Identity Based Strong Bi-Designated Verifier Proxy Signature Schemes

Sunder Lal and Vandani Verma
Department of Mathematics, Dr. B.R.A. (Agra), University,
Agra-282002 (UP), India.
E-mail- sunder_lal2@rediffmail.com, verma_vandani@rediffmail.com

Abstract: Proxy signature schemes allow delegation of signing rights. The paper proposes the notion of Identity Based Strong Bi-Designated Verifier Proxy Signature (ID-SBDVPS) schemes. In such schemes, only the two designated verifiers can verify that the proxy signer on behalf of the original signer signed the message but none of them is able to convince anyone else of this fact. The paper proposes nine such schemes and analyses the computational efficiency of each.

Keywords: ID based cryptography, designated verifiers, proxy signatures, Diffie Hellman problems, bilinear pairing, hash functions.

1. Introduction

Identity based cryptography was first proposed by Shamir [13] in 1984, to simplify key management procedure of certificate based public key infrastructure. In ID-PKC an entity's public key is derived from certain aspects of his identity and a trusted third party called a private key generator (PKG) generates secret keys for the entities. Since then many ID-based crypto primitives [1, 11] have been proposed one of them is a proxy signature. In 1996, Mambo et al [10] introduced the concept of proxy signatures. In such schemes an original signer delegates his signing authority to proxy signer in such a way that the proxy signer can sign any message on behalf of the original signer. In 1996, Jakobsson et al [4] introduced a new primitive called designated verifier signatures (DVS). In such schemes only the designated verifier can check the validity of the signatures. However, Saeednia et al [12] in 2003 introduced the concept of strong designated verifier signatures (SDVS), which forces the designated verifier to use his secret key at the time of verification. Since then several schemes [5] based on single designated verifier have been proposed. However, Desmedt [3] raised the problem of generalizing the designated verifier signature (DVS) concept to multi-designated verifier signatures. Laguillamie and Vergnaud [8] proposed the first Bi-designated verifier signatures scheme based on bilinear maps in 2004. Wang [14], Dia et al [2] and Lu and Cao [9], proposed designated verifier proxy signature schemes. Lal et al [6], also proposed ID-based strong designated verifier proxy signatures schemes. Lal et al [7], proposed the concept of ID based strong bi-designated verifier signature schemes.

In this paper, we combine the ideas of ID-based cryptography, strong designated verifier (with two verifiers) and proxy signatures and propose ID-based strong bi-designated verifier proxy signatures schemes. In such schemes, the original signer delegates her signing capability to proxy signer so that he can generate strong bi-designated verifier proxy signatures for the two designated verifiers. The signatures are generated in such a manner that only the designated verifiers can check the validity of the proxy signatures and they are unable to convince anyone else of this fact. In our schemes we do not require that the two designated verifiers to know each other. As an example consider a situation where Alice a corporate manager has to sign an important document with company XYZ but due to some urgent work he has to leave the station for a week. He gives his proxy signing capability to his assistant Bob. Bob on behalf of Alice generates the designated verifier signatures for company XYZ to be verified by their representatives Cindy and Tom

respectively. In this situation, it is not necessary that the two representatives Cindy and Tom know each other. The paper proposes nine ID-SBDVPS schemes based on this concept and also analyses the computational efficiency of these schemes.

Rest of the paper is organized as follows: in section 2 we briefly recall the concept of bilinear pairings and some related problems. In section 3, we present the phases of our proposed ID-SBDVPS schemes. In section 4, we present nine new ID-SBDVPS schemes and analyze the computational efficiency and security in section 5 and section 6 respectively. Finally, we conclude in section 7.

2. Background Concepts

In this section, we briefly review the concepts of bilinear pairings and some related mathematical problems.

2.1 Bilinear pairings

Let G_1 be a cyclic additive group generated by P, whose order is a large prime number q and G_2 be a cyclic multiplicative group with the same order q. Let e: $G_1 \times G_1 \rightarrow G_2$ be a map with the following properties:

***Bilinearity*:** $e(aP, bQ) = e(P, Q)^{ab}$ $\forall$ $P, Q \in G_1$ and $a, b \in Z_q^*$.

***Non-degeneracy*:** $\exists$ $P, Q \in G_1$, such that $e(P, Q) \neq 1$, the identity of G_2.

***Computability*:** There is an efficient algorithm to compute $e(P, Q)$ $\forall$ $P, Q \in G_1$.

Such pairings may be obtained by suitable modification in the Weil-pairing or the Tate-pairing on an elliptic curve defined over a finite field.

2.2 Computational problems

Here we present some computational hard problems, which form the basic security of our schemes.

***Discrete Logarithm Problem (DLP)*:** Given $Q \in G_1$, find an integer $a \in Z_q^*$, such that $Q = aP$, P is a generator of G_1.

***Decisional Diffie-Hellman Problem (DDHP)*:** Given P, aP, bP, cP in G_1, decide whether $c = ab \bmod q$.

***Computational Diffie-Hellman Problem (CDHP)*:** Given P, aP, bP, compute abP

Bilinear Diffie-Hellman Problem (BDHP): Given P, aP, bP, cP compute $e(P, P)^{abc}$.

***Gap Diffie-Hellman Problem (GDHP)*:** A class of problems, where DDHP can be solved in polynomial time but no probabilistic algorithm exists that can solve CDHP in polynomial time.

3. Phases of the proposed scheme:

Our proposed schemes are divided into following five phases.

- **Setup phase:** Given security parameters *k*, this phase outputs the public parameters.
- **Key generation phase:** Given a user identity and the public parameters, this phase computes the secret key of the user.
- **Proxy key generation:** Given original signers purported signatures and proxy signers secret key this phase computes proxy secret key.
- **Proxy signature generation:** Given proxy secret key, designated verifiers public key and random numbers this phase outputs a bi-designated verifier proxy signature.
- **Proxy signature verification:** On receiving the bi-designated verifier proxy signature, the private key of any of the designated verifiers, this phase tests the validity of the proxy signatures.

4. Description of ID-SBDVPS schemes

In this section we propose nine ID based SBDVPS schemes. These schemes are the extension of our previous work [6, 7]. We have introduced one more verifier to our earlier schemes proposed in [6] and the concept of proxy in schemes proposed in [7].

4.1. First ID-SBDVPS scheme

This scheme is the extension of scheme 4.1 proposed in [6]. The scheme works as follows:

1. **Setup:** In this phase, PKG chooses a generator $P \in G_1$, a random number $s \in Z_q^*$ and computes $P_{pub} = sP$. PKG also chooses two cryptographic hash functions H_1 and H_2, $H_1 : \{0,1\}^* \rightarrow Z_q^*$, and $H_2 : \{0,1\}^* \times G_2 \rightarrow G_1$. The system parameters (G_1, G_2, P, P_{pub}, H_1, H_2, e) are made public and s is kept secret with KGC.

2. **Key generation:** Given an identity ID_U, this phase generates $S_{IDU} = s^{-1} H_1(ID_U).P$ as the secret key and sends it to the user U in a secure manner. Thus, $Q_{IDU} = H_1(ID_U)$ is the public key of the user while $S_{IDU} = s^{-1} H_1(ID_U).P$ is the secret key of the user.

3. **Proxy key generation:** The original signer A generates the warrant m_w on message 'm' containing the identities of A, the proxy signer B, the designated verifiers C and D and the period of delegation. A generates the signature on message 'm' as follows:
   He chooses three random numbers $r_1, r_2, r_3 \in Z_q^*$
   computes $U_1 = r_1 Q_{IDB}.P$, $U_2 = r_2 Q_{IDA} P$, $U_3 = r_3 U_1$, $V = r_3 H + r_1^{-1} S_{IDA}$
   Here $H = H_2(m_w , e(P , S_{IDA})^{r_2 Q_{IDB}})$.
   A sends $\sigma = (m_w, U_1, U_2, U_3, V)$ to the proxy signer B.
   On receiving σ, B computes $H = H_2(m_w, e(U_2 , S_{IDB}))$.
   B accepts the signature iff $e(U_1 , V) = e(U_3 , H) e(S_{IDB} , P)^{Q_{IDA}}$.
   Now, B computes the proxy secret key $S_{IDP} = V + S_{IDB}$

4. **Proxy signature generation:** The proxy signer B computes the proxy signature on message 'm' as follows: B chooses three random numbers $t_1, t_2, t_3 \in Z_q^*$ and computes
   $R = Q_{IDC} Q_{IDT}$, $X_1 = t_1.R.P$, $X_2 = t_2 S_{IDP}$, $X_3 = t_3 X_1$, and $X = t_3 H^1 + t_1^{-1} S_{IDP}$
   Here $H^1 = H_2(m_w, e(X_2, P)^R)$.
   B sends $\sigma' = (m_w, R, X_1, X_2, X_3, X, V)$ to the designated verifiers C and T.

5. **Proxy signature verification:** On receiving σ' the two designated verifier C and T performs as follows:
   - Checks whether the message 'm' confirms to the warrant m_w. If not, stops Otherwise, continues.
   - Checks whether A and B are specified as the original signer and the proxy signer in the warrant m_w, respectively.
   - If all validation passes, C computes $H^1 = H_2(m_w, e(X_2 , P)^R)$, $Q_{IDT} = Q_{IDC}^{-1}.R$ and accepts the signature iff $e(X_1, X) = e(X_3, H^1) e(P, RV + Q_{IDB.} Q_{IDT}.S_{IDC})$
     Similarly, T can also verify the signatures as follows: Tom computes $H^1 = H_2(m_w, e(X_2 , P)^R)$, $Q_{IDC} = Q_{IDT}^{-1}.R$ and he accepts the signature iff $e(X_1, X) = e(X_3, H^1)\, e(P, RV + Q_{IDB.} Q_{IDC}.S_{IDT})$.
     But if the verification procedure fails then either C (or T) are not the designated verifiers or σ' is not correct.

6. Correctness:

$e(X_1, X)$

$= e(t_1.R.P , t_3H^1 + t_1^{-1}S_{IDP})$

$= e(t_1t_3.R.P, H^1)\ e(R.P , S_{IDP})$

$= e(t_1t_3.R.P , H^1)\ e(R.P , V + S_{IDB})$

$= e(X_3, H^1)\ e(P, RV)\ e(P , Q_{IDC}.\ Q_{IDT}.s^{-1}\ Q_{IDB}.P)$

$= e(X_3, H^1)\ e(P , RV)\ e(P , Q_{IDB}.\ Q_{IDT}.S_{IDC})$

$= e(X_3, H^1)\ e(P, RV + Q_{IDB}.Q_{IDT}.S_{IDC})$

A similar correctness equation for T verification equation can also be given.

4.2. Second ID-SBDVPS scheme

This scheme is based on the scheme 4.2 [6] and formed by introducing the concept of bi-designated verifier in the scheme.

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the two cryptographic hash functions H_1 and H_2. $H_1: \{0,1\}^* \rightarrow Z_q^*$ and $H_2: \{0,1\}^* \times G_1 \rightarrow Z_q^*$

2. **Key Generation:** Same as previous scheme.

3. **Proxy key generation phase:** A chooses two random numbers $r_1, r_2 \in Z_q^*$
   Computes $U = e(P, P)^{r_1 Q_{IDB}}$, $V = r_1 r_2^{-1} P - H_1(m_w) S_{IDA}$
   He sends $\sigma = (m_w, r_2, U, V)$ to B.
   On receiving σ B accepts the warrant iff $[e(V, P)^{Q_{IDB}}\ e(P, S_{IDB})^{H_1(m_w)Q_{IDA}}]^{r_2} = U$
   B computes the proxy secret key, $S_{IDP} = V + S_{IDB}$

4. **Proxy signature generation:** To generate a valid proxy signature on the message 'm' B chooses two random numbers $t_1, t_2 \in Z_q^*$ and computes
   $R = Q_{IDC}\ Q_{IDT}$, $X_1 = t_1^{-1}RP$, $X_2 = t_2\ Q_{IDB}.P$, $X_3 = H_2(m_w, X_1)$
   $X_4 = (t_2 + X_3)V$, $X = t_1(t_2 + X_3)\ S_{IDP}$
   He sends $\sigma' = (m_w, X_1, X_2, X_4, X)$ to the designated verifiers C and T.

5. **Proxy signature verification:** On receiving σ' the designated verifier C operates as follows:
   - Checks whether the message m confirms to the warrant m_w. If not, stops. Otherwise, continues.
   - Checks whether A and B are specified as the original signer and the proxy signer in the warrant m_w, respectively.
   - Computes $Q_{IDC} = Q_{IDT}^{-1}.R$, $X_3 = H_2(m_w, X_1)$ and accepts the signature iff
     $e(X_1, X) = e(P, X_4)^R\ e(S_{IDC}, X_2 + X_3\ Q_{IDB}.P)^{Q_{IDT}}$
     But if the verification procedure fails then either C is not the designated verifier or σ' is not correct.

   Similarly, T can verify the signatures using his secret key.

6. **Correctness:**

$e(X_1, X)$

$= e(t_1^{-1}RP, t_1(t_2 + X_3)\, S_{IDP})$

$= e(RP, (t_2 + X_3)(V + S_{IDB}))$

$= e(RP, (t_2 + X_3)\, V + (t_2 + X_3)S_{IDB})$

$= e(RP, X_4 + (t_2 + X_3)\, s^{-1} Q_{IDB}.P)$

$= e(P, X_4)^R\, e(s^{-1} Q_{IDC}\, P, (t_2 + X_3)\, Q_{IDB}\, Q_{IDT}\, P)$

$= e(P, X_4)^R\, e(S_{IDC}, t_2\, Q_{IDB}.P + X_3\, Q_{IDB.}P)^{Q_{IDT}}$

$= e(P, X_4)^R\, e(S_{IDC,}\, X_2 + X_3\, Q_{IDB.}P)^{Q_{IDT}}$

4.3. Third ID-SBDVPS scheme

The following scheme is the extension of scheme 4.3 [6]. In this scheme we have used three hash functions instead of two hash functions used in scheme 4.3 [6].

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the cryptographic hash functions H_1, H_2 and H_3 H_1: $\{0,1\}^* \rightarrow Z_q^*$, $H_2 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$ and $H_3 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$

2. **Key Generation:** Same as previous scheme.

3. **Proxy Key Generation:** To generate a proxy key S_{IDP}, the original signer A and the proxy signer B execute the following protocol jointly.
   - A chooses a random value $k_A \in_R Z_q^*$, computes $r_A = k_A.P$, and sends to B. Similarly, B chooses a random value $r_B = k_B.P$ and sends r_B to A.
   - On receiving r_B, A computes $r_p = r_A + r_B$, $y = H_2(m_w, r_p)$, $S_A = S_{IDA}.\, k_A\, .y$, and sends S_A to B.
   - Upon receiving (r_A , S_A), B computes $r_p = r_A + r_B$, $y = H_2(m_w, r_p)$ and checks whether $e(S_A , r_B)^{Q_{IDB}} = e(r_A , S_{IDB})^{Q_{IDA} k_B y}$. If all validation passes, B computes, $S_B = S_{IDB}\, .\, k_{B.}.y$, Then, computes $S_{IDP} = S_A + S_B$ as proxy secret key.

