J. Appl. Math. & Informatics Vol. **31***(2013), No. 3 - 4, pp. 425 - 441*
Website: http://www.kcam.biz

AN EFFICIENT AND SECURE STRONG DESIGNATED VERIFIER SIGNATURE SCHEME WITHOUT BILINEAR PAIRINGS[†]

SK HAFIZUL ISLAM* AND G. P. BISWAS

Abstract. In literature, several strong designated verifier signature (SDVS) schemes have been devised using elliptic curve bilinear pairing and map-to-point (MTP) hash function. The bilinear pairing requires a super-singular elliptic curve group having large number of elements and the relative computation cost of it is approximately two to three times higher than that of elliptic curve point multiplication, which indicates that bilinear pairing is an expensive operation. Moreover, the MTP function, which maps a user identity into an elliptic curve point, is more expensive than an elliptic curve scalar point multiplication. Hence, the SDVS schemes from bilinear pairing and MTP hash function are not efficient in real environments. Thus, a cost-efficient SDVS scheme using elliptic curve cryptography with pairing-free operation is proposed in this paper that instead of MTP hash function uses a general cryptographic hash function. The security analysis shows that our scheme is secure in the random oracle model with the hardness assumption of CDH problem. In addition, the formal security validation of the proposed scheme is done using AVISPA tool (Automated Validation of Internet Security Protocols and Applications) that demonstrated that our scheme is unforgeable against passive and active attacks. Our scheme also satisfies the different properties of an SDVS scheme including *strongness*, *source hiding*, *non-transferability* and *unforgeability.* The comparison of our scheme with others are given, which shows that it outperforms in terms of security, computation cost and bandwidth requirement.

AMS Mathematics Subject Classification : 94A60, 14H52, 11G05, 97P99.
Key words and phrases: Elliptic curve cryptography, random oracle model, designated verifier, signature, formal security, AVISPA tool.

Received October 12, 2012. Revised January 24, 2013. Accepted January 29, 2013. *Corresponding author. [†]This research work is supported by the Department of Science and Technology (DST), Govt. of India under the INSPIRE fellowship Ph.D program (Reg. No. IF10247) and the Department of Information Technology (DIT), Ministry of Communication and Information Technology, Govt. of India under the Information Security Education and Awareness (ISEA) program (Project No. MIT(2)/2006-08/189/CSE). The authors would also like to express their gratitude and heartiest thanks to the Department of Computer Science and Engineering, Indian School of Mines, Dhanbad, India for providing their research support, without which this work could not be carried out.

1. Introduction

The notion of designated verifier signature (DVS) schemes was proposed by Jakobsson et al. [1]. In a DVS scheme, a signer *Alice* generates a signature and the legitimacy of it can be verified by the designated verifier *Bob*, a specified user, but he/she cannot prove to a third-party *Cindy* that *Alice* signs the message. It happens since *Bob* can create another valid signature designated for him, which is indistinguishable from the signatures generated by *Alice*. Since the DVS scheme provides message authentication with the *non-repudiation*, it has many applications such as in software licensing, e-voting, call for tenders, electronic auction etc. In [1], Jakobsson et al. also introduced a signature scheme, called stronger designated verifier signature (SDVS). In an SDVS scheme, anyone who does not have the verifier's private key cannot verify the validity of the signature; since verifier's private key is strictly required in the verifying phase.

1.1. Related Works. In 2003, Sadeednia et al. [2] proposed an SDVS scheme based on public key infrastructure (PKI). However, Lee and Chang [3] pointed out that Sadeednia's signature can be verified not only using verifier's private key, but also with the singer's private key. If the signer's private is leaked and the signature is captured by an adversary before the verifier received it, then the adversary can verify the signature and convinced about the original signer. This could make the signer's identity revealed. After Sadeednia's work, several identity-based SDVS (ID-SDVS) schemes [4, 5, 6, 7, 8] have been proposed using identity-based cryptosystem (IBC) [9] and bilinear pairing [10]. To achieve strong security and to satisfies all the properties of an SDVS scheme, Zhang and Mao [11] proposed a novel ID-SDVS scheme based on bilinear pairings, which was broken by Kang et al. [12]. They demonstrated that Zhang-Mao's scheme fails to offer the *strongness* property of SDVS scheme since anyone who intercepts one signature can get some information and verify subsequent signatures. In the same paper [12], Kang et al. proposed a new and efficient ID-SDVS scheme to remove the abovementioned security flaw. In 2010, Lee et al. [13] demonstrated that signature scheme of Kang et al. [12] is vulnerable to a *universal forgery attack* and Kumar et al.'s scheme [7] lacks the *strongness* property of an SDVS scheme. In order to offer low computation and communication costs, Kang et al. [14] proposed a new SDVS scheme using elliptic curve and bilinear pairings. They claimed that their scheme provides the *strongness* and *unforgeability* properties; however Kang et al.'s scheme is susceptible to the *universal forgery attack* and does not have the property of *strongness* as proved by Du and Wen [15]. That is, in Kang et al.'s [14] scheme, an adversary can forge a signature on any message after having an old designated verifier signature. In 2012, Tian et al. [16] proposed an efficient *non-delegatable* SDVS scheme using elliptic curve cryptography [17, 18]. Also, the scheme achieves *non-delegatability* and *signer ambiguity* properties. They also prove that the scheme is secure in the random oracle model [19] provided the Computational Diffie-Hellman (CDH) problem is intractable by any polynomial time bounded algorithm. In 2013, Islam and

Biswas [20] devised a bilinear paring-based SDVS scheme using elliptic curve cryptography and certificateless public key cryptography (CL-PKC) [21], and shows that the scheme is unforgeable under the adaptive chosen message and identity attacks against various adversaries in the random oracle model based on the intractability of BDH (Bilinear Diffie-Hellman) and CDH assumptions [10]. Furthermore, they claimed that it satisfies different trust levels defined by Girault [22] and the necessary security properties of an SDVS scheme.

1.2. Motivations and Contribution. The SDVS schemes proposed in [4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16] used the elliptic curve bilinear pairings that requires approximately two to three times more multiplications in the underlying finite field than an elliptic curve scalar point multiplication does in the same field [23, 24]. In spite of the significant improvements in the computation speed, the bilinear paring is still regarded as the most expensive operation in cryptography. In addition, most of the SDVS schemes [4, 5, 6, 7, 8, 11, 12, 13, 14, 15, 16] need a time-consuming special hash function, called MTP hash function that converts a user's identity into an elliptic curve point. The MTP hash function is usually implemented as a probabilistic algorithm and is more expensive than an elliptic curve scalar point multiplication. Thus, in terms of computation efficiency, SDVS scheme without bilinear pairing and MTP hash function would be more appealing in practice. In this paper, we proposed a pairing-free PKI-based SDVS scheme using elliptic curve cryptography (ECC) and the general cryptographic hash function (e.g., SHA-1) instead of MTP hash function. The proposed scheme is secure in the random oracle model [19] with the hardness assumption of CDH problem and computation efficient than other schemes. In order to validate the formal security, we have designed our SDVS scheme in AVISPA tool [25, 26], which is a well-known and widely used strong security attack model checker. The simulation results of our scheme on AVISPA software confirmed that it can prevent attacks from both active and passive adversaries.

1.3. Organization of the Paper. The rest of the paper is organized as follows. In Section 2, we describe some preliminaries such as elliptic curve and computational problems on it. We present our pairing-free SDVS scheme in Section 3. The security and efficiency analyses of the proposed scheme are discussed in Section 4. Finally, the Section 5 concludes the paper.

2. Preliminaries

2.1. Elliptic Curve Cryptography. Let E/F_q denote the set of elliptic curve points $E_q(a,\, b)$ over the prime field F_q, defined by the non-singular elliptic curve equation:

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

with $x, y, a, b \in F_q$ satisfying the equation

$$\Delta = (4a^3 + 27b^2) \bmod q \neq 0 \tag{2}$$

The additive elliptic curve cyclic group defined as $G_q = \{(x, y) : x, y \in F_q$ and $(x, y) \in E/F_q\} \cup \{O\}$, where the point O is known as "*point at infinity*" and it acts as identity element of the group G_q.The scalar multiplication on G_q defined as $k{\cdot}P{=}P + P + \cdots + P$ (k times). A point P has order n if $n{\cdot}P{=}O$ for smallest integer $n{>}0$. Initially, the ECC was proposed by Miller [17] and Koblitz [18] in 1985. Compared with other public key cryptosystem, ECC-based public key cryptosystem has many advantages such as smaller key size, low computation cost, low storage space cost etc. It is known that the elliptic curve discrete logarithm problem (ECDLP) of a random elliptic curve element with respect to a publicly known base point is harder than the discrete logarithm problem (DLP) of a finite set of elements F_q. Besides, an ECC-based cryptosystem offers the same level of security compared to an RSA-based cryptosystem with a large modulus e.g., 160-bit ECC-based public key provides the same security to a 1024-bit RSA public key. The details of elliptic curve group and its properties are given in [27].

2.2. Computational Problems. The security of our signature scheme relies on the hardness of the CDH problem in the elliptic curve group and we briefly review this mathematical problem as given below.

Definition 2.1 (Computational Diffie-Hellman (CDH) Problem)**.** Given $(P, aP, bP) {\in} G_q$ for $a, b \in Z_q^*$, computation of abP is hard to the group G_q.

Definition 2.2 (Computational Diffie-Hellman (CDH) Assumption)**.** A probabilistic polynomial time bounded adversary $\mathscr{A}$ is said to break the CDH problem with negligible probability, if given an instance $(P, aP, bP) {\in} G_q$ of CDH problem, where $a, b \in Z_q^*$, then the advantage $Adv_{\mathscr{A},G_q}^{CDH} = Pr[\mathscr{A}(P, aP, bP) = abP : a, b \in Z_q^*]$ of $\mathscr{A}$ in solving CDH problem is negligible.

2.3. Definition of an SDVS Scheme. In this section, we define the formal security model of an SDVS scheme.

Definition 2.3. An SDVS scheme consists of five algorithms: *Setup*, *Extract*, *SDVS-Sign*, *SDVS-Verify* and *SDVS-Simulation.* Now we can describe these algorithms as follows.

Setup: This is a probabilistic polynomial time (PPT) algorithm that takes a security parameter $k \in Z^+$ as input and returns a list of system's parameter Ω, i.e., $\Omega \leftarrow Setup(k)$.
Extract: This is also a PPT algorithm runs by every user in the system, which takes system's parameter Ω and an identity ID_i of a user as inputs and outputs the private key x_i of ID_i i.e., $x_i \leftarrow Extract(ID_i, \Omega)$.
SDVS-Sign: This is a PPT algorithm runs by a signer ID_i, which takes a message $m_i \in \{0, 1\}^*$, private key x_i of ID_i and public key P_j of the designated verifier ID_j as inputs and returns a signature σ_i on the message m_i i.e., $\sigma_i {\leftarrow} SDVS\text{-}Sign(ID_i, ID_j, x_i, P_j, m_i)$.

SDVS-Verify: This is a deterministic polynomial time (DPT) algorithm runs by the designated verifier ID_j, which takes public key P_i of ID_i, private key x_j of ID_j, the signed message m_i and the signature σ_i as inputs, then it outputs either *accept* or *reject* as the verification decision e.g.,
$\{accept, reject\} \leftarrow SDVS\text{-}Verify(ID_i, ID_j, x_j, P_i, m_i, \sigma_i)$.
SDVS-Simulation: The designated verifier ID_j runs this PPT algorithm to generate identically distributed signatures that are indistinguishable from the signatures generated by the original signer. It takes public key P_i of ID_i, private key x_j of ID_j, and a signed message m_i as inputs, and then outputs a simulated signature $\hat{\sigma}$ i.e., $\hat{\sigma} \leftarrow SDVS\text{-}Simulation(ID_i, ID_j, x_j, P_i, m_i)$.

2.4. Security Properties of SDVS Scheme. In this section, we discuses different security properties of an SDVS scheme.
Correctness: Assume that the signer ID_i computes a signature σ_i on a message m_i for the designated verifier ID_j based on the signing algorithm *SDVS-Sign*, then ID_j must accepts the signature σ_i i.e., for $\Omega \leftarrow Setup(k)$, any $ID_i, ID_j \in \{0,1\}^*$, $x_i \leftarrow Extract(ID_i, \Omega)$, a message $m_i \in \{0,1\}^*$, let us assume that $\sigma_i \leftarrow SDVS\text{-}Simulation(ID_i,\ ID_j,\ x_i,\ P_j,\ m_i)$ and therefore
$SDVS\text{-}Verify(ID_i, ID_j, x_j, P_i, m_i, \sigma_i) = accept$ must holds.
Strongness: A properly designed strong designated verifier signature can be verified only by the designated verifier, but not by any third-party who does not have knowledge about the verifier's private key. This indicates that the private key of the designated verifier must be involved in the verifying phase i.e., if $x_j \neq x_j^*$ then $SDVS\text{-}Verify(ID_i, ID_j, x_j^*, P_i, m_i, \sigma_i) = reject$.
Source hiding: A strong designated verifier signature σ_i on a message m_i, the original signer or the designated verifier must not be identified by the third-party, even if all the private keys are known to him.
Non-transferability: The designated verifier can convinced the validity of the signature, but he could not transfer the conviction to any third-party. It means the designated verifier cannot prove to a third-party that the signature was produced by the real signer or designated verifier. This is because, the verifier is able to generate an indistinguishable signature from that one generated by the real signer.

Definition 2.4. An SDVS scheme is called *non-transferable* if the signature generated by the signer is indistinguishable from the signature simulated by the designated verifier that is,
$[\sigma \leftarrow SDVS\text{-}Sign(ID_i, ID_j, x_i, P_j, m_i) \approx \hat{\sigma} \leftarrow SDVS\text{-}Simulation(ID_i, ID_j, x_j, P_i, m_i)]$

Unforgeability: It is impossible for an adversary to construct a valid SDVS without having the private key of either the signer or the designated verifier.
The formal security model of SDVS scheme under the adaptively chosen message attack in the random oracle model is the following challenge-response game between the adversary $\mathscr{A}$ and a simulator $\mathscr{C}$.

- **Setup:** $\mathscr{C}$ runs this algorithm to generate the list of system's parameter Ω, which takes the security parameter $k \in Z^+$ as input and returns Ω to $\mathscr{A}$.
- **Extract queries:** $\mathscr{A}$ issues this query to $\mathscr{C}$ for the private key of a user whose identity ID_i, $\mathscr{C}$ then computes $x_i \leftarrow Extract(ID_i, \Omega)$ and returns the private key x_i to $\mathscr{A}$.
- **SDVS-Sign queries:** $\mathscr{A}$ submits this query with (ID_i, ID_j, m_i) to $\mathscr{C}$ to get a valid signature on the adaptively chosen message m_i, then $\mathscr{C}$ computes a signature σ_i on m_i, which is valid with respect to ID_i and ID_j, and returns it to $\mathscr{A}$.
- **SDVS-Verify queries:** To verify the signature σ_i of the message m_i, $\mathscr{A}$ submits a query of the form $(ID_i, ID_j, m_i, \sigma_i)$ to $\mathscr{C}$ and then $\mathscr{C}$ returns *accept* if σ_i is valid signature on m_i, and *reject* otherwise.
- **Forgery:** Finally, $\mathscr{A}$ outputs a forged signature σ_i^* on the message m_i^* with the signer's identity ID_i^* and the designated verifier's identity ID_j^* provided:
  - **(i)** $ID_i^* \neq ID_j^*$ holds.
  - **(ii)** $\mathscr{A}$ did not make any *Extract* query on ID_i^* and ID_j^*.
  - **(iii)** $\mathscr{A}$ did not make any *SDVS-Sign* query on m_i^* with ID_i^* and ID_j^*.
  - **(iv)** The signature σ_i^* on the message m_i^* is valid with respect to the signer's identity ID_i^* and the designated verifier's identity ID_j^* i.e., $SDVS\text{-}Verify(ID_i^*, ID_j^*, x_j^*, P_i^*, m_i^*, \sigma_i^*) = accept.$

The advantage of the polynomial time bounded adversary $\mathscr{A}$ to win this game is defined as $Adv_{SDVS,\mathscr{A}}^{CMA}(k)$.

Definition 2.5. The SDVS scheme is existentially unforgeable under the adaptive chosen message attack if a polynomial time bounded adversary $\mathscr{A}$ wins the above game with negligible advantage i.e., $Adv_{SDVS,\mathscr{A}}^{CMA}(k)$ is negligible.

3. Proposed Pairing-free SDVS Scheme

In this section, we proposed an efficient and pairing-free SDVS scheme from the elliptic curve cryptography. The bilinear pairing and the MTP hash function are not employed in our scheme, so that it can speed up in both signature generation and verification phases. The proposed scheme consists of the following algorithms.

3.1. Setup. On input of a security parameter $k \in Z^+$, this algorithm outputs system's parameter $\Omega = \{F_q, E/F_q, G_q, P, H_1\}$, where q denotes k-bit prime number, P is the base point of the group G_q and H_1 is a one-way and secure cryptographic hash function.

3.2. Extract. Assume that *Alice* with identifier ID_A is the signer and *Bob* with identifier ID_B is the designated verifier. To generate a private key, every user in the system executes this algorithm. The user ID_i for $i \in \{A, B\}$ picks a number $x_i \in_R Z_q^*$ as his private key and then computes the corresponding public key as $P_i = x_i P$.

3.3. SDVS-Sign. To generate a SDVS on the message $m \in \{0,1\}^*$ for the designated verifier *Bob*, *Alice* selects a number $r \in_R Z_q^*$ and performs the following operations:

- $U = rP_A$ (3)
- $h = H_1(m||U)$ (4)
- $\sigma = x_A(r+h)P_B$ (5)

Then *Alice* sends the signature (U, σ) with the message m to *Bob* for verification.

3.4. SDVS-Verify. Given a signature (U, σ), a message m, the private key x_B of *Bob*, public key P_A of *Alice*, respectively, *Bob* executes the following operations to *accept* or *reject* the signature.

- $\hat{h} = H_1(m||U)$ (6)
- $\hat{\sigma} = x_B(U + \hat{h}P_A)$ (7)
- If $\hat{\sigma} = \sigma$ holds, *Bob* accepts the signature (U, σ), otherwise rejects it.

3.5. SDVS-Simulation. To simulate a transcript on a given message m, *Bob* selects a number $\overline{r} \in_R Z_q^*$ and computes the following:

- $\overline{U} = \overline{r}P_A$ (8)
- $\overline{h} = H_1(m||\overline{U})$ (9)
- $\overline{\sigma} = x_B(\overline{r} + \overline{h})P_A$ (10)

Clearly, the simulated signature $(\overline{U}, \overline{\sigma})$ on the message satisfies the verifying equation.

4. Analysis of the Proposed SDVS Scheme

4.1. Security Analysis. This section provides the provable security analysis in the random oracle model [19] against the adaptive chosen message attack and formal security validation in AVISPA tool [25, 26] of our SDVS scheme. It can be noted that our scheme also satisfies *strongness*, *source hiding*, *non-transferability* and *unforgeability* properties.

Theorem 4.1. *The signer ID_A generates the signature (U, σ) on a given message m with the proposed SDVS scheme for the designated verifier ID_B is correct if the condition $\hat{\sigma} = \sigma$ holds.*

Proof. The correctness of the verification phase is proved as follows:
From the equations (4) and (6) we have

$$\begin{aligned} \hat{h} &= H_1(m||U) \\ &= h \end{aligned} \tag{11}$$

From the equations (7) and (11) we have,

$$\begin{aligned} \hat{\sigma} &= x_B(U + \hat{h}P_A) \\ &= x_B(rP_A + \hat{h}P_A) \ [\because U = rP_A, equ.(3)] \\ &= x_B(r + \hat{h})P_A \end{aligned}$$

$$
\begin{aligned}
&= x_B(r+\hat{h})x_AP \ [\because P_A = x_AP] \\
&= x_A(r+\hat{h})x_BP \\
&= x_A(r+\hat{h})P_B \ [\because P_B = x_BP] \\
&= x_A(r+h)P_B \ [\because \hat{h} = h, equ.(11)] \\
&= \sigma
\end{aligned}
$$

Therefore, we have $\hat{\sigma} = \sigma$. Thus, *Bob* accepts (U, σ) as valid signature of the message m. □

