in Z_q^*$, $\hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$ holds.
- **Non-degenerate:** For all $P, Q \in G_q$ such that $\hat{e}(P, Q) \neq 1_m$, where 1_m is the identity element of the group G_m.
- **Computability:** There must be a polynomial time-bounded algorithm that can easily execute $\hat{e}(P, Q)$ for all $P, Q \in G_q$.

For the efficient implementation of pairing-based protocol, Weil pairing and Ate pairing on elliptic curves over prime fields have been considered. In pairing-based protocols, the elliptic curve group $E(F_q)$ is constructed from the supersingular elliptic curve $y^2 \bmod q = (x^3 + ax + b) \bmod q$, where F_q be a prime field with order $q = p^n$ and p is a large prime number [32]. The group $E(F_q)$ is a the multiplicative group of the extension field F_{q^k}, where k is called the *embedding degree* of the elliptic curve given above. The pairing is said to be secure if the computational problems are computationally hard both in the groups $E(F_q)$ and $F_{q^k}^*$.

In order to obtain computation and security efficiencies, q and k should be chosen so that the computational problems are hard by any polynomial time algorithm, and the group order denoted by $\mathcal{O}(E(F_q))$ must have a large prime factor r. Suppose that for a large prime number r such that it divides $\mathcal{O}(E(F_q))$, then for the smallest integer k (embedding degree) such that r divides $(q^k - 1)$. It is proven that the pairing is secure when $r \approx 2^{160}$ and $k \approx 6 - 10$. In order to achieve the enhanced security of pairing-based protocols, Barreto and Naehrig [33] proposed an efficient and powerful method that can easily calculate pairing-friendly elliptic curves over a field Z_q of prime order q, and with the embedding degree $k = 12$ [33]. The equation of the curve is $E(Z_q)$: $y^2 = x^3 + b$, with $b \neq 0$, called Barreto–Naehrig curve.

In our construction, the map $\hat{e}$ will be derived either from Weil pairing or Ate pairing over the prime order elliptic curve group $E(Z_q)$ defined over the prime field Z_q [33,34]. According to the explanations given in [35], the bilinear pairing discussed above is a symmetric pairing of Type 1. For this type of pairing, the group G_q is a subgroup of $E(Z_q)$ and there is a *distortion map* defined as $\psi : G_q \rightarrow E(Z_{q^k})$. The pairing of $P, Q \in G_q$ can be computed efficiently by executing $\hat{e}(P, \psi(Q))$.

2.3 Computational Problems

In this section, we described some computational problems and hardness assumptions.

Definition 1 (*Bilinear Diffie–Hellman (BDH) problem*) Given a random tuple $\langle P, aP, bP, cP \rangle \in G_q$, where $a, b, c \in Z_q^*$, it is hard to compute $\hat{e}(P, P)^{abc}$ by a probabilistic polynomial time-bounded algorithm $\mathcal{B}$. The probability that $\mathcal{B}$ can solve the BDH problem is defined as $\text{Adv}_{\mathcal{B}}^{\text{BDH}}(k) = Pr[\mathcal{B}(P, aP, bP, cP) = \hat{e}(P, P)^{abc} : a, b, c \in Z_q^*]$.

Definition 2 (*Bilinear Diffie–Hellman (BDH) assumption*)) Given a random tuple $\langle P, aP, bP, cP \rangle \in G_q$, where $a, b, c \in Z_q^*$ and for every $\mathcal{B}$, $\text{Adv}_{\mathcal{B}}^{\text{BDH}}(k)$ is negligible.

Definition 3 (*Co-Diffie–Hellman (Co-DH) problem*) Given a random tuple $\langle P, Q, aP\rangle \in G_q$, where $a \in Z_q^*$, it is hard to compute aQ by a probabilistic polynomial time-bounded algorithm $\mathcal{B}$. The probability that $\mathcal{B}$ can solve the Co-BDH problem is defined as $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{Co-DH}}(k) = Pr[\mathcal{B}(P, aP, Q) = aQ : a \in Z_q^*]$.

Definition 4 (*Co-Diffie–Hellman (Co-DH) assumption*) Given a random tuple $\langle P, Q, aP\rangle \in G_q$, where $a \in Z_q^*$ and for every $\mathcal{B}$, $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{Co-DH}}(k)$ is negligible.

Definition 5 (*Co-Bilinear Diffie–Hellman (Co-BDH) problem*) Given a random tuple $\langle P, Q, aP, bP\rangle \in G_q$, where $a, b \in Z_q^*$, it is hard to compute $\hat{e}(P, Q)^{ab}$ by a probabilistic polynomial time-bounded algorithm $\mathcal{B}$. The probability that $\mathcal{B}$ can solve the Co-BDH problem is defined as $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{Co-BDH}}(k) = Pr[\mathcal{B}(P, aP, bP, Q) = \hat{e}(P, Q)^{ab} : a, b \in Z_q^*]$.

Definition 6 (*Co-Bilinear Diffie–Hellman (Co-BDH) assumption*) Given a random tuple $\langle P, Q, aP, bP\rangle \in G_q$, where $a, b \in Z_q^*$ and for every $\mathcal{B}$, $\mathrm{Adv}_{\mathcal{B}}^{\mathrm{Co-BDH}}(k)$ is negligible.

3 Formal Definition of SDVSMR Scheme

In this section, we present the formal definition of SDVSMR scheme. We assume ID_i is the identity of a user i, who may be the signer or designated verifier. Let us assume that the tuple $\langle x_i, P_i\rangle$ denotes the private key and public key pair of the user ID_i. The scheme SDVSMR has the following polynomial time-bounded algorithms, called **Setup**, **Keygen**, **Sign**, **Verify** and **Sig-sim**.

- **Setup:** The input of this probabilistic polynomial time (PPT) algorithm is a security parameter 1^k, and the output is the system's parameter Ω.
- **Keygen:** The system's parameter Ω is the input of this PPT algorithm, and the output is $\langle x_i, P_i\rangle$, where x_i is the private key and P_i is the public key of the user ID_i.
- **Sign:** This PPT algorithm takes a message $m_i \in \{0, 1\}^k$, private key x_i of the signer ID_i and public key P_j of the designated verifier ID_j as input and produces a signature σ_i for m_i.
- **Sig-sim:** The designated verifier ID_j executes this deterministic polynomial time-bounded algorithm to calculate an identically distributed signature, which is indistinguishable from the signature produced by the signer ID_i. This algorithm takes public key P_i of the signer ID_i, private key x_j of the designated verifier ID_j and a message $m_i \in \{0, 1\}^k$ as input and then outputs a simulated signature $\hat{\sigma_i}$ on m_i.
- **Verify:** This deterministic polynomial time-bounded algorithm takes signer's public key P_i, designated verifier's private key x_j and the signature σ_i as input; then, it recovers m_i from σ_i and outputs *true* if $\langle m_i, \sigma_i\rangle$ is valid and *false* otherwise.

4 Security Properties of SDVSMR Scheme

The following security properties must be satisfied by any SDVSMR scheme.

4.1 Correctness

If the signer ID_i properly computes a signature σ_i on a message m_i, then the designated verifier ID_j must be able to recover the message m_i from the signature σ_i and verifies the correctness of the message-signature pair $\langle m_i, \sigma_i\rangle$. That is, for $\Omega \leftarrow$ **Setup**(1^k), for any $ID_i, ID_j \in \{0, 1\}^*$, $\langle x_i, P_i\rangle \leftarrow$ **Keygen**(ID_i, Ω), $\langle x_j, P_j\rangle \leftarrow$ **Keygen**(ID_j, Ω) for any message $m_i \in \{0, 1\}^k$, if $\sigma_i \leftarrow$ **Sign**$(ID_i, ID_j, x_i, P_j, m_i)$ and $\hat{\sigma_i} \leftarrow$ **Sign-sim**$(ID_i, ID_j, x_j, P_i, m_i)$, therefore **Verify** $(ID_i, ID_j, x_j, P_i, m_i, \sigma_i) = true$ and **Verify**$(ID_i, ID_j, x_j, P_i, m_i, \hat{\sigma_i}) = true$ hold.

4.2 Strongness

A genuine signature σ_i can be verified and the correct message m_i from it can be recovered only by the designated verifier ID_j, but not by any outsider ID_l who does not have knowledge about the verifier's private key. That is, for $\Omega \leftarrow$ **Setup**(1^k), for any $ID_i, ID_j \in \{0, 1\}^*$, $\langle x_i, P_i\rangle \leftarrow$ **Keygen**(ID_i, Ω), $\langle x_j, P_j\rangle \leftarrow$ **Keygen**(ID_j, Ω) for any message $m_i \in \{0, 1\}^k$, if $\sigma_i \leftarrow$ **Sign**$(ID_i, ID_j, x_i, P_j, m_i)$ and $\hat{\sigma_i} \leftarrow$ **Sign-sim**$(ID_i, ID_j, x_j, P_i, m_i)$, therefore **Verify** $(ID_i, ID_j, x_l, P_i, m_i, \sigma_i) = false$ and **Verify**$(ID_i, ID_j, x_l, P_i, m_i, \hat{\sigma_i}) = false$ hold provided $x_j \neq x_l$, where x_l is the private key of the outsider ID_l.

4.3 Source Hiding

Suppose all the private keys of the signer ID_i and the designated verifier ID_j are known to an outsider; however, he cannot identify that ID_i is the signer or ID_j is the signer for a given message-signature pair $\langle m_i, \sigma_i\rangle$. That is, an outsider ID_l cannot distinguished the signature $\hat{\sigma}$ simulated by the verifier ID_j and the signature σ generated by the signer ID_i within polynomial time bound. That is, for $\Omega \leftarrow$ **Setup**(1^k), for any $ID_i, ID_j \in \{0, 1\}^*$, $\langle x_i, P_i\rangle \leftarrow$ **Keygen**(ID_i, Ω), $\langle x_j, P_j\rangle \leftarrow$ **Keygen**(ID_j, Ω), for any message $m_i \in \{0, 1\}^k$, then $\sigma_i \leftarrow$ **Sign**$(ID_i, ID_j, x_i, P_j, m_i)$ $\approx \hat{\sigma_i} \leftarrow$ **Sign-sim**$(ID_i, ID_j, x_j, P_i, m_i)$.

4.4 Non-delegatability

The *non-delegatable* property of an SDVSMR scheme state that an adversary $\mathcal{A}$ cannot generate a valid signature even if either the signer ID_i or the designated verifier ID_j delegates his/her signing capability to $\mathcal{A}$ without disclosing the secret key. That is, in a delegatable SDVSMR scheme, the signer ID_i disclose some side information of the secret key to $\mathcal{A}$ without disclosing the secret key x_i so that $\mathcal{A}$ can produce a valid signature on behalf of ID_i and this signature can be verified only by the designated verifier ID_j. Similarly, the designated verifier ID_j may disclose some side information to $\mathcal{A}$ such that $\mathcal{A}$ can produce a valid simulated signature. The formal definition [36] of non-delegatable property of an SDVSMR scheme is given as follows:

Definition 7 Suppose $\mathcal{K}$ be the knowledge extractor and $\xi \in [0, 1]$ is the knowledge error. An IBSDVS scheme is (t, ξ) non-delegatable if there is a $\mathcal{K}$; for every simulator $\mathcal{C}$ that runs in polynomial time, t, satisfies the following condition:
For $\Omega \leftarrow$ **Setup**(1^k), for every $ID_i, ID_j \in \{0,1\}^*$, $\langle x_i, P_i\rangle \leftarrow$ **Keygen**(ID_i, Ω), $\langle x_j, P_j\rangle \leftarrow$ **Keygen**(ID_j, Ω) and every message $m_i \in \{0, 1\}^k$, if $\mathcal{C}$ produces a valid signature σ_i on m_i against $\langle ID_i, ID_j\rangle$ with negligible probability $\epsilon > \xi$, then on input m_i and on oracle access to $\mathcal{C}$, $\mathcal{K}$ produces either x_i or x_j with in the time $\frac{t}{\epsilon-\xi}$, without considering the time to make oracle queries.

4.5 Unforgeability

An adversary $\mathcal{A}$ cannot compute a valid signature σ_i on a message $m_i \in \{0, 1\}^k$ chosen by himself without the private key x_i of the signer ID_i or the private key x_j of the designated verifier ID_j.

The formal unforgeability model of a SDVSMR scheme under the adaptively chosen message attack is defined by the following challenge-response game. This game is executed cooperatively by a polynomial time-bounded adversary $\mathcal{A}$ with a polynomial time-bounded algorithm/challenger $\mathcal{C}$.

- **Setup:** The challenger $\mathcal{C}$ executes the **Setup** algorithm. It takes a security parameter 1^k as input and then given the system's parameter Ω to $\mathcal{A}$ as output.
- **Keygen queries:** To obtain the private key of the user ID_i, $\mathcal{A}$ submit this query and then $\mathcal{C}$ returns $\langle x_i, P_i\rangle$ to $\mathcal{A}$, where x_i is the private key and P_i is the public key of ID_i.
- **Hash queries to H_i:** $\mathcal{C}$ maintains the initial-empty list $L_{H_i}^{\text{list}}$ for the oracle $H_i (i = 1, 2)$ and it includes the tuple $\langle c_i, d_i\rangle$. If $\mathcal{A}$ asks a H_i query with the input c_i, then $\mathcal{C}$ returns d_i, if a tuple $\langle c_i, d_i\rangle$ is in $L_{H_i}^{\text{list}}$. Otherwise, $\mathcal{C}$ chooses a number $d_i \in_R Z_q^*$ such that the tuple $\langle\cdot, d_i\rangle$ is not in $L_{H_i}^{\text{list}}$, then returns d_i as answer and incorporates $\langle c_i, d_i\rangle$ into the list $L_{H_i}^{\text{list}}$.
- **Sign queries:** To obtain a signature for an adaptively chosen message $m_i \in \{0, 1\}^k$, $\mathcal{A}$ asks a **Sign** query with the tuple $\langle ID_i, ID_j, m_i\rangle$, $\mathcal{C}$ then produces a signature σ_i and sends it to $\mathcal{A}$.
- **Verify queries:** Suppose $\mathcal{A}$ asks to verify $\langle ID_i, ID_j, \sigma_i\rangle$, $\mathcal{C}$ executes the **Verify** algorithm, then returns *true* if σ_i is valid and the recovered message m_i is correct, and returns *false* otherwise.
- **Forgery:** Finally, $\mathcal{A}$ stops and outputs a forged signature σ_i^* on m_i^* with the signer's identity ID_i^* and designated verifier's identity ID_j^*. The adversary $\mathcal{A}$ wins the game if the following holds:
  - $ID_i^* \neq ID_j^*$.
  - $\mathcal{A}$ did not make any **Keygen** queries on ID_i^* and ID_j^*.
  - $\mathcal{A}$ did not make any **Sign** queries with $\langle ID_i^*, ID_j^*, m_i^*\rangle$.
  - Signature σ_i^* of m_i^* is valid against ID_i^* and ID_j^*.

Definition 8 The advantage to win the above challenge-response game by a probabilistic polynomial time-bounded adversary with the help of $\mathcal{C}$ is defined as $\text{Adv}_{\mathcal{A},\text{UF}}^{\text{SDVSMR}}(k)$.

Definition 9 A SDVSMR scheme is existentially unforgeable in the random oracle model under the adaptively chosen message attack if $\text{Adv}_{\mathcal{A},\text{UF}}^{\text{SDVSMR}}(k)$ is negligible.

4.6 Non-transferability

It is impossible for the designated verifier ID_j to prove to an outsider $\mathcal{A}$ that σ_i is actually generated by the signer ID_i. Because, the designated verifier ID_j also has the ability to generate a simulated signature $\hat{\sigma}_i$, which indistinguishable from the signature σ_i generated by ID_i.

We can formally define the non-transferability of SDV SMR scheme against adaptive chosen message attack by the following challenge-response game, which is executed by a polynomial time-bounded adversary $\mathcal{A}$ and a simulator $\mathcal{C}$.

- **Setup:** This query is executed as described in the unforgeability game.
- **Keygen queries:** This query is executed as described in the unforgeability game.
- **Hash queries to H_i:** This query is executed as described in the unforgeability game.
- **Sign queries:** This query is executed as described in the unforgeability game.
- **Sign-sim queries:** To obtain a simulated signature on m_i (same message chosen in the **Sign** phase), $\mathcal{A}$ asks a **Sign-sim** query with the tuple $\langle ID_i, ID_j, m_i\rangle$, $\mathcal{C}$ outputs a simulated signature $\hat{\sigma}_i$ to $\mathcal{A}$.
- **Verify queries:** Suppose $\mathcal{A}$ asks to verify $\langle ID_i, ID_j, \sigma_i\rangle$ (or $\langle ID_i, ID_j, \hat{\sigma}_i\rangle$), $\mathcal{C}$ executes the **Verify** algorithm, then

returns *true* if σ_i (or $\hat{\sigma}_i$) is valid and the recovered message m_i is correct, and returns *false* otherwise.

- **Forgery:** Finally, $\mathcal{A}$ stops and outputs two forged signature σ_i^* on m_i^* and $\hat{\sigma}_i^*$ on m_i^* against the signer ID_i^* and the designated verifier ID_j^*. We can say that $\mathcal{A}$ wins this game if the following holds:
  - $ID_i^* \neq ID_j^*$.
  - $\mathcal{A}$ did not make any **Keygen** queries on ID_i^* and ID_j^*.
  - $\mathcal{A}$ did not make any **Sign** and **Sign-sim** queries with $\langle ID_i^*, ID_j^*, m_i^* \rangle$.
  - Both the signatures σ_i^* and $\hat{\sigma}$ on m_i^* are valid against ID_i^* and ID_j^*.

Definition 10 The advantage to win the above challenge-response game by a probabilistic polynomial time-bounded adversary with the help of $\mathcal{C}$ is defined as $Adv_{\mathcal{A},\text{NT}}^{\text{SDVSMR}}(k)$.

Definition 11 A SDVSMR scheme is non-transferable in the random oracle model against the adaptive chosen message attack if $\text{Adv}_{\mathcal{A},\text{NT}}^{\text{SDVSMR}}(k)$ is negligible.

5 The Proposed SDVSMR Scheme

The concrete description of the proposed SDVSMR scheme using elliptic curve and bilinear pairing is presented in this section. Here, we assumed that the original signer is identified with the identity ID_A and that the designated verifier is identified with the identity ID_B. The proposed scheme is the collection of the following algorithms:

5.1 Setup

On input a security parameter 1^k, this algorithm produces the system's parameter $\Omega = \langle F_q, E(F_q), G_q, P, Q, H_1, H_2 \rangle$, where q denotes *k-bit* prime number, P and Q are two generators of G_q, and $H_1, H_2 : \{0,1\}^* \to Z_q^*$ are two secure and one-way cryptographic hash functions.

5.2 Keygen

The user $ID_i, i \in \{A, B\}$ picks a number $x_i \in_R Z_q^*$ as his/her private key and publishes $P_i = x_i P$ as his/her public key.

5.3 Sign

To compute the signature $\sigma = \langle R, t, g \rangle$, the signer ID_A chooses a message $m \in \{0,1\}^k$ and then calculates the following:

(i) Choose $r \in_R Z_q^*$ and compute $R = rP_A$. (2)

(ii) Compute $l = H_1(\hat{e}(Q, P_B)^{rx_A})$. (3)

(iii) Compute $t = l \oplus m (\text{mod } q)$. (4)

(iv) Compute $h = H_2(m, t, l)$. (5)

(v) Compute $s = (r + h)x_A (\text{mod } q)$. If $s = 0$ go to step (i), otherwise proceed to the next step. (6)

(vi) Compute $g = \hat{e}(Q, P_B)^s$. (7)

(vii) Output the signature $\sigma = \langle R, t, g \rangle$.

5.4 Verify

On receiving the signature $\sigma = \langle R, t, g \rangle$, the designated verifier ID_B does as follows:

(i) Compute $l' = H_1\left(\hat{e}(x_B Q, R)\right)$. (8)

(ii) Compute $m' = t \oplus l' (\text{mod } q)$. (9)

(iii) Compute $h' = H_2\left(m', t, l'\right)$. (10)

(iv) Compute $g' = \hat{e}\left(x_B Q, R + h' P_A\right)$. (11)

(v) Accept the signature $\sigma = \langle R, t, g \rangle$ and the message m is correct i.e., $m' = m$ if $g' = g$ holds, otherwise reject the signature $\sigma = \langle R, t, g \rangle$.

5.5 Sig-Sim

To generate a simulated signature, the designated verifier ID_B selects a message $m \in \{0,1\}^k$ and then calculates the following:

(i) Choose a number $\hat{r} \in_R Z_q^*$ and compute $\hat{R} = \hat{r} P_A$.

(ii) Compute $\hat{l} = H_1\left(\hat{e}(x_B Q, \hat{R})\right)$.

(iii) Compute $\hat{t} = \hat{l} \oplus m (\text{mod } q)$.

(iv) Compute $\hat{h} = H_2\left(m, \hat{t}, \hat{l}\right)$.

(v) Compute $\hat{s} = \left(\hat{r} + \hat{h}\right) x_B (\text{mod } q)$. If $\hat{s} = 0$ go to step (i), otherwise proceed to the next step.

(vi) Compute $\hat{g} = \hat{e}(Q, P_A)^{\hat{s}}$.

It is to be noted that the simulated signature $\hat{\sigma} = \langle \hat{R}, \hat{t}, \hat{g} \rangle$ is also a valid signature.

In Figs. 1 and 2, we further illustrated the signature computation and verification phases of the proposed SVDSMR.

6 Security Analysis of the Proposed Scheme

Here, we evaluated all the security requirements of the proposed SDVSMR scheme. We will also demonstrated that our scheme is unforgeable against the adaptive chosen message attack in the random oracle model.

Theorem 1 *If the signer computes the strong designated verifier signature $\sigma = \langle R, t, g \rangle$ on a message m for the designated verifier, then the signature σ is correct and consistent,*

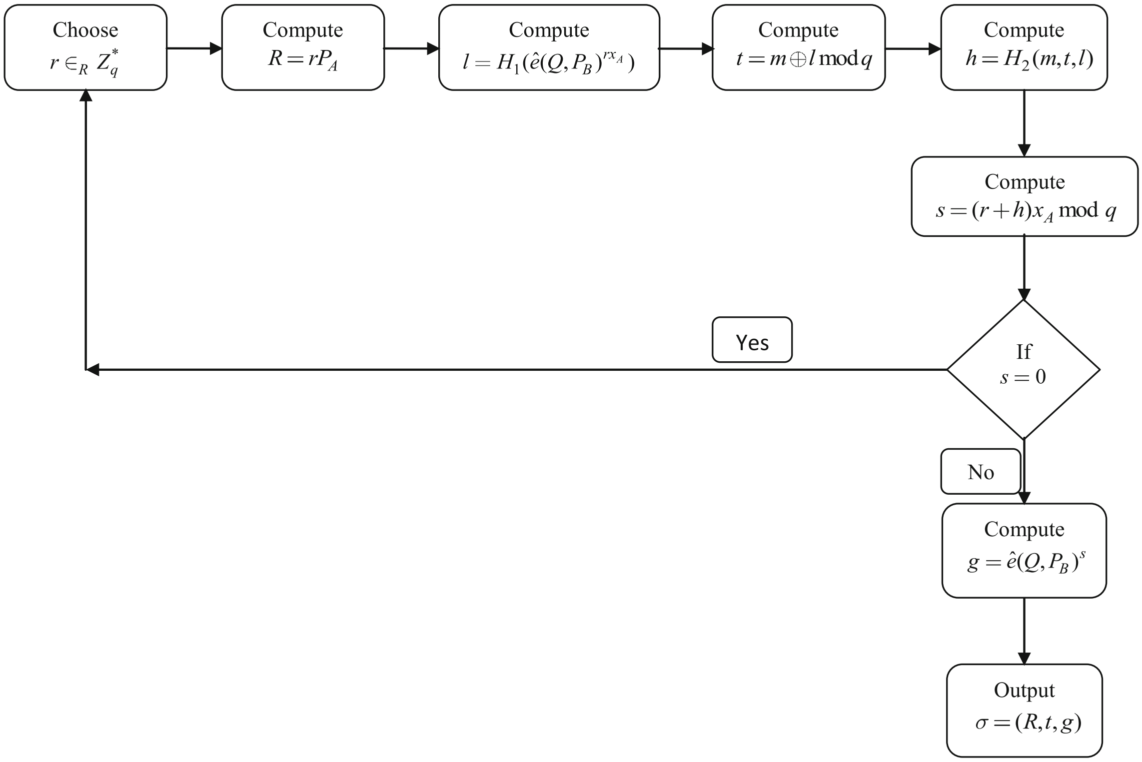

Fig. 1 Signature generation process of the proposed SDVSMR scheme

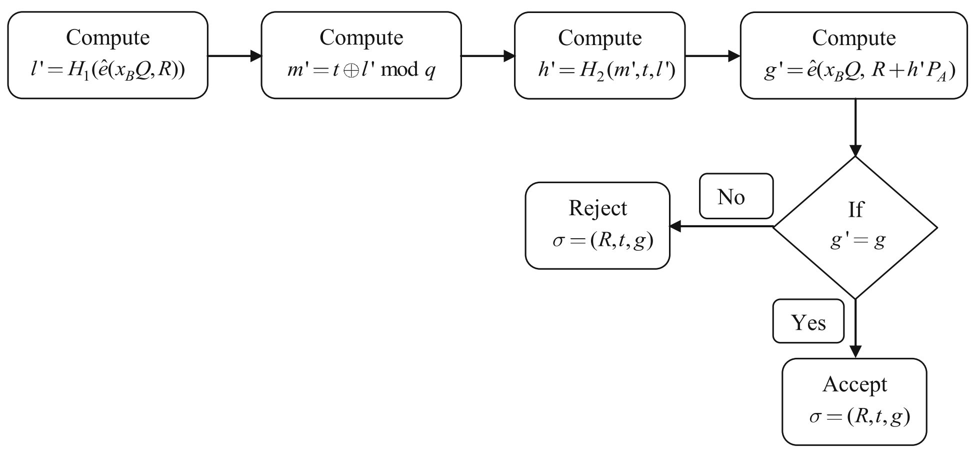

Fig. 2 Signature verification process of the proposed SDVSMR scheme

and the message m only can be recovered by the designated verifier.

Proof From Eqs. (3) and (8), we have

$$
\begin{aligned}
l' &= H_1\left(\hat{e}(x_B Q, R)\right) \\
&= H_1\left(\hat{e}(Q, r x_A P)^{x_B}\right) \\
&= H_1\left(\hat{e}(Q, P)^{r x_A x_B}\right) \\
&= H_1\left(\hat{e}(Q, x_B P)^{r x_A}\right) \\
&= H_1\left(\hat{e}(Q, P_B)^{r x_A}\right) \\
&= l
\end{aligned} \tag{12}
$$

From Eqs. (4), (9) and (12), we obtained

$$
\begin{aligned}
m' &= t \oplus l' \\
&= m \oplus l \oplus l \\
&= m
\end{aligned} \tag{13}
$$

From Eqs. (10), (12) and (13), we get

$$
\begin{aligned}
h' &= H_2\left(m', t, l'\right) \\
&= H_2(m, t, l) \\
&= h
\end{aligned} \tag{14}
$$

From Eq. (11), we derived

$$
\begin{aligned}
g' &= \hat{e}\left(x_B Q, r x_A P + h' x_A P\right) && \text{[Eqs. (2) and (14)]}\\
&= \hat{e}\left(x_B Q, (r+h) x_A P\right)\\
&= \hat{e}(x_B Q, sP) && \text{[Eq. (6)]}\\
&= \hat{e}(Q, P)^{x_B s} && \text{[Bilinearity]}\\
&= \hat{e}(Q, x_B P)^{s} && \text{[Bilinearity]}\\
&= \hat{e}(Q, P_B)^{s} && \text{[Bilinearity]}\\
&= g && \text{[Eq. (7)]}
\end{aligned}
$$

Therefore, the signature $\sigma = \langle R, t, g\rangle$ is valid and the recovered message m is correct.

Theorem 2 *The proposed SDVSMR scheme is a strong designated verifier signature scheme.*

Proof In the following, we proved that our SDVSMR scheme satisfies the *strongness* property. Assume that the signer ID_A generates a valid signature $\sigma = \langle R, t, g\rangle$ for the designated verifier ID_B. For an outsider identified with the identity ID_C, there is no way to obtain the information about the private keys x_A and x_B of ID_A and ID_B from $\sigma = \langle R, t, g\rangle$. Moreover, from the verification equation $g' = \hat{e}(x_B Q, R + h' P_A) = g$, we observed that x_B is strictly required to check the validity of $\sigma = \langle R, t, g\rangle$ and to recover the message m correctly. As a result, the outsider ID_C cannot recover m and verify $\sigma = \langle R, t, g\rangle$ without x_B. Thus, only the designated verifier ID_B can verify message-signature pair $\langle m, \sigma\rangle$. □

Theorem 3 *The proposed SDVSMR scheme satisfies the source-hiding property.*

Proof The *source-hiding* property of an strong designated verifier signature scheme states that the outsider ID_C cannot recognize whether a given signature $\sigma = \langle R, t, g\rangle$ for a message m is produced by the signer ID_A or the designated verifier ID_B, even if the private keys of ID_A and ID_B are disclosed to the outsider ID_C. Let us define $\mathcal{S}$ be the set of signatures generated by the signer ID_A for the designated verifier ID_B and $\hat{\mathcal{S}}$ be the set of simulated signatures computed by the designated verifier ID_B for himself.

Let the signature $\sigma'' = \langle R'', t'', g''\rangle$ for some message $m \in \{0,1\}^k$ is chosen randomly from $\mathcal{S}$; thus,

$$
\begin{aligned}
Pr[(R,t,g) = \left(R'', t'', g''\right)] &= \left\{\begin{array}{l} r \in_R Z_q^*,\ R = rP_A = R'' \\ l = H_1(\hat{e}(Q, P_B)^{r x_A}) = l'' \\ t = l \oplus m (\text{mod } q) = t'' \\ h = H_2(m, t, l) = h'' \\ s = (r+h) x_A (\text{mod } q) = s'' \\ g = \hat{e}(Q, P_B)^s = g'' \end{array}\right\} \\
&= \frac{1}{q^3}
\end{aligned}
$$

Since r and t are chosen randomly from a uniformed set Z_q^* of order q and g is selected from the group G_m of order q, let the signature $\hat{\sigma} = \langle \hat{R}, \hat{t}, \hat{g}\rangle$ on the same message m is chosen randomly from $\hat{\mathcal{S}}$; thus,

$$
\begin{aligned}
Pr[(R,t,g) = (\hat{R}, \hat{t}, \hat{g})] &= \left\{\begin{array}{l} r \in_R Z_q^*,\ R = rP_A = \hat{R} \\ l = H_1(\hat{e}(Q, P_B)^{r x_A}) = \hat{l} \\ t = l \oplus m (\text{mod } q) = \hat{t} \\ h = H_2(m, t, l) = \hat{h} \\ s = (r+h) x_A (\text{mod } q) = \hat{s} \\ g = \hat{e}(Q, P_B)^s = \hat{g} \end{array}\right\} \\
&= \frac{1}{q^3}
\end{aligned}
$$

Therefore, from the above two equations, we can say that the signature $\hat{\sigma}$ simulated by the verifier ID_B and the signature σ generated by the signer ID_A are statistically indistinguishable from each other. Accordingly, a polynomial time-bounded adversary $\mathcal{A}$ cannot distinguish the simulated signatures from the real signatures. Thus, the proposed signature scheme achieves the source-hiding property. □

Theorem 4 *The proposed SDVSMR scheme is non-delegatable in the random oracle model.*

Proof Here, we will prove that the proposed SDVSMR scheme is non-delegatable in the programmable random oracle model as described in [36]. Assume that $\epsilon > \xi = \frac{1}{q}$, and there exists a polynomial time-bounded knowledge extractor $\mathcal{K}$ that on input of a signature $\sigma = \langle R, t, g\rangle$ and on oracle access to the adversary $\mathcal{A}$ can produce either the private key x_A of the signer ID_A or the private key x_B of the designated verifier ID_B within the time bound $\tau' \le \frac{56\tau}{\epsilon}$ and with probability 1, where $\mathcal{A}$ has the ability to constructs two valid strong designated verifier signatures within time bound τ and with probability ϵ. Assume that $\mathcal{A}_m$ be a forger with the input m. Consider two executions of $\mathcal{A}_m$ by $\mathcal{K}$ with the same random input. In both cases, $\mathcal{K}$ executes $\mathcal{A}_m$ step-by-step, except that $\mathcal{K}$ returns two valid signatures $\sigma = \langle R, t, g\rangle$ and $\sigma' = \langle R', t', g'\rangle$ with two different hash values h and h'. Since σ and σ' are valid, therefore, we have $s x_A^{-1} - h \equiv \hat{s} x_A^{-1} - \hat{h}$ (mod q) and $s' x_B^{-1} - h' \equiv \hat{s}' x_B^{-1} - \hat{h}'$ (mod q). Therefore, $\mathcal{A}_m$ computes $x_A = \frac{s - \hat{s}}{h - \hat{h}}$ and $x_B = \frac{s' - \hat{s}'}{h' - \hat{h}'}$.

According to [37], there exists an algorithm **Rewind**, on oracle access to the adversary $\mathcal{A}_m$, in time τ, outputs two correct signatures $\sigma = \langle R, t, g\rangle$ and $\sigma' = \langle R', t', g'\rangle$ such that $h \ne h'$, but $\langle R, t, g\rangle = \langle R', t', g'\rangle$ holds. Accordingly, $\mathcal{A}_m$ can compute either the private key x_A of the signer ID_A or the private key x_B of the designated verifier ID_B within the time bound $\tau' \le \frac{56\tau}{\epsilon}$ and with probability 1. □

Theorem 5 *The proposed SDVSMR scheme is secure against the adaptive chosen message attack in the random oracle model based on the infeasibility of the Co-BDH problem.*

Proof Assume that the proposed SDVSMR scheme can be forged by the probabilistic polynomial time-bounded adversary $\mathcal{A}$; then, it is possible to construct a challenger $\mathcal{C}$ which helps $\mathcal{A}$ to solve the Co-BDH problem, i.e., $\mathcal{A}$ produces $\hat{e}(P, Q)^{ab}$ from the given Co-BDH problem instance $\langle P, Q, aP, bP\rangle$, where $a, b \in Z_q^*$ are unknown to $\mathcal{A}$. In order to breach the unforgeability of our scheme, $\mathcal{C}$ sets $P_A = aP$ and $P_B = bP$, respectively, and then gives $\Omega = \langle F_q, E(F_q), G_q, \hat{e}, P, Q, P_A = aP, P_B = bP, H_1, H_2\rangle$ to $\mathcal{A}$. $\mathcal{C}$ maintains the following lists in order to achieve the consistency between queries made by $\mathcal{A}$:

- L_{H1}^{list}: This is an initial-empty list, and it consists the tuple of type $\langle r_i, P_j, l_i\rangle$.
- L_{H2}^{list}: This is an initial-empty list, and it consists the tuple of type $\langle m_i, t_i, h_i\rangle$.
- L_{pk}^{list}: This is an initial-empty list, and it consists the tuple of type $\langle ID_i, x_i, P_i\rangle$.

Now $\mathcal{C}$ answers $\mathcal{A}$'s queries in the following ways:

- **Keygen queries:** If $\mathcal{A}$ asked an **Keygen** query for the user ID_i, then $\mathcal{C}$ responds as follows:
  - If $ID_i = ID_A$, output the tuple $\langle ID_A, \perp, P_A = aP\rangle$.
  - If $ID_i = ID_B$, output the tuple $\langle ID_B, \perp, P_B = bP\rangle$.
  - Else, choose $x_i \in_R Z_q^*$, compute $P_i = x_i P$ and returns $\langle ID_i, x_i, P_i\rangle$ as answer.

  Finally, $\mathcal{C}$ incorporates the tuple $\langle ID_i, x_i, P_i\rangle$ into the list L_{pk}^{list}.
- **Hash queries to** H_1**:** Suppose $\mathcal{A}$ asks a H_1 query with the input $\langle r_i, P_j\rangle$, $\mathcal{C}$, then replies with the previous l_i if a tuple $\langle r_i, P_j, l_i\rangle$ is found in L_{H1}^{list}. Otherwise, $\mathcal{C}$ selects a number $l_i \in_R Z_q^*$ such that there is no item $\langle\cdot, \cdot, l_i\rangle$ in L_{H1}^{list} and returns l_i to $\mathcal{A}$, and includes $\langle r_i, P_j, l_i\rangle$ into L_{H1}^{list}.
- **Hash queries to** H_2**:** Suppose $\mathcal{A}$ asks a H_2 query with the input $\langle m_i, t_i\rangle$, $\mathcal{C}$ then replies with the previous h_i if a tuple $\langle m_i, t_i, h_i\rangle$ is found in L_{H2}^{list}. Otherwise, $\mathcal{C}$ selects a number $h_i \in_R Z_q^*$ such that there is no tuple $\langle\cdot, \cdot, h_i\rangle$ in L_{H2}^{list}, returns h_i to $\mathcal{A}$ and includes $\langle m_i, t_i, h_i\rangle$ into L_{H2}^{list}.
- **Sign queries:** Suppose that $\mathcal{A}$ asks to produce a signature on an adaptively chosen message $m_i \in \{0, 1\}^k$ for the signer ID_i and the designated verifier ID_j. $\mathcal{C}$ executes the following:

  (i) If $\langle ID_i, ID_j\rangle = \langle ID_A, ID_B\rangle$ or $\langle ID_i, ID_j\rangle = \langle ID_B, ID_A\rangle$, $\mathcal{C}$ outputs *failure* and aborts the simulation.

  (ii) Otherwise, $\mathcal{C}$ uses the private key x_i of ID_i and then performs the following:
  - Choose $r_i \in_R Z_q^*$.
  - Compute $R_i = r_i P_i$ and $l_i = H_1\left(\hat{e}(Q, P_j)^{r_i x_i}\right)$.
  - Compute $t_i = m_i \oplus l_i$ and $h_i = H_2(m_i, t_i, l_i)$.
  - Compute $s_i = (r_i + h_i)x_i$ and $g_i = \hat{e}(Q, P_j)^{s_i}$.
  - Output $\sigma_i = \langle R_i, t_i, g_i\rangle$.
- **Verify queries:** If $\mathcal{A}$ asks to verify a signature $\sigma_i = \langle R_i, t_i, g_i\rangle$ and to recover m_i for the signer ID_i and the designated verifier ID_j, $\mathcal{C}$ then does as follows:

  (i) If $\langle ID_i, ID_j\rangle = \langle ID_A, ID_B\rangle$ or $\langle ID_i, ID_j\rangle = \langle ID_B, ID_A\rangle$ holds, then terminate the protocol simulation.

  (ii) Otherwise, use the private key x_j of ID_j and verifies $\sigma_i = \langle R_i, t_i, g_i\rangle$ using the **Verify** algorithm of our scheme.
- **Forgery:** Finally, $\mathcal{C}$ stops the protocol execution and outputs a signature $\sigma = \langle R, t, g\rangle$ with the hash value h of the message m if $\langle ID_i, ID_j\rangle = \langle ID_A, ID_B\rangle$ (or $\langle ID_i, ID_j\rangle = \langle ID_B, ID_A\rangle$) hold. Based on the forking lemma [38], $\mathcal{C}$ finds the tuples $\langle r_i, P_j, l_i\rangle$ and $\langle m_i, t_i, h_i\rangle$ from the lists L_{H1}^{list} and L_{H2}^{list} and another valid signature $\sigma' = \langle R', t', g'\rangle$ with the hash value h' on m such that $h \neq h'$, $g \neq g'$ and $R = R'$. Since both $\sigma = \langle R, t, g\rangle$ and $\sigma' = \langle R', t', g'\rangle$ are valid signatures on the message m. Therefore, we can write $g = \hat{e}(d_B Q, R + hP_A)$ and $g' = \hat{e}(d_B Q, R' + h'P_A)$. We have

$$\frac{g'}{g} = \frac{\hat{e}\left(x_B Q, R + h'P_A\right)}{\hat{e}\left(x_B Q, R + hP_A\right)}$$
$$= \hat{e}\left(x_B Q, (h' - h)P_A\right)$$
$$= \hat{e}(P_A, x_B Q)^{(h'-h)}$$

That is,

$$\left(\frac{g'}{g}\right)^{\frac{1}{(h'-h)}} = \hat{e}(P_A, x_B Q)$$
$$= \hat{e}(aP, bQ)$$
$$= \hat{e}(P, Q)^{ab}$$

Hence, $\mathcal{C}$ solves the Co-BDH problem as $\hat{e}(P, Q)^{ab} = \left(\frac{g'}{g}\right)^{\frac{1}{(h'-h)}}$ and it contradicts that the Co-BDH problem is computationally hard. Therefore, our SDVSMR scheme is existentially unforgeable in the random oracle model against the adaptive chosen message attack.

Theorem 6 *The proposed SDVSMR scheme is non-transferable against the adaptive chosen message attack in the random oracle model based on the infeasibility of the Co-BDH problem.*

Proof Suppose that a probabilistic polynomial time-bounded adversary $\mathcal{A}$ breaches the non-transferability of our scheme.

If this happen, then there must be a polynomial time-bounded challenger $\mathcal{C}$ which helps $\mathcal{A}$ to solve the Co-BDH problem, i.e., $\mathcal{A}$ can compute $\hat{e}(P, Q)^{ab}$ from a given tuple $\langle P, Q, aP, bP \rangle$, where $a, b \in Z_q^*$ are not known to $\mathcal{A}$. Now, $\mathcal{C}$ sets $P_A = aP$ and $P_B = bP$, respectively, and then returns $\Omega = \langle F_q, E(F_q), G_q, \hat{e}, P, Q, P_A = aP, P_B = bP, H_1, H_2 \rangle$ to $\mathcal{A}$. Similar to the Theorem 5, $\mathcal{C}$ maintains the following lists $L_{H1}^{\text{list}}, L_{H2}^{\text{list}}$ and L_{pk}^{list}, respectively. The challenger $\mathcal{C}$ returns output based on $\mathcal{A}$'s queries as follows:

- **Keygen queries:** This query is same as given in Theorem 5.
- **Hash queries to H_1:** This query is same as given in Theorem 5.
- **Hash queries to H_2:** This query is same as given in Theorem 5.
- **Sign queries:** This query is same as given in Theorem 5.
- **Sign-sim queries:** Suppose that $\mathcal{A}$ asks to produce a simulated signature on an adaptively chosen message $m_i \in \{0, 1\}^k$ (as chosen in **Sign** phase) for the signer ID_i and the designated verifier ID_j. $\mathcal{C}$ does as follows:
  - (i) If $\langle ID_i, ID_j \rangle = \langle ID_A, ID_B \rangle$ or $\langle ID_i, ID_j \rangle = \langle ID_B, ID_A \rangle$, $\mathcal{C}$ outputs *failure* and aborts the simulation.
  - (ii) Otherwise, $\mathcal{C}$ uses the private key x_j of ID_j and then does as follows:
    - Choose $\hat{r}_i \in_R Z_q^*$.
    - Compute $\hat{R}_i = r_i P_i$ and $\hat{l}_i = H_1(\hat{e}(Q, P_i)^{\hat{r}_i x_j}$.
    - Compute $\hat{t}_i = m_i \oplus \hat{l}_i$ and $\hat{h}_i = H_2(m_i, \hat{t}_i, \hat{l}_i)$.
    - Compute $\hat{s}_i = (\hat{r} + \hat{h})x_j$ and $\hat{g}_i = \hat{e}(Q, P_i)^{\hat{s}_i}$.
    - Output $\hat{\sigma}_i = \langle \hat{R}_i, \hat{t}_i, \hat{g}_i \rangle$.
- **Verify queries:** If $\mathcal{A}$ asks to verify $\sigma_i = \langle R_i, t_i, g_i \rangle$ or $\hat{\sigma}_i = \langle \hat{R}_i, \hat{t}_i, \hat{g}_i \rangle$ and to recover m_i, $\mathcal{C}$ then does as follows:
  - (i) If $\langle ID_i, ID_j \rangle = \langle ID_A, ID_B \rangle$ or $\langle ID_i, ID_j \rangle = \langle ID_B, ID_A \rangle$ holds, terminate the protocol execution.
  - (ii) Otherwise, use the private key x_j of ID_j and verify $\sigma_i = \langle R_i, t_i, g_i \rangle$ (or $\hat{\sigma}_i = \langle \hat{R}_i, \hat{t}_i, \hat{g}_i \rangle$) using the **Verify** algorithm of our scheme.
- **Forgery:** Finally, $\mathcal{C}$ terminates the protocol simulation and produces a signature $\sigma = \langle R, t, g \rangle$ with the hash value h of the message m if $\langle ID_i, ID_j \rangle = \langle ID_A, ID_B \rangle$ (or $\langle ID_i, ID_j \rangle = \langle ID_B, ID_A \rangle$) holds. Moreover, $\mathcal{C}$ finds the tuples $\langle r_i, P_j, l_i \rangle$ and $\langle m_i, t_i, h_i \rangle$ from L_{H1}^{list} and L_{H2}^{list} and can produce a simulated signature $\hat{\sigma}_i = \langle \hat{R}_i, \hat{t}_i, \hat{g}_i \rangle$ with the hash value $\hat{h}$ on m according to the **Sign-sim** algorithm such that $h \neq \hat{h}, g \neq \hat{g}$ and $R = \hat{R}$ holds. Since $\sigma = \langle R, t, g \rangle$ and $\hat{\sigma} = \langle \hat{R}, \hat{t}, \hat{g} \rangle$ are valid for m. Accordingly, $g = \hat{e}(d_B Q, R + hP_A)$ and $\hat{g} = \hat{e}(d_B Q, \hat{R} + \hat{h} P_A)$ hold. We have

$$\frac{\hat{g}}{g} = \frac{\hat{e}\left(x_B Q, R + \hat{h} P_A\right)}{\hat{e}\left(x_B Q, R + h P_A\right)} = \hat{e}\left(x_B Q, (\hat{h} - h) P_A\right) = \hat{e}(P_A, x_B Q)^{(\hat{h}-h)}$$

That is,

$$\left(\frac{\hat{g}}{g}\right)^{\frac{1}{(\hat{h}-h)}} = \hat{e}(P_A, x_B Q) = \hat{e}(aP, bQ) = \hat{e}(P, Q)^{ab}$$

Therefore, $\mathcal{C}$ solves the Co-BDH problem as $\hat{e}(P, Q)^{ab} = \left(\frac{\hat{g}}{g}\right)^{\frac{1}{(\hat{h}-h)}}$, and thus, our SDVSMR scheme is non-transferable in the random oracle model. □

7 Efficiency Comparison of our SDVSMR Scheme with Others

In this section, we illustrated the performance comparisons of our scheme with the related schemes [18–22] from the computation and communication (signature length) costs point of view. For this purpose, in Table 1, we define some computational time complexity and their conversions [39,40] in terms of T_{ML}.

As discussed in [41], to achieve the comparable security with 1,024-bit RSA key, bilinear pairing-based schemes execute Ate pairing on a supersingular elliptic curve $E(F_q)$: $y^2 = x^3 + x$ with embedding degree 2 and the large prime order q, which is a 160-bit Solinas prime of the form $q = 2^{159} + 2^{17} + 1$ and p is at least 512-bit prime number that satisfies $p + 1 = 12qr$ [42]. To achieve the same level of

Table 1 Different notations and their meanings

Notations	Definition and conversion
T_{ML}	Time needed to execute the modular multiplication operation
T_{EX}	Time needed to execute modular exponentiation operation, $T_{\text{EX}} \approx 240 T_{\text{ML}}$
T_{EM}	Time needed to execute the elliptic curve point multiplication operation, $T_{\text{EM}} \approx 29 T_{ML}$
T_{BP}	Time needed to execute the bilinear pairing operation, $T_{\text{BP}} \approx 87 T_{\text{ML}}$
T_{PX}	Time needed to execute the pairing-based exponentiation operation, $T_{\text{PX}} \approx 43.5 T_{ML}$
T_{IN}	Time needed to execute the modular inversion operation, $T_{\text{IN}} \approx 11.6 T_{\text{ML}}$

Table 2 Computation cost comparison of the different schemes

Scheme	Signature length (bits)	Signing cost	Verification cost	Total cost
Lee and Chang [18]	$4 \times 1{,}024$	$3T_{EX} + T_{IN}$	$5T_{EX}$	$8T_{EX} + T_{IN} \approx 1{,}931T_{ML}$
Saeednia et al. [19]	$3 \times 1{,}024 + \|m\|$	T_{EX}	$3T_{EX}$	$4T_{EX} \approx 960T_{ML}$
Lee and Chang [20]	$2 \times 1{,}024 + \|m\|$	$2T_{EX}$	$2T_{EX}$	$4T_{EX} \approx 960T_{ML}$
Yang and Liao [21]	$1{,}024 + 160$	T_{EX}	T_{EX}	$2T_{EX} \approx 480T_{ML}$
Shim [22]	$4 \times 1{,}024 + \|m\|$	$5T_{EX}$	$T_{EX} + 4T_{BP}$	$6T_{EX} + 4T_{BP} \approx 1{,}798T_{ML}$
Proposed	$2 \times 512 + 160$	$2T_{PX} + T_{EM}$	$2T_{BP} + 2T_{EM}$	$2T_{BP} + 2T_{PX} + 3T_{EM} \approx 348T_{ML}$

security, pairing-free elliptic curve-based schemes execute operations on Koblitz curve defined as $y^2 = x^3 + ax^2 + b$ on $F_{2^{163}}$ with $a = 1$ and b is a 163-bit random prime number. Thus, the security provided by a 512-bit random number in a pairing-based scheme is equivalent to a 160-bit random number in a pairing-free scheme and 1,024-bit number in RSA type scheme. Here, we also assume that the output length of the hash function is 160 bits. Therefore, the length of the signature of our scheme is $(2 \times 512 + 160)$ bits = 1,184 bits.

Due to the lightweight feature of the hash function (H_1 and H_2) and the elliptic curve point addition operation, we ignore these computations in our comparison. It is assumed that the order of G_q and G_m is a large prime number q (512 bits) and $|G_q| = |G_m| = 512$ bits; $|G_q|$ denotes the bit length of the element of G_q. In our scheme, the signer can pre-compute $\hat{e}(Q, P_B)$, and thus, to compute $l = H_1(\hat{e}(Q, P_B)^{rd_A})$ and $g = \hat{e}(Q, P_B)^s$, he/she has to execute only two pairing-based exponentiations ($2T_{PX}$). Therefore, the signature generation phase and the verification phase involve ($2T_{PX} + T_{EM}$) and ($2T_{BP} + 2T_{EM}$) amount of time. Thus, the total computation cost of our scheme is ($2T_{BP} + 2T_{PX} + 3T_{EM} \approx 348T_{ML}$) amount of time, whereas other schemes need more. We conducted a comparison in Table 2 of different schemes [18–22] with respect to computation and communication costs.

It is clear that our scheme bears benefit of message recovery and the length of the signature in the proposed scheme is reduced compared with other related schemes. Similar to the scheme [22], our scheme is provably secure in the random oracle model. However, the schemes [18–21] are not provably secured. From the Table 2, we have seen that, the communication cost of our scheme is 28 % of Lee and Chang's scheme [18], 38 % of Saeednia et al.'s scheme [19], 14 % of Lee and Chang's scheme [20], 100 % of Yang and Liao's scheme [21] and 28 % of Shim's scheme [22], respectively. Based on the Table 2, we observed that the computation cost of our scheme is 18 % of Lee and Chang's scheme [18], 36 % of Saeednia et al.'s scheme [19], 36 % of Lee and Chang's scheme [20], 72 % of Yang and Liao's scheme [21] and 20 % of Shim's scheme [22], respectively.

8 Conclusion and Future Scope

This paper proposed a provably secure SDVSMR scheme using bilinear pairing and elliptic curve. The security analysis demonstrates that our scheme provides unforgeablility in the random oracle model and its security against adaptive chosen message adversary is based on the Co-BDH assumption. Furthermore, our scheme is shown to be more efficient than the earlier schemes from the perspective of computation and communication costs. Thus, our scheme will be more useful in resource-constrained and small message applications where confidentiality, integrity, authentication and non-repudiation of the message are needed.

Although the proposed SDVSMR scheme is implemented with symmetric bilinear pairing on supersingular elliptic curve; however, it needs a global public key infrastructure to authenticate the public keys of the signer and the designated verifier. In addition, our scheme requires more computation costs due to bilinear pairing compared with the pairing-free schemes. As a result, our scheme experiences additional overhead due to the public key infrastructure and bilinear pairing. Thus, we will study an efficient identity-based SDVSMR scheme with bilinear pairing-free concept in the near future.

Acknowledgments The work is supported by the Outstanding Potential for Excellence in Research and Academics (OPERA) award, Birla Institute of Technology and Science (BITS) Pilani, Pilani Campus, Rajasthan 333031, India. The authors would like to acknowledge the many helpful suggestions of the anonymous reviewers and the Editor-in-Chief, which have improved the content and the presentation of this paper.

References

1. Jakobsson, M.; Sako, K.; Impagliazzo, R.: Designated verifier proofs and their applications. In: Advances in Cryptology (Eurocrypt '96). Lecture Notes in Computer Science, vol. 1070, pp. 143–154. Springer, Berlin (1996)
2. Miller, V.S.: Use of elliptic curves in cryptography. In: Proceedings of the Advances in Cryptology (Crypto '85). Lecture Notes in Computer Science, pp. 417–426. Springer, Berlin (1985)
3. Koblitz, N.: Elliptic curve cryptosystem. J. Math. Comput. **48**, 203–209 (1987)

4. Hankerson, D.; Menezes, A.; Vanstone, S.: Guide to Elliptic Curve Cryptography. Springer, New York (2004)
5. Boneh, D.; Franklin, M.K.: Identity-based encryption from the Weil pairing. SIAM J. Comput. **32**, 586–615 (2003)
6. Boneh, D.; Lynn, B.; Shacham, H.: Short signatures from the Weil pairing. J. Cryptol. **17**(4), 297–319 (2001)
7. Zhang, J.; Mao, J.: A novel ID-based designated verifier signature scheme. Inf. Sci. **178**, 766–773 (2008)
8. Kang, B.; Boyd, C.; Dawson, E.: Identity-based strong designated verifier signature schemes: attacks and new construction. Comput. Electr. Eng. **35**, 49–53 (2009)
9. Lee, J.S.; Chang, J.H.; Lee, D.H.: Forgery attacks on Kang et al.'s identity-based strong designated verifier signature scheme and its improvement with security proof. Comput. Electr. Eng. **36**, 948–954 (2010)
10. Kumar, K.; Shailaja, G.; Saxena, A.: Identity Based Strong Designated Verifier Signature Scheme. Cryptography eprint archive report 2006/134. International Association for Cryptologic Research. http://eprint.iacr.org/omplete/2006/134
11. Kang, B.; Boyd, C.; Dawson, E.: A novel identity-based strong designated verifier signature scheme. J. Syst. Softw. **82**, 270–273 (2009)
12. Du, H.; Wen, Q.: Attack on Kang et al.'s Identity-Based Strong Designated Verifier Signature Scheme. Cryptography eprint archive report 2008/297. International Association for Cryptologic Research. http://eprint.iacr.org/2008/297
13. Yang, B.; Xiao, Z.; Hu, Z.: A secure ID-based strong designated verifier signature scheme. In: Proceedings of the International Conference on Network Infrastructure and Digital, pp. 543–547. IEEE, Beijing, China (2009)
14. Sun, S.; Wen, Q.; Jin, Z.; Zhang, H.: A New Efficient ID-based Strong Designated Verifier Signature Scheme. In: Proceedings of the Third International Symposium on Information Science and Engineering, pp. 137–141. IEEE, Shanghai, China (2010)
15. Huang, Q.; Yang, G.; Wong, D.S.; Susilo, W.: Identity-based strong designated verifier signature revisited. J. Syst. Softw. **84**, 120–129 (2011)
16. Nyberg, K.; Rueppel, A.R.: Message recovery for signature schemes based on the discrete logarithm problem. In: Advances in Cryptology (Eurocrypt '94). Lecture Notes in Computer Science, vol. 950, pp. 175–190. Springer, Berlin (1994)
17. Tseng, S.-F.; Hwang, M.-S.: Digital signature with message recovery and its variant based on elliptic curve discrete logarithm problem. Comput. Stand. Interfaces **26**, 61–71 (2004)
18. Lee, J.-S.; Chang, J.H.: Strong designated verifier signature scheme with message recovery, In: Proceedings of the Advanced Communication Technology, vol. 1, pp. 801–803. IEEE, Gangwon-Do (2007)
19. Saeednia, S.; Kremer, S.; Markowitch, O.: An efficient strong designated verifier signature scheme. In: Information Security and Cryptology - ICISC 2003. Lecture Notes in Computer Science, vol. 2971, pp. 40–54. Springer, Berlin (2004)
20. Lee, J.-S.; Chang, J.H.: Comment on Saeednia et al.'s strong designated verifier signature scheme. Comput. Stand. Interfaces **31**(1), 258–260 (2009)
21. Yang, F.-Y.; Liao, C-M.: A provably secure and efficient strong designated verifier signature scheme. Int. J. Netw. Secur. **10**(3), 220–224 (2010)
22. Shim, K.-A.: A strong designated verifier signature scheme tightly related to the LRSW assumption. Int. J. Comput. Math. **90**(2), 163–171 (2013)
23. Lysyanskaya, A.; Rivest, R.; Sahai, A.; Wolf, S.: Pseudonym systems. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 1758, pp. 184–199. Springer, Berlin (1999)
24. Kang, B.; Xu, H.; Niu, Y.: On delegatability of some strong designated verifier signature schemes. Math. Probl. Eng. doi:10.1155/2014/761487 (2014)
25. Susilo, W.; Zhang, F.; Mu, Y.: Identity-based strong designated verifier signature schemes. In: Information Security and Privacy. Lecture Notes in Computer Science, vol. 3108, pp. 313–324. Springer, Berlin (2004)
26. Shamir, A.: Identity-based cryptosystems and signature schemes. In: Advances in Cryptology (Crypto '84). Lecture Notes in Computer Science, pp. 47–53. Springer, Berlin (1984)
27. Shao, Z.: Improvement of digital signature with message recovery and its variants based on elliptic curve discrete logarithm problem. Comput. Stand. Interfaces **27**, 61–69 (2004)
28. Zhang, F.; Susilo, W.; Mu, Y.: Identity-based partial message recovery signatures (or how to shorten ID-based signatures). In: Financial Cryptography and Data Security. Lecture Notes in Computer Science, vol. 3570, pp. 45–56. Springer, Berlin (2005)
29. Tso, R.; Gu, C.; Okamoto, T.; Okamoto, E.: An efficient ID-based digital signature scheme with message recovery. In: Cryptology and Network Security. Lecture Notes in Computer Science, vol. 4856, pp. 47–59. Springer, Berlin (2007)
30. Li, Y.; Chen, H.: Efficient identity-based signature scheme with partial message recovery. In: Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, vol. 01, pp. 883–888. IEEE, Qingdao (2007)
31. Kalkan, S.; Kaya, K.; Selcuk, A.A.: Generalized ID-based ElGamal signatures with message recovery. In: Proceedings of the Information Security and Cryptology Conference, pp. 1–6. IEEE, Istanbul (2007)
32. Boyen, X.: A tapestry of identity-based encryption: practical frameworks compared. Int. J. Appl. Cryptogr. **1**(1), 3–21 (2008)
33. Barreto, P.S.L.M.; Naehrig, M.: Pairing-friendly elliptic curves of prime order. In: Selected Areas in Cryptography. Lecture Notes in Computer Science, vol. 3897, pp. 319–331. Springer, Berlin (2006)
34. Devegili, A.J.; Scott, M.; Dahab, R.: Implementing Cryptographic Pairings over Barreto–Naehrig Curves. Cryptology ePrint archive, report 2007/390. International Association for Cryptologic Research. https://eprint.iacr.org/2007/390.pdf
35. Galbraith, S.D.; Paterson, K.G.; Smart, N.P.: Pairings for cryptographers. Discret. Appl. Math. **156**, 3113–3121 (2008)
36. Lipmaa, H.; Wang, G.; Bao, F.: Designated verifier signature schemes: attacks, new security notions and new construction. In: Automata, Languages and Programming. Lecture Notes in Computer Science, vol. 3580, pp. 459–471. Springer, Berlin (2005)
37. Damgard, I.; Fujisaki, E.: A statistically-hiding integer commitment scheme based on groups with hidden order. In: Advances on Cryptology (Asiacrypt '02). Lecture Notes in Computer Science, vol. 2501, pp. 125–142. Springer, Berlin (2002)
38. Pointcheval, D.; Stern, J.: Security arguments for digital signatures and blind signatures. J. Cryptol. **13**, 361–396 (2000)
39. Islam, S.H.; Biswas, G.P.: A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile network. Ann. Telecommun. **67**(11–12), 547–558 (2012)
40. Islam, S.H.; Biswas, G.P.: Provably secure and pairing-free certificateless digital signature scheme using elliptic curve cryptography. Int. J. Comput. Math. **90**(11), 2244–2258 (2013)
41. Cao, X.; Kou, W.; Du, X.: A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges. Inf. Sci. **180**(15), 2895–2903 (2010)
42. Solinas, J.A.: Generalized Mersenne Prime: Encyclopedia of Cryptography and Security, pp. 509–510, 2nd edn. Springer, New York (2011)