A Certified Three Way Authentication Scheme For Trust Establishment In Online Social Networking Communication System

Shresta N M *Dept of CSE, BIT, VTU, India,* **,Sowmya T** *Dept of CSE, BIT, VTU, India,* **Shravanthi T** *Dept of CSE, BIT, VTU,India,*

Abstract- **Existing Online social networks (OSNs) such as Facebook, Twitter etc, provided direct communication to unknown users leading to security and privacy issues on OSNs , we propose a certified three way authentication scheme for authenticating multiple users to improve the efficiency and security of OSNs. In the proposed authentication scheme, three batch authentication protocols are proposed, adopting the one-way hash function, proxy encryption, and certificates as the underlying cryptosystems. The hash-based authentication protocol requires lower computational cost and is suitable for resource-limited devices. The proxy-based protocol is based on asymmetric encryption and can be used to exchange more information among users. The certificate based protocol guarantees nonrepudiation of transactions by signatures. Without a centralized authentication server, the proposed authentication scheme therefore facilitate the extension of an OSN with batched verifications. In this paper, we formally prove that the proposed batch authentication protocols are secure against both passive adversaries and impersonator attacks, can offer implicit key authentication, and require fewer messages to authenticate multiple users. We also show that our protocols can meet important security requirements, including mutual authentication, reputation, community authenticity, nonrepudiation, and flexibility. With these effective security features, our framework is appropriate for use in P2P-based OSNs.**

Keywords- Authentication protocol, batch authentication, Online social networks (OSNs), Peer to peer (P2P).

I. INTRODUCTION

Online social networks (OSNs) such as Facebook, Twitter are increasingly popular services. People can share information and pictures with old acquaintances, as well as relationships with friends. It is estimated that half a billion registered users interact with friends over OSNs. However, the weak authentication and registration process of current OSNs may lead to some security attacks. With the rapid growth of OSNs, more valuable information is stored on OSNs. Hence, the privacy and security issues inherent to OSNs have attracted much attention[1]. Peer-to-peer (P2P) technology is considered in the design of next-generation OSNs. As described in, a P2P-based OSN consists of the following three levels:

1) The social network level represents members and their relationships;
2) The application service level implements the P2P-based application infrastructure;
3) The communication and transport level provides transport services over networks such as the Internet or mobile ad hoc networks.

Relying on the cooperation between a number of independent parties who are also OSN users[2], a decentralized P2P architecture can be adopted with merits, including strong privacy protection, better scalability, and a lowered requirement for continuous Internet connectivity. Furthermore, a P2P architecture can take advantage of real social networks and geographic proximity to support local services.1 P2P-based OSNs is a relatively new trend.

Existing protocols suffer from the following weaknesses.
1) Most of the current security protocols for P2P-based OSNs lack specific procedures.
2) In current security protocols for P2P-based OSNs, each user has to be authenticated by OOB methods, which may impede the extension speed of social networks.
3) Most of the existing protocols support only one-to-one authentication.
4) The existing protocols do not consider the restrictions of underlying devices such as computing power and memory limitations.

This paper proposes a framework to take advantage of the P2P architecture, including geographic proximity. Under the proposed framework, three batch authentication protocols are designed, using different cryptographic primitives [5], for different devices in P2P-based OSNs.
The novel contributions of this paper are listed as follows
• The proposed framework reduces the communication cost required for authenticating users.
• Due to their different security properties, the proposed protocols can be realized on a variety of devices such as personal digital assistants (PDAs), mobile phones, and laptops.
• By incorporating different trust levels, the proposed protocols allow a user with a high trust level to help

authenticate other users and achieve the extensibility of a social network.

• The proposed protocols support a one-to-many authentication, which is the basis of batch authentication, to simultaneously authenticate multiple users. To the best of our knowledge, this paper is the first study that offers oneto- many batch authentication in P2P-based OSNs [7]. The proposed protocols are proved to be capable of mutually authenticating communication peers and remain secure against passive adversaries.

II. OVERVIEW OF THE PROPOSED THREE WAY AUTHENTICATION

The proposed batch authentication protocols, which are composed of three roles, a requester U_R, an authenticator U_A, and a user group $\hat{}U$, are operated based on the following assumptions.

1) The requester U_R and authenticator U_A have negotiated a shared key by face-to-face preauthentication through a
Location-limited channel.
2) The authenticator U_A is trusted by all his/her friends who are involved in the batch authentication.
3) If two users U_X and U_Y are friends, they have shared a secret key K_{XY}.

