Secret Handshakes based on Shortened Versions of DSS

Preeti Kulshrestha
Department of Mathematics, Statistics and Computer Science
G.B. Pant University of Agriculture and Technology
Pantnagar, India

Arun Kumar
Department of Mathematics, Statistics and Computer Science
G.B. Pant University of Agriculture and Technology
Pantnagar, India

ABSTRACT
Balfanz et al. in 2003 introduced secret handshakes as mechanisms designed to prove group membership and share a secret key between two fellow group members. A secret handshake protocol allows two users to mutually verify another's authenticity without revealing their own identity. In a secret handshake *Verification* if the verification succeeds the users may compute a common shared key for further communication. Thus secret handshakes can be appropriately turned into an authenticated key exchange protocol. The present paper proposes two secret handshakes scheme based on variations DSS-1 and DSS-2 of DSS signature. It is shown that proposed schemes are secure under the random oracle model along with comparison of computational complexity of proposed schemes with existing schemes.

General Terms
Secret Handshakes, Signature, Security, Random Oracle Model, Computational Complexity.

Keywords
Secret Handshakes, Credential, ElGamal, DSA, DSS-1, DSS-2, Computational Complexity.

1. INTRODUCTION
A secret handshake (SH) between two users was first introduced by Balfanz et al. [2] in 2003, to simultaneously prove to each other possession of membership of a certain group. In SH two participating users authenticate each other in a way that no one reveals his own membership or credential unless the peer's legitimacy was already ensured of and share a common key for further communication. A SH can be appropriately turned into an authenticated key exchange but an authenticated key exchange does not necessarily imply a SH. Users are not able to perform a successful handshake without the appropriate credentials. Protocol exchanges are often untraceable and anonymous. The protocol makes sure that an outsider or an illegitimate group member does not learn anything by interacting with a legitimate user or by eavesdropping on protocol exchanges. In a SH *verification* is only possible by legitimate group member because it relies on unique SH.

Balfanz et al. [2] introduced the notion of privacy in public key based authentication schemes and proposed the first two-party SH schemes based on bilinear maps secure under the Gap Diffie- Hellman (GDH) assumption. Using CA-Oblivious public key encryption Castellucia et al. [3] developed an efficient SH scheme secure under the Computational Diffie-Hellman (CDH) assumption. Vergnaud [9] presented two SH schemes inspired by two RSA-based key agreement protocols first introduced byOkamoto-Tanaka [8] and second by Girault [5]. Zhou et al. [12] proposed three round SH schemes based on ElGamal signature and extends their scheme to a DSA based SH which also requires only three rounds. Wen et al. [10] proposed two party SH schemes from ID-based message recovery signature (MRS). In all these schemes, the players use one time certificates to achieve unlinkability. If the players re use their certificates it's possible to trace multiple occurrences of the same party. Ateniese et al. [1] extended the SH with dynamic matching in which each party can reuse their credential. Inspired by [1] Kulshrestha et al. [7] proposed a SH with dynamic matching which is based on ZSS signature.

In this paper two new SH schemes based on variations DSS-1 and DSS-2 of DSS signature [11] are proposed. In this work computational complexity of proposed schemes along with comparison with known SH schemes based on ElGamal and DSA by Zhou et al. [12] is discussed. The present study is arranged in the following manner: section 2 defines basic terminology and brief account of the work of Zhou. In section 3 two new SH schemes based on shortened versions of DSS along with security has been discussed. Section 4 compares the computational complexity of all the schemes.

2. SECRET HANDSHAKES SCHEMES
In SH scheme their exists three entities for a group G, a user, a member which is a user which belongs to the group and a group administrator (GA) who creates and adds members into the group, and issues certificate in a form of secret key to members.

2.1 The SH scheme consists of three following algorithms
Create Group is an algorithm run by a GA, which takes Params (a set of parameters) as input and generates a key pair GP_k (group public key) and GS_k (group secret key).

Add User is an algorithm between a user U and the GA of some group. It takes Params and GA's secret GS_k as input and outputs a public key P_k and secret key S_k for U and makes U a valid member of the group.