4. **Proxy Signature Generation:** To generate proxy signature on message 'm' B chooses two random numbers $t_1, t_2 \in Z_q^*$ and computes

   $R = Q_{IDC}\, Q_{IDT}$, $X_1 = e(P, P)^{t_1 R}$, $X_2 = H_3(m_w, X_1)$,

   $X_3 = (Q_{IDA}.r_A + Q_{IDB}.r_B).y$, $X = t_2^{-1} t_1\, P - X_2 S_{IDP}$.

   He sends $\sigma' = (m_w, R, X_1, X_3, X)$ to the designated verifier C and T.

5. **Proxy Signature Verification:** To verify the validity of the signatures the designated verifier C operates as follows:
   - Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
   - Checks whether A and B are specified as the original signer and the proxy signer in the warrant w, respectively.
   - Computes $Q_{IDT} = Q_{IDC}^{-1}.R$, $X_2 = H_3(m_w, X_1)$ and he accepts the signature iff

     $[e(X, P)^R\, e(X_{3,}\, S_{IDC})^{X_2 Q_{IDT}}]^{t_2} = X_1$

6. **Correctness:** The following equation gives the correctness of the signature equation for C

$[e(X, P)^R e(X_{3,} S_{IDC})^{X_2 Q_{IDT}}]^{t_2}$

$= [e(t_2^{-1} t_1 P - X_2 S_{IDP}, RP) e(X_2 X_{3,} Q_{IDT} S_{IDC})]^{t_2}$

$= [e(t_2^{-1} t_1 P - X_2 S_{IDP}, RP) e(X_2 X_{3,} Q_{IDT} s^{-1} Q_{IDC} P)]^{t_2}$

$= [e(t_2^{-1} t_1 P - X_2 S_{IDP}, RP) e(X_2 S_{IDP,} RP)]^{t_2}$

$= e(t_2^{-1} t_1 P_{,} RP)^{t_2}$

$= e(P, P)^{t_1 R}$

$= X_1$

4.4 Forth ID- SBDVPS scheme

In this section we proposed another new ID based SBDVPS scheme based on 4.4 [6]. The scheme works as follows:

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the two cryptographic hash functions H_1 and H_2. $H_1: \{0,1\}^* \rightarrow Z_q^*$ and $H_2: \{0,1\}^* \times G_1 \rightarrow G_1$

.

2. **Key Generation:** Same as the previous scheme.

3. **Proxy key generation phase:** A chooses a random numbers $r \in Z_q^*$ and computes $U = rP$, $V = r H_1(m_w) S_{IDA} + U$. He sends $\sigma = (m_w, U, V)$ to B. On receiving σ, B accepts the warrant iff $e(P_{pub}, V) = e(H_1(m_w).Q_{IDA}.P + P_{pub}, U)$. Then he computes the proxy secret key $S_{IDP} = V + S_{IDB}$.

4. **Proxy signature generation:** To sign a message 'm' B performs as follows:
   He chooses a random value $t \in Z_q^*$ and computes
   $R = Q_{IDC}.Q_{IDT,}$ $X_1 = tRP$, $X_2 = H_2(m_w, X_1)$, $X = t^{-1}(S_{IDP} + X_2)$.
   Sends $\sigma' = (m_w, V, X, X_1, X_2)$ to C and T.

5. **Proxy signature verification:** On receiving σ' the designated verifier C operates as follows:
   - Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
   - Checks whether A and B are specified as the original signer and the proxy signer in the warrant w, respectively.
   - Computes $Q_{IDT} = Q_{IDC}^{-1}.R$. He accepts the signature iff

     $e(X_1, X) = e(P, V + X_2)^R e(S_{IDC}, P)^{Q_{IDT} Q_{IDB}}$

   But if the verification procedure fails then either C is not the designated verifier or σ' is not correct. Similarly, T can verify the signatures using his secret key.

6. **Correctness:**

$e(X_1, X)$

$= e(tRP, t^{-1}(S_{IDP} + X_2))$

$= e(RP, S_{IDP} + X_2)$

$= e(Q_{IDC}.Q_{IDT,} P, V + S_{IDB} + X_2)$

$= e(P, V + X_2)^R e(S_{IDC}, P)^{Q_{IDT} Q_{IDB}}$

4.5 Fifth ID- SBDVPS scheme

This scheme is based on the scheme 4.1 [7]. This scheme is formed by introducing the concept of proxy signatures in 4.1[7]. The scheme works as follows:

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the two cryptographic hash functions H_1 and H_2. H_1: $\{0,1\}^* \rightarrow Z_q^*$ and H_2: $\{0,1\}^* \times G_2 \rightarrow Z_q^*$

2. **Key Generation:** Same as the previous scheme.

3. **Proxy key generation:** A chooses a random numbers $r \in Z_q^*$ and computes $U = r^{-1} Q_{IDB} P$, $V = r H_1(m_w) S_{IDA}$. He sends $\sigma = (m_w, U, V)$ to Bob. On receiving σ, B accepts the warrant iff $e(U, V) = e(S_{IDB}, P)^{Q_{IDA} H_1(m_w)}$. Then he computes the proxy secret key $S_{IDP} = V + S_{IDB}$.

4. **Proxy Signature generation:** B generates the signature on message 'm' as follows: chooses random numbers $(t_1,t_2) \in Z_q^*$ and computes
   $R = Q_{IDC} + Q_{IDT}$, $X_1 = e(S_{IDB}, P)^{Q_{IDT} Q_{IDC}}$, $X_2 = H_2(m_w, X_1^{t_2})$,
   $X_3 = t_1.Q_{IDB}.P$, $X_4 = V(X_2 - t_1R)$, $X = S_{IDP}(X_2 - t_1R)$,
   B sends (m, σ') as the signatures on message 'm' to the designated verifiers C and T where $\sigma' = (m_w, t_2, R, X_3, X_4, X)$.

5. **Proxy signature verification:** On receiving (m, σ'), the designated verifier C first computes the public key of the other designated verifier T from U_1 and then computes
   $X_1 = e(S_{IDC}, P)^{Q_{IDB} Q_{IDT}}$, $X_2 = H_2(m, X_1^{t_2})$.
   He accepts the signatures iff $e(X, P)^{Q_{IDC}} \; e(X_3, S_{IDC})^R = e(P, X_4 Q_{IDC} + X_2 Q_{IDB}.S_{IDC})$.
   But if the verification procedure fails then either C is not the designated verifier or σ' is not correct. Similarly, T can verify the signatures using his secret key.

6. **Correctness:** The following equation gives the correctness of the verification for the designated verifier C.
   $e(X, P)^{Q_{IDC}} \; e(X_3, S_{IDC})^R$
   $= e(S_{IDP}(X_2 - t_1R), Q_{IDC}.P) \; e(t_1 Q_{IDB} RP, S_{IDC})$
   $= e((V + S_{IDB})(X_2 - t_1R), Q_{IDC}.P) \; e(t_1 Q_{IDB} RP, S_{IDC})$
   $= e(V(X_2 - t_1R), Q_{IDC}.P) \; e(X_2 Q_{IDB} P - t_1 Q_{IDB}.RP, S_{IDC}) \; e(t_1 Q_{IDB} RP, S_{IDC})$
   $= e(X_4, Q_{IDC}.P) \; e(X_2 Q_{IDB}.P, S_{IDC})$
   $= e(P, X_4 Q_{IDC} + X_2 Q_{IDB}.S_{IDC})$.

4.6 Sixth ID- SBDVPS scheme

This scheme is based on the scheme 4.4 [7]. We have introduced the concept of proxy signatures in the scheme to form our sixth ID-SBDVPS scheme.

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the two cryptographic hash functions H_1 and H_2. H_1: $\{0,1\}^* \rightarrow Z_q^*$ and H_2: $G_1 \rightarrow Z_q^*$

2. **Key Generation:** Same as the previous scheme.

3. **Proxy key generation:** A chooses a random number r_1, $r_2 \in Z_q^*$ and computes
   $U = r_1.Q_{IDB}.P$, $V = r_1^{-1}[H_1(m_w).P + H_2(U).S_{IDA}]$.
   A sends $\sigma = (m_w, U, V)$ to B as the signatures on message 'm'.
   On receiving σ, B accepts the signatures iff $e(U, V) = e(P, P)^{Q_{IDB}H_1(m_w)}\, e(S_{IDB}, P)^{Q_{IDA}H_2(U)}$
   Then, he computes the proxy key as $S_{IDP} = V + H_2(U_1)\, S_{IDB}$, where $U_1 = r_2P$

4. **Proxy signature generation:** B chooses a random number $t \in Z_q^*$ and computes
   $R = Q_{IDC}.Q_{IDT}$, $X_1 = tRP$, $X = t^{-1}[H_1(m_w).P + H_2(X_1).S_{IDP}]$.
   B sends $\sigma' = (m_w, V, R, U_1, X_1, X)$ to C and T as the signatures on message 'm'.

5. **Proxy signature verification:** On receiving (m, σ), C first computes $Q_{IDC} = Q_{IDB}^{-1}$ R and accepts the signature iff $e(X_1, X) = e(P, [H_1(m_w)P + H_2(X_1)V].R)\, e(S_{IDC}, P)^{Q_{IDB}\, Q_{IDT} H_2(U_1)}$.
   Similarly, T can check the trueness of the signatures by using his secret key.

