Smart Grid with Multilayer Consensus ECC based PAKE Protocol

Meera Jadhav[1], Shilpa R Patil[2], Madhusudhan M.V[3]

[1,2,3]Asst Professor,Dept of CSE, SaIT, Bangalore-India, VTU University

Abstract—**The paper aims in providing a securely communicating of data between the devices which are nothing but appliance of Home Area Network (HAN) by a set of controller outside the HAN. The packets generated by the controller should be delivered fast without any interruption. The role of an HAN controller is like a gateway, whose role is to filter the incoming packet. Due to this HAN controller, the data in the packets should not be modified & it should not cause any delay in encryption & decryption of the packets. Based on the level of security and quality of service, we design the protocol with an Elliptic Curve Cryptography (ECC) approach. Password Authenticated Key Exchange (PAKE) protocol is improved & implemented. This consists of two phases. We propose an ECC version of PAKE & later extend this protocol to a multilayer consensus ECC based Password Authenticated Key Exchange (PAKE) protocol for smart grid. This protocol uses only one hash function & it utilizes a primitive password between the appliance and HAN controller. The four layer individual consensus password-authenticated symmetric key established between the appliance and upstream controllers during only 12 packets.It provides high security level with small key size when compared to RSA algorithm key size which is big with comparison to our proposed protocol. Proposed protocol is resistant against many attacks. It improves & reduces the delay caused by the security process by more than one half. Finally we show that our approach decreases the number of packets & improves the security in smart grid.**

Key words—**Access control, BAN, Consensus, ECC, HAN, NAN, EPAKE, Security, Smart grid.**

I INTRODUCTION

Due to the rapid development in the area of smart grid (SG) environment in recent years have led to many technical issues have drawn the attention for systems, communications and security research communities. Due to the wide spread of smart grid to wireless technology has led to different levels of harm to the SG system. The major challenge for smart grid system is security and privacy levels. Among this security plays a major key challenge for SG [2]. In this paper, we propose a protocol for key agreement to securely access control in a hierarchical architecture for the SG communication infrastructure with different layers between smart appliances in users premises and upstream controllers of the Home Area Networks (HANs), Building Area Networks (BANs) and Neighbor Area Networks (NANs) & Smart Grid Central Controllers (SGCC), where these devices are located in distribution networks or substations [1]. HAN controller is a smart meter (SM) that serves as the gateway for user premises. To access and control the smart appliances in user's premises this protocol provides a secured means for controllers upstream of the HAN controllers. Various existing controlling commands that are sent to a smart appliance from outside the HAN have been considered in [4].

For instance, a NAN controller which is located outside a HAN may supervise electric charging of a plug-in electric vehicle located inside the HAN. SGCC may need to remotely turn off low-priority high-demand appliances in the case of disaster or an emergency. In such cases, HAN controllers should not interfere & should not have much delay in decrypting & re-encrypting the corresponding packets. So we need to address the appropriate secrecy level in the SG control system design while providing the quality of service (QoS) required in terms of keeping the command response delay within an acceptable limit.

In this paper, we propose two protocols. One is ECC based Password Authenticated Key Exchange (PAKE) protocol & the second one is Multilayer Consensus EPAK protocol which is developed for communications in the SG control system.

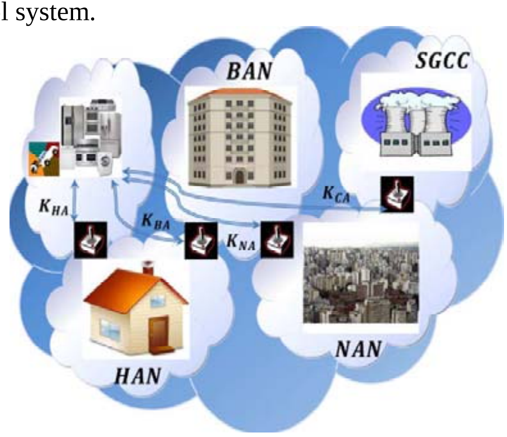


Fig 1: Symmetric key

Fig 1. shows the communication between four layers which is home appliance A_N, Home Area Networks controller H_C, Building Area Networks controller B_C, Neighbor Area Networks controller N_C & Smart Grid Central Controllers C_C. The SG controllers with the hierarchical architecture share common secrets with each other are designed to be authenticated. The controllers are authenticated to both upstream and downstream & can communicate in a secure fashion with neighbors. Any smart appliance which wants to join HAN it needs to share a password with HAN controller for it to be trusted in the HAN. In our proposed protocol each controller needs to setup a secure & private communication channel with home appliance with any other relay controller that just acts as a part of communication connection without participating in the security operations. Primitive password is shared between hope appliance & home controller, four individual consensuses password-authenticated symmetric keys between home appliance & upstream controllers. A symmetric key agreement is based on the Diffie and Hellman algorithm. Sharing a pre-shared password for mutual authentication is known as the PAKE protocol. T he proposed protocol is based on the Diffie & Hellman algorithm [3].

II. Related Work and Background Review

Many solutions have been proposed over years for symmetric key & asymmetric key. Many of them were based on the Diffie-Hellman algorithm for symmetric key. For asymmetric key ElGamal algorithm & RSA algorithm are used. Crypto-system has been proposed to resists attacks against like Man-In-The-Middle (MITM). Many protocols were proposed to prevent against MITM. One among them is Simple Authentication and Key Agreement (SAKA) which is resistant against passwords guessing attack. In this protocol both parties convert their shared password into a number. In SAKA each party randomly selects a number & multiplies it to the shared number to be used in the Diffie-Hellman algorithm. Another protocol PAKE is used for smart card (SC) which identifies & delivers an entity server with mutual authentication. It requires SC managements & supports one smart card per device. This SC is used for authentication between both parties which limits its usefulness in smart grid systems.

1) X.1035 Standard

It specifies a protocol, establishing a symmetric cryptographic key via Diffie-Hellman exchange that ensures a mutual authentication between both parties. Diffie-Hellman exchange provides a perfect forward secrecy. With this authentication method the exchange is protected from the man-in-the-middle attack. This authentication purely relies on a pre-shared secret (e.g., password), which is protected which remains unrevealed to an eavesdropper which prevents an off-line dictionary attack. Thus, this protocol can be used in a wide variety of applications based on password sharing. There are many methods used to resolve such attacks. Some of them rely on public key cryptography & others rely on shared key cryptography (passwords).

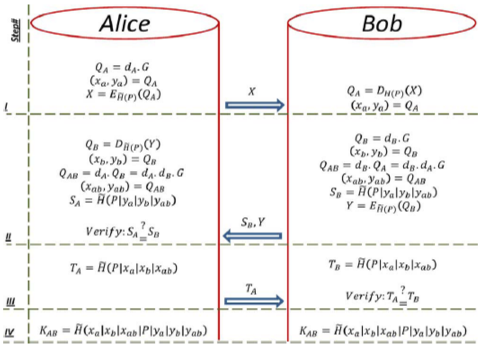

Step#
Alice
Bob
$Q_A = d_A.G$
$(x_a, y_a) = Q_A$
$X = E_{\tilde{H}(P)}(Q_A)$
X
$Q_A = D_{H(P)}(X)$
$(x_a, y_a) = Q_A$
I
$Q_B = D_{\tilde{H}(P)}(Y)$
$(x_b, y_b) = Q_B$
$Q_{AB} = d_A.Q_B = d_A.d_B.G$
$(x_{ab}, y_{ab}) = Q_{AB}$
$S_A = \tilde{H}(P|y_a|y_b|y_{ab})$
$Q_B = d_B.G$
$(x_b, y_b) = Q_B$
$Q_{AB} = d_B.Q_A = d_B.d_A.G$
$(x_{ab}, y_{ab}) = Q_{AB}$
$S_B = \tilde{H}(P|y_a|y_b|y_{ab})$
$Y = E_{\tilde{H}(P)}(Q_B)$
S_B, Y
$Verify: S_A \overset{?}{=} S_B$
II
$T_A = \tilde{H}(P|x_a|x_b|x_{ab})$
$T_B = \tilde{H}(P|x_a|x_b|x_{ab})$
T_A
$Verify: T_A \overset{?}{=} T_B$
III
IV
$K_{AB} = \tilde{H}(x_a|x_b|x_{ab}|P|y_a|y_b|y_{ab})$
$K_{AB} = \tilde{H}(x_a|x_b|x_{ab}|P|y_a|y_b|y_{ab})$

Fig. 2. PAKE Protocol: X.1035 Standard.

PAKE protocols advantages are listed below:

- Provides strong key exchange with weak passwords.
- Foils the man-in-the-middle attack.
- Provides explicit mutual authentication.
- Ensures perfect forward secrecy.

Password-authenticated key exchange (PAK) protocol that meets the following requirements:

- ✓ Provides mutual authentication based on a pre-shared password.
- ✓ Provides protection against a man-in-the-middle (MITM) & against offline dictionary attacks.

PAKE protocol presented in the X.1035 standard assumes that the two parties share a password (*pw*). Using D-H algorithm X.1035 standard constructs a symmetric cryptographic which is four-phase mutual authentication that uses D-H values *g* & *p* and five shared hash functions H_1-H_5. Fig. 2. shows the following phases ID_A & ID_B are the IDs of two parties Alice & Bob respectively. $P=(ID_A|ID_B|pw)$, and R_A & R_B are the random numbers chosen by them respective:

The series of steps are shown below:

Step 1: Alice obtains X via (I) and forwards it to Bob:

$$X = H_1(P) . (g^{R_A} \text{ mod p}) \qquad (1)$$

On other side, Bob extracts "g^{R_A}mod p" from X by (2)

$$\frac{X}{H_1(P)} = \frac{H_1(P).(g^{R_A} \text{ mod p})}{H_1(P)} = g^{R_A} \text{ mod p} \qquad (2)$$

Step 2: Bob computes Y and S_B following (3) and (4) and send them to Alice

$$Y = \text{H}_2(\text{P}) . (g^{R_B} \text{mod p}) \qquad (3)$$

$$S_B = H_3 \text{ (P}|g^{R_A}\text{mod p}|g^{R_B}\text{mod p}|g^{R_A R_B}\text{mod p)} \qquad (4)$$

Alice also similar obtains "g^{R_A}mod p" from *Y* per (5) and then calculate S_A per (6) for the verification

$$\frac{Y}{H_2(P)} = \frac{H_2(P).(g^{R_B} \text{ mod p})}{H_2(P)} = g^{R_B} \text{ mod p} \qquad (5)$$

$$S_A = H_3 \text{ (P}|\ g^{R_A}\text{mod p}|g^{R_B}\text{mod p}|g^{R_A R_B}\text{mod p)} \qquad (6)$$

Step 3: Alice computes T_A via (7) and sends it to Bob

$$T_A = H_4 \text{ (P}|\ g^{R_A}\text{mod p}|g^{R_B}\text{mod p}|g^{R_A R_B}\text{mod p)} \qquad (7)$$

Then, Bob calculate $T_B$via (8) for the verification:

$$T_B = H_4 \text{ (P}|\ g^{R_A}\text{mod p}|g^{R_B}\text{mod p}|g^{R_A R_B}\text{mod p)} \qquad (8)$$

Step 4: The verification of S_A, S_B, T_A & T_B by Alice and Bob means a mutual authentication derived by *pw*. Using the above values, Alice and Bob can obtain the symmetric key *K* through (9):

$$K = H_5 \text{ (P}|\ g^{R_A}\text{mod p}|g^{R_B}\text{mod p}|g^{R_A R_B}\text{mod p)} \qquad (9)$$

2) EPAK: ECC-Based Password Authenticated Key-exchange Protocol

EPAK protocol that is the designed as an ECC version of PAKE protocol presented in the X.1035 standard. Here we consider there is a pre-shared password (pw) agreement between Alice & Bob [10]. We define P= ($ID_A|ID_B|pw$) as similar to the X.1035 standard. Both Alice & bob have the knows the e EC parameters set {a,b,p,G,n,h} & H the hash function. Table 5.1 presents the list of parameters and their definitions [6], [7], [8].

TABLE 1 EPAK parameters

Parameter	Description
a & b	Two field elements that define the equation EC.
p	The field size
G	An ECC point that generates the subgroup of order n
n	The order of the point G
h	The order of EC divided by n
xw & yw	Two elements of the finite field of size p (in the range of $[0, p-1]$), which are the x & y coordinate of point W
dw	Private key of party W,which are integers in range [2,$n-1$]
Qw	Public key of party W
Sw & Tw	Verifiers generated by party W
$U = E_{\text{ke}}\ (V)$	U in encryption of V using key K_{e}
$V = D_{\text{kd}}\ (U)$	V in encryption of U using key K_{d}

EPAK Protocol: EPAKE protocol has the following steps:

Step 1:

a) *Alice*: Alice is the initiator, were she picks a random number d_A€ [2,n-1] (as her private key) & multiply it to G t which is a group generator to obtain her public key Q_A via (10) & with EC point as (x_a , y_a) via (11). Finally, she computes $\widetilde{H}(P)$ *to* obtain a symmetric key with which she encrypts Q_A as X via (12) than sends it to Bob.

$$Q_A = d_A . \mathrm{G} \quad (10)$$

$$(x_a, y_a) = Q_A \quad (11)$$

$$Q_{A=} E_{\widetilde{H}(P)}(Q_A) \quad (12)$$

b) *Bob*: Once Alice the packets it reaches Bob. Bob uses $\widetilde{H}(P)$ to decrypt and obtain Q_A following (13), & appropriate EC point aligned with the value shown by (11)

$$Q_{A=} D_{\widetilde{H}(P)}(\mathrm{X}) \quad (13)$$

Step 2:

c) *Bob*: d_B a random number which Bob chooses € [2, n-1] (as his private key) & then multiply this number with group generator G in order to obtain his public key Q_B via (14). Calculates EC point (x_b ,y_b) aligned with the Q_B based on the value from (15):

$$Q_B = d_B . \mathrm{G} \quad (14)$$

$$(x_b, y_b) = Q_A \quad (15)$$

Now, we multiplies his private key to Alice's public key to obtain Q_{AB} which is a shared value through (16), than find appropriate EC points (x_{ab} ,y_{ab}) as per (17). Bob computes S_B in order to verify the values Q_A, Q_B & Q_{AB}, through (18) & finally, uses $\widetilde{H}(P)$ to encrypt Q_B via (19), and sends it to Alice

$$Q_{AB} = d_B . Q_A = d_B . d_A . \mathrm{G} \quad (16)$$

$$(x_{ab}, y_{ab}) = Q_{AB} \quad (17)$$

$$S_B = \widetilde{H}(P|y_a|y_b|y_{ab}) \quad (18)$$

$$\mathrm{Y} = E_{\widetilde{H}(P)}(Q_B) \quad (19)$$

d) *Alice*: Alice uses $\widetilde{H}(P)$ in order to decrypt Y & obtains Q_B through (20), & computes EC point (x_b ,y_b) aligned with the given by (15). Which later she multiplies her private key to Bob's public key Q_B to obtain shared value Q_{AB} via (21), followed by (x_{ab} ,y_{ab}) given by (17).

Alice computes S_A for verification of having the values of Q_A, Q_B, Q_{AB} through (22). If the verification holds that she can be sure that Bob has the required values

$$Q_{A=} D_{\widetilde{H}(P)}(\mathrm{Y}) \quad (20)$$

$$Q_{AB} = d_A . Q_B = d_B . d_A . \mathrm{G} \quad (21)$$

$$S_B = \widetilde{H}(P|x_a|x_b|x_{ab}) \quad (22)$$

Step 3:

e) *Alice*: Alice needs to make Bob assure that she has the values as well. Then Alice performs (23) to calculate T_A out of Q_A, Q_B, and QAB which is later sends it to Bob

$$T_A = \widetilde{H}(P|x_a|x_b|x_{ab}) \quad (23)$$

f) *Bob*: On the other side, Bob calculates T_B via (24) and compares it with T_A. If the verification holds than Bob is assured that Alice has the required values as well

$$T_B = \widetilde{H}(P|y_a|y_b|y_{ab}) \quad (24)$$

Step 4:

Here the both parties have the required parameters & have verified each other. Finally, they perform (25) to calculate the shared symmetric key

$$K_{AB} = \widetilde{H}(x_a|x_b|x_{ab}|P|y_a|y_b|y_{ab}) \quad (25)$$

We try to eliminate the fixed initial private key when in comparison to the previous models. Here a random number is chosen to get the private key by both the parties. Key is constructed via one multiplication from (16) & (21) & one hash function from (25).

3) SGMCEP: Smart Grid with Multilayer Consensus ECC based PAKE Protocol

The appliance A_N knows at least the ID of the HAN later it can obtain ID of the H_C. Using four phase mechanism A_N gains access controllers. Now we assume:

- Appliance A_N & Home controller H_C share a predefined secret password *pw*.
- ECC parameter set {a, b, p, G, n, h} and $\widetilde{H}(.)$ are known & shared among all the parties.
- In order to have secure communications among them, controllers H_C, B_C, N_C and C_C have already been authenticated to upstream & downstream controllers
- Controllers are trusted parties that are controlled by the grid domain. The appliance which belongs to the customer domain is controlled by the customer.

- The symmetric keys exist:
  — k_{hb}: Shared between H_C and B_C.
  — k_{bn}: Shared between B_C and N_C.
  — k_{nc}: Shared between N_C and C_C.

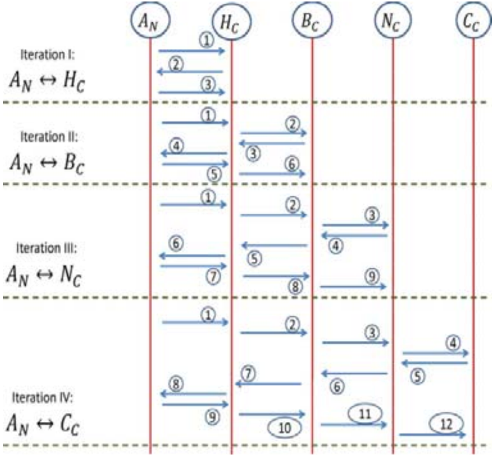

A_N
H_C
B_C
N_C
C_C
Iteration I:
$A_N \leftrightarrow H_C$
Iteration II:
$A_N \leftrightarrow B_C$
Iteration III:
$A_N \leftrightarrow N_C$
Iteration IV:
$A_N \leftrightarrow C_C$
1
2
3
4
5
6
7
8
9
10
11
12

Fig. 3: Four keys construction in SGMCEP.

The following are the sequence of steps that takes place in SGMCEP.

- ✓ A_N sends a request to send the packets to Home controller H_C. Next H_C accept the request & sends an acknowledgment to AN. At this point, A_N starts sending the packets to H_C.
- ✓ A_N sends a request to send the packets to Building controller B_C via H_C. Next B_C accept the request & sends an acknowledgment to A_N via H_C. At this point, A_N starts sending the packets to B_C via H_C.
- ✓ A_N sends a request to send the packets to Neighbour controller N_C via H_C & B_C. Next N_C accept the request & sends an acknowledgment to A_N via H_C & B_C. At this point, A_N starts sending the packets to B_C.
- ✓ In a similar way, A_N sends a request to send the packets to Central controller C_C via H_C, B_C & N_C. Next C_C accept the request & sends an acknowledgment to A_N via H_C, B_C & N_C. At this point, A_N starts sending the packets to C_C.

In this model, we need to have a predefined shared password between the two parties i.e., appliance and one of the controllers. At the end SGMCEP protocol decreases the number of packets and improves the security for smart grid.

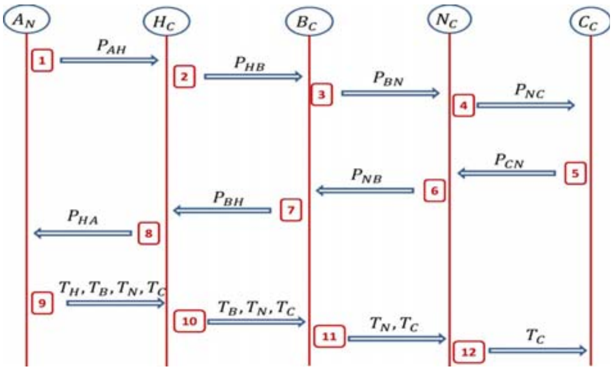


Fig. 4: SGMCEP Protocol phases and packets transfer.

Here we, introduce a new vector $\widehat{V}$ (entities identification set), which contains the IDs of the entities involved in SGMCEP as a part of the information exchanged between them. The four phases in SGMCEP protocol depicted in Fig.4 consists of the following steps.

Phase I (Initial Flow): In SGMCEP, A_N initiates the keys establishment process:

1) First Packet: A_N follows (26) to utilize the initial password *pw* shared by H_C to calculate temporary key $k^t{}_{ah}$

$$k^t{}_{ah}=\widetilde{H}(ID_A| \, pw \, |ID_H \,) \qquad (26)$$

A_N also picks a random number d_A€ [2,n-1], then computes Q_{AH} via (27) & appropriate coordinates $(x_a \, , \, y_a)$ given by (28)

$$Q_{AH} = d_A.\,\mathrm{G} \qquad (27)$$

$$(x_a \quad, y_a) = \; Q_{AH} \qquad (28)$$

A_N put its own ID in field A of $\widehat{V}$ given by (29).Finally, A_N forms packets P_{AH} by Q_{AH} and $\widehat{V}$ all encrypted by $k^t{}_{ah}$ key as per (30), and then sends the packet from to H_C

$$\widehat{V}.[A] \;\leftarrow\; ID_A \qquad (29)$$

$$P_{AH} = E_{k^t{}_{ah}} \;(Q_{AH}, \widehat{V}) \qquad (30)$$

2) Second Packet: First, H_C calculates temporary key $k^t{}_{ah}$ by performing (26) & decrypts received packet from A_N by way of (31) to obtain Q_{AH} and $\widehat{V}$.

$$(Q_{AH}, \widehat{V}\,) = \; D_{k^t{}_{ah}} \;(P_{AH}) \qquad (31)$$

H_C picks a random number d_H € [2,n-1] and computes Q_{HB} through (32)

$$Q_{HB} = (Q_{AH}).\mathrm{d} = (d_A.G).\, \mathrm{d_H}$$

$$= d_A.\, \mathrm{d_H}\, .\mathrm{G} \qquad (32)$$

$$(x_{hb}, y_{hb}) = Q_{HB} \qquad (33)$$

H_C puts its own ID into field *H* of $\widehat{V}$ by way of (34),and also computes pw_b via (35)

$$\widehat{V}.[H] \;\leftarrow\; ID_H \qquad (34)$$

$$pw_b = \widetilde{H}(\, k_{ah}{}^t | \;\; ID_A) \qquad (35)$$

Finally, H_C dispatches $\widehat{V}$ along with Q_{HB} and pw_b to B_C, all encrypted with the k_{hb} shared key following (36)

$$P_{HB} = E_{k_{hb}}(Q_{HB}, \widehat{V}, pw_b) \qquad (36)$$

3)Third Packet: First, B_C obtains Q_{HB}, $\widehat{V}$ and pw_b by decryption of the received packets P_{HB} from H_C via (37):

$$(Q_{HB}, \widehat{V}, pw_b) = \; D_{k_{hb}}(P_{HB}) \qquad (37)$$

Then, B_C chooses random number d_B € [2,n-1] and computes Q_{BN} through (38):

$$Q_{BN} = (Q_{HB}).\, d_B = d_A.\, \mathrm{d_H}\, .\, d_B.\mathrm{G} \qquad (38)$$

$$(x_{bn} \;, y_{bn}) = \; Q_{BN} \qquad (39)$$

Then, B_C copies its own ID into the $\widehat{V}$ field *B* in (40), and computes pw_n via (41).Finally, B_C forwards $\widehat{V}$, Q_{BN} and pw_n to N_C ,all encrypted with the predefined shared key of k_{hb} through (42)

$$\hat{V}.[B] \leftarrow ID_B \tag{40}$$

$$pw_n = \widetilde{H}(pw_b| \;\; ID_N) \tag{41}$$

$$P_{BN} = E_{k_{hb}}(Q_{BN}, \hat{V}, pw_n) \tag{42}$$

4) Fourth Packet: Firstly, N_C follows (43) to obtain Q_{BN}, $\hat{V}$ and pwn from the packet P_{BN} received from B_C:

$$(Q_{BN}, \hat{V}, pw_n) = D_{k_{hb}}(P_{BN}) \tag{43}$$

Then, N_C chooses random number d_N € [2, n-1] to obtain Q_{CN} via (44)

$$Q_{CN}= (Q_{BN}).\, d_N = d_A.\, d_H\,.\, d_B.\, d_N\,.G \tag{44}$$

$$(x_{bn}\ , y_{bn})= \ Q_{BN}$$

Then, N_C updates $\hat{V}$ field N with its own ID as depicted by (46), also computes pw_c through (47)

$$\hat{V}.[N] \leftarrow ID_N \tag{46}$$

$$pw_c = \widetilde{H}(pw_n|\, ID_C) \tag{47}$$

Finally, N_C forms packet P_{NC} out of $\hat{V}$, Q_{CN} and pw_c as shown by (48),encrypts its by k_{nc}, and forwards it to C_C

$$P_{NC} = E_{k_{nc}}(Q_{CN}, \hat{V}, pw_c) \tag{48}$$

Phase II (Response Flow): This flow starts with C_C replying to the fourth packets above.

5) Fifth packet: First, C_C obtains the Q_{CN}, $\hat{V}$ and pw_c values by decryption of the packets P_{NC} received from the N_C following (49)

$$(Q_{CN}, \hat{V}, pw_c) = D_{k_{hc}}(P_{NC}) \tag{49}$$

Then, C_C extracts ID of any of the controller as well as ID of the appliance ID_A from $\hat{V}$ *($\hat{V}$. [A])*, and also calculates $k^t{}_{ca}$ through (50).Beside, C_C inserts its own ID into field *C* of *V* as presented by (51)

$$k^t{}_{ca}= \widetilde{H}(ID_C|pw_c\,|ID_A) \tag{50}$$

$$\hat{V}.[C] \leftarrow ID_C \tag{51}$$

Then, C_C picks a random number d_C € [2, n-1] to obtain Q_C and Q_{CC} following (52) and (53) respectively

$$Q_C=d_A.G \tag{52}$$

$$Q_{CC}= (Q_{CN}).\, d_C \tag{53}$$

$$(x_c\ , y_c)= Q_{CC} \tag{54}$$

Then, C_C obtains coordinates (x_c, y_c) as shown by (54) and (x_{nc}, y_{nc}) as depicted by (45), and then computes S_{CN} via (55) for verification purpose

$$S_{CN} = \widetilde{H}(k^t{}_{ca} \;|y_{nc}|y_c\,) \tag{55}$$

Finally, C_C follows (56) to form P_{CN} from S_{CN}, Q_C and $\hat{V}$, in which C_C encrypts the packets by k_{nc} as shown in (56)

$$P_{CN} = E_{k_{nc}} \ (S_{CN}, Q_C, \hat{V}) \tag{56}$$

6) Sixth Packet: First, N_C decrypts the packets received from C_C to obtain the S_{CN}, Q_C and *V* values following (57). Then, N_C calculate $k^t{}_{na}$ through (58)

$$(S_{CN}, Q_C, \hat{V}) = D_{k_{nc}}(P_{CN}) \tag{57}$$

$$k^t{}_{ca}= \widetilde{H}(ID_N|pw_n\,|ID_A) \tag{58}$$

Then, N_C utilizes its own random number d_N to calculate Q_N via (59), and Q_{NC} via (60). Then, N_C follows (61) to calculate S_{NB} for the verification purpose

$$Q_N=d_N.G \tag{59}$$

$$Q_{NC}= (Q_C\,).d_N \;=\; d_N\,.d_C\,.G \tag{60}$$

$$S_{NB}= S_{CN} \oplus \widetilde{H}(k^t{}_{na}\,|y_{bn}|y_{nc}\,) \tag{61}$$

Finally, N_C forms P_{NB} out of S_{NB}, Q_N, Q_{NC} and $\hat{V}$ and encrypts the packet by k_{bn} as shown in (62) to be sent to the BAN controller (B_C)

$$P_{NB} = E_{k_{bn}} \ (S_{NB}, Q_N, Q_{NC}, \hat{V}) \tag{62}$$

7) Seventh Packet: Firstly, B_C obtains the parameters S_{NB}, Q_N, Q_{NC} and $\hat{V}$ as presented by (63) by decrypting packet received from N_C. Then, B_C calculates the $k^t{}_{ba}$ key via (64)

$$(S_{NB}, Q_N, Q_{NC}, \hat{V}) = D_{k_{bn}}(P_{NB}) \tag{63}$$

$$k^t{}_{ba}=\widetilde{H}(ID_B|pw_b|ID_A) \tag{64}$$

Then, B_C uses its own random number d_B to obtain the Q_B via (65), Q_{NC} through (66) and Q_{BNC} via (67)

$$Q_B=d_B.G \tag{65}$$

$$Q_{BN}= (Q_N\,).d_B \;=\; d_N\,.d_B\,.G \tag{66}$$

$$Q_{BNC}= (Q_{NC}\,).d_B \;= d_B.\, d_N\,.d_C\,.G \tag{67}$$

Then, B_C obtains coordinates (x_{nc}, y_{nc}) and (x_{bn} , y_{bn}) as shown by (45) and (39) respectively, and calculates S_{BH} through (68) for verification

$$S_{BH}= S_{NB} \oplus \widetilde{H}(k^t{}_{ba}\,|y_{hb}|y_{bn}\,) \tag{68}$$

Finally, B_C forms P_{BH} packets by S_{BH}, Q_N, Q_{NC}, Q_{BNC} and $\hat{V}$, encrypted by k_{nc} as shown (69), and sends the packet to H_C

$$P_{BH} = E_{k_{hb}} \ (S_{BH}, Q_B, Q_{BN}, \ Q_{BNC}, \hat{V}) \tag{69}$$

8) Eighth Packet: First, H_C decrypts the packet received from B_C and obtains S_{BH}, Q_N, Q_{NC}, Q_{BNC} and $\hat{V}$ depicted by (70).Then, H_C calculates $k^t{}_{ha}$ through (71)

$$(S_{BH}, Q_B, Q_{BN}, \ Q_{BNC}, \hat{V}) =D_{k_{hb}}(\ P_{BH}\ ) \tag{70}$$

$$k^t{}_{ha}= \widetilde{H}(ID_H|pw|ID_A) \tag{71}$$

Then, H_C utilize its own random number d_H to compute Q_H via (72), Q_{HB} through (73), Q_{HBN} via (74) and Q_{HBNC} through (75)

$$Q_H=d_H.G \tag{72}$$

$$Q_{HB}= (Q_B).d_H \;=\; d_H\,.d_B\,.G \tag{73}$$

$$Q_{HBN}= (Q_{BN}\,.G).d_H = d_H.\, d_B.d_N\,.G \tag{74}$$

$$\begin{aligned} Q_{HBNC}= (Q_{HBN}).\ d_H &=(\, d_C.\ d_B.d_N\,.G) \\ &=(\, d_H.\ d_B.d_N.\ d_C.G) \end{aligned} \tag{75}$$

H_C obtains coordinates ($x_{bn}y_{bn}$) and (x_{hb} , y_{hb}) as depicted by (39) and (33), respectively, and then computes S_{HA} using (76) for verification

$$S_{HA} = S_{BH} \oplus \widetilde{H}(k^t{}_{ha}|y_a|y_{hb}\,) \tag{76}$$

Finally, H_C forms PHA packet out of S_{HA}, Q_H, Q_{HB},Q_{HBN} and Q_{HBNC} $\hat{V}$, encrypted by $k^t{}_{ha}$ as shown by (77), and sends the packet to A_N

$$P_{HA}=E_{k^t{}_{ha}} \ (S_{HA}, Q_H, Q_{HB}, Q_{HBN}, Q_{HBNC}, \hat{V}) \tag{77}$$

Phase III: Verification

9) Ninth Packet (Appliance): In this phase, A_N verifies the received values and dispatches the confirmation to the upstream controllers. First, A_N computes the $k^t_{ha.}$ Temporary key via (71), to decrypt the received packet P_{HA} from H_C in order to obtain S_{HA}, $Q_{H,}$ $Q_{HB,}Q_{HBN}$ and Q_{HBNC} and $\widehat{V}$ following (78)

$$(S_{HA}, Q_H, Q_{HB}, Q_{HBN}, Q_{HBNC}, \widehat{V}) \quad = D_{k^t{}_{ha}}\,(P_{HA}) \qquad (78)$$

Then, A_N utilizes its own random number d_A to calculate Q_{HB} via (79), Q_{BN} through (80), Q_{NC} via (81) and Q_{CC} through (82), which are shared by H_C, B_C, N_C and C_C respectively

$$(Q_H).\, d_A\text{=(} d_H.G).\ d_A = d_A.\ d_H.G\text{=} \qquad (79)$$

$$(Q_{HB}).\, d_A\text{=}(d_H.\, d_B.G\ ).\, d_A\text{=}d_A.\, d_H.\, d_B.G = Q_{BN} \qquad (80)$$

$$(Q_{HBN}).\, d_A\text{= }(d_H.\, d_B.\, d_N.\, G).\, d_A\text{=}d_A.\, d_H.\, d_B.\, d_N.\, G\text{= }Q_{CN} \qquad (81)$$

$$(Q_{HBNC}).\, d_A\text{= }(d_H.\, d_B.\, d_N.\, d_C.\, G).\, d_A\text{=}d_A.\, d_H.\, d_B.\, d_N.\, d_C.\, G\text{= }Q_{CC} \qquad (82)$$

Then, A_N uses the above shared values to obtain coordinates (x_c, y_c), (x_{nc}, y_{nc}), (x_{bn}, y_{bn}), (x_{hb}, y_{hb}) as shown in (54), (45), (39) and (33) respectively. Then, A_N utilize the coordinates and performs (55), (61), (68) and (76) to substantiate S_{HA}. If the verification holds, A_N proceeds to next step. Note that, since A_N has *pw*, it is able to obtain pw_b, pw_n, pw_c based upon (35),(41) and (47). Finally, A_N generates four values T_{AH} via (83) for H_C, T_{AB} through (85) for B_C, T_{AN} via (87) for N_C and T_{AC} through (89) for C_C, as verifiers of the shared values, and forwards them to H_C

$$T_{AH}\text{= }\widetilde{H}(k^t{}_{ah}|x_a|x_{hb}) \qquad (83)$$

$$k^t{}_{ab}\text{= }\widetilde{H}(ID_A|pw_{ab}\,|ID_B) \qquad (84)$$

$$T_{AB}\text{= }\widetilde{H}(k^t{}_{ab}|x_{hb}|x_{bn}) \qquad (85)$$

$$k^t{}_{an}\text{= }\widetilde{H}(ID_A|pw_{an}\,|ID_N) \qquad (86)$$

$$T_{AN}\text{= }\widetilde{H}(k^t{}_{an}|x_{bn}|x_{nc}) \qquad (87)$$

$$k^t{}_{ca}\text{= }\widetilde{H}(ID_A|pw_{ac}\,|ID_C) \qquad (88)$$

$$T_{AC}\text{= }\widetilde{H}(k^t{}_{ac}|x_{nc}|x_c) \qquad (89)$$

10) Tenth Packet (HAN controller): H_C receives the above substantiation values and then verifies T_{AH} based upon (83). If the verification holds, H_C relays the other values to B_C.

11) Eleventh Packet (BAN controller): B_C receives the above values and then verifies T_{AB} following (85). If the verification holds, B_C relays the values to N_C.

12) Twelfth Packet (NAN controller): N_C receives the eleventh packet & then verifies T_{AN} through (87), If the verification holds, N_C relays the other values to C_C. SGCC controller: C_C receives the twelfth packet and then verifies T_{AC} via (89).

Phase IV: Keys Calculation

Thus far, all parties have their verified shared values. Finally, they can generate their appropriate symmetric keys per (90), (91), (92) and (93)

$$A_N\ \&\ H_C\text{:}K_{HA}\text{= }\widetilde{H}(x_a|x_{hb}|k^t{}_{ha}|k^t{}_{ah}|y_a|y_{hb}) \qquad (90)$$

$$A_N\ \&\ B_C\text{: }K_{HA}\text{= }\widetilde{H}(x_{hb}|x_{bn}|k^t{}_{ba}|k^t{}_{ab}|y_{hb}|y_{bn}) \qquad (91)$$

$$A_N\ \&\ N_C\text{: }K_{NA}\text{= }\widetilde{H}(x_{bn}|x_{nc}|k^t{}_{na}|k^t{}_{an}|y_{bn}|y_{nc}) \qquad (92)$$

$$A_N\ \&\ C_C\ \text{: }K_{CA}\text{= }\widetilde{H}(x_{nc}|x_c|k^t{}_{ca}|k^t{}_{ac}|y_{nc}|y_c) \qquad (93)$$

V RESULTS

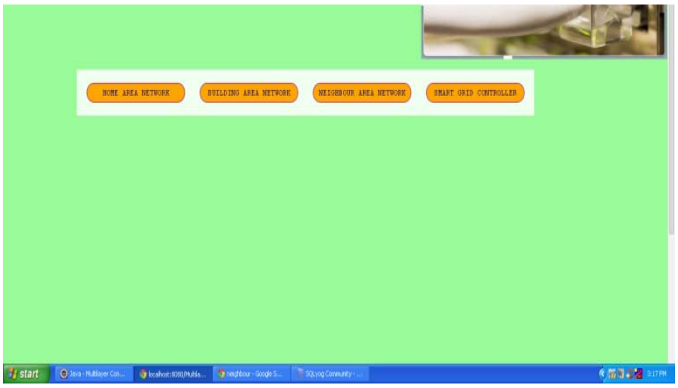
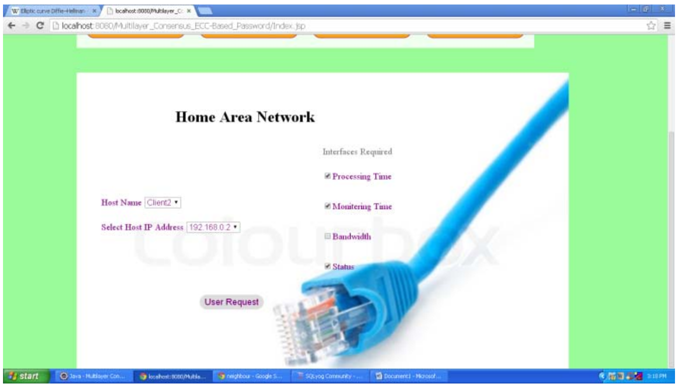


Fig 1: SGMCEP (Smart Grid with Multilayer Consensus ECC based PAKE Protocol home page.


localhost:8080/Multilayer_Consensus_ECC-Based_Password/Index.jsp

Home Area Network

Host Name: Client2

Select Host IP Address: 192.168.0.2

Interfaces Required

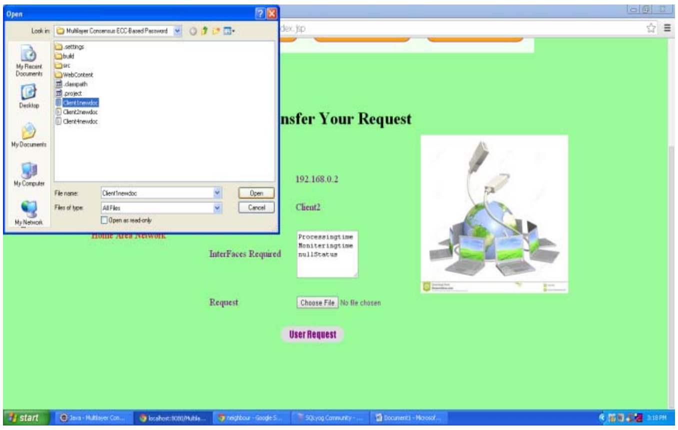
☑ Processing Time

☑ Monitering Time

☐ Bandwidth

☑ Status

Fig 2: Requesting Page


Open

Look in: Multilayer Consensus ECC-Based Password

.settings
build
src
WebContent
.classpath
.project
Client1newdoc
Client2newdoc
Client4newdoc

File name: Client1newdoc

Files of type: All Files

☐ Open as read-only

nsfer Your Request

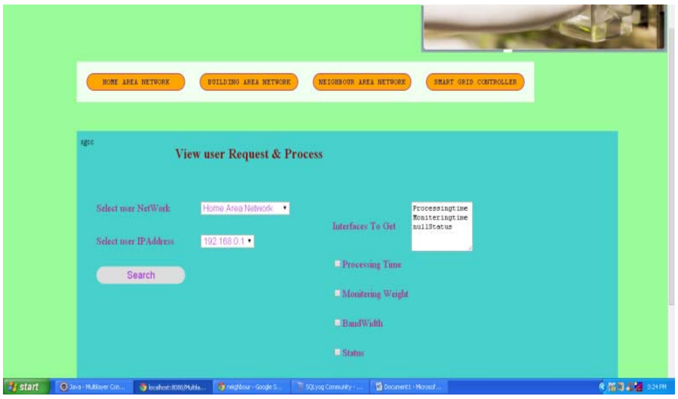
192.168.0.2

Client2

InterFaces Required:
Processingtime
Moniteringtime
nullStatus

Request: Choose File No file chosen

Fig 3: File Transformation



View user Request & Process

Select user NetWork: Home Area Network

Select user IPAddress: 192.168.0.1

Interfaces To Get:
Processingtime
Moniteringtime
nullStatus

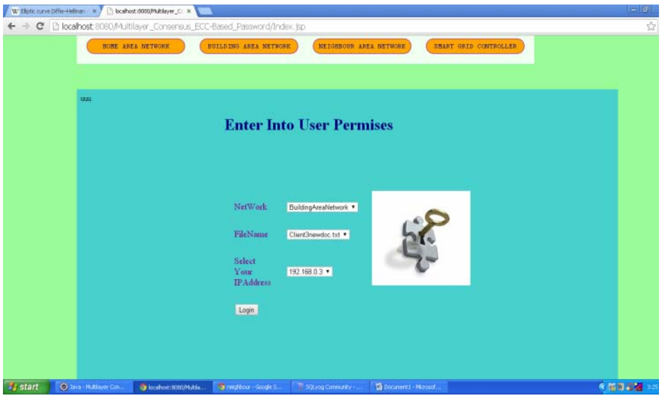
☐ Processing Time

☐ Monitering Weight

☐ BandWidth

☐ Status

Fig 4: Key generation by the Smart Grid



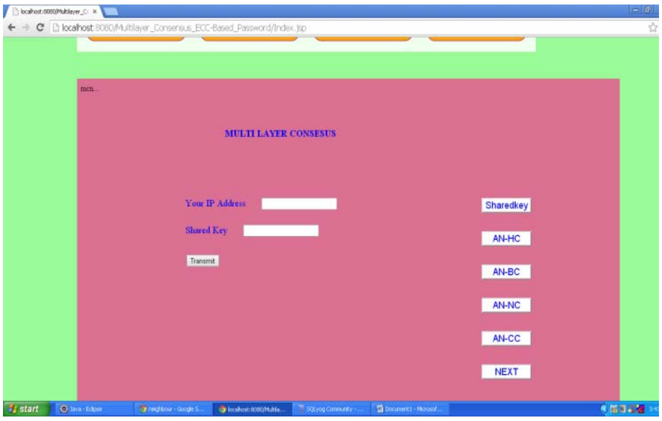
Enter Into User Permises

NetWork: BuildingAreaNetwork

FileName: Client3newdoc.txt

Select Your IPAddress: 192.168.0.3

Fig 5: User Premises

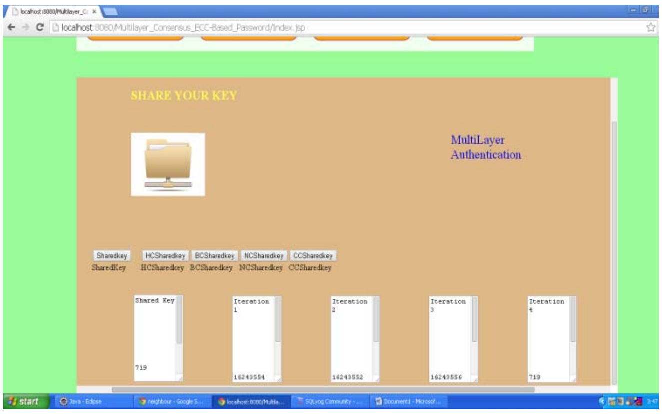

localhost:8080/Multilayer_Consensus_ECC-Based_Password/Index.jsp
Transmit
Sharedkey
AN-HC
AN-BC
AN-NC
AN-CC
NEXT

MULTI LAYER CONSESUS

Your IP Address

Shared Key

Fig 6: Multi Layer


localhost:8080/Multilayer_Consensus_ECC-Based_Password/Index.jsp
Sharedkey
HCSharedkey
BCSharedkey
NCSharedkey
CCSharedkey

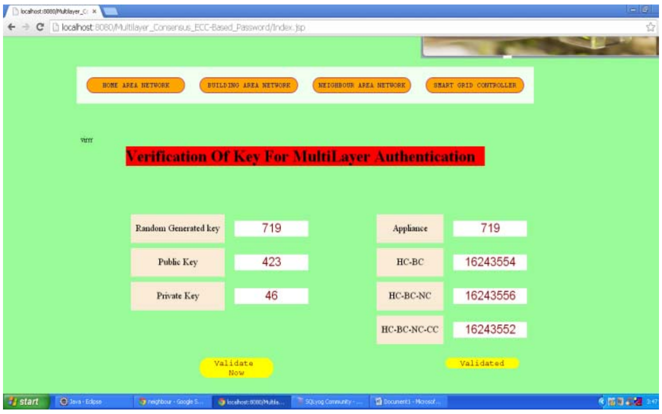
SHARE YOUR KEY

MultiLayer Authentication

SharedKey	HCSharedkey	BCSharedkey	NCSharedkey	CCSharedkey

Shared Key	Iteration 1	Iteration 2	Iteration 3	Iteration 4
719	16243554	16243552	16243556	719

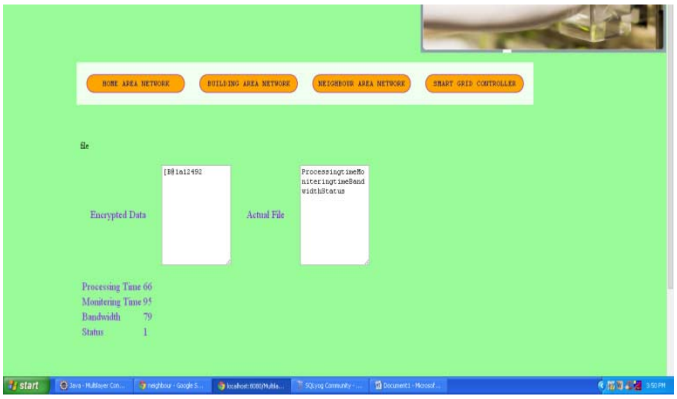
Fig 7: Multi Layer Authentication



Verification Of Key For MultiLayer Authentication

Random Generated key	719	Appliance	719
Public Key	423	HC-BC	16243554
Private Key	46	HC-BC-NC	16243556
		HC-BC-NC-CC	16243552

Fig 8: Verification of Key



file

Encrypted Data: [B@1a12492

Actual File: ProcessingtimeMoniteringtimeBandwidthStatus

Processing Time	66
Monitering Time	95
Bandwidth	79
Status	1

Fig 9: Final Output

VI CONCLUSION

The proposed protocol enables secure communications between a home appliance and different layers of the SG control system. This protocol establishes four multilayer consensus password-authenticated symmetric keys between an appliance & upper layer controllers in order to provide a hierarchical authority over the appliances. The SGMCEP protocol relies on ECC to provide a high security level with a small key size while addressing the resource constraints in the devices. The protocol is easily implemented by adaptation of the X.1035 standard & applying ECC approach. SGMCEP protocol can be extended to a larger number of layers if required. The proposed protocol reduces the system security overhead against most of the well-known attacks. When compared to the X.1035 standard, SGMCEP involves a lower load for computations of the hash function & it requires passwords. EPAK protocol presented in this paper can be used & implemented in any application & environment outside of any smart grid system. We have developed EPAK, which is based on ECC that a high security level with a small key size.

REFERENCES

[1] M.M. Fouda, Z.M. Fadlullah,N. Kato,R.Lu, andX.Shen, "Towards a light-weight message authentication mechanism tailored for smart grid communications," in *Proc. IFIP SCNC Workshop*, Shanghai, China, Apr. 2011.

[2] "Introduction to NISTIR 7628 guidelines for smart grid cyber security," National Institute of Standards and Technology (NIST), 2010 [Online]. Available: http://www.nist.gov/smartgrid/upload/nistir-7628_total.pdf

[3] H. Nicanfar and V. C. M. Leung, "Smart grid multilayer consensus password-authenticated key exchange protocol," in *Proc. IEEE SFCS Workshop*, Ottawa, ON, Canada, Jun. 2012.

[4] Q. Li and G. Cao, "Multicast authentication in the smart grid with onetime signature," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 686–696, Dec. 2011.

[5] A. P. Muniyandi, R. Ramasamy, and Indrani, "Password based remote authentication scheme using ECC for smart card," in *Proc. ICCCS*, Orisa, India, Feb. 2011.

[6] P. Jokar, H. Nicanfar, and V. C. M. Leung, "Specification-based intrusion detection for home area networks in smart grids," in *Proc. IEEE SmartGridComm*, Brussels, Belgium, Oct. 2011.

[7] X. Ding, C. Ma, and Q. Cheng, "Password authenticated key exchange protocol with stronger security," in *Proc. ETCS Workshop*, Wuhan, Hubei, China, Mar. 2009.

[8] H. Boumerzoug, B. A. Bensaber, and I. Biskri, "A keys management method based on an AVL tree and ECC cryptography for wireless sensor networks," in *Proc. Q2SWinet*, Miami Beach, FL, Oct.–Nov. 2011.

[9] E. J. Yooni and K. Y. Yoo, “A new elliptic curve Diffie-Hellman two party key agreement protocol,” in *Proc. ICSSSM Conf.*, Tokyo, Japan, Jun. 2010.

[10] S. Wang, Z. Cao, M. A. Strangio, and L. Wang, “Cryptanalysis and improvement of an elliptic curve Diffie-Hellman key agreement protocol,” *IEEE Commun. Lett.*, vol. 12, no. 2, pp. 149–151, Feb. 2008.