Handshake is executed between users, say, A and B, who want to authenticate each other on the public inputs ID_A , ID_B and Params. The private input of each party is their secret credential, and the output of the protocol for either party is either *reject* or *accept*.

2.3 The SH scheme have the following security properties
Completeness/ Correctness:
If two honest members belonging to the same group and perform handshake protocol with valid credentials, then both members always output *accept*.

Impersonator Resistance
The impersonator resistance property is violated if an honest member V of group G authenticates a non member $\mathcal{A}$ as a group member, with non negligible probability. For this property to hold, we must have

$Pr[\mathcal{A}\ succeeds\ in\ making\ V\ output$ accept $| V \in G\ and\ \mathcal{A} \notin G] \leq \varepsilon$, where ε is negligible.

Detector Resistance
An adversary $\mathcal{A}$ violates the detector resistance property if it can decide with some non negligible probability, whether some honest party V is a member of some group G by determining the relationship between the public message of the member and the public key of the group, even though $\mathcal{A}$ is not a member of G. For this property to hold, we must have $r[\mathcal{A}\ Knows\ whether\ V\ is\ the\ valid\ member\ and\ \mathcal{A} \notin G] \leq \varepsilon$, where ε is negligible.

2.4 Known SH schemes

Here two SH schemes given by Zhou et al. [12] are discussed. The first scheme is based on ElGamal signature and the second one is based on DSA.

2.4.1 ElGamal based SH Scheme [12]:

ElGamal Signatures [4] are generated as follows:

Key Generation: Chooses a large prime p and a generator g of group $\mathbb{Z}_p^*$, select a random number $s, 1 < s < p-1$ as the secret.Compute $y = g^s \ mod\ p$.Then the public key is$\{p, g, y\}$, and private key is s.

Signature Generation: To sign a message M, the signer chooses $r \in_R \mathbb{Z}_p^*$ such that $\gcd(r, p-1) = 1$. Compute the pair (α, β) as $\alpha = g^r \ mod\ p$ and $\beta = (M - \alpha * s) * r^{-1}\ mod(p-1)$, as signature on M.

Verification: Signature, are valid iff $g^M = y^\alpha \alpha^\beta \ mod\ p$.

The SH scheme runs as follows:

Create Group:

The GA runs the ElGamal key generation algorithm to create keys {p, q, g, y, s} where p is large prime and q is a prime divisor of $p-1$ and g is generator $i.e., g^q = 1\ mod\ p$ and $y = g^s\ mod\ p$ is public key of GA and s is the secret key of GA. $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_2: \{0,1\}^* \rightarrow \{0,1\}^n$ are two cryptographic hash functions.

Add User:

To add a user U to the group, GA allocates ID_U to user and chooses a random nonce $r_U \in_R \mathbb{Z}_q^*$. GA then Computes $h_U = H_1(ID_U), \alpha_U = g^{r_U}\ mod\ p$, and $\beta_U = (h_U - \alpha_U * s) * r_U^{-1}\ mod\ q$. Secret key for user U is (α_U, β_U).

Handshake:

Two users A and B conduct the secret handshake as follows:

- **B → A : (ID_B , ζ_B, η_B),**

where $k_B \in_R \mathbb{Z}_q^*$, $\zeta_B = \alpha_B^{(k_B+1)}\ mod\ (p.q)$, and $\eta_B = \beta_B * (k_B+1)^{-1} * \alpha_B^{k_B}\ mod\ q$.

- **A → B : (ID_A , ζ_A, η_A, V_o),**

where $k_A \in_R \mathbb{Z}_q^*$, $\zeta_A = \alpha_A^{(k_A+1)}\ mod\ (p.q)$, $\eta_A = \beta_A * (k_A+1)^{-1} * \alpha_A^{k_A}\ mod\ q$, and

$$V_o = H_2\left(\left(\left(y^{(\zeta_B mod\ q)} * (\zeta_B\ mod\ p)^{\eta_B}\right)^{h_B^{-1}}\right)^{\alpha_A^{k_A}}\ mod\ p \parallel ID_A \parallel ID_B \parallel 0\right)$$

- **B → A : (V1) , where**

$$V_1 = H_2\left(\left(\left(y^{(\zeta_A\ mod\ q)} * (\zeta_A\ mod\ p)^{\eta_A}\right)^{h_A^{-1}}\right)^{\alpha_B^{k_B}}\ mod\ p \parallel ID_A \parallel ID_B \parallel 1\right)$$

- **A verifies, if**

$$V_1 = H_2\left(\left(\left(y^{(\zeta_B\ mod\ q)} * (\zeta_B\ mod\ p)^{\eta_B}\right)^{h_B^{-1}}\right)^{\alpha_A^{k_A}}\ mod\ p \parallel ID_A \parallel ID_B \parallel 1\right)$$

- **B verifies, if**

$$V_1 = H_2\left(\left(\left(y^{(\zeta_A\ mod\ q)} * (\zeta_A\ mod\ p)^{\eta_A}\right)^{h_A^{-1}}\right)^{\alpha_B^{k_B}}\ mod\ p \parallel ID_A \parallel ID_B \parallel 0\right)$$

2.4.2 DSA based SH Scheme [12]:

DSA generates signature as follows:

Key Generation: Choose a large prime p, a prime divisor q of $p-1$ and a generator $g\ mod\ p$ of order q. Pick s as random such that $1 < s < q$ and compute $y = g^s\ mod\ p$. Then the public key is$\{p, q, g, y\}$, and private key is s.

Signature Generation: To sign a message M signer chooses a random number $r < q$. Compute the pair (α, β) where $\alpha = (g^r\ mod\ p)\ mod\ q$ and $\beta = (M + \alpha * s) * r^{-1}\ mod\ q$.$(\alpha, \beta)$ as a signature on M.

Verification: To verify the signature, the receiver first computes $\omega = \beta^{-1}\ mod\ q$, $Z_1 = (M * \omega)\ mod\ q$ and $Z_2 = \alpha * \omega\ mod\ q$. Then output true if the following equation hold $\alpha = \left((g^{Z_1} * y^{Z_2})\ mod\ p\right) mod q$.

The SH scheme runs as follows:

Create Group:

The GA runs the DSA key generation algorithm to create keys {p, q, g, y, s} where p is large prime and q is a prime divisor of $p-1$ and g is generator$i.e., g^q = 1\ mod\ p$ and $y = g^s\ mod\ p$ is public key of GA and s is the secret key of GA. $H_1: \{0,1\}^* \rightarrow \mathbb{Z}_q^*$ and $H_2: \{0,1\}^* \rightarrow \{0,1\}^n$ are two cryptographic hash functions.

Add User:

To add a user U to the group, GA allocates ID_U to user, and then computes$\alpha_U = (g^{r_U} mod\ p)\ mod\ q$, and $\beta_U = (h_U + \alpha_U * s) * r_U^{-1}\ mod\ q$, where $h_U = H_1(ID_U)$ and $r_U \in_R \mathbb{Z}_q^*$. Secret key for user U is (α_U, β_U).

Handshake:

Users A and B conduct the SH as follows:

- **B → A : (ID_B , ζ_B, η_B),where**

$\gamma_B = (g^{h_B} * y^{\alpha_B})^{\beta_B^{-1}}\ mod\ p, \zeta_B = \gamma_B^{(k_B+1)}\ mod\ (p.q)$, and $\eta_B = \beta_B * (k_B+1)^{-1} * \gamma_B^{k_B}\ mod\ q$.

- **A → B : (ID_A , ζ_A, η_A, V_o), where**

$\gamma_A = (g^{h_A} * y^{\alpha_A})^{\beta_A^{-1}} \ mod\ p, \quad \zeta_A = \gamma_A^{(k_A+1)} \ mod\ (p.q),$
$\eta_A = \beta_A * (k_A + 1)^{-1} * \gamma_A^{k_A} \ mod\ q$, and

$$V_o = H_2\left(\left(\left(y^{(-\zeta_B \bmod q)} * (\zeta_B \bmod p)^{\eta_B}\right)^{h_B^{-1}}\right)^{\gamma_A^{k_A}} \bmod p \parallel ID_A \parallel ID_B \parallel 0\right)$$