6. **Correctness:** The following equation gives the correctness of the scheme for C
   $e(X_1, X)$
   $= e(tRP, t^{-1}[H_1(m_w).P + H_2(X_1).S_{IDP}])$
   $= e(RP, H_1(m_w).P + H_2(X_1).S_{IDP})$
   $= e(P, P)^{RH_1(m_w)}\, e(P, V + H_2(U_1)\, S_{IDB})^{RH_2(X_1)}$
   $= e(P, P)^{RH_1(m_w)}\, e(P, V)^{RH_2(X_1)}\, e(s^{-1}Q_{IDC}P, Q_{IDT}.\, Q_{IDB.}P.H_2(R_1))$
   $= e(P, [H_1(m_w)P + H_2(X_1)V].R)\, e(S_{IDC}, P)^{Q_{IDB}\, Q_{IDT} H_2(U_1)}$
   Similar correctness equation can also be given for C.

4.7 Seventh ID- SBDVPS scheme

This phase proposes the extension of scheme 4.5 [7] to ID-SBDVPS scheme.

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the cryptographic hash functions H_1, H_2 and H_3. H_1: $\{0,1\}^* \rightarrow Z_q^*$ and $H_2 : \{0,1\}^* \times G_1 \rightarrow Z_q^*$ and $H_3 : \{0,1\}^* \times G_2 \rightarrow Z_q^*$

2. **Key Generation:** Same as previous scheme.

3. **Proxy key generation:** A chooses a random number r_1, $r_2 \in Z_q^*$ and computes $U_1 = r_1 Q_{IDA}\, P$, $U_2 = H_2(m_w, U_1)$, $V = (r_1 + U_2)\, S_{IDA}$
   A sends $\sigma = (m_w, U_1, V)$ as the signature on message 'm' to B.
   On receiving σ B computes $U_2 = H_2(m_w, U_1)$, $W = U_1 + U_2\, Q_{IDA}P$ and accepts σ iff $e(Q_{IDB}, V) = e(S_{IDB}, W)$. Then, he computes the proxy secret key as $S_{IDP} = r_2(S_{IDB} + V)$

4. **Proxy signature generation:** To generate signatures on message 'm' B computes $R = Q_{IDC}\, Q_{IDT}$, $X_1 = H_3(m_w, e(P , S_{IDP})^R)$, $X_2 = r_2\, (W + Q_{IDB}P)$, $X = X_1 S_{IDP}$
   B sends $\sigma' = (m_w, R, X_1, X_2, X)$ as the signature on message 'm' to C and T.

5. **Proxy signature verification:** On receiving (m, σ'), C first computes $Q_{IDT} = Q_{IDC}^{-1} R$ $X_1 = H_3(m_w, e(S_{IDC}, X_2)^{Q_{IDT}})$, and accepts the signature iff $e(P, X)^{Q_{IDB}} = e(S_{IDC}, V_2)^{X_1}$. Similarly, C can check the trueness of the signatures by using his secret key.

6. **Correctness:** The following equation gives the correctness of the scheme for C.
$e(P, X)^{Q_{IDC}}$
$= e(Q_{IDC}.P, X_1 S_{IDP})$
$= e(Q_{IDC}.P, X_1 r_2(S_{IDB} + V))$
$= e(Q_{IDC} P, X_1 r_2(S_{IDB} + (r + U_2)S_{IDA}))$
$= e(Q_{IDC} P, X_1 r_2 s^{-1}(Q_{IDB} P + r Q_{IDA}.P + U_2 Q_{IDA} P))$
$= e(S_{IDC}, X_1 r_2(Q_{IDB} P + W))$
$= e(S_{IDC}, X_1 V_2)$
$= e(S_{IDC}, V_2)^{X_1}$.
Similar correctness equation can also be given for Tom.

4.8 Eight ID- SBDVPS scheme

This phase proposes the extension of scheme 4.8 [7] to ID-SBDVPS scheme.

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the cryptographic hash functions H_1, H_2 and H_3. H_1: $\{0,1\}^* \rightarrow Z_q^*$ and H_2 : $\{0,1\}^* \times G_1 \rightarrow Z_q^*$ and H_3 : $\{0,1\}^* \times G_2 \rightarrow Z_q^*$

2. **Key Generation:** Same as previous scheme.

3. **Proxy key generation:** A chooses a random number $r \in Z_q^*$ and computes $U_1 = r^{-1}.Q_{IDB}.P$, $U_2 = r Q_{IDA} P$, $V = r. H_2(m_w, U_2). S_{IDA}$.
A sends $\sigma = (m_w, U_1, U_2, V)$ as the signature on message 'm' to B. On receiving σ, B checks whether $e(U_1, V) = e(S_{IDB}, P)^{Q_{IDA} H_2(m_w, U_2)}$. If true then, he computes the proxy secret key as $S_{IDP} = V + H_2(m_w, U_2). S_{IDB}$

4. **Proxy signature generation:** B generates the signature on message 'm' as follows: computes $R = Q_{IDC} Q_{IDT}$, $X_1 = e(P, S_{IDP})^R$, $X = H_3(m_w, X_1)$
B sends $\sigma' = (m_w, U_2, R, X)$ as the signature on message 'm' to C and T.

5. **Signature verification:** On receiving (m, σ'), C first computes $Q_{IDT} = Q_{IDC}^{-1} R$ and accepts the signature as valid signature on message 'm' iff
$X = H_3(m_w, e(S_{IDC}, U_2 + Q_{IDB}.P)^{H_2(m_w, U_2) Q_{IDT}})$
Similarly, T can check the validity of the signatures.

6. **Correctness:** The following equation gives the correctness of the scheme for C.
X
$= H_3(m_w, X_1)$
$= H_3(m_w, e(P, S_{IDP})^R)$
$= H_3(m_w, e(P, V + H_2(m_w, U_2). S_{IDB})^R)$
$= H_3(m_w, e(P, [r. H_2(m_w, U_2). S_{IDA} + H_2(m_w, U_2). S_{IDB}]^R)$

$= H_3(m_w, e(Q_{IDB}\, Q_{IDC}\, P, s^{-1}\, [rQ_{IDA}\, P + Q_{IDB}.P]^{H_2(m_w, U_2)})$

$= H_3(m_w, e(S_{IDC}, U_2 + Q_{IDB}.P)^{H_2(m_w, U_2) Q_{IDT}}$

Similar correctness equation can also be given for T.

4.9 Ninth ID- SBDVPS scheme

In this section we propose a new ID-SBDVPS scheme. This scheme is an independent scheme.

1. **Setup:** In this phase, the setup is same as the first proposed scheme except for the cryptographic hash functions H_1, and H_2 H_1: $\{0,1\}^* \rightarrow Z_q^*$ and H_2 : $\{0,1\}^* \times G_1 \rightarrow Z_q^*$.

2. **Key Generation:** Same as previous scheme.

3. **Proxy key generation:** The original signer A computes $V = H_1(m_w).S_{IDA}$ and sends $\sigma = (m_w, V)$ to B. B on receiving σ checks whether $e(V, P)^{Q_{IDB}} = e(P, S_{IDB})^{Q_{IDA} H_1(m_w)}$. If true then chooses a random value $r \in Z_q^*$ and computes the proxy secret key as follows: $S_{IDP} = r\, V + S_{IDB}\, H_1(m_w)$

4. **Proxy signature generation:** B generates the signature on message 'm' as follows: chooses $t \in Z_q^*$ and computes $R = Q_{IDC}\, Q_{IDT}$, $X_1 = t^{-1}RP$, $X_2 = r\, Q_{IDA}P$, $X = t\, H_2(m_w, X_1)S_{IDP}$. Sends $\sigma' = (m_w, R, X_1, X_2, X)$ as the signature on message 'm' to C and T.

5. **Proxy signature verification:** C verifies the proxy signature σ' as follows: computes $Q_{IDT} = Q_{IDC}^{-1}\, R$ and accepts the signature as valid signature on message 'm' iff
$e(X_1, X) = e(S_{IDC}, X_2 + Q_{IDB}\, P)^{H_1(m_w) H_2(m_w, X_1) Q_{IDT}}$

6. **Correctness:** The following equation gives the correctness of the scheme for C.
$e(X_1, X)$
$= e(t^{-1}RP, t\, H_2(m_w, X_1)S_{IDP})$
$= e(Q_{IDC}\, Q_{IDT}\, P, H_2(m_w, X_1)S_{IDP})$
$= e(Q_{IDC}\, Q_{IDT}\, P, H_2(m_w, X_1)\, (r\, V + S_{IDB}\, H_1(m_w))$
$= e(Q_{IDC}\, Q_{IDT}\, P, H_2(m_w, X_1)\, (r\, H_1(m_w).S_{IDA} + S_{IDB}\, H_1(m_w))$
$= e(Q_{IDC}\, Q_{IDT}\, P, H_2(m_w, X_1)\, H_1(m_w).\, s^{-1}\, (r\, Q_{IDA}\, P + Q_{IDB}P))$
$= e(S_{IDC}, X_2 + Q_{IDB}\, P)^{H_1(m_w) H_2(m_w, X_1) Q_{IDT}}$
Similar correctness equation can also be given for T.

5. Computational aspects:

We observe that the formation of the proposed schemes require the operations of hashing, multiplication, pairing evaluation, exponentiation and taking the inverse. In this section, we compare the proposed nine schemes discussed above by counting the number of the hash, multiplication, exponentiation, pairing and inverse required in signature generation and signature verification in each scheme. The following table gives the computational complexity of the schemes at a glance:

Proposed schemes	*Proxy Key Generation*					*Proxy Signature Generation*					*Proxy Signature Verification*				
	H	***M***	***P***	***E***	***I***	***H***	***M***	***P***	***E***	***I***	***H***	***M***	***P***	***E***	***I***
4.1	2	8	5	2	1	1	7	1	1	1	1	4	4	1	1
4.2	2	5	3	4	1	1	8	-	-	1	1	3	3	2	1
4.3	2	8	2	2	-	1	8	1	1	1	1	2	2	3	1
4.4	2	5	2	-	-	1	4	-	-	1	1	2	3	2	1
4.5	2	5	2	1	1	1	7	1	2	-	1	4	4	4	-
4.6	5	9	3	2	1	2	6	-	-	1	3	5	3	1	1
4.7	2	6	2	-	-	1	4	1	1	-	1	1	3	3	1
4.8	2	8	2	1	1	1	1	1	1	-	1	3	1	1	1
4.9	3	4	2	2	-	1	7	-	-	1	2	4	2	1	1

Here **H** = Hash, **M** = Multiplication, **E** = Exponential, **P** = Pairing, **I** = Inverse.

Proposed schemes	*Total computations in ID-SBDVS schemes*				
	Hash	***Multiplication***	***Pairings***	***Exponential***	***Inverse***
4.1	4	19	10	4	3
4.2	4	16	6	6	2
4.3	4	18	5	6	2
4.4	4	11	5	2	2
4.5	4	16	7	7	1
4.6	10	20	6	3	3
4.7	4	11	6	4	1
4.8	4	12	4	3	2
4.9	6	15	4	3	1

From the table it is clear that all the proposed schemes except 4.1, 4.2 and 4.6 require two pairings for proxy key generation. Scheme 4.8 and 4.9 requires equal number of pairings (4) and exponential (3) to produce an ID-SBDVS scheme. Scheme 4.4 requires least number of exponential operations and 4.5 requires least number of inverse computations. Scheme 4.8 is the most efficient scheme as compared to the others proposed schemes.

6. Security analyses:

In this section we analyze the security of the proposed ID-SBDVPS schemes.

6.1 Strongness: In each of the propose schemes proxy signatures are generated in such a manner that only the designated verifier can check the validity of the signatures using his secret key. Hence, our schemes provide the strongness property.

6.2 Proxy protected: Each of the proposed scheme is proxy protected, as the original signer cannot generate a valid proxy signatures on behalf of the proxy signer.

6.3 Unforgeability: It is not possible to construct proxy signatures without the knowledge of the proxy secret key and proxy secret key cannot be generated even knowing the secrets of the original signer and the proxy signer. Thus, the signatures are unforgeable.

6.4 Secrecy: In all the proposed schemes, the proxy key cannot be derived even knowing the secrets of the original signer and the proxy signer. Hence, our schemes are secure.

7. Conclusion:

In this paper, we have presented a new concept of Identity based strong bi-designated verifier proxy signature schemes and proposed nine schemes based on this concept. We have also analyzed the security of our schemes and the computational efficiency of each of the proposed schemes. Our proposed scheme in section 4.8 is computationally most efficient as compared to the other proposed schemes.

References:

1. **J.C.Cha, J.H Cheon.** An identity based signature from gap Diffie-Hellman groups. Public key cryptography PKC 2003, LNCS #2567, Springer-Verlag, 1990, 18-30.
2. **J. Dai, X.Yang, J.Dong.** Designated receiver proxy signature scheme for e-commerce. Proc.of IEEE International Conference on System, Man and Cybernetic, IEEE-2003, 384-389.
3. **Y.Desmedt.** Verifier-Designated Signatures, Rump Session, Crypto'03 (2003).
4. **M.Jakobsson, K.Sako, K.R.Impaliazzo.** Designated verifier proofs and their applications. Eurocrypt 1996, LNCS #1070, Springer-Verlag, 1996, 142-154.
5. **K.P Kumar, G.Shailaja, Ashutosh Saxena.** Identity based strong designated verifier signature scheme. Cryptography eprint Archive Report 2006/134. Available at http://eprint.iacr.org/2006/134.pdf
6. **Sunder Lal, Vandani Verma.** Identity based strong designated verifier proxy signature scheme. Cryptography eprint Archive Report 2006/394. Available at http://eprint.iacr.org/2006/394.pdf
7. **Sunder Lal, Vandani Verma.** Some identity based strong bi-designated verifier signature scheme. Cryptography eprint Archive Report 2007/193. Available at http://eprint.iacr.org/2007/193.pdf
8. **F.Laguillaumie, D.Vergnaud.** Multi-Designated Verifiers Signatures. ICICS 2004, LNCS #3269 Springer-Verlag, 2004, 495-507.
9. **R.Lu, Z.Cao.** Designated verifier proxy scheme with message recovery. Applied Mathematics and Computation, 169(2), 2005, 1237-1246.
10. **M. Mambo, K. Usuda, and E. Okamoto.** Proxy signatures, revisited, In Proc. Of ICICS' 97, LNCS 1334, Springer-Verlag, 1997, 223-232.
11. **K.G.Paterson.** ID-based signatures from pairings on elliptic curves. Cryptography eprint Archive Report 2006/134. Available at http://eprint.iacr.org/2002/004.pdf
12. **S.Saeednia, S.Kreme, O.Markotwich.** An efficient strong designated verifier signature scheme. ICICS 2003, LNCS #2971, Springer-Verlag, 2003, 40-54.
13. **A. Shamir.** ID based cryptosystems and signature scheme. Crypto'84, LNCS #196, Springer-Verlag, 1984, 47-53.
14. **G. Wang.** Designated verifier proxy signature for e-commerce. IEEE International Conferences on Multimedia and Expo (ICME 2004) CD-ROM, ISBN- 0-7803-8604-3, Taipei, Taiwan, 2004, 27-30.