In the proposed protocol, U_A helps U_R authenticate the user group $\hat{}U$, in which all users are friends of U_A. After successful authentication, U_R establishes a shared key KR_i with each user U_i in the group ($U_i \in \hat{}U$). We briefly explain our design concept by the following two cases.

In the first case, we introduce a user group with only one user U_1 ($\hat{}U = \{U_1\}$), as shown in Fig. 1(a). The message flow is given as follows.

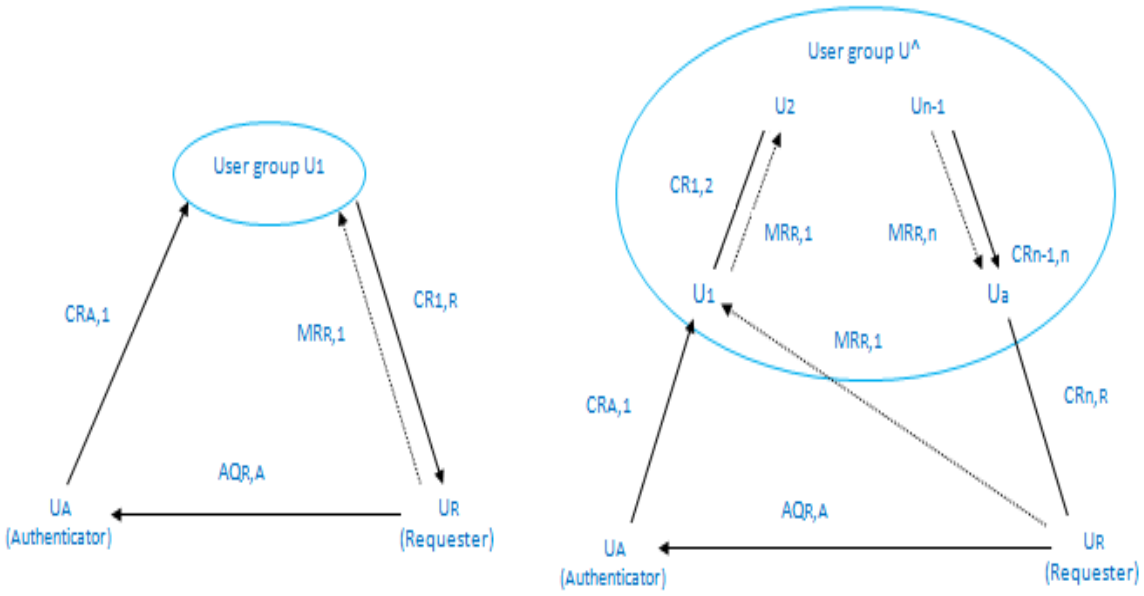
1) $U_R \rightarrow U_A : AQ_{R,A}$.
2) $U_A \rightarrow U_1 : CR_{A,1}$.
3) $U_1 \rightarrow U_R : CR_{1,R}$.
4) $U_R \rightarrow U_1 : MR_{R,1}$.

The requester U_R initiates a request to the authenticator U_A. Then, U_A helps contribute some parameters to U_R and U_1 at Steps 2 and 3. Finally, U_R replies a message ($MR_{R,1}$) at Step 4 for mutual authentication.
The second case scales up the user group to n users ($\hat{}U = \{U_1, U_2, \ldots, U_n\}$, and $|\hat{}U| = n$),2 as shown in Fig. 1(b). The message flow is given as follows.

1) $U_R \rightarrow U_A : AQ_{R,A}$.
2) $U_A \rightarrow U_1 : CR_{A,1}$.
3) $U_{i-1} \rightarrow U_i : CR_{i-1,i}$, where $2 \leq i \leq |\hat{}U|$.
4) $U|_{\hat{}U|} \rightarrow U_R : CR_{|\hat{}U|,R}$.
5) $U_R \rightarrow U_i : MR_{R,i}$, where $1 \leq i \leq |\hat{}U|$.

Similarly, U_R sends a request to U_A. The authentication requests (chain reply $CR_{i,i+1}$) are then passed through U_1, U_2, . . . to Un at Steps 2 and 3. At Step 4, $U|_{\hat{}U|}$ sends back the chain reply to UR. For mutual authentication, U_R sends $MR_{R,i}$ to users $U_i \in \hat{}U$.

Fig 1. Message flows of batch authentication for (a) only one member and (b) several members in case n = 3.



Parameters and Notations

Symbol	Description
q , p	Large primes such that p = 2q + 1.
G	The primitive root of prime q.
RK_i	The private key of U_i.
PK_i	The public key of U_i. PK_i is used for ElGamal encryption such that $PK_i = g^{RK_i} \bmod p$.
E_{KRA}	The symmetric encryption function with secrete key K_{RA}.
B_n	Representing n positive integers that are pairwise relatively primes used in CRT.
H()	Collision – resistant one-way hash functions with length of l, $H() : \{0,1\}^* \rightarrow \{0,1\}^l$
H (r,msg)	A length extension hash function with r-times hash operation for message msg. For instance, $H(r,msg) = H(msg) \,||\, H^2(msg) \ldots \,||\, H^r(msg)$, where r is determined by the required lengths.
MAC	The message authentication code generated by a keyed hash function.
Requester (U_R)	A user who requests batch authentication.
Authenticator (U_A)	A user who assists U_R for the batch authentication.
$\hat{G}$	The set of all participants involved in the batch aunthentication. $\hat{G}=\{U_R, U_A, U_1, U_2, \ldots, U_n\}$
$\|\hat{G}\|$	The number of all participants involved in the batch authentication
$\hat{U}$	A user group to be authenticated $\hat{U}=\hat{G} - \{U_A, U_R\}=\{U_1, U_2, \ldots, U_n\}$
$\|\hat{U}\|$	The number of user group.
U I D	The set of $\hat{U}$'s identities in this batch aunthentication session $UID = \{ID_1, ID_2, \ldots, ID_n\}$
N_i	A nonce picked by U_i.
W_i, t_i	The random numbers that have the same bit lengths as H().
S	A random number serving as a seed of ElGamal proxy encryption key.
T_i	The U_i's certificate.
T	The set of $\hat{U}$'s certificates $T=\{T_1, T_2, \ldots, T_n\}$
KP_R	The set of key parameters sent from U_R to $\hat{U}$ for key agreement. $KP_R = \{g^{m1}, g^{m2}, \ldots, g^{mn}\}$.
KP_0	The set of key parameters sent from $\hat{U}$ to U_R. $KP_0 = \{g^{n1}, g^{n2}, \ldots, g^{nn}\}$.
$QR_{R,A}$	The authentication request message transmitted from U_R to U_A.
CR	The chain-reply messages passed through users in a user group.
M R	The reply messages for mutual authentication.

III. PROPOSED PROTOCOLS

A. Message Integrity verifier protocol

Step 1

U_R sends $AQ_{R,A}$ to U_A . $AQ_{R,A}$ is composed of U_R 's identification (ID_R), a nonce (N_R), the user group identification (UID = $\{ID_1 , ID_2 , \ldots , ID_{|U|}\}$), and the parameters of key agreement ($KP_R = \{g^{m1}, g^{m2}, \ldots, g^{m_{|U|}}\}$, where $m_i \in Z_p$). The nonce is protected by a secret key K_{RA} that is shared by U_R and U_A . The group identification and key parameters are protected by the nonce. In addition, a message authentication code $MAC_R = H(ID_R , \{KR_A \oplus N_R\}, U\ ID \oplus H(r, (N_R +1)), KP_R \oplus H(r, (N_R + 2)), (N_R + 3))$ is attended to ensure the integrity of message.

Step 2

Upon the receipt of $AQ_{R,A}$, U_A derives N_R by performing $K_{RA} \oplus \{K_{RA} \oplus N_R\}$ and checks the validity of MAC_R . If $AQ_{R,A}$ is correct, the following steps are implemented.

U_A randomly generates an initial value h0 and a sequence of random numbers w_i (for $0 \le i \le \hat{}|U| - 1$). Then, U_A constructs and maintains a chain of one-way hash values ($h_i = H(h_{i-1} \oplus w_{i-1})$ for $1 \le i \le |U|$)

U_A derives the user group identification UID and the key parameters KP_R by N_R.

U_A computes V_0 for U_R , where

$V_0 = \{IDA, H(r, (KRA||t0))((h0||H(KRA)||NA|| U_{j=0}^{|\hat{U}|-1} wj), t0)\}$

Note that the unequal-bit-length problem can be solved by the specific length extension hash function H(r, msg) and V0 should be regarded as a single element from the view of calculation. As mentioned in the previous section, KRA is the shared key between UR and UA , NA and t0 are random challenges from UA , and $U_{j=0}^{|\hat{U}|-1}$ wjis a concatenation of w0 , w1 , . . . , $w_{|U|-1}$.