- **B → A : (V1), where**

$$V_1 = H_2\left(\left(\left(y^{(-\zeta_A \bmod q)} * (\zeta_A \bmod p)^{\eta_A}\right)^{h_A^{-1}}\right)^{\gamma_B^{k_B}} \bmod p \parallel ID_A \parallel ID_B \parallel 1\right)$$

- **A verifies, if**

$$V_1 = H_2\left(\left(\left(y^{(-\zeta_B \bmod q)} * (\zeta_B \bmod p)^{\eta_B}\right)^{h_B^{-1}}\right)^{\gamma_A^{k_A}} \bmod p \parallel ID_A \parallel ID_B \parallel 0\right)$$

- **B verifies, if**

$$V_o = H_2\left(\left(\left(y^{(-\zeta_A \bmod q)} * (\zeta_A \bmod p)^{\eta_A}\right)^{h_A^{-1}}\right)^{\gamma_B^{k_B}} \bmod p \parallel ID_A \parallel ID_B \parallel 0\right)$$

3. PROPOSED SCHEMES

3.1 SH Scheme based on DSS-1:

DSS -1 generates signature [11] as follows:

Key Generation: Generate random distinct secret prime p and a generator g of $\mathbb{Z}_p^*$ select a random integer x_A s.t.,

$1 \le x_A \le p-2$. Compute $y = g^{x_A} \ mod\ p$. Then the public key is $\{p, q, g, y\}$ and private key is x_A.

Signature Generation: To sign a message M signer chooses a random number x s.t. $1 \le x_A \le p-2$ and gcd$(x, p-1) = 1$. Compute the pair (r, s) as $r = h(g^x, M)$ and $s = x * (r + x_A)^{-1} mod(p-1)$ is signature on M.

Verification: To verify the signature, the receiver first verifies that $1 \le x \le p-1$, if not reject the signature. Otherwise verify $(y * g^r)^s = g^x$.

The SH scheme runs as follows:

Create Group

The GA runs the DSS key generation algorithm to create keys (p, q, g, y, s), where p is large prime and q is a prime divisor of p-1 and g is generator $i.e., g^q = 1 \ mod\ p$ and $y = g^s \ mod\ p$ is group public key and s is the group secret key. $H_1: \{0,1\}^* \to \mathbb{Z}_q^*$ and $H_2: \{0,1\}^* \to \{0,1\}^n$ are two cryptographic hash functions.

Add User

To add a user U to the group G, the GA allocates a unique identity ID_U to the user and computes:

$\alpha_U = g^{r_U} mod\ p$, where $r_U \in_R \mathbb{Z}_q^*$, $\beta_U = H_1(ID_U, \alpha_U) \in \mathbb{Z}_q^*$ and $\gamma_U = r_U * (\beta_U + s)^{-1} \ mod\ q$. Here (α_U, γ_U) is the signature of GA in form of DSS-1 Signature.

Handshake:

Users A and B run the SH Protocol as follows:

- **B → A : (ID_B, ω_B, n_B, ζ_B), where** $k_B \in_R \mathbb{Z}_q^*$

$$\omega_B = \left(\gamma_B{}^{-1} * \beta_B^{-1} * \ (k_B + 1)^{-1} * \alpha_B^{k_B}\right) mod\ q,$$
$\eta_B = \alpha_B^{(k_B+1)} \ mod\ (pq),\ \zeta_B = \gamma_B * (k_B + 1) \ mod\ q.$

- **A → B: (ID_A, ω_A, n_A, ζ_A , V_o),**

$\omega_A = \left(\gamma_A{}^{-1} * \beta_A^{-1} * (k_A + 1)^{-1} * \alpha_A^{k_A}\right) mod\ q,$
$\eta_A = \alpha_A^{(k_A+1)} \ mod\ (pq),$
$\zeta_A = \gamma_A * (k_A + 1) \ mod\ q$, where $k_A \in_R \mathbb{Z}_q^*$ and

$$V_o = H_2\left(\left(\left(\eta_B \ mod\ p * y^{(-\zeta_B \bmod q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} \bmod p \parallel ID_A \parallel ID_B \parallel 0\right)$$

- **B → A : (V1), where**

$$V_1 = H_2\left(\left(\left(\eta_A \ mod\ p * y^{(-\zeta_A \bmod q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} \bmod p \parallel ID_A \parallel ID_B \parallel 1\right)$$

- **A verifies, if**

$$V_1 = H_2\left(\left(\left(\eta_B \ mod\ p * y^{(-\zeta_B \bmod q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} \bmod p \parallel ID_A \parallel ID_B \parallel 1\right)$$

- **B verifies, if**

$$V_o = H_2\left(\left(\left(\eta_A \ mod\ p * y^{(-\zeta_A \bmod q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} \bmod p \parallel ID_A \parallel ID_B \parallel 0\right)$$

If both the verification succeeds then A and B finish all the steps of the SH protocol and the handshake has been successful. A and B now can compute the shared key:
A computes

$$K_A = H_2\left(\left(\left(\eta_B \ mod\ p * y^{(-\zeta_B \bmod q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} \bmod p \parallel ID_A \parallel ID_B \parallel 2\right)$$

and B computes

$$K_B = H_2\left(\left(\left(\eta_A \ mod\ p * y^{(-\zeta_A \bmod q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} \bmod p \parallel ID_A \parallel ID_B \parallel 2\right)$$

Correctness:

To see that $K_A = K_B$, we observe that

$$= \left(\left(\eta_B \ mod\ p * y^{(-\zeta_B \bmod q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}}$$

$$= \left(\left(\alpha_B^{(k_B+1)} * y^{-\gamma_B * (k_B+1)}\right)^{\gamma_B{}^{-1} * \beta_B^{-1} * (k_B+1)^{-1} * \alpha_B^{k_B}}\right)^{\alpha_A{}^{k_A}}$$

$= \left(g^{(r_B - s*\gamma_B)*{\gamma_B}^{-1}*\beta_B^{-1}*\alpha_B^{k_B}}\right)^{\alpha_A{}^{k_A}}$

$= g^{\alpha_B^{k_B} * \alpha_A^{k_A}}$

Similarly for B.

3.2 SH Scheme based on DSS-2:

DSS -2 generates signature [11] as follows:

Key Generation: Same as DSS-1.

Signature Generation: To sign a message M signer chooses a random number x s.t., $1 \le x_A \le p-2$ and gcd$(x, p-1) = 1$.Compute the pair (r,s) as $r = h(g^x, M)$ and $s = x*(1 + r*x_A)^{-1} mod(p-1)$ is signature on M.

Verification: To verify the signature, the receiver first verifies that $1 \le x \le p-1$, if not reject the signature. Otherwise verify $(g*y^r)^s = g^x$.

The SH scheme runs as follows:

Create Group: Same as DSS-1

Add User: To add a user U to the group G, the GA allocates a unique identity ID_U to the user and computes:

$\alpha_U = g^{r_U} mod\ p$, where $r_U \in_R \mathbb{Z}_q^*$ and

$\beta_U = H_1(ID_U, \alpha_U) \in_R \mathbb{Z}_q^*, \gamma_U = r_U * (1 + \beta_U * s)^{-1}\ mod\ q$

Here (α_U, γ_U) is the signature of GA in form of DSS-2 Signature.

Handshake:

Users A and B run the SH Protocol as follows:

- **B → A :** $(\boldsymbol{ID_B},\ \boldsymbol{\omega_B},\ \boldsymbol{n_B},\ \boldsymbol{\zeta_B})$

$\omega_B = \left(\gamma_B{}^{-1} *\ \ (k_B + 1)^{-1} * \alpha_B^{k_B}\right) mod\ q,$

$\eta_B = \ \alpha_B^{(k_B+1)}\ mod\ (pq),$

$\zeta_B = \gamma_B * \beta_B * (k_B + 1)\ mod\ q$, where $k_B \in_R \mathbb{Z}_q^*$.

- **A → B:** $(\ \boldsymbol{ID_A},\ \boldsymbol{\omega_A},\ \boldsymbol{n_A},\ \boldsymbol{\zeta_A}\ ,\ \boldsymbol{V_o}\ )$

$\omega_A = \left(\gamma_A{}^{-1} * (k_A + 1)^{-1} * \alpha_A^{k_A}\right) mod\ q,$

$\eta_A = \ \alpha_A^{(k_A+1)}\ mod\ (pq),$

$\zeta_A = \gamma_A * \beta_A * (k_A + 1)\ mod\ q$, where $k_A \in_R \mathbb{Z}_q^*$ and

$V_o = H_2\left(\left(\left(\eta_B\ mod\ p * y^{(-\zeta_B\ mod\ q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} mod\ p \parallel ID_A \parallel ID_B \parallel 0\right)$

- **B → A : (V1), where**

$V_1 = H_2\left(\left(\left(\eta_A\ mod\ p * y^{(-\zeta_A\ mod\ q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} mod\ p \parallel ID_A \parallel ID_B \parallel 1\right)$

- **A verifies, if**

$V_1 = H_2\left(\left(\left(\eta_B\ mod\ p * y^{(-\zeta_B\ mod\ q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} mod\ p \parallel ID_A \parallel ID_B \parallel 1\right)$

- **B verifies, if**

$V_o = H_2\left(\left(\left(\eta_A\ mod\ p * y^{(-\zeta_A\ mod\ q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} mod\ p \parallel ID_A \parallel ID_B \parallel 0\right)$

If both the verification succeeds then A and B finish all the steps of the SH protocol and the handshake has been successful. A and B now can compute the shared key:

A computes

$K_A = H_2\left(\left(\left(\eta_B\ mod\ p * y^{(-\zeta_B\ mod\ q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}} mod\ p \parallel ID_A \parallel ID_B \parallel 2\right)$

and B compute

$K_B = H_2\left(\left(\left(\eta_A\ mod\ p * y^{(-\zeta_A\ mod\ q)}\right)^{\omega_A}\right)^{\alpha_B{}^{k_B}} mod\ p \parallel ID_A \parallel ID_B \parallel 2\right)$

Correctness

To see that $K_A = K_B$, we observe that

$= \left(\left(\eta_B\ mod\ p * y^{(-\zeta_B\ mod\ q)}\right)^{\omega_B}\right)^{\alpha_A{}^{k_A}}$

$= \left(\left(\alpha_B^{(k_B+1)} * y^{-\gamma_B*\beta_B*(k_B+1)}\right)^{{\gamma_B}^{-1}*(k_B+1)^{-1}*\alpha_B^{k_B}}\right)^{\alpha_A{}^{k_A}}$

$= \left(g^{(r_B - s*\gamma_B*\beta_B){\gamma_B}^{-1}*\alpha_B^{k_B}}\right)^{\alpha_A{}^{k_A}}$

$= g^{\alpha_B^{k_B} * \alpha_A^{k_A}}$

Similarly for B.

3.3 Security

We adopt the security definitions of Zhou et a.l [12] to show the security of proposed schemes.

Theorem: The proposed SH scheme based on DSS-1 is impersonator resistant under the assumption that DSS-1signature is existentially unforgeable in the random oracle model.

Proof: The proposed SH scheme is impersonator resistant if no polynomial bounded adversary $\mathcal{A}$ wins the following game against the challenger with non-negligible probability:

- The challenger random pick a public key$(\ p,\ q,\ g, y\ )$, and send it to adversary$\mathcal{A}$.
- The adversary responds with an ID_A
- The Challenger then picks random pair $\langle \zeta_A,\ \eta_A, w_A \rangle$, where $\zeta_A \in_R \mathbb{Z}_{p*q}$, and $\eta_A, w_A \in_R \mathbb{Z}_q$ and send to $\mathcal{A}$.
- Then adversary outputs $k_A^{/} \in_R \mathbb{Z}_q$.
- The adversary wins the game if $(g)^{k_A^{/}} = (\ y^{-\zeta_A} * \eta_A)^{w_A}\ \ modp$

Given an attacker $\mathcal{A}$ that wins the above game with probability $\mathcal{E}$ then an another attacker $\mathcal{B}$ can be constructed to successfully forge the DSS-1signature with probability $\mathcal{E}$.

- $\mathcal{B}$, when given the DSS-1 public key (g, p, q, y) and send to $\mathcal{A}$.

- $\mathcal{A}$ Responds with ID_A.
- $\mathcal{B}$ Picks a random pair $\langle \zeta_A, \eta_A, w_A \rangle$ and send to $\mathcal{A}$.
- Then $\mathcal{A}$ output $k_A^{/} \in_R \mathbb{Z}_q$ and send to $\mathcal{B}$.
- Since($y^{-\zeta_A} * \eta_A)^{w_A} \ modp = (g)^{k_A^{/}}$. Hence the pair $\langle \zeta_A, \eta_A, w_A \rangle$ can be viewed as the DSS-1 signature on the message $k_A^{/}$ in (g, p, q, y).

Then $\mathcal{B}$ succeeds in forging the signature if and only if $\mathcal{A}$ wins the above game.

Hence, if the Adversary $\mathcal{A}$ can impersonate a user with valid credential, a polynomial time algorithm can be constructed to forge the DSS-1 signature. But the assumption is that DSS-1 signature is existentially unforgeable. If this assumption holds, the probability $\mathcal{E}$ that $\mathcal{A}$ can impersonate a valid user in the protocol should be negligible in value.

Theorem: The proposed SH scheme is detector resistant under the Computational Diffie-Hellman (CDH) assumption in the random oracle model.

Proof: The CDH assumptions define as: Given a cyclic group G, a generator $g \in G$, and group element g^a, g^b the probability to compute g^{ab} is negligible.

The proposed SH scheme is detector resistant if no polynomially bounded adversary wins the following game against the challenger with non-negligible probability:

- The GA holds a key for DSS-1$\langle g, p, q, y, s \rangle$, and the challenger gets the$\langle g, p, q \rangle$, and gives it to the adversary $\mathcal{A}$.
- The Challenger asks the member ($ID_A, \zeta_A, \eta_A, w_A$),where $\eta_A = \alpha_A^{(k_A+1)} (mod\ pq)$, $\zeta_A = \gamma_A * (k_A + 1)\ (mod\ q)$ and $\omega_A = \left(\gamma_A{}^{-1} * (k_A + 1)^{-1} * \alpha_A^{k_A}\right)\ (mod\ q)$,for adversary $\mathcal{A}$.$(\alpha_U, \gamma_U)$is the DSS-1 signature on ID_A.
- The adversary $\mathcal{A}$ outputs $y^{/} \in \mathbb{Z}_p$.

The adversary wins the game if $y^{/} = y$.

Given an attacker $\mathcal{A}$ that wins the above game with probability ε then an attacker $\mathcal{B}$ can be constructed to successfully break the CDH assumption with probability ε.

- Given$\langle g, p, q \rangle$, $\mathcal{B}$ passes to $\mathcal{A}$.
- Given$\langle \zeta_A, \eta_A, w_A \rangle$, $\mathcal{B}$ can compute:
- $g^{\eta_A^{-1}} = g^{\alpha_A{}^{-(k_A+1)}}$ and $\left(\eta_A \ y^{-\zeta_A}\right)^{w_A} = g^{\alpha_A^{k_A}}$
- Let a be $\alpha_A{}^{-(k_A+1)} mod q$ and b be $\alpha_A^{k_A}\ mod q$ as defined in the CDH problem.
- $\mathcal{B}$ Send $\langle \zeta_A, \eta_A, w_A \rangle$to $\mathcal{A}$. Subsequently, $\mathcal{B}$ obtains y from $\mathcal{A}$.
- $\mathcal{B}$ Can compute $g^{\alpha_A^{-1}} = \left(\eta_A y^{-\zeta_A}\right)^{w_A \eta_A{}^{-1}}$.

Then $\mathcal{B}$ has successfully broken the CDH assumption with probability ε.

Thus if CDH assumption holds the probability ε that $\mathcal{A}$ can violate the detector resistance property should be a negligible value.

Security of SH based on DSS-2 can be discussed in similar manner.

