International Journal of Computer Networks and Communications Security

VOL. 3, NO. 3, MARCH 2015, 117–123
Available online at: www.ijcncs.org
E-ISSN 2308-9830 (Online) / ISSN 2410-0595 (Print)

An Efficient ECC-based Authentication and Key Agreement Protocol

Mojtaba Mohammadpoor[1] and Abbas Mehdizadeh[2]

[1] Department of Electrical Engineering, University of Gonabad, Gonabad, Iran

[2] Department of Computing, Nilai University, 71800 Putra Nilai, Negeri Sembilan, Malaysia

E-mail: [1]m.mohammadpoor@gmail.com, [2]mehdiizadeh@ieee.org

ABSTRACT

Public-key cryptography is commonly used to authenticate communicating entities in some networks. One of the key tools in this way is to use the elliptic curves cryptography (ECC) which is relatively lightweight due to its shorter key size compared to the conventional River-Shamir-Adleman (RSA) method. This paper is proposing an efficient protocol by analysing two variants of ECC-based wireless authentication protocol, namely, Aydos-Savas-Koc's wireless authentication protocol (ASK-WAP) and user authentication protocol (UAP) from various security aspects and communication concerns. We show that although UAP is able to address some of ASK-WAP vulnerabilities, it is confined to one-way communication where the authentication can only be initialized by users and not the server. In light of their limitations, we suggest several possible improvements to both ASK-WAP and UAP. The proposed solutions focus on applying encryption methods to the transmitted keys and enabling two-way communication on UAP. From performance evaluation, we show that our proposed methods are able to address the security concerns of ASK-WAP and UAP, while at the same time achieving acceptable communication overheads.

Keywords: *TElliptic Curves Cryptography, ASK-WAP, UAP.*

1 INTRODUCTION

Providing adequate security is a challenging issue for many types of communication networks. Among them, whose that are using air interface or internet (such as wireless communications, wireless sensor networks, voice over IP, etc.) are more prone to various types of security attacks. In general, wireless networks do not provide the same level of protection as wired networks. Due to using air interface, they are vulnerable to several attacks, including: unauthorized use of resources, masquerading, unauthorized disclosure and flow data information, unauthorized alteration of resources and data information, repudiation of actions, and denial of service attacks.

To enhance the security of wireless networks, public-key cryptography can be used for authentication. In this regard, securing communications between users and certification authority (CA) is one of the concerns in wireless networks. Compared to private-key/symmetric-key systems, public-key systems (i.e., based on certificates) ensure stronger security, but suffer from more computational cost and power [1]. To balance the security and efficiency, an efficient authentication is therefore required for inclusion in wireless networks.

One promising approach is to use elliptic curve cryptography (ECC) which provides striking advantage of shorter key size compared to conventional algorithm (e.g., RSA algorithm), while preserving the equivalent security level. Additionally, ECC has been accepted as IEEE P1363 Standard for Public Key Cryptography [2]. Recently, some authentication and key agreement protocol based on ECC have been proposed [3-6]. On the other hand, several EC-based cryptography methods are proposed for wireless sensor networks [7] as well as session initiation protocol [8].

In [9] Aydos *et al.* an ECC-based wireless authentication and key agreement protocol called ASK-WAP have been proposed for securing user-server communication. While their proposed ASK-WAP has several advantages in terms of storage requirements, bandwidth, and computational burden, it is also vulnerable to several attacks such as man-in-the-middle attack [10], lack of mutual authentication [2] and forward secrecy [11], and prone to forging certificate attack [12]. This led to a new wireless authentication protocol called user authentication protocol (UAP) [10][13].

In this paper, we analyze both ASK-WAP and UAP schemes and show that although UAP is able to overcome some of the security concerns faced by ASK-WAP, has another problem and isn't the best solution for ASK-WAP. One of the biggest problems of UAP is its one-way nature where the communication can only be initialized by users (i.e., server has no control over the communication). Therefore, it is also prone to denial of service attack, known-text and chosen-text attacks, as well as exhaustive attack. We improve both ASK-WAP and UAP by proposing some derivations to solve the security and communication problems.

The rest of the paper is structured as follows. Some common definitions and abbreviations are summarized as bellow. In Section 2, two existing protocols, namely, ASK-WAP and UAP are discussed in details followed by highlighting their efficiencies and weaknesses. Our proposed algorithms are explained in Section 3. The performance evaluation is discussed in Section 4, followed by some concluding remarks.