UA also computes Vi for Ui ∈ U , ($1 \le i \le \hat{}|U|$), where

$V_i = \{IDA, H(r, (KRAi||ti)) \oplus ((hi||H(KAi)||NRNA||gmi), ti)\}$

In V_i ($i \neq 0$), g^{mi} is used for negotiating session keys K_{Ri} between UR and U_i in the end of the batch authentication.

To eliminate the bandwidth requirements, we adopt the Chinese reminder theory (CRT) [17] to accommodate messages in a single message. Let B0 , B1 , B2 , . . . , B|U | be |U |+1 positive integers that are pairwise relative primes and A0 , A1 , A2 , . . . , A|U | be the multiplicative inverses of B0 , B1 , B2 , . . . , B|U | . UA

computes a common solution X for the following congruous equations:
X ≡ V0 mod B0 (for UR)
X ≡ Vimod Bi (for Ui ∈ U, 1 ≤ i ≤ |U |).
By the CRT , we have
$X = (\sum_{i=0}^{|\hat{U}|} L/Bi \times Vi \times Ai) mod\ L$, where L = $\prod_{i=0}^{|\hat{U}|} Bi$
$A_i \times (L/B_i) \bmod B_i \equiv 1$.
UA calculates M ACA = H(X, NR + NA) and sends the chain reply CRA,1 = {X, M ACA } to the first user in the group (U1).

Step 3

After receiving $CR_{A,1}$ = {X, M AC_A }, the following steps are implemented.
U_1 retrieves V_1 by calculating X mod B_1 . Next, U_1 obtains
H(r, (K_{A1}|| t_1)) ⊕ (h_1|| H(K_{A1}) N_R + N_A|| g^{m1}) and t_1 from V_1 . U_1 then uses the shared keys K_{A1} and t1 to derive h_i , H(K_{A1}), N_R + N_A , and g^{m1} .
The validity of V1 and $CR_{A,1}$ can be verified by H(K_{A1}) and M AC_A , respectively.
The request is dropped when any invalidity is detected. Then, U1 computes
M1 = H((NR +NA)⊕ h1) and adds a key parameter g n1 to KPU .ˆ
U1 generates M AC1 = H(M1 , X, KPU(NR +N_A)) and sends message CR1,2 = {M1 ,X, KPU , MAC1 } to the next group user
U2 .For Ui ∈ U (2 < i ≤ |U |), the following steps repeat until the
chain reply passes through all group users.
U_i get $CR_{i-1,i}$ = {M_{i-1} , X, KP_U ,M AC_{i-1} } from U_{i-1} .
U_i extracts V_i by X mod B_i . Similarly, U_i can obtain h_i ,H(K_{Ai}), N_R + N_A , g^{mi} from V_i by the shared key K_{Ai} and random challenge t_i.
The validity of V_i and $CR_{i-1,i}$ can be verified by H(K_{Ai}) and
MAC_{i-1} , respectively When any invalidity is detected, the request is dropped, and U_i reports the failure to U_A . Then, U_i computes M_i = M_{i-1} ⊕ H((N_R + N_A) ⊕ h_i) and adds a key parameter g ni to KP_U
U_i generates M AC_i = H(M_i , X, KP_U , (N_R + N_A)) and sends $CR_{i,i+1}$ = {M_i , X, KP_U ,MAC_i } to the next group user U_{i+1}