4. COMPARISON TABLE

This section compares proposed schemes with two known schemes namely SH based on ElGamal and DSA by L. Zhou et al. [12].

The following table shows the number of multiplications **(M)**, the number of inversions **(I),** the number of exponentiation **(E),** and the number of hash evaluation **(H)** to complete the respective schemes.

In the Add User phase DSS -2 based SH scheme is as good as ElGamal and DSA. For multiplication DSS-1 based SH scheme is superior to ElGamal and DSA. For inversion, exponentiation and evaluation of hash proposed schemes are comparable to ElGamal and DSA.

Scheme	**Add User**				**Secret Handshakes**			
	M	**I**	**E**	**H**	**M**	**I**	**E**	**H**
ElGmal	2	1	1	1	8	4	10	4
DSA	2	1	1	1	10	8	16	6
DSS-1	1	1	1	1	12	8	8	2
DSS-2	2	1	1	1	12	6	8	2

During the Secret Handshake phase proposed schemes for exponentiation are superior to ElGamal and DSA. For inversion DSS-1 based SH scheme is comparable to DSA while DSS-2 based SH scheme is superior to DSA. For multiplication, our schemes are not as good as ElGamal and DSA. For evaluation of hash functions proposed schemes are superior to ElGamal and DSA.

5. CONCLUSION

In this paper two SH schemes based on a variation of DSA signature are proposed which are inspired by Zhou [12] and the comparison of the computational complexity of the new schemes with Zhou's schemes are shown. The proposed schemes are comparable to known schemes for most operations. An interesting future work is to construct a SH scheme for multiple group member from other existentially unforgeable signature and from other computationally infeasible problem.

6. ACKNOWLEDGMENT

The authors express sincere thanks to Professor Sunder Lal and Mr. Manmohan Singh Chauhan for their help and encouragement.

7. REFERENCES

[1] Ateniese G., Blanton M. and Kirsch J. 2007. Secret handshakes with dynamic and fuzzy matching. In Network and Distributed System Security Symposium, NDSS (2007), 159-177.

[2] Balfanz D., Durfee G., Shankar N., Smetters D., Staddon J., and Wong H. C. 2003. Secret handshakes from pairing

based key agreement. In IEEE Symposium on Security and Privacy, (2003), 180-196.

[3] Castelluccia C., Jarecki S., and Tsudik G. 2004. Secret handshake from ca-oblivious encryption. In ASIACRYPT, (2004), 293-307.

[4] Elgamal T. 1985. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, (July 1985), Vol. IT-31.

[5] Girault M. 1991. Self certified public keys. Proceeding in EUROCRYPT' 91, LNCS #547, Springer- Verlag, (1991), 490-497.

[6] Kulshrestha P., Pal A. K., and Chauhan M. S. 2015. Cryptanalysis of efficient unlinkable secret handshakes for anonymous communications. IOSR Journal of Computer Engineering, (2015), vol. 17, issue II, 71-74.

[7] Kulshrestha P. and Pal A. K. 2015. A new secret handshakes scheme with dynamic matching based on ZSS. International Journal of Network Security and its Applications, (2015), vol. 7, no. 1, p. 67-78.

[8] Okamoto E. and Tanaka K. 1989. Key distribution systems based on identification information". IEEE Journal on Selected Areas in Communications, (1989), 481-485.

[9] Vergnaud D. 2005. RSA-based secret handshakes. Proceedings in WCC 2005, LNCS #3969, Springer-Verlag, (2005), 252-274

[10] Wen Y., Zhang F. and Xu L. 2012.Secret handshakes from id-based message recovery signatures: a generic approach. Computers and Electrical Engineering Vol. 38, (2012), 96-104.

[11] Zheng Y. 1997. Digital signcryption or how to achieve cost (signature & encryption) <cost (signature) + cost (encryption)>. CRYPTO'97, LNCS # 1294, (1997), Springer-Verlag, 165-179..

[12] Zhou L., Susilo W. and Mu Y. 2006. Three round secret handshakes based on ElGamal and DSA. Proceedings in ISPEC (2006), LNCS #3903, Springer-Verlag, 332-342.