Definitions:

Subscripts:
- **U**: User
- **CA**: Certificate Authority
- **S**: Server

d: Private Key (an integer).
H(.): One-way Hash function.
E(*k,m*): Encrypt plaintext *m* by key *k.*
D(*k,m*): Decrypt cipher text *m* by key *k.*
t: time stamp
I: identifier
M: A point (M.x , M.y) on the Elliptic Curve that is chosen as Public Key.
B: base point on the Elliptic Curves by order n that is known for all parties.
d×B: multiply integer *d* by point *B* using ECC's specific rules.
g: Random number.
(r,s): special certificates for use in Elliptic Curve Digital Signature Algorithm (ECDSA).

2 EXISTING PROTOCOLS

2.1 ASK-WAP

ASK-WAP uses the Elliptic Curve Diffie-Hellman (ECDH) for key substitute/exchange and Elliptic Curve Digital signature Algorithm (ECDSA) for signing and verification. In the initialization section the users and the server obtain their certificates *(r,s)* for use in ECDSA, identifiers (I), and expiration dates from the Certification Authority (CA) throughout a secure channel. This is done just one time during the expiration time.

To apply Elliptic Curves, a curve over a Galois field $GF(p)$ where p is a prime number or over $GF(2^q)$ where q is an integer should be defined. The first one is suitable to be implemented in software while the second one is compatible with hardware [11]. To prevent known attacks, one must pay enough attention in choosing curve's coefficients. The National Institute of Standards and Technology (NIST) has recommended some prime numbers in Federal Information Processing Standard 186-2 (FIPS 186-2) standard [14]. In an Elliptic Curve group a base point $B\ (B_x,B_y)$ of large order n should be selected and made public to all parties. In the initialization parts, user can select a random number d_U as its private key and calculate its public key $M_U=d_U\times B$ by performing point doubling and multiplying rules. Also server can calculate $M_S=d_S\times B$ in a similar manner.

In the key agreement part, user and server can exchange their public key. User can calculate $d_U \times M_S = (d_U\ d_S) \times B$, server also can calculate $d_S \times M_U = (d_S\ d_U) \times B$. Now they can agree on $(d_U\ d_S) \times B$ as their mutual key.

2.2 UAP

Because of ASK-WAP's weaknesses such as its vulnerability against man-in-the-middle attack (to be discussed in the next subsection), Mangipudi *et al.* proposed a variant of it in [10]. They simply calculate another random numbers g_U and g_S in user and server parts. User sends $M_R = g_U \times M_S = (g_U\ d_S) \times B$ to the server but agrees on $g_U \times B$. Server can calculate $d_S^{-1}\times M_R = (d_S^{-1}\ g_U\ d_S) \times B= g_U \times B$ to get the mutually agreed key.

2.3 Analysis of ASK-WAP and UAP

Several security requirements for an authentication and key agreement protocol in wireless communication are defined in [9], which are nonrepudiation of service, mutual authentication, confidentiality, and anonymity of user.

ASK-WAP doesn't provide mutual authentication [2] due to using Diffie-Hellman (DH) key exchange protocol in its initialization phase, which is vulnerable against man-in-the-middle and impersonating attacks. Since ASK-WAP and UAP both offers digital signature, the non-reputation of services can be achieved. Both protocols achieved confidentiality by protecting data transmitted between two parties. They also have the capability of using a temporary identity assigned by CA to the user, so they can meet the anonymity of user.

Both ASK-WAP and UAP do not meet known-key security, since the long-term session key $(M_K.x)$ can be easily compromised. In addition, ASK-WAP doesn't provide forward secrecy [11], and neither does UAP because by compromising the server's private key all session keys can be recovered as well.

2.4 Other Weaknesses of UAP

In addition, UAP has other weaknesses as follow:

- Only user can initialize the communication. It should be noted that in certain cases server needs to do it as well, e.g., for call terminating. Although it is not a security weakness, it can be regarded as a major communication problem, and thus the protocol may not be feasible to be implemented at all.
- At the beginning of the initialization session, users do not send any information (for example its permanent public key or its certificate) to the server. Even in the next steps user's identifier is never used, thus an adversary/attacker can easily launch a Denial-of-Service (DoS) attack. Attacker can send huge number of requests to the server and confusing it by calculating mutually agreed keys.
- The algorithm uses the mutual key as the encryption key and a part of the plain-text of the block cipher simultaneously to encrypt the text. This accelerates the known-text and chosen-text attacks against the block cipher. An adversary can easily break it knowing that the major part of the plain-text is the same as the key.
- Suppose that the long term session key, i.e., mutual key, is compromised by knowing one's M_R that is available because it is sent clear. The adversary can recover the server's private key. Because $M_R = (g_U d_S) \times B$, and adversary knows the g_U, thus, adversary can even launch the exhaustive attack easily to find a multiplicative of g_U that meets the above equation.

3 IMPROVEMENTS AND SUGGESTIONS

In this section, we suggest three new possible ways to improve the ASK-WAP. Then we introduce a robust way to improve UAP protocol.

3.1 First Proposal - EKE-ASK

Pre-shared password idea was used by P. Koduri [12] called EC-EKE (Elliptic Curve Encrypted Key Exchange) which considers a predetermined block cipher and a pre-shared password 's' as its key, namely:

User 1 computes $M_A=d_A \times B$ and sends $E(s,M_A)$ to user 2.
User 2 computes $M_B=d_B \times B$ and sends $E(s,M_B)$ to user 1.

If we apply EC-EKE idea to ASK-WAP, by encrypting M_U and M_S before exchanging them, we will solve almost all its problems. It will foil man-in-the-middle attack (i.e., one of the main weaknesses of ASK-WAP). The two remained problems are; i) possessing and saving shared password 's' in subscriber Identity Module (SIM) of both sides, and ii) encrypting a point by the block cipher algorithm. To solve this problem, we can represent the points in compressed forms as in [15]. Figure 1 illustrates its mutual authentication phase, and the main changes to ASK-WAP are shown in bold.

User	Server
1. Send $E(s,M_U)$	Receive $E(s,M_U)$
	$M_U=D(s,M_U)$
2.	Generate a random number $g_S \in \{2, n-2\}$
3. Receive $E(s,M_S, g_S)$	Send $E(s,M_S, g_S)$
$M_S, g_S = D(s,M_S, g_S)$	
4. $M_K = d_U \times M_S = (d_U d_S) \times B$	$M_K = d_S \times M_U = (d_S d_U) \times B$
5.	$M_K.x$: Mutually agreed key

Fig. 1. EKE-ASK algorithm's Mutual key agreement phase

3.2 Second Proposal - SPE-ASK

We use SPECKE (Simple Password Elliptic Curve Key Exchange) method introduced in [12]. We calculate $M_U = (sd_U) \times B$ instead of $M_U = d_U \times$

B , using pre-shared password 's' and send it to the other side. Also in the other side we use $M_S = (sd_S) \times B$ instead of $M_S = d_S \times B$. These changes guarantee its resilience against all possible attacks mentioned above. Figure 2 shows mutual authentication phase, and the main changes to ASK-WAP are shown in bold.

User	Server
1. $M_U=(sd_u)\times B$	$M_S=(sd_S)\times B$
2. Send M_U	Receive M_U
3.	Generate a random number $g_S \in \{2, n-2\}$
4. Receive M_S, g_S	Send M_S, g_S
5. $M_K = d_U \times M_S$ $= (sd_U\, d_S) \times B$	$M_K = d_S \times M_U$ $= (sd_S\, d_U) \times B$
6. $M_K.x$: Mutually agreed key	

Fig. 2. SPE-ASK algorithm's Mutual key agreement phase

3.3 Third Proposal - VER-ASK

Another approach to prevent man-in-the-middle attack in ASK-WAP is by encrypting M_U and M_S or at least one of them (due to tradeoff between its performance and security goals) using one of ECC encryption/decryption algorithms as in [16][17]. Figure 3 shows the mutual authentication phase, and again the main changes to ASK-WAP are shown in bold.

User	Server
1. Randomly generate a number $g_U \in \{2, n-2\}$	
2. Send $\{g_UB, M_U+g_UMs\}$	Receive $\{g_UB, M_U+g_UMs\}$
3.	$M_U=(M_U+g_uM_S)-d_S\times(g_UB)$
4.	Random-generate a number $g_S \in \{2, n-2\}$
5. Receive $\{g_SB, M_S+g_SM_U\}$	Send $\{g_SB, M_S+g_SM_U\}$
6. $Ms= (M_S+g_SM_U)-d_U\times(g_SB)$	
7. $M_K = d_U \times M_S$ $= (d_U\, d_S) \times B$	$M_K = d_S \times M_U$ $= (d_S\, d_U) \times B$
8. $M_K.x$: Mutually agreed key	

Fig. 3. VER-ASK mutual authentication phase

3.4 Fourth Proposal - VER-UAP

In regards to UAP weaknesses, if we impose UAP idea to both sides (i.e., user and server) we will be able to remove most of its problems. Figures 4, 5, and 6 illustrate the entire algorithms:

User	Server
1. Select $d_S \in \{2, n-2\}$ $M_S= d_S\times B$	
3. Send M_S	Receive M_S
4. Compute d_S^{-1}	Select unique I_S and t_S
5. Receive M_U, I_S, t_S	Send M_U , I_S, t_S
6. Store d_S^{-1}, M_U, I_S, t_S	Store M_S, t_S

Fig. 4. VER-UAP server authentication protocol

User	Server
1. Select $d_U \in \{2, n-2\}$	Select $k_U \in \{2, n-2\}$
2. $M_U= d_U\times B$	$R_U= k_U \times B$
3. Send M_U	Receive M_S
4. Compute d_U^{-1}	Select unique I_U and t_U
5. Receive M_S, I_U, t_U	Send M_S , I_U, t_U
6. Store d_U^{-1}, M_S, I_U, t_U	Store M_U, t_U

Fig. 5. VER-UAP user authentication protocol

User	Server
1. Randomly generate a number $g_U \in \{2, n-2\}$	$g_S \in \{2, n-2\}$
2. $M_M = g_U\times M_S$ $= (g_U\, d_S) \times B$	$M_N = g_S\times M_U$ $= (g_S\, d_U) \times B$
3. Send M_M	Receive M_M
4. Receive M_N	Send M_N
5. $M_K = d_U^{-1}\times M_N + g_U \times B$ $= (d_U^{-1}\, g_S\, d_U) \times B + g_U \times B$ $= g_U \times B + g_S \times B$	$M_K = d_S^{-1}\times M_M + g_S \times B$ $= (d_S^{-1}\, g_U\, d_S) \times B + g_S \times B$ $= g_S \times B + g_U \times B$
6. $M_K.x$: Mutually agreed key	
7.	$C_0= E(M_K.x, I_S,, t_S)$
8. Receive C_0	Send C_0
9. $D(M_K.x, C_0)$: Valid I_S, t_S?	
10. $C_1= E(M_K.x, I_U, t_U)$	
11. Send C_1	Receive C_1
12.	$D(M_K.x, C_1)$: Valid I_U, t_U ?

13. $k_M = H(M_k.x, g_S,g_U)$ $k_M = H(M_k.x, g_S,g_U)$

k_M is the unique session key

Fig. 6. VER-UAP authentication and key agreement protocol

In this protocol we don't need ECDSA in any side; therefore, we have maximum performance in comparison with previous proposed protocols.

4 PERFORMANCE EVALUATION

First we analyze the proposed methods based on various security aspects and compare them according to their performances.

4.1 Security of ASK-WAP Improvements

Due to encryption of transmitted public keys in the first three proposed methods:

i. They provide mutual authentication.

ii. They foil man-in-the-middle attack.

iii. They foil impersonating attack.

iv. They meet known key security and prefect forward secrecy.

Note that the only vulnerability left for SPE-ASK and VER-ASK protocols is 'known key-share attack'. In this attack an adversary can change their mutual key by multiplying the transmitting public keys with a predetermined integer in order to force them to agree on a desired value. Fortunately, this attack is aborted in the following phases where the ECDSA scheme is used.

4.2 Security of VER-UAP

The security of VER-UAP can be summarized as follows:

i. It provides mutual authentication as same as UAP.

ii. It foils man-in-the-middle and impersonating attacks as same as UAP.

iii. It meets known key security. Compromising $M_K.x$ leads adversary to achieve (g_U+g_S) for and he/she cannot calculate random numbers g_U or g_S separately.

iv. It meets prefect forward secrecy. Compromising private keys d_U and/or d_S doesn't jeopardize entire system's security due to choosing independent random numbers g_U and g_S.

v. Both users and server can initialize communication and there is no difference between them.

vi. $M_K.x$ is used just as the key of symmetric encryption or decryption algorithm.

4.3 Bandwidth, Storage and Computational Load

In order to compare the proposed protocols, we consider three parameters, namely *bandwidth, storage requirements* and *computational load.* We use the following values, as mentioned in [11].

M_U, M_S: 161 bits
e_U, e_S: 160 bits
(r_U, s_U) , (r_S, s_S): 320 bits
t_U, t_S, g_U, g_S: 64 bits

Aydos et al. [9] achieved 1666 bits for transmitting data and 1440 bits for storing data in their protocol.

By calculating the number of transmitting and storing bits in mutual authentication parts of each protocol, we measure the number of storing and transmitting bits in different protocols, as shown in Figure 7 and 8, respectively. For example, by looking at Figure 6 we can see that in stage 3 each party sends a 161 bits point on the elliptic curve (namely M_N , M_M). After it in stages 6 and 10 they send encrypted I and t (I_u ,t_u for user and I_s ,t_s for server) that each of them is 64 bits, thus we send totally 2*64 + 161 = 289 bits.

In regards to the storage cost, VER-UAP doesn't need to store *(r,s)* pair that costs 320 bits. But each party needs to store the public key of another part (i.e., 161 bits). Therefore, there is 320-161=159 bits reduction for storage cost. The total number of storing bits regarding to ASK-WAP is 1440 – 159 = 1281 bits.

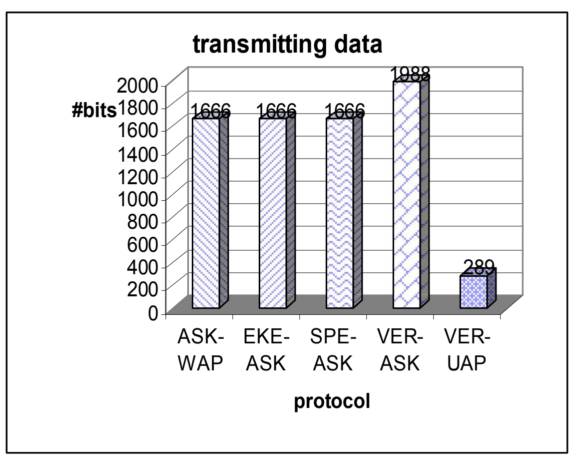

Fig. 7. Number of transmitting bits in different protocols

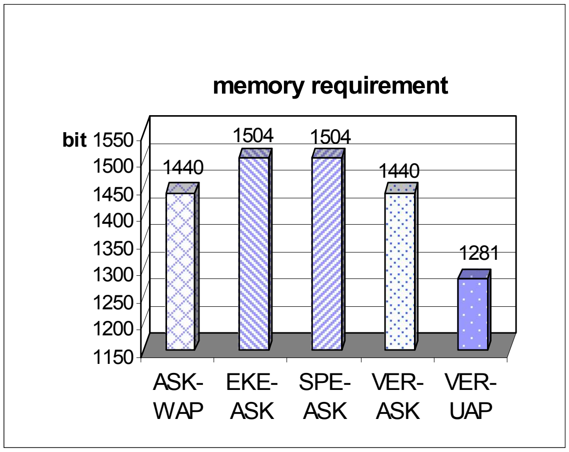

Fig. 8. Number of storing bits in different protocols

To compare their computational load, we consider the following abbreviations:

eP: *Point Multiplication.*
ECDSAV: *Elliptic Curve Digital Signature Algorithm Verification.*
SKE: *Secret Key Encryption or Decryption.*

We have:

ASK-WAP: 1 eP(160 bits) +1 ECDSAV(160 bits)
+2 SKE (800 bits data)
EKE-ASK: 1 eP(160 bits) +1 ECDSAV(160 bits)
+4 SKE (800 bits data)
SPE-ASK: 2 eP(160 bits) +1 ECDSAV(160 bits)
+2 SKE (800 bits data)
VER-ASK: 4 eP(160 bits) +1 ECDSAV(160 bits)
+2 SKE (800 bits data)
+ 2 Point Addition (161 bits)
VER-UAP: 3 eP(160 bits) +2 SKE (128 bits data)
+ 1 Point Addition (161 bits)

As can be seen VER-UAP protocol has acceptable performance while meeting security goals as compared to other methods.

5 CONCLUSION

We analyzed ASK-WAP and UAP, categorized their advantages and disadvantages. We showed that while UAP is compromising some of ASK-WAP weaknesses, it has some security and communication problems. For example, server cannot initialize the communication, it doesn't meet prefect forward secrecy, and its block cipher is vulnerable because it uses $M_k.x$ as the main portion of its plain text and its key, simultaneously. By exposing one session key, the server's private key is discovered. So we proposed three derivations of ASK-WAP, namely, EKE-ASK, SPE-ASK and VER-ASK, and a derivation of UAP called VER-UAP. The EKE-ASK and SPE-ASK are based on pre-shared password, while the VER-ASK is enhancement of ASK-WAP. The VER-UAP adds UAP idea to both sides. The obtained results show the efficiency of our proposed methods while meet the desired security goals. We compared them and showed that VER-UAP has the best performance.

7 REFERENCES

[1] R. S. Douglas, "Cryptography theory and practice," Chapman & Hall/CRC, 3rd Ed., 2006.
[2] H.M. Sun, B.T. Hsieh, and S-M. Tseng, "Cryptanalysis of Aydos et al.'s ECC-based wireless authentication protocol," IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE '04), pp. 563-566, 2004.
[3] Z. Juan, and D. Fangmin, "The authentication and key agreement protocol based on ECC for wireless communications," International Conference on Management and Service Science 2009 (MASS '09), pp. 1-4, 2009.
[4] A. Chandrasekar, V. Rajasekar, and V. Vasudevan, "Improved authentication and key agreement protocol using elliptic curve cryptography," International Journal of Computer Science and Security (IJCSS), vol. 3, p. 325, 2009.
[5] Khalique, Aqeel, K. Singh, and S. Sood. "A password-authenticated key agreement scheme based on ECC using smart cards." International Journal of Computer Applications, Vol.2, No.3, pp. 26-30, 2010.
[6] Nicanfar, Hasen, and Victor CM Leung. "Multilayer consensus ecc-based password authenticated key-exchange (mcepak) protocol for smart grid system." IEEE Transactions on Smart Grid, Vol.4, no.1, pp. 253-264, 2013.
[7] Nam J, Kim M, Paik J, Lee Y, Won D. "A Provably-Secure ECC-Based Authentication Scheme for Wireless Sensor Networks." Sensors. 2014
[8] S. S. Mousavi-nik, M H Yaghmaee-moghaddam and M B Ghaznavi-ghoushchi. "Proposed secureSIP Authentication Scheme based on Elliptic Curve Cryptography."

International Journal of Computer Applications 58(8):25-30, November 2012.
[9] M. Aydos, B. Sunar, and C. K. Koc, "An elliptic curve cryptography based authentication and key agreement protocol for wireless communication," 2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications, Dallas, Texas, 1998.
[10] K. Mangipudi, N. Malneedi, R. Katti, and H. Fu, "Attacks and solutions on Aydos-Savas-Koc's wireless authentication protocol," IEEE Global Telecommunications Conference 2004 (GLOBECOM '04), vol.4, pp. 2229-2234, 2004.
[11] W. Stallings, Cryptography and network security, vol. 3: Prentice Hall New Jersey, 2003.
[12] L. Yongliang, W. Gao, H. Yao, and X. Yu, "Elliptic curve cryptography based wireless authentication protocol," International Journal of Network Security, vol. 5, pp. 327–337, 2007.
[13] Indra, Gaurav, and Renu Taneja. "An ECC-Time Stamp based Mutual Authentication and Key Management Scheme for WSNs." 27th International Conference on IEEE Advanced Information Networking and Applications Workshops (WAINA), 2013.
[14] P. FIPS, "186-2: Digital signature standard (DSS)," National Institute of Standards and Technology (NIST), 2000.
[15] P. Koduri, A. Mahajan, P. Montague, and P. Moseley, "An elliptic curve authenticated key exchange based approach to key infrastructure," 3rd International Conference on Knowledge-Based Intelligent Information Engineering Systems, pp.513-517, 1999.
[16] E. Pierre, A. M'hamed, E. L. H. Bachar, and M. Mokhtari, "Secure authenticated and key agreement protocols with access control for mobile environments," International Journal of Computer Science, vol. 6, pp. 170-183, 2009.
[17] P. E. Abi-Char, A. Mhamed, and B. El-Hassan, "A fast and secure elliptic curve based authenticated key agreement protocol for low power mobile communications," The International Conference on Next Generