Identity Based Strong Designated Verifier Proxy Signature Schemes

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

Abstract: The paper proposes four new ID based strong designated verifier proxy signature (SDVPS) scheme. The schemes are formed by introducing proxy in ID based SDVS, ID based in SDVPS and ID based proxy in SDVS. We have also analyzed the security of the schemes and their computation aspects.

Keywords: ID based cryptography, proxy signature, designated verifier, bilinear pairing, Diffie-Hellman problems, hash functions.

1. Introduction

Strong designated verifier proxy signature (SDVPS) is special type of proxy signatures in which the designated verifier alone can check the validity of the proxy signatures. Such signatures consist of three main phases – proxy key generation, proxy signature generation and proxy signature verification by the designated verifier. The first two phases are carried out on the concept of proxy signatures and the last phase is carried out on the outline of strong designated verifier signatures i.e in the first two phases the original signer Alice delegates her signing power to proxy signer Bob to generate proxy signatures for the designated verifier Cindy and in the last phase Cindy checks the validity of the proxy signatures by using his secret key. As an example consider a situation where Alice a corporate manager is on a vacation for one a week and in her absence some urgent business contract is to be signed with Cindy. So, she assigned Bob (her assistant manager) as her representative to negotiate the business contract with Cindy in this period. Bob signs the contract documents on behalf of Alice in such a manner that Cindy can only validate the corresponding signatures. Cindy uses his secret key to check the validity of the signatures. Another application of SDVPS is in on-line shopping. In such schemes primitives are proxy signature generation, strong designated verification and identity based. We introduce these primitives in three schemes to get three ID based SDVPS schemes. For example, by introducing proxy signature in Kumar's ID based strong designated verifier signature scheme we get one ID based SDVPS scheme. Similarly, by introducing 'proxy' and 'identity based' in Wang's scheme we get another ID based SDVPS scheme. We introduced 'proxy' and 'identity based' to Saeednia's scheme and get yet another ID based SDVPS scheme. Finally, we introduce one more ID based SDVPS scheme. We then, compare the computational efficiency of these schemes. We also make security analyses of these schemes.

Jakobsson et al [3] first proposed the designated verifier signatures (DVS) at Eurocrypt'96. Such signatures provide message authentication without non-repudiation and with the property that only one specified recipient could check their validity. These signatures have several applications such as E-voting and software licensing. Saeednia et al [7], introduced strong designated verifier signatures (SDVS), which forces the designated verifier to use his secret key at the time of verification. Saeednia's scheme is very efficient as

compared to Jakobsson et al [3] in terms of communication and computation. Rivest et al [6], introduced ring signatures that leads to DVS when the group size is reduced to two. However, these schemes cannot be SDVS. Dia et al [1], Lu and Cao [5] proposed designated verifier proxy signature schemes.

Shamir [8] first proposed the idea of ID based public key cryptography. The ID based public key systems allows some public information of the user such as name, address etc to be used as his public key. The private key of the user is calculated by a trusted party called key generating center (KGC) and sent to the user via a secure channel. In 2005, An ID based DVS scheme was proposed by Huang et al [2].

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 group of order a large prime number q and G_2 be a multiplicative subgroup of a finite field F of same order and P be a generator of G_1. A map e: $G_1 \times G_1 \rightarrow G_2$ is called a bilinear map if it has 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*:** $\forall$ P, Q $\in$ G_1 there is an efficient algorithm to compute e(P, Q).

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 basis 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, for a, b, c $\in$ Z_q^*, decide whether c = ab mod q.

***Computational Diffie-Hellman Problem (CDHP)*:** For any a, b $\in$ Z_q^*, given P, aP, bP, compute abP.

Bilinear Diffie-Hellman Problem (BDHP): Given P, aP, bP, cP, for a, b, c $\in$ Z_q^*, 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 time algorithm exists which can solve CDHP.

3. Phases of the proposed scheme:

Our proposed schemes are divided into five phases. Here we have assumed Alice as the original signer, Bob as the proxy signer and Cindy as the designated verifier.

- **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 (private) 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 designated verifier proxy signature.
- **Proxy signature verification:** On receiving the designated verifier proxy signature, the private key of the designated verifier, this phase tests whether proxy signatures are valid or not.

4. Description of SDVPS schemes

In this section we propose four ID based SDVPS schemes and also give the reviews of the schemes on which they are based.

4.1. SDVPS scheme derived from Kumar, Shailaja and Saxena [4]

In this section, firstly we shall give the review of Kumar's ID based strong designated verifier signature scheme and then the model of the new ID based SDVPS formed by using this scheme.

Review of the Kumar et al scheme

- **Setup:** In this phase, KGC chooses a generator $P \in G_1$, a random number $s \in Z_q^*$ and computes $P_{pub} = sP$. KGC also chooses two cryptographic hash functions H_1 and H_2
  $H_1 : \{0,1\}^* \rightarrow G_1$; $H_2 : \{0,1\}^* \times G_2 \rightarrow G_1$.
  The system parameters (G_1, G_2, P, P_{pub}, H_1, H_2, e) are published and s is kept secret with KGC.
- **Key generation:** Given an identity ID_U of a user U, this phase generates $Q_{IDU} = H_1(ID_U)$ as the public key of the user. Further, KGC computes and $S_{IDU} = sH_1(ID_U)$ as the secret key of the user and communicates through the secure channel.
- **Signature generation:** To generate signature on the message M which can be verified by the user B. T he signer A chooses three random numbers $r_1, r_2, r_3 \in Z_q^*$ and computes
  $U_1 = r_1 Q_{IDB}$, $U_2 = r_2 Q_{IDA}$
  $U_3 = r_3 U_1$, $V = r_3 H + r_1^{-1} S_{IDA}$ where $H = H_2(M, e(r_2 Q_{IDB}, S_{IDA}))$.
  Signer A sends (M, U_1, U_2, U_3, V) to the designated verifier B.
- **Signature verification:** On receiving (M, U_1, U_2, U_3, V) the designated verifier B computes $H = H_2(M, e(U_2, S_{IDB}))$
  B accepts the signature iff $e(U_1, V) = e(U_3, H)e(S_{IDB}, Q_{IDA})$.

In this scheme, the secret key S_{IDB} is used in the verification phase, therefore only the designated verifier can check the validity of the proxy signature. We now modify this scheme to get our first ID-based SDVPS scheme. In this scheme, the first two phases are carried out at key generation centre (KGC), the third phase is carried out jointly by the original signer and the proxy signer, the forth and the fifth phases are carried out by the proxy signer and the designated verifier respectively. As defined earlier, G_1 denotes a group of order a large prime number q, G_2 is a multiplicative subgroup of a finite field F of same order and $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear pairing.

Proposed scheme

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

2. **Key generation:** Given an identity, this phase generates $S = sH_1$ 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 $S_{IDU} = sH_1(ID_U)$ is the secret (private) key of the user.

3. **Proxy key generation:** The original signer Alice computes the signature on message M as follows: Alice chooses three random numbers $r_1, r_2, r_3 \in Z_q^*$ and a warrant W and computes $U_1 = r_1Q_{IDB}$, $U_2 = r_2Q_{IDA}$, $U_3 = r_3U_1$ and $V = r_3H + r_1^{-1}S_{IDA}$
   Here $H = H_2(M, W, e(r_2Q_{IDB}, S_{IDA}))$.
   Alice sends $\sigma = (M,W,U_1,U_2,U_3,V)$ to the proxy signer Bob.
   On receiving σ Bob computes $H = H_2(M, e(U_2, S_{IDB}))$.
   Bob accepts the signature iff $e(U_1, V) = e(U_3, H)e(S_{IDB}, Q_{IDA})$.
   Now, Bob computes the proxy secret key $S_{IDP} = V + S_{IDB}$

4. **Proxy signature generation:** The proxy signer Bob computes the proxy signature on message M as follows: Bob chooses three random numbers $t_1, t_2, t_3 \in Z_q^*$ and computes
   $X_1 = t_1Q_{IDC}$, $X_2 = t_2S_{IDP}$, $X_3 = t_3X_1$, and $X = t_3H^1 + t_1^{-1}S_{IDP}$
   Here $H^1 = H_2(M, W, e(t_2Q_{IDC}, S_{IDP}))$.
   Bob sends (M, W, X_1, X_2, X_3, X, V) to the designated proxy verifier Cindy.

5. **Proxy signature verification:** On receiving (M, W, X_1, X_2, X_3, X, V) the designated verifier Cindy performs as follows:
   - Checks whether the message M confirms to the warrant W. If not, stops Otherwise, continues.
   - Checks whether Alice and Bob are specified as the original signer and the proxy signer in the warrant W, respectively.
   - If all validation passes, Cindy computes $H^1 = H_2(M, W, e(X_2, Q_{IDC}))$
     Cindy accepts the signature iff $e(X_1, X) = e(X_3, H^1)e(S_{IDC}, Q_{IDB})e(Q_{IDC}, V)$

6. **Correctness:**

$$
\begin{aligned}
e(X_1, X) &= e(t_1Q_{IDC}, t_3H^1 + t_1^{-1}S_{IDP}) \\
&= e(t_1Q_{IDC}, t_3H^1)\, e(t_1Q_{IDC}, t_1^{-1}S_{IDP}) \\
&= e(t_1t_3Q_{IDC}, H^1)\, e(Q_{IDC}, V + S_{IDB}) \\
&= e(X_3, H^1)\, e(Q_{IDC}, V)\, e(Q_{IDC}, S_{IDB}) \\
&= e(X_3, H^1)\, e(Q_{IDC}, V)\, e(S_{IDC}, Q_{IDB})
\end{aligned}
$$

4.2. SDVPS scheme derived from Saeednia et al [7] scheme

Saeednia et al scheme is SDVS scheme based on the discrete logarithmic problem. The scheme forces the designated verifier to use his secret key at the time of verification.

Review of Saeednia et al scheme

- **Setup:** A large prime p, a prime factor p-1 , a generator $g \in Z_q^*$ of order q and a one way hash function h are assumed to be some common parameters initially shared between the users.

- **Key generation:** Each user i chooses a secret key $x_i \in Z_q$ and the corresponding public key $y_i = g^{x_i} \bmod p$ is made public.
- **Signature generation:** To sign a message m for Bob, Alice selects two random numbers k, $t \in Z_q$ and computes $c = y_b^k \bmod p$, $r = h(m, c)$, $s = kt^{-1} - rx_a \bmod q$
  Alice sends (r, s, t) as signature on the message m to Bob.
- **Signature verification:** Bob accepts (r, s, t) as Alice signature on the message m iff
  $h(m, (g^s y_a^r)^{tx_b} \bmod p) = r$

Now, we introduce the two primitives that of 'proxy' and of 'identity based' to the above scheme to form our second ID-based SDVPS scheme.

Proposed scheme:

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

2. **Key Generation:** A user U identity ID_U, generates $Q_{IDU} = H_1(ID_U)$ as his public key and $S_{IDU} = sQ_{IDU}$ as the secret key.

3. **Proxy key generation phase:** Alice chooses two random numbers $t \in Z_q^*$ and $k \in Z_q$
   Computes $r = e(P, Q_{IDB})^k$, $U = H_1[rH_1(m, w)]$, $V = t^{-1}kP - U S_{IDA}$
   He sends σ = (m, w, t, r, U, V) to Bob.
   On receiving σ Bob computes $U = H_2[rH_1(m, w)]$ and accepts the warrant iff
   $[e(V, Q_{IDB})\, e(Q_{IDA}, S_{IDB})^U]^t = r$
   Bob computes the proxy secret key, $S_{IDP} = V + S_{IDB}$

4. **Proxy signature generation:** To generate a valid proxy signature on the message m Bob chooses two random numbers $t_1 \in Z_q^*$ and $x \in Z_q$. Computes
   $U_1 = t_1^{-1} Q_{IDC}$, $U_2 = x\, Q_{IDB}$, $h = H_2(m, w, U_1)$
   $V_1 = t_1(x + h)\, S_{IDP}$, $V_2 = (x + h)V$
   He sends σ_1 = (m, w, U_1, U_2, h, V_1, V_2) to the designated verifier Cindy.

5. **Proxy signature verification:** On receiving σ_1 the designated verifier Cindy operates as follows:
   - Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
   - Checks whether Alice and Bob are specified as the original signer and the proxy signer in the warrant w, respectively.
   - Computes $h = H_2(m, w, U_1)$ and accepts the signature iff
     $e(U_1, V_1) = e(Q_{IDC}, V_2)\, e(S_{IDC}, U_2 + hQ_{IDB})$

6. **Correctness:**
   $e(U_1, V_1) = e(t_1^{-1} Q_{IDC}, t_1(x + h)S_{IDP})$
   $= e(Q_{IDC}, (x + h)V + (x + h)S_{IDB})$
   $= e(Q_{IDC}, V_2)\, e(Q_{IDC}, (x + h)sQ_{IDB})$
   $= e(Q_{IDC}, V_2)\, e(S_{IDC}, U_2 + hQ_{IDB})$

4.3. SDVPS scheme derived from Wang's scheme

In his paper Wang [9] first proposed a provably secure proxy signature scheme based on two party Schnorr signature scheme then extended it to designated verifier proxy signature scheme using Saeednia et al.'s [7] strong designated verifier signature scheme.

Review of Wang's scheme

- **Setup:** Let p and q be two large primes such that $q \mid (p-1)$ and $G_q = < g >$ is a multiplicative subgroup of Z_p^* generated by an element $g \in Z_p^*$. Let h (.) And h^1 (.) be two cryptographic publicly known hash functions.
- **Key Generation:** Every user chooses $x \in Z_q^*$ and computes $y = g^x \bmod p$, x is the users secret key; y is the users public key.
- **Proxy Key Generation:** To generate a proxy key pair (x_p, y_p), the original signer Alice and the proxy signer Bob execute the following protocol jointly.
  - **a.** Alice chooses a random number $k_A \in_R Z_q^*$, and computes $r_A = g^{k_A} \bmod p$ and $c = h^1(r_A)$, and sends c to Bob.
  - **b.** Similarly, Bob chooses a random number $r_B = g^{k_B} \bmod p$ and sends r_B to Alice.
  - **c.** On receiving r_B Alice checks $r_B^q = 1 \bmod p$. If all validation passes, computes $r_p = r_A . r_B \bmod p$, $S_A = k_A + x_A h(w, r_p) \bmod q$, where w is the warrant on message m. Alice sends (r_A , S_A) to Bob.
  - **d.** Upon receiving (r_A , S_A), Bob computes $r_p = r_A . r_B \bmod p$ and then checks whether $r_A^q = 1 \bmod p$, $c = h^1(r_A)$, and $g^{S_A} = (y_A)^{h(w, r_p)} . r_A \bmod p$.

    If all validation pass computes $S_B = k_B + x_B h(w, r_p) \bmod q$ and
    Computes, $x_P = S_A + S_B \bmod q$ as the proxy secret key
    And, $y_p = g^{x_P} \bmod p$ as the proxy public key.
- **Proxy Signature Generation:** To generate a proxy signature on a message m that confirms to the warrant w.
  Bob chooses two random numbers $k \in Z_q$ and $t \in Z_q^*$ and computes
  $r = y_c^k \bmod p$, $c = h(m, w, r)$, $s = kt^{-1} - x_p . c \bmod q$
  Sends (w, r_p, c, s, t) to the designated verifier Cindy.
- **Proxy Signature Verification:** To verify the validity of the signatures the designate verifier Cindy operates as follows:
  - ✓ Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
  - ✓ Checks whether Alice and Bob are specified as the original signer and the proxy signer in the warrant w, respectively.
  - ✓ Computes $r^1 = (g^s y_p^c)^{tx\,c} \bmod p$, here $y_p = (y_A . y_B)^{h(w, r_p)} . r_p \bmod p$

    Cindy accept the proxy signature iff $h(m, w, r^1) = c$.

Now, we shall introduce the 'identity based' in Wang's scheme to form another ID based SDVPS scheme.

Proposed scheme

Let G_1 be a group of order a large prime number q and G_2 be a multiplicative subgroup of a finite field F of same order. $e : G_1 \times G_1 \to G_2$ is a bilinear pairing.

1. **Setup:** In this phase, KGC chooses a generator $P \in G_1$, a random number $s \in Z_q^*$ and computes $P_{pub} = sP$. KGC also chooses two cryptographic hash functions H_1 and H_2.

$H_1 : \{0,1\}^* \rightarrow Z_q^*$ and $H_2 : G_1 \rightarrow Z_q^*$

The system parameters (G_1, G_2, P, P_{pub}, H_1, H_2, e) are public and s is kept with KGC.

2. **Key Generation:** For a user U with identity ID_U this phase generates $Q_{IDU} = H_1(ID_U)$ as the public key of the user, and $S_{IDU} = s^{-1}.Q_{IDU}.P$ as the secret key of the user.

3. **Proxy Key Generation:** To generate a proxy key pair (x_p , y_p) for the proxy signer Bob, the original signer Alice and the proxy signer Bob execute the following protocol jointly.
   - Alice chooses a random value $k_A \in_R Z_q^*$, computes $r_A = k_A.P$, $c = H_2(r_A.Q_{IDA}) \in Z_q^*$. He sends c to Bob.
   - Similarly, Bob chooses a random value $r_B = k_B.P$ and sends r_B to Alice.
   - On receiving r_B Alice checks $H_2(r_B.Q_{IDB}) \in Z_q^*$. If all validation passes computes $r_p = r_A + r_B$, $y = H_2[r_p.H_1(m, w)]$, $S_A = S_{IDA}. k_A.y$, where w is the warrant on message m**.** Alice sends (r_A , S_A) to Bob.
   - Upon receiving (r_A , S_A), Bob computes $r_p = r_A + r_B$, $y = H_2[r_p.H_1(m, w)]$ $H_2(r_A.Q_{IDA}) \in Z_q^*$, $c = H_2(r_A.Q_{IDA})$, and checks $e(S_A , r_B.Q_{IDB}) = e(Q_{IDA}.r_A.H_2[r_A.H_1(m, w)] , k_B . S_{IDB})$ If all validation passes, Bob computes, $S_B = S_{IDB} . k_B.y$, Then, computes $x_p = S_A + S_B$ as proxy secret key.

4. **Proxy Signature Generation:** To generate proxy signature on message m Bob chooses two random numbers $k \in Z_q$ and $t \in Z_q^*$ and computes $r = e(P, P)^{kQ_{IDC}}$, $U = H_2(H_1(m, w, r).P)$, $r_1 = (Q_{IDA}.r_A + Q_{IDB}.r_B).y$, $V = t^{-1}kP - x_p. U$. He sends $\sigma = (t, m, w, r_p, r, r_1, V, U)$ to the designated verifier Cindy.

5. **Proxy Signature Verification :**To verify the validity of the signatures the designated verifier Cindy operates as follows:
   - Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
   - Checks whether Alice and Bob are specified as the original signer and the proxy signer in the warrant w, respectively.
   - Computes $U = H_2(H_1(m, w, r).P)$ and he accepts the signature iff $[e(V, P.Q_{IDC})e(r_1, S_{IDC})^U]^t = r$

6. **Correctness:**

$$
\begin{aligned}
&[e(V, P.Q_{IDC})\, e(r_1, S_{IDC})^U]^t \\
&= [e(t^{-1}.kP - U.x_p , P.Q_{IDC})\, e(r_1.U , s^{-1}Q_{IDC}.P)]^t \\
&= [e(t^{-1}.kP - U.x_p , P.Q_{IDC})\, e(U.x_p, P.Q_{IDC})]^t \\
&= e(t^{-1}.kP, P.Q_{IDC})^t \\
&= e(P, P)^{kQ_{IDC}} \\
&= r
\end{aligned}
$$

4.4 Another ID-based SDVPS scheme

In this section we proposed another new ID based SDVPS Scheme:

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

2. Key Generation: Given an identity, this phase generates $Q_{IDU} = H_1 (ID_U)$as the public key of the user, and $S_{IDU} = s^{-}1Q_{IDU.}P$ as the secret key of the user.

3. Proxy key generation phase: Alice chooses a random numbers $r_w \in Z_q^*$ and computes $U_w = r_w . P$, $h_w = H_1(ID_A , m, U_w)$, $V_w = S_{IDA} . r_w + h_w .P$
He sends $\sigma = (V_w , U_w)$ to Bob. On receiving σ, Bob computes $h_w = H_1(ID_A , m_w, U_w)$ and he accepts the warrant iff $e(P_{pub} , V_w) = e(P, U_w)^{Q_{IDA}} e(P_{pub} , P)^{h_w}$. Then he computes the proxy secret key $S_{IDP} = V_w + H_2[H_1(ID_A , ID_B , m_w, U_w).P] . S_{IDB}$.

4. Proxy signature generation: To sign a message m Bob performs as follows:
He chooses a random value $r_p \in Z_q^*$ and
Computes $U_p = r_p .Q_{IDC} . P$, $h_P = H_1(ID_B , m_w, U_p)$, $V_p = r_p^{-1}(S_{IDA} + h_p .P)$.
Sends $(ID_A , ID_B, U_W , U_p , V_p , h_p , m_w)$ to Cindy.

5. Proxy signature verification: On receiving $(ID_A , ID_B, U_W , U_p , V_p , h_p , m_w)$ the designated verifier Cindy operates as follows:

- Checks whether the message m confirms to the warrant w. If not, stops. Otherwise, continues.
- Checks whether Alice and Bob are specified as the original signer and the proxy signer in the warrant w, respectively.
- Computes $h_p = H_1(ID_B , m_w, U_p)$, $h_w = H_1(ID_A , m_w, U_w)$, $H_2[H_1(ID_A , ID_B , m_w, U_w).P]$. He accepts the signature iff

$e(U_p , V_p)$
$= e(S_{IDC}, Q_{IDA} .U_w + Q_{IDB} . H_2[H_1(ID_A , ID_B , m_w, U_w).P]) e(P, h_P.Q_{IDC}.U_p + h_w.Q_{IDC}.P)$

6. Correctness:
$e(U_p, V_p)$
$= e(P.Q_{IDC}, S_{IDP} + h_pU_p)$
$= e(P.Q_{IDC}, V_w + H_2[H_1(ID_A, ID_B, m, w).P].S_{IDB}) e(P.Q_{IDC}, h_pU_p)$
$= e(P.Q_{IDC}, S_{IDA}. r_w + h_w.P)e(P.Q_{IDC}, H_2[H_1(ID_A, ID_B, m, w).P].s^{-1} Q_{IDB}.)e(P,Q_{IDC}.h_pU_p)$
$= e(P.Q_{IDC}, s^{-1}Q_{IDA}.P.r_w)e(P, h_wQ_{IDC}. P) e(S_{IDC}, H_2[H_1(ID_A, ID_B, m, w). P] .Q_{IDB}.P).e(P,Q_{IDC}.h_pU_p)$
$= e(S_{IDC}, Q_{IDA} U_w) e(P, h_wQ_{IDC}. P) e(S_{IDC}, H_2[H_1(ID_A, ID_B, m, w).P] .Q_{IDB}.P).e(P, Q_{IDC}.h_pU_p)$
$= e(S_{IDC}, Q_{IDA} . U_w + Q_{IDB} . H_2[H_1(ID_A , ID_B , m_w, U_w).P]) \times e(P, h_P.Q_{IDC}.U_p + h_w.Q_{IDC}.P)$

5. Computation aspects

We observe that the implementation of the above schemes require the operations of the hashing, multiplication, pairing evaluation, exponentiation and taking the inverse. In this section, we compare the four schemes discussed above and count the hash, multiplication, exponentiation, pairing and inverse for each of them.

Scheme based on Kumar et al	Hash	Multiplication	Pairing	Exponential	Inverse
Signature generation Alice	1	5	1	-	1
Verification Bob	1	-	3	-	-
Proxy signature generation	1	5	1	-	1
Signature verification Cindy	1	-	4	-	-

Scheme based on Saeednia et al	Hash	Multiplication	Pairing	Exponential	Inverse
Signature generation Alice	2	4	1	1	1
Verification Bob	2	1	2	2	-
Proxy signature generation	1	4	-	-	1
Signature verification Cindy	1	-	3	-	-

Scheme based on Wang's Scheme	Hash	Multiplication	Pairing	Exponential	Inverse
Signature generation Alice	4	5	-	-	-
Verification Bob	1	4	2	-	-
Proxy signature generation	2	7	1	1	1
Signature verification Cindy	2	1	2	2	-

New proposed Scheme	Hash	Multiplication	Pairing	Exponential	Inverse
Signature generation Alice	1	3	-	-	1
Verification Bob	1	-	3	2	-
Proxy signature generation	1	4	-	-	1
Signature verification Cindy	3	8	3	-	-

The following table gives the computational complexity of the schemes at a glance. Scheme based on Kumar et al uses least number of hashing, but maximum number of pairing evaluation. Scheme based on Wang uses least number of pairing, but maximum number of hashing.

Schemes	Hash	Multiplication	Pairing	Exponential	Inverse
Kumar's	4	10	9	-	2
Saeednia's	6	9	6	3	2
Wang's	9	17	5	3	1
New	9	16	6	2	1

6. Security analysis:

6.1 **Secrecy:** In all the described schemes an intruder cannot derive the proxy key even knowing the secrets of the original signer Alice and the proxy signer Bob. Hence, our schemes are secure.

6.2 **Strongness:** Designated verifier Cindy has to use his secret key at the time of verification of the proxy signature. So, only the designated verifier can check the validity of the proxy signatures.

6.3 **Unforgeability:** In each of the presented schemes, the proxy signatures cannot be generated without the knowledge of the proxy secret key and anybody (including Alice and Bob) cannot generate a valid proxy secret key independently. Thus, the signatures are unforgeable.

6.4 Proxy protected: The original signer Alice cannot generate a valid proxy signature on behalf of the proxy signer Bob since Alice does not have any information about the secret key of Bob. Thus, each scheme is proxy protected.

7. Conclusion: In this paper we proposed four new ID based strong designated verifier proxy signature schemes. We have analyzed the security of our schemes and compared them in terms of computation efficiency. Wang's scheme is more efficient than all the proposed schemes in terms of bilinear pairing.

References:

1. **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.
2. **X. Mu Huang, W.Y.Susilo, F.Zhan.** Short designated verifier proxy signature from pairings, International Workshop on Security in Ubiquitous Computing Systems. LNCS #3823, Springer-Verlag, 2005, 835-844.
3. **M.Jakobsson, K.Sako, K.R.Impaliazzo.** Designated verifier proofs and their applications. Eurocrypt 1996, LNCS #1070, Springer-Verlag, 1996, 142-154.
4. **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
5. **R.Lu, Z.Cao.** Designated verifier proxy scheme with message recovery. Applied Mathematics and Computation, 169(2), 2005, 1237-1246.
6. **Ronald Rivest, Adi Shamir, Yaee Tauman.** How to leak a secret. ASIACRYPT'01, LNCS #2248, Springer-Verlag, 2001.
7. **S. Saeednia, S.Kreme, O.Markotwich.** An efficient strong designated verifier signature scheme. ICICS 2003, LNCS #2971, Springer-Verlag, 2003, 40-54.
8. **A. Shamir.** ID based cryptosystems and signature scheme. Crypto'84, LNCS #196, Springer-Verlag, 1984, 47-53.
9. **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.