Step 4

Upon the receipt of the last chain reply $CR_{|U|,R}$ = {$M_{|U|}$, X, KP_U , M $AC_{|U|}$ }, the following steps are implemented.
U_R computes X and B_0 and obtains V_0 . With the shared key K_{RA} and random challenge t_0, U_R derives h_0 , H(K_{RA}), N_A , and $\bigcup_{j=0}^{|U|-1} Wj$ from V_0.
Similarly, the authenticity of V0 and M $AC_{|U|}$ can be verified by H(K_{RA}) and M $AC_{|U|}$ .If validated, U_R derives hi (1 ≤ i ≤ |U |) by h0 $\bigcup_{j=0}^{|U|-1} Wj$ from V_0.
U_R also computes $M_{|U|}$ = H((N_R + N_A) h_1) ⊕ H((N_R + N_A)⊕ h_2) ⊕ · · · ⊕ H((N_R +N_A) ⊕ $h_{|U|}$) and compares it with $M_{|U|}$. If matched, the user group U is authenticated. Otherwise, at least one of the users fails the authentication, and the session terminates.\
After the successful batch authentication, U_R computes session keys SK_{Ri} = (g^{ni})^{mi} for ˆU_i (1 ≤ i ≤ |U |).
For mutual authentication, U_R calculates replie S_i = H((N_R + N_A + 1) ⊕hi) mod B_i . Again, by applying the CRT [17], we can find a common solution for
Y ≡ S1 mod B1
Y ≡ S2 mod B2
.
.
.
Y ≡ S|U | mod B|U |
Then, U_R generates M AC_R = H(Y, (N_R +N_A)) and sends M $R_{R,i}$ = {Y, M AC_R }toˆU_i (1 ≤ i ≤ |U |). In the case that U_R cannot directly reach U_i , U_A can be involved to help forward the messages.

Step 5

After receiving M $R_{R,i}$ from U_R (or U_{i-1}), the following steps are implemented.
U_i first checks the validity of MAC_R.
The session is dropped if MAC_R fails the check otherwise, U_i computes S_i = Y mod B_i and checks the equality of S_i , where S_i =ˆH((N_R + N_A + 1) ⊕ h_i) (1 ≤ i ≤ |U |). If the equality holds, U_R is authenticated; otherwise the session is terminated.
After the successful batch authentication, U_i computes the session key SK_{Ri} = (g^{mi})^{ni} Subsequent communications between U_R and U_i can be protected by SK_{Ri}

B. Assymetric proxy encryption protocol

Step 1

The requester U_R sets the shared key K_{RA} as the seed of the ElGamal proxy encryption key and then starts the batch authentication protocol as follows.
U_R sends the authentication request $AQ_{R,A}$ = {ID_R , {C_1, C_{2R} }, K_{RA} ⊕ N_R , UID ⊕ H(r,(N_R + 1)), MAC_R } to U_A .

Step 2

Upon the receipt of $AQ_{R,A}$, the following steps are implemented.
U_A first derives N_R by the shared key K_{RA} and extracts the UID by N_R.

Next, U_A verifies MAC_R and checks whether each U_i 's trust level that is maintained by himself is higher than the predefined trust threshold. If one of the verifications fail, U_A drops this session. Otherwise, U_A computes V_0 for U_R and V_i for $U_i \in U$ as

$V_0 = \{ ID_A, E_{KRA}(N_A, \sum_{J=1}^{|U|} K_{AJ}, H(K_{RA}))\}$

$V_i = \{ ID_A, E_{KAi}(K_{RA}+N_R+N_A, \sum_{\substack{J=1 \\ j \neq i}}^{|U|} K_{AJ}, H(K_{Ai}))\}$

Similarly, by applying the CRT [17], we can accommodate all replies in a single message as

$X \equiv V0 \bmod B0$ (for UR)

$X \equiv V1 \bmod B1$ (for U1)

.

.

$X \equiv Vi \bmod Bi$ (for $Ui \in U$).

As mentioned in section IV-A, by the CRT, we obtain

$X = (\sum_{J=1}^{|U|} \frac{L}{Bi} * Vi * Ai) \bmod L.$

based on the ElGamal proxy encryption schema, U_A calculates

$C2A = (C2R \times C1NR) \bmod p$

$= \xi\beta\, r \times g\, r(NR) \bmod p$

$= \xi g\, (KRA)r \times g\, r(NR) \bmod p$

$= \xi g\, r(KRA +NR) \bmod p$

UA generates the message authentication code to protect the integrity of the message, where

$MAC_A = H(C_1, C_{2A}, X, (K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{AJ}))$

U_A sends $CR_{A,1} = \{C_1, C_{2A}, X, M\ AC_A\}$ to U_1

Step 3

After receiving $CR_{A,1}$, the following steps are implemented.

U_1 extracts $V_1 = X \bmod B_1$ and decrypts

$E_{KAi}(K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{Aj},\ H(K_{A1}))$ by K_{A1}.

U_1 verifies the integrity of V_1 and $CR_{A,1}$ by checking $H(K_{A1})$ and MAC_A respectively. The request is dropped when any invalidity is detected.

U_1 calculates

$C_{21} = C_{2A} \times C_1^{(K_{A1})} \bmod p$

$= \xi g^{r}(K_{RA} + N_R) \times g^{r(K_{A1})} \bmod p$

$= \xi g^{r}(K_{RA} + N_R + K_{A1}) \bmod p$

U_1 selects the parameter of the session key $KP_U = \{g^{n1}\}$.

Because K_{A1} is shared with U_A and U_1,only legitimate U_1 can decrypt V_1 , add K_{A1} with

$K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{Aj}$, and compute the message authentication code

$MAC_1 = H(C_1, C_{21}, X, K_{PU}, K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{Aj})$.

U_1 sends $CR_{1,2} = \{C_1, C_{21}, X, KP_U, MAC_1\}$ to U_2

For $U_i \in U$ $(2 < i \leq |U|)$, the following steps repeat until the chain reply passes through all group users.

Upon the receipt of $CR_{i-1,i}$, U_i derives $Vi = X \bmod Bi$ and decrypts

$EK_{Ai}(K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{Aj}, H(K_{Ai}))$ by K_{Ai}

U_i checks the validity of $H(K_{Ai})$ and MAC_{i-1} . The session is dropped if any invalidity is detected; otherwise

Upon the receipt of $CR_{i-1,i}$, U_i derives $V_i = X \bmod B_i$ and decrypts $EK_{Ai}(K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K_{Aj}, H(K_{Ai}))$ by K_{Ai}.

U_i checks the validity of $H(K_{Ai})$ and MAC_{i-1} . The session is dropped if any invalidity is detected. otherwise, U_i computes

$C2_i = C2_{i-1} \times C_1^{(KAi)} \bmod p = (\xi g^{r(K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K^{Aj})} g^{r(Kai)}) \bmod p = (\xi g^{r(K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K^{Aj})}) \bmod p$

U_i selects a parameter of session key g ni and attaches it to KPU . Then, Ui generates

$MAC_i = H(C_1, C2_i, X, KP_U, K_{RA} + N_R + N_A + \sum_{J=1}^{|U|} K^{Aj})$.

U_i sends $CR_{i,i+1} = \{C_1, C2_i, X, KP_U, MAC_i\}$ to the next user U_{i+1} .

Step 4

After receiving $CR_{|U|,R}$, the following steps are implemented.

U_R computes$V_0 = X \bmod B_0$ and decrypts V_0 by K_{RA} to obtain$(N_A + \sum_{J=1}^{|U|} K_{Aj}, H(K_{Ai}))$.

U_R checks the validity of V_0 by $H(K_{RA})$ and $MAC_{|U|}$. If valid, U_R computes

$\xi' = C2_{|U|} \times (C_1^{(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})})^{-1}$

$= (\xi g^{r(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})}) \times (g^{r(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})})^{-1} \bmod p$

Once U is authenticated, U_R can extract the key parameters of session key g^{ni} from KP_U and negotiate session keys with $U_i \in U$. The session keys can be obtained by

$SK_{Ri} = (g^{ni})^{mi}$.

For mutual authentication and key agreement, U_R computes $C2' = C2_{|U|} \times C1^{NA} \bmod p$ and $MAC'_R = H(C2'.K_{RA}+N_R+N_A+\sum_{J=1}^{|U|} K^{Aj})$.Then, the message {C2 , M ACR } is sent to Ui $(1 \leq i \leq |U|)$. In the case that UR cannot directly reach Ui , UA can be involved to help forward the messages.

Step 5

After receiving $MR_{R,i}$ from U_R , the following steps are implemented.

U_i verifies the authenticity of MAC_R and computes

$\xi^n = C2' \times (C_1^{(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})})^{-1} \bmod p$

$= (\xi g^{r(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})}) \times (g^{r(KRA+NA+\sum_{J=1}^{|U|} K^{Aj})})^{-1} \bmod p.$

U_i also checks whether ID_R is included in ξ^n If yes, U_R is authenticated.

Then, U_i generates the session key $SK_{Ri} = (g^{mi})^{ni}$ to protect the communication between U^R and U^i .

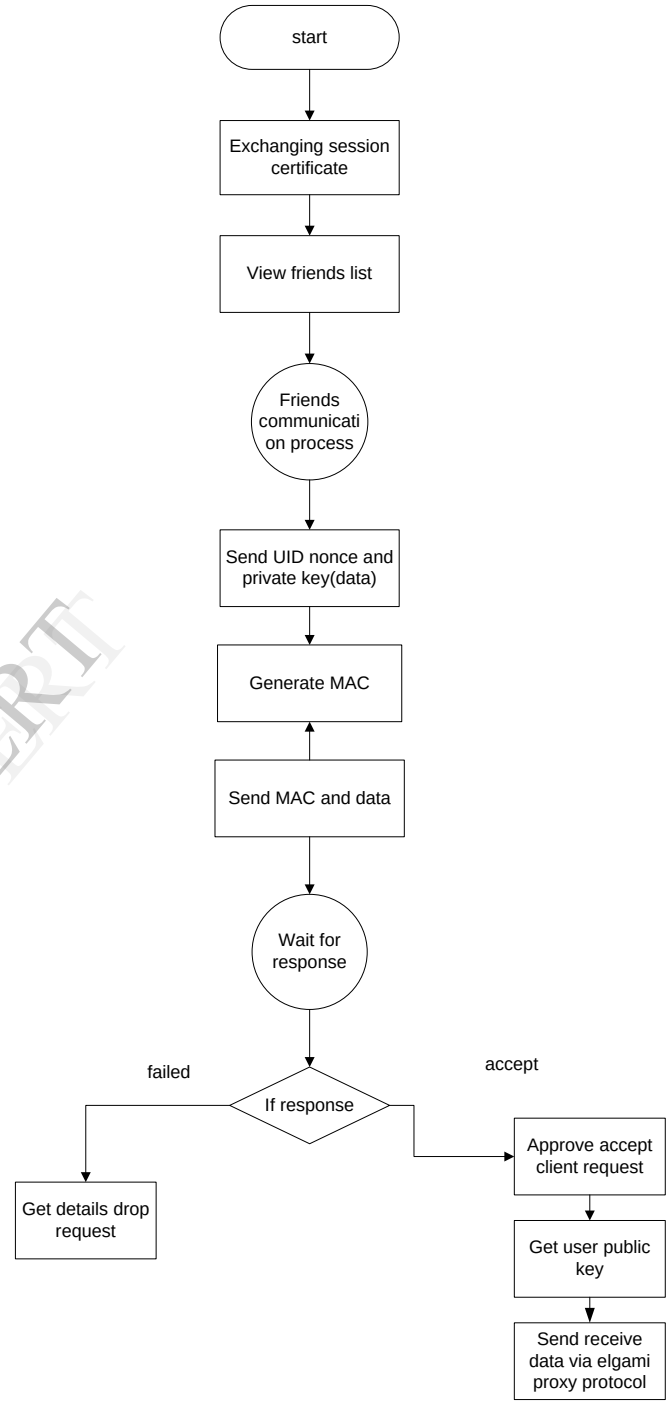
C. Online/Offline certification management Protocol

The Online/Offline certification management protocol is proposed to guarantee the nonrepudiation of a transaction. In this protocol, we adopt the Shamir–Tauman online/offline signature [1] to enhance the security property. The authenticator UA , behaving as a local trusted certificate authority, helps deliver and verify certificates for the group users (Ui ∈ U).

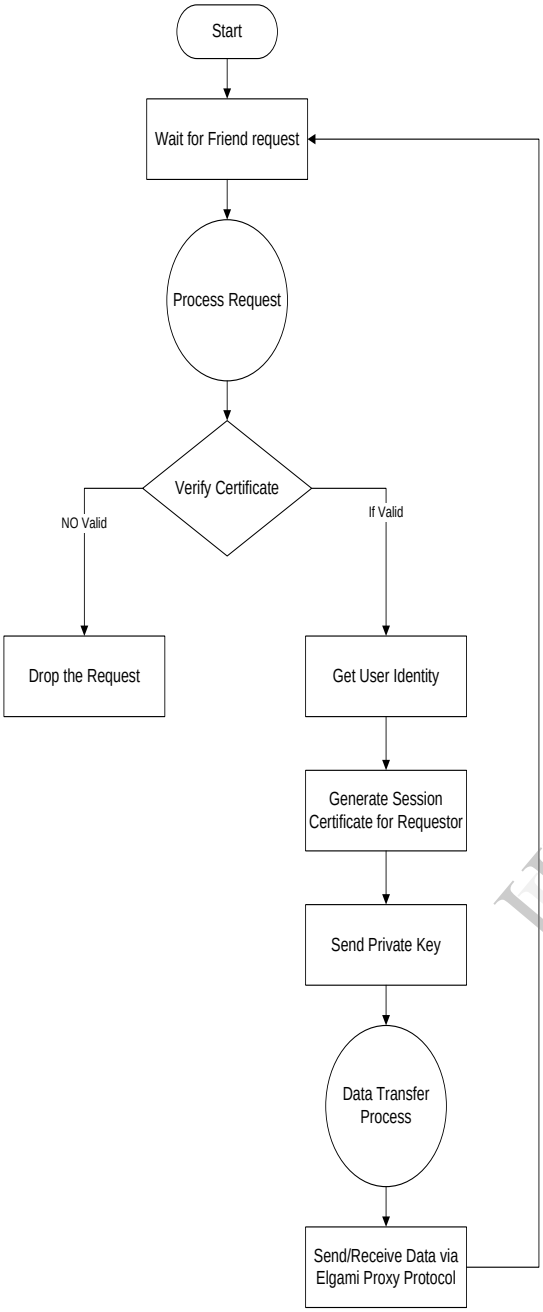
1) $U_R \rightarrow U_A : AQ_{R,A} = \{P\ K_A\ \{ID_R , N_R , U\ ID\}, MAC_R\}$.
2) $U_A \rightarrow U_R : AR_{A,R} = \{P\ K_R\ \{N_R + 1, T\}, M\ AC_A\}$.
3) $U_R \rightarrow U_1 : CR_{R,1} = \{C_1, X, MAC_A\}$.
4) $U_{i-1} \rightarrow U_i : CR_{i-1,i} = \{C_1, C2_i , X, KP_U , M\ AC_i\}$,where $2 \leq i \leq |U|$
5) $U_{|U|} \rightarrow U_R : CR_{|U|,R} = \{C_1, C2_{|U|} , X, KP_U , MAC_{|U|}\}$.

IV. FLOW CHART

A. Requestor flow chart



B. Users Flow graph



CONCLUSION

In this paper, we have designed Certified three way authentication schema which establish trust management for OSNs. We have also designed three way authentication protocols using the one-way hash function, ElGamal proxy encryption, and certificates for different situations and purposes. The massage integrity verifier protocol adopts light weight cryptosystems to reduce the computational costs. To offer higher security properties, the asymmetric proxy encryption protocol and Online/Offline certification management protocol are based on asymmetric encryptions and signature methods to fulfill the security requirements of sensitive transactions.

REFERENCES

[1] C. Zhang, J. Sun, X. Zhu, and Y. Fang, "Privacy and security for online social networks: Challenges and opportunities," *IEEE Netw.*, vol. 24, no. 4, pp. 13–18, Jul./Aug. 2010.

[2] D. Niyato, P. Wang, W. Saad, and A. Hjorungnes, "Controlled coalitional games for cooperative mobile social networks," *IEEE Trans. on Vehi. Tech.*, vol. 60, no. 4, pp. 1812–1824, May 2011.

[3] M. Ge, K.-Y. Lam, X. Wang, Z. Wan, and B. Jiang, "VisualSec: A secure message delivery scheme for online social networks based on profile images," in *Proc. IEEE GLOBECOM*, 2009, pp. 1–6.

[4] S. Buchegger and A. Datta, "A case for P2P infrastructure for social networks Opportunities and challenges," in *Proc. WONS*, 2009, pp. 161–168.

[5] S. Buchegger, D. Schioberg, L. H. Vu, and A. Datta, "PeerSoN—P2P social networking: Early experiences and insights," in *Proc. SocialNets*, 2009, pp. 46–52.

[6] L. A. Cutillo and R.Molva, "Safebook: A privacy-preserving online social network leveraging on real-life trust," *IEEE Commun. Mag.*, vol. 47, no. 12, pp. 94–101, Dec. 2009.

[7] U. Lee, J. Sewook, C. Dae-Ki, A. Chang, C. Junho, and M. Gerla, "P2P content distribution to mobile Bluetooth users," *IEEE Trans. Veh. Technol.*, vol. 59, no. 1, pp. 356–367, Jan. 2010.

[8] S. Gokhale and P. Dasgupta, "Distributed authentication for peer-topeer networks," in *Proc. Appl. Internet Workshops*, Jan. 27–31, 2003, pp. 347–353.

[9] H. Lee and K. Kim, "An adaptive authentication protocol based on reputation for peer-to-peer system," in *Proc. Symp. Crypto. Info. Sec.*, 2003, pp. 661–666.