Lemma 4.2. *The following distributions* $S = (U, \sigma) = \left\{ \begin{array}{l} r \in_R Z_q^*, U = rP_A \\ h = H_1(m||U) \\ \sigma = x_A(r+h)P_B \end{array} \right\}$ *and* $\overline{S} = (\overline{U}, \overline{\sigma}) = \left\{ \begin{array}{l} \overline{r} \in_R Z_q^*, \overline{U} = \overline{r}P_A \\ \overline{h} = H_1(m||\overline{U}) \\ \overline{\sigma} = x_B(\overline{r}+\overline{h})P_A \end{array} \right\}$ *are identical.*

Proof. Let us choose a signature (U^*, σ^*) randomly from the set of all valid designated verifier signature of the *Alice* intended for *Bob*. Then the probability

$Pr[(U, \sigma) = (U^*, \sigma^*)] = \left\{ \begin{array}{l} r \in_R Z_q^*, U = rP_A = U^* \\ h = H_1(m||U) \\ \sigma = x_A(r+h)P_B = \sigma^* \end{array} \right\} = \frac{1}{q-1}$, because (U, σ) is generated with the random number $r \in_R Z_q^*$. Similarly, the probability $Pr[(\overline{U}, \overline{\sigma}) = (U^*, \sigma^*)] = \left\{ \begin{array}{l} \overline{r} \in_R Z_q^*, \overline{U} = \overline{r}P_A = U^* \\ \overline{h} = H_1(m||\overline{U}) \\ \overline{\sigma} = x_B(\overline{r}+\overline{h})P_A = \sigma^* \end{array} \right\} = \frac{1}{q-1}$, because the simulated signature $(\overline{U}, \overline{\sigma})$ is also depends on a random number $\overline{r} \in_R Z_q^*$. This proves that both the distributions are identical. □

Theorem 4.3. *The proposed SDVS scheme is non-transferable i.e., the designated verifier cannot convinced to a third-party that the signature* (U, σ) *of the message* m *was signed by the signer.*

Proof. The *non-transferability* property can be achieved in the proposed SDVS scheme. That is, *Bob* cannot prove to a third-party *Cindy* that the signature (U, σ) of the message m was generated by *Alice* or *Bob* himself. This is because *Bob* has the ability to produce an indistinguishable signature intended for him from the one generated by *Alice*. Let $(U^{''}, \sigma^{''})$ is a signature randomly chosen from the set of all valid signature created by *Alice* intended for *Bob*. Then from the lemma 4.2, we have $Pr[(U, \sigma){=}(U^{''}, \sigma^{''})]{=}1/(q-1)$ and $Pr[(\overline{U}, \overline{\sigma}){=}(U^{''}, \sigma^{''})]{=}1/(q-1)$. That is, the transcripts simulated by *Bob* are indistinguishable from those that he receives from *Alice*. □

Theorem 4.4. *The proposed SDVS is a strong designated verifier signature scheme i.e., the verification of a properly formatted signature (U, σ) on the message m may not be possible without designated verifier's private key.*

Proof. The proposed SDVS scheme satisfies the *strongness* property. To verify the signature (U, σ), *Bob*'s private key x_B must be used in the verification phase. Since for the verification purpose, *Bob* computes $U = rP_A$, $h = H_1(m||U)$ and $\sigma = x_A(r + h)P_A$, and then checks whether the validity condition $\hat{\sigma} = \sigma$ holds. However, it is impossible to compute $\hat{\sigma}$ without *Bob*'s private key x_B. Therefore, only *Bob* can verify the correctness of the signature (U, σ). □

Theorem 4.5. *The proposed SDVS scheme satisfies the source hiding property.*

Proof. The source hiding property states that, for a given signature (U, σ) of the message m, a third-party *Cindy* cannot identify the original signer *Alice* and the designated verifier *Bob* even if the private keys of them are known to *Cindy.* Assume that the private keys x_A of *Alice* and x_B of *Bob* are exposed to *Cindy.* However, *Cindy* has no ability to identify whether *Alice* or *Bob* has been produced the signature (U, σ) for the message m. This is because *Cindy* cannot identify whether x_A or x_B has been used in the construction of the terms $U = rP_A$ and $\sigma = x_A(r + h)P_A$ since he does not have the knowledge about the random number r. Therefore, to identify the signature (U, σ), *Cindy* has to know x_A and x_B as well as r those are used to generate the terms $U = rP_A = rx_AP$, $h = H_1(m||U)$ and $\sigma = x_A(r+h)P_B$. Besides, the random number r is protected under the ECDLP problem in U as well as the private key of the *Alice* or *Bob* in σ. □

Theorem 4.6. *If there is a polynomial time bounded adversary $\mathscr{A}$ that breaks our proposed SDVS scheme under the adaptively chosen message attack, then there must be an algorithm $\mathscr{C}$ that can solve the CDH problem in the random oracle model.*

Proof. Assume that the proposed SDVS scheme can be forged under the adaptive chosen message attack by a polynomial time adversary $\mathscr{A}$, then it is possible to construct a simulator $\mathscr{C}$ that helps $\mathscr{A}$ to solve an instance of CDH problem that is, $\mathscr{A}$ outputs abP from the input (P, aP, bP), where $a, b \in_R Z_q^*$.
Setup: To solve an instance of CDH problem, $\mathscr{C}$ sets (x_A=a, P_A=aP) and (x_B=b, P_B=bP) as the private/public key pairs of *Alice* and *Bob*, respectively, where $a, b \in_R Z_q^*$. Finally, $\mathscr{C}$ gives $\mathscr{A}$ the system's parameter $\Omega = \{F_q, G_q, P, P_A = aP, P_B = bP, H_1\}$ and responses to the queries made by $\mathscr{A}$ as follows:
Extract queries: For the private key of a user ID_i, $\mathscr{A}$ executes this query . If $\mathscr{A}$ asks an *Extract* query on ID_i, $\mathscr{C}$ then responds as given below:

$$x_i = \left\{ \begin{array}{ll} \perp & \text{for } ID_i = ID_A, ID_B \\ y_i & \text{otherwise, } y_i \in_R Z_q^* \end{array} \right\}$$

Hash queries to H_1: To respond with the H_1 queries, $\mathscr{C}$ maintains an initial-empty list L_{H1}^{list}. Each entry in the list L_{H1}^{list} is a tuple of the form (m_i, U_i, h_i).

For each query (m_i, U_i) issued by $\mathscr{A}$ to the oracle H_1, $\mathscr{C}$ replies the previous value defined in L_{H1}^{list}. Otherwise, $\mathscr{C}$ selects a number $h_i \in_R Z_q^*$ such that there is no item $(\cdot, \cdot, h_i)$ in L_{H1}^{list}, $\mathscr{C}$ sets $H_1(m_i||U_i) \leftarrow h_i$ and returns h_i to $\mathscr{A}$. Then, $\mathscr{C}$ includes the tuple (m_i, U_i, h_i) in the list L_{H1}^{list}.

SDVS-Sign queries: Suppose that, $\mathscr{A}$ submits a signature query for the signature of a message m_i with the signer's identity ID_i and the designated verifier's identity ID_j, then $\mathscr{C}$ first searches the list L_{H1}^{list} and then generates the signature as given below:

(i) If $ID_i \neq ID_A$ or ID_B, $\mathscr{C}$ selects a number $r_i \in_R Z_q^*$, computes the private keys $x_i = y_i$ and $x_j = y_j$ for ID_i and ID_j by executing the *Extract* algorithm and then performs the followings:
- Compute $U_i = r_i P_i$
- Compute $h_i = H_1(m_i||U_i)$
- Compute $\sigma_i = y_i(r_i + h_i)P_j$

(ii) If $ID_j \neq ID_B$ or ID_A, $\mathscr{C}$ selects a number $r_i \in_R Z_q^*$, computes the private keys $x_j = y_j$ and $x_i = y_i$ for ID_j and ID_i, respectively by executing the *Extract* query and then performs the followings:
- Compute $U_i = r_i P_i$
- Compute $h_i = H_1(m_i||U_i)$
- Compute $\sigma_i = y_j(r_i + h_i)P_i$

(iii) Otherwise, $\mathscr{C}$ outputs "failure" and aborts the protocol execution.

Finally, $\mathscr{C}$ returns a signature (U_i, σ_i) of the message m_i to $\mathscr{A}$ for the signer ID_i and the designated verifier ID_j.

SDVS-Verify queries: When $\mathscr{A}$ makes this query to $\mathscr{C}$ for the verification of the signature (U_i, σ_i) of the message m_i for the signer ID_i and the designated verifier ID_j, $\mathscr{C}$ then first verifies that whether $(ID_i, ID_j) = (ID_A, ID_B)$ or $(ID_i, ID_j) = (ID_B, ID_A)$ holds.

(i) If it holds, $\mathscr{C}$ terminates the simulation and reports "failure".

(ii) Otherwise, $\mathscr{C}$ recovers the private key x_j of ID_j and verifies the signature (U_i, σ_i) using the verification algorithm.

Forgery: Finally, $\mathscr{A}$ outputs a valid signature (U^*, σ^*) with on the message m^* with the signer's identity ID_i and the designated verifier's identity ID_j. If $ID_i = ID_A$ and $ID_j = ID_B$ (or $ID_i = ID_B$ and $ID_j = ID_A$), then $\mathscr{C}$ outputs the signature (U^*, σ^*) with h^* and m^*. According to the *forking lemma* [28], $\mathscr{C}$ finds a tuple (m_i, U_i, h_i) from the list L_{H1}^{list} and then replies with the same random tape, but different choices of hash value H_1. Therefore, $\mathscr{C}$ finds another valid signature $(U^*, \sigma^{'})$ with $h^{'}$ on the same message m^* such that $\sigma^* \neq \sigma^{'}$ and $h^* \neq h^{'}$. Since both (U^*, σ^*) and $(U^*, \sigma^{'})$ are valid signatures on the message m^*. Therefore, we can write

$$\sigma^* = x_B(U^* + h^* P_A) \tag{12}$$

$$\sigma^{'} = x_B(U^* + h^{'} P_A) \tag{13}$$

Subtracting the equations (12) and (13), we have

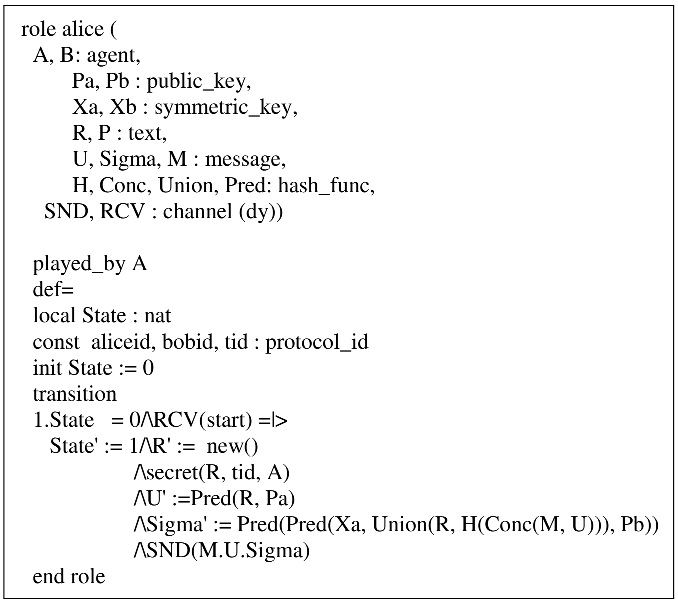
$$\begin{aligned}
\sigma^* - \sigma^{'} &= x_B(U^* + h^* P_A) - x_B(U^* + h^{'} P_A) \\
&= x_B(h^* - h^{'})P_A \\
&= b(h^* - h^{'})aP \\
&= (h^* - h^{'})abP \\
\Rightarrow abP &= (\sigma^* - \sigma^{'})/(h^* - h^{'})
\end{aligned}$$

Hence, $\mathscr{C}$ outputs the solution of the CDH problem as $abP = (\sigma^* - \sigma^{'})/(h^* - h^{'})$ that leads to a contradiction that the CDH problem is hard for any polynomial time bounded algorithm. Thus, we conclude that the proposed SDVS scheme is existentially unforgeable under the adaptive chosen message attack in the random oracle model. □

4.2. Formal Security Analysis using AVISPA Tool. Recently, AVISPA tool [25] is widely used by many researchers [29, 30, 31] for the automated validation of Internet security protocols and applications. The AVISPA is a push-button tool designed by University of Geneva, Italy using the concept of Dolev and Yao intruder model [32], where the network is controlled by an intruder (Active and passive); however he is not allowed to crack the underlying cryptography. The AVISPA tool supports High Level Protocol Specification Language (HLPSL) based on which the cryptographic protocols are to be implemented and analyzed. It has four model checkers/back-ends, called OFMC (On-the-fly Model-Checker), CL-AtSe (Constraint-Logic-based Attack Searcher), SATMC (SAT-based Model-Checker) and TA4SP (Tree Automata-based Protocol Analyzer). The back-end OFMC is helpful for the verification of the protocols and detection of attacks. The back-end CL-AtSe also can detect the attacks on the protocol with the help of a set of constraints which are obtained after the translation of the protocol specification written in Intermediate Format (IF). The back-end SATMC explores the state space of the protocol using symbolic techniques. The back-end TA4SP approximates the intruder knowledge (over or under) with an unbounded number of sessions using propositional formula and regular tree languages. The details description about AVISPA and HLPSL can be found in [26]. The role specification of the signer and the designated verifier of our SDVS scheme are implemented using HLPSL and these are shown in Tables 1 and 2. The simulation results using OFMC and CL-AtSe back-ends are given in Tables 3 and 4, which validates that the proposed SDVS scheme is unforgeable against both the passive and active adversaries.

4.3. Performance Analysis. In this section, we give a performance comparison of our pairing-free SDVS scheme with other relevant schemes (i.e., based on elliptic curve) including Susilo et al. [4], Yang et al. [6], Kumar et al. [7], Wang [8], Zhang-Mao [11], Kang et al. [12], Lee et al. [13], Kang et al. [14], Tian et al. [16] and Islam-Biswas [20] in terms of computation cost and signature length. All

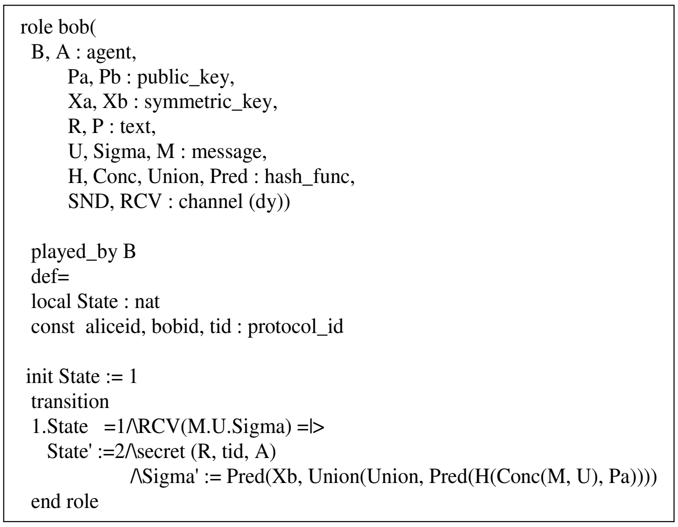
Table 1. Role specification of the signer.



```
role alice (
  A, B: agent,
        Pa, Pb : public_key,
        Xa, Xb : symmetric_key,
        R, P : text,
        U, Sigma, M : message,
        H, Conc, Union, Pred: hash_func,
    SND, RCV : channel (dy))

  played_by A
  def=
  local State : nat
  const  aliceid, bobid, tid : protocol_id
  init State := 0
  transition
  1.State   = 0/\RCV(start) =|>
     State' := 1/\R' :=  new()
                  /\secret(R, tid, A)
                  /\U' :=Pred(R, Pa)
                  /\Sigma' := Pred(Pred(Xa, Union(R, H(Conc(M, U))), Pb))
                  /\SND(M.U.Sigma)
  end role
```

Table 2. Role specification of the designated verifier.



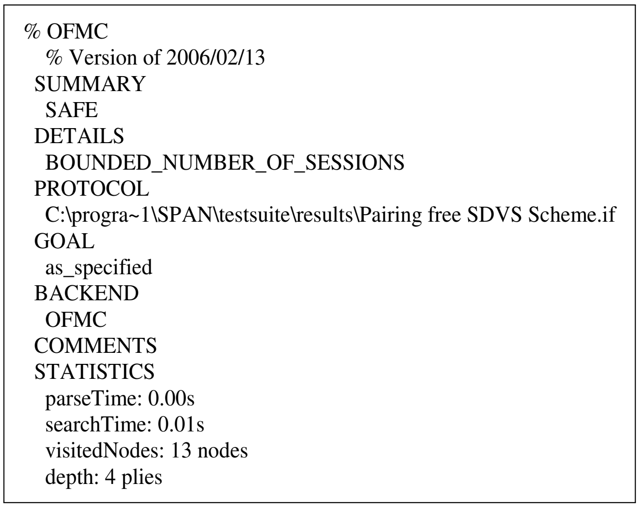
```
role bob(
  B, A : agent,
        Pa, Pb : public_key,
        Xa, Xb : symmetric_key,
        R, P : text,
        U, Sigma, M : message,
        H, Conc, Union, Pred : hash_func,
        SND, RCV : channel (dy))

  played_by B
  def=
  local State : nat
  const  aliceid, bobid, tid : protocol_id

 init State := 1
  transition
  1.State   =1/\RCV(M.U.Sigma) =|>
     State' :=2/\secret (R, tid, A)
                  /\Sigma' := Pred(Xb, Union(Union, Pred(H(Conc(M, U), Pa))))
  end role
```

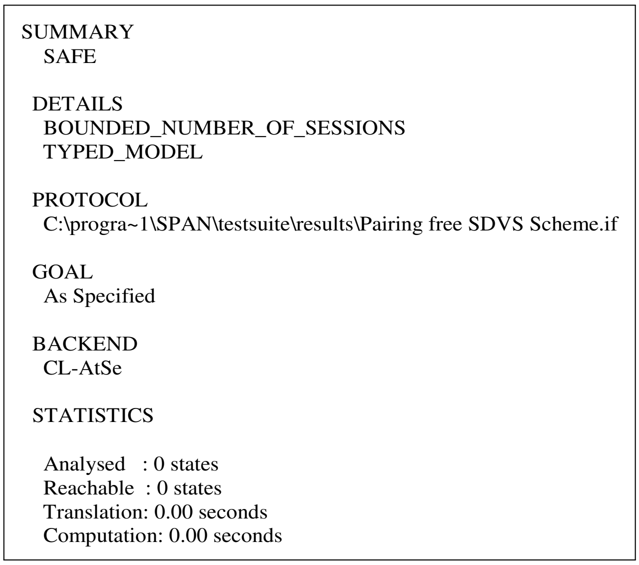
the above mentioned schemes used a bilinear paring $\hat{e} : G_q \times G_q \rightarrow G_m$, where G_q is an additive elliptic curve cyclic group of prime order q and G_m is another multiplicative cyclic group of order q. Let us suppose that the bit length of the

Table 3. Simulation result of our scheme on OFMC model checker.



```
% OFMC
  % Version of 2006/02/13
 SUMMARY
  SAFE
 DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
 PROTOCOL
  C:\progra~1\SPAN\testsuite\results\Pairing free SDVS Scheme.if
 GOAL
  as_specified
 BACKEND
  OFMC
 COMMENTS
 STATISTICS
  parseTime: 0.00s
  searchTime: 0.01s
  visitedNodes: 13 nodes
  depth: 4 plies
```

Table 4. Simulation result of our scheme on CL-AtSe model checker.



```
SUMMARY
  SAFE

 DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

 PROTOCOL
  C:\progra~1\SPAN\testsuite\results\Pairing free SDVS Scheme.if

 GOAL
  As Specified

 BACKEND
  CL-AtSe

 STATISTICS

  Analysed   : 0 states
  Reachable  : 0 states
  Translation: 0.00 seconds
  Computation: 0.00 seconds
```

elements in G_q is $|G_q|$ (assume that $|G_q|=|G_m|$). Now we define and consider the time complexity of different operations and their relationships [33, 34, 35, 36] given in Table 5. The comparative results of the proposed scheme with others are provided in Table 6, which demonstrated that our scheme has less computation cost in both signature generation and verification phases. Therefore, it is clear that the proposed SDVS scheme can substantially raise the efficiency of the signature generation and verification phases.

Table 5. Definition of different cryptographic operations.

Notations	Descriptions
T_{ML}	Time complexity for executing the modular multiplication
T_{EX}	Time complexity for executing the modular exponentiation, $1T_{EX} \approx 240T_{ML}$
T_{EM}	Time complexity for executing the elliptic curve scalar point multiplication, $1T_{EM} \approx 29T_{ML}$
T_{BP}	Time complexity for executing the bilinear pairing operation, $1T_{BP} \approx 3T_{EM} \approx 87T_{ML}$
T_{PX}	Time complexity for executing pairing-based exponentiation, $1T_{PX} \approx 1/2T_{BL} \approx 43.5T_{ML}$
T_{MTP}	Time complexity for executing the map-to-point function, $1T_{MTP} \approx 1T_{EM} \approx 29T_{ML}$
T_{IN}	Time complexity for executing the modular inversion operation, $1T_{IN} \approx 11.6T_{ML}$
T_{EA}	Time complexity for executing the addition of two elliptic curve points, $1T_{EA} \approx 0.12T_{ML}$
T_H	Time complexity for executing the simple hash function, which is negligible

Table 6. Comparison of the proposed scheme with others.

Scheme	Length	Signing Cost	Verifying Cost	Total	Assumption
Susilo et al. [4]	$2\|G_q\|+\|Z_q\|$	$T_{BP} + T_{PX} + 2T_{EM} + T_{MTP} + T_{IN} \approx 229.1T_{ML}$	$2T_{BP} + 2T_{PX} + T_{EM} + T_{MTP} \approx 321T_{ML}$	$550T_{ML}$	BDH
Yang et al. [6]	$3\|G_q\|$	$T_{BP} + T_{MTP} + 3T_{EM} + T_{IN} \approx 214.6T_{ML}$	$3T_{BP} + T_{PX} + T_{MTP} \approx 333.5T_{ML}$	$548T_{ML}$	BDH
Kumar et al. [7]	$4\|G_q\|$	$T_{BP} + 5T_{EM} + T_{MTP} + T_{IN} + T_{EA} \approx 272.72T_{ML}$	$4T_{BP} + T_{MTP} \approx 377T_{ML}$	$650T_{ML}$	BDH
Wang [8]	$4\|G_q\|$	$2T_{BP} + T_{MTP} + 5T_{EM} \approx 348T_{ML}$	$3T_{BP} + 2T_{EM} \approx 319T_{ML}$	$667T_{ML}$	GBDH
Zhang-Mao [11]	$3\|G_q\|$	$4T_{EM} + T_{MTP} + T_{IN} + T_{EA} \approx 156.72T_{ML}$	$3T_{BP} + T_{MTP} \approx 290T_{ML}$	$447T_{ML}$	BDH
Kang et al. [12]	$2\|G_q\|$	$T_{BP} + 2T_{EM} + T_{MTP} \approx 174T_{ML}$	$T_{BP} + T_{MTP} \approx 116T_{ML}$	$290T_{ML}$	BDH
Lee et al. [13]	$2\|G_q\|$	$2T_{BP} + T_{PX} + T_{MTP} + T_{EM} \approx 275.5T_{ML}$	$2T_{BP} + T_{MTP} \approx 203T_{ML}$	$278T_{ML}$	BDH
Kang et al. [14]	$2\|G_q\|$	$2T_{BP} + T_{PX} + 2T_{EM} + T_{MTP} \approx 304.5T_{ML}$	$T_{BP} + T_{PX} + T_{EM} + T_{MTP} \approx 188.5T_{ML}$	$493T_{ML}$	BDH
Tian et al. [16]	$3\|G_q\|$	$3T_{EM} + T_{MTP} \approx 116T_{ML}$	$3T_{EM} + T_{MTP} \approx 116T_{ML}$	$232T_{ML}$	BDH
Islam-Biswas [20]	$2\|G_q\|$	$1T_{BP} + 2T_{EM} + T_{PX} \approx 188.5T_{ML}$	$2T_{EM} + T_{PX} \approx 101.5T_{ML}$	$290T_{ML}$	BDH & CDH
Proposed	$2\|G_q\|$	$2T_{ML} + T_H \approx 58T_{ML}$	$2T_{EM} + T_{EA} + T_H \approx 58.12T_{ML}$	$116T_{ML}$	CDH

5. Conclusions

In this paper, we proposed a PKI-based computation efficient SDVS scheme based on elliptic curve cryptosystem. The main features are (1) The proposed scheme is rigorously analyzed in the random oracle model and it is proven to be provably secure against the adaptive chosen message attack, (2) It is secure based on a weak computational assumption, called CDH problem, whereas other relevant schemes are assumed to be secured with the strong computational assumption, named as BDH problem, (3) It is implemented on AVISPA tool for automated security validation and the simulation results proved that the active and passive attacks are prevented, (4) It employs general cryptographic hash function only and free from bilinear paring and MTP hash function, (5) It has low computation and communication costs, that is, the overall computation cost is $116T_{ML}$ and the size of the signature is $2|G_q|$, which are much lower than other existing schemes and (6) It has more applicability than the existing schemes, especially in the environments where low computation and communication costs are two main constraints. Owing the bilinear pairing and MTP hash function free realization, this paper efficiently devised a new elliptic curve-based strong designated verifier signature scheme; however, it needs a Public Key Infrastructure (PKI) to support the users' public key certificates for authentication of the public keys, thus, the storing and maintaining of the public key certificates are required in our proposed system. Although IBC can be implemented without using public key certificate, the present work using IBC may be further extended.

References

1. M. Jakobsson, K. Sako and R. Impagliazzo, *Designated verifier proofs and their applications*, In: Proceedings of the Advances in Cryptology (EUROCRYPT'96), LNCS, Springer-Verlag **1070**(1996), 143-154.
2. S. Saeednia, S. Kremer and O. Markowitch, *An efficient strong designated verifier signature scheme*, In: Proceedings of the Information Security and Cryptology (ICISC'03), LNCS, Springer-Verlag **2971**(2004), 40-54.
3. J-S. Lee, and J. H. Chang, *Comment on Saeednia et al.'s strong designated verifier signature scheme*, Computer Standards & Interfaces **31**(2009), 258-260.
4. W. Susilo, F. Zhang and Y. Mu, *Identity-based strong designated verifier signature schemes*, In: Proceedings of the Information Security and Privacy (ISP'04), LNCS, Springer-Verlag **3108**(2004), 313-324.
5. Q. Huang, G. Yang, D. S. Wong and W. Susilo, *Identity-based strong designated verifier signature revisited*, The Journal of Systems and Software **84**(2011), 120-129.
6. B. Yang, Z. Xia and Z. Hu, *A secure ID-based strong designated verifier signature scheme*, In: Proceedings of the International Conference on Network Infrastructure and Digital (2009), 543-547.
7. K. P. Kumar, G. Shailaja and A. Saxena, *Identity based strong designated verifier signature scheme*, Cryptography ePrint Archive Report 2006/134. Available at: http://eprint.iacr.org/complete/2006/134.pdf.

8. B. Wang, *A non-delegatable identity-based strong designated verifier signature scheme*, Cryptography ePrint Archive Report 2008/507. Available at: http://eprint.iacr.org/2008/507.pdf.
9. A. Shamir, *Identity-based cryptosystems and signature schemes*, In: Proceedings of the Advances in Cryptology (CRYPTO'84), LNCS, Springer-Verlag **196**(1984); 47-53.
10. D. Boneh and M. K. Franklin, *Identity-based encryption from the Weil pairing*, In: Proceedings of the Advances in Cryptology (CRYPTO'01), LNCS, Springer-Verlag **2139**(2001), 213-229.
11. J. Zhang and J. Mao, *A novel ID-based designated verifier signature scheme*, Information Sciences **178**(2008), 766-773.
12. B. Kang, C. Boyd and E. Dawson, *Identity-based strong designated verifier signature schemes: Attacks and new construction*, Computers & Electrical Engineering **35**(2009), 49-53.
13. J-S. Lee, J. H. Chang and D. H. Lee, *Forgery attacks on Kang et al.'s identity-based strong designated verifier signature scheme and its improvement with security proof*, Computers & Electrical Engineering **36**(2010), 948-954.
14. B. Kang, C. Boyd and E. Dawson, *A novel identity-based strong designated verifier signature scheme*, The Journal of Systems and Software **82**(2009), 270-273.
15. H. Du and Q. Wen, *Attack on Kang et al.'s Identity-Based Strong Designated Verifier Signature Scheme*, Cryptography ePrint Archive Report 2008/297. Available at: http://eprint.iacr.org/2008/297.pdf.
16. H. Tian, X. Chen, Z. Jiang and Y. Du, *Non-delegatable strong designated verifier signature on elliptic curves*, In: Proceedings of the Information Security and Cryptology (ISC'11), LNCS, Springer-Verlag **7259**(2012), 219-234.
17. V. S. Miller, *Use of elliptic curves in cryptography*, In: Proceedings of the CRYPTO'85, LNCS, Springer-Verlag **218**(1985), 417-426.
18. N. Koblitz, *Elliptic curve cryptosystem*, Journal of Mathematics of Computation **48**(1987), 203-209.
19. M. Ballare and P. Rogaway, *Random oracles are practical: a paradigm for designing efficient protocols*, In: Proceedings of the 1st ACM Conference on Computer and Communications Security (CCS'93), 62-73, 1993.
20. S. H. Islam and G. P. Biswas, *Provably secure certificateless strong designated verifier signature scheme based on elliptic curve bilinear pairings*, Journal of King Saud University-Computer and Information Sciences **25**(2013), 51-61.
21. S. Al-Riyami and K. Paterson, *Certificateless public key cryptography*, In: Proceedings of the ASIACRYPT'03, LNCS **2894**(2003), 452-473.
22. M. Girault, *Self-certified public keys*, In: Proceedings of the Advances in Cryptology (EUROCRYPT'91), LNCS, Springer-Verlag **547**(1992), 490-497.
23. P. Barreto, H. Kim, and B. Lynn, and M. Scott, *Efficient algorithms for pairing-based cryptosystems*, In: Proceedings of the Advances in Cryptology (CRYPTO'02), LNCS, Springer-Verlag **2442**(2002), 354-368.
24. P. Barreto, B. Lynn and M. Scott, *On the selection of pairing-friendly groups*, In: Proceedings of the Selected Areas in Cryptography (SAC'03), LNCS, Springer-Verlag **3006**(2004), 17-25.
25. AVISPA Web tool, *Automated Validation of Internet Security Protocols and Applications*, Available at http://www.avispa-project.org/web-interface/. (Accessed on January, 2013).
26. AVISPA, *The AVISPA User Manual* (2005). Available at http://www.avispa-project.org/publications.html.
27. D. Hankerson, A. Menezes and S. Vanstone, *Guide to elliptic curve cryptography*, 2nd edition, Springer-Verlag, New York, USA 2004.
28. D. Pointcheval and J. Stern, *Security arguments for digital signatures and blind signatures*, Journal of Cryptology **13**(2000) 361-396.

29. A. Basu, I. Sengupta and J. K. Sing, *Formal Security Verification of Secured ECC Based Signcryption Scheme*, In: Proceedings of the Advances in Computer Science, Engineering & Applications, LNCS, Springer-Verlag, **167**(2012), 713-725.
30. A. K. Das, *A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications*, Networking Science (2012). DOI: 10.1007/s13119-012-0009-8.
31. A. K. Das, A. Massand and S. Patil, *A novel proxy signature scheme based on user hierarchical access control policy*, Journal of King Saud University-Computer and Information Sciences (2013). DOI: 10.1016/j.jksuci.2012.12.001.
32. D. Dolev and A. C. Yao, *On the Security of Public-Key Protocols*, IEEE Transactions on Information Theory **2** (29) (1983), 198-208.
33. Y-F. Chung, K-H. Huang, F. Lai, and T. S. Chen, *ID-based digital signature scheme on the elliptic curve cryptosystem*, Computer Standards & Interfaces **29**(2007), 601-604.
34. N. McCullagh and P. S. L. M. Barreto, *A new two-party identity-based authenticated key agreement*, In: Proceedings of the Topics in Cryptology (CT-RSA'05), LNCS, Springer-Verlag **3376**(2005), 262-274.
35. X. Cao, W. Kou and X. Du, *A pairing-free identity-based authenticated key agreement protocol with minimal message exchanges*, Information Sciences **180**(2010), 2895-2903.
36. S. H. Islam and G. P. Biswas, *A pairing-free identity-based authenticated group key agreement protocol for imbalanced mobile networks*, Annals of Telecommunications **67**(11-12) (2012), 547-558.

SK Hafizul Islam is pursuing Ph.D in Computer Science and Engineering from Indian School of Mines Dhanbad, under the **INSPIRE Fellowship** (funded by DST, Govt. of India) and **Information Security Education and Awareness** (ISEA) program (funded by Department of Information Technology (DIT), Ministry of Communication and Information Technology, Govt. of India, No. MIT (2)/2006-08/189/CSE). Mr. Islam received his B.Sc (Hons.) in Mathematics and M.Sc in Applied Mathematics from Vidyasagar University, West Bengal, India in 2004 and 2006, and M.Tech from ISM Dhanbad in 2009, respectively. He has around four years of teaching and research experiences, and published around 15 research papers in Journals and Conference Proceedings of International reputes. His research interest includes Cryptography, Network/Information Security and Computer Networks.

Department of Computer Science & Engineering, Indian School of Mines, Dhanbad, Jharkhand-826004, India.
e-mail: hafi786@gmail.com, hafizul.ism@gmail.com

G. P. Biswas received B.Sc (Engg.) and M.Sc (Engg.) degrees in Electrical & Electronics Engineering and Computer Science & Engineering, respectively. He completed his PhD degree in Computer Science & Engineering from Indian Institute of Technology, Kharagpur, India. He is currently working as a Professor in the Department of Computer Science & Engineering, ISM Dhanbad, India. He has around 20 years of teaching and research experiences, and published around 90 research papers in Journals, Conferences and Seminar Proceedings. His main research interests include Cryptography, Computer Network and Security, Cellular Automata, VLSI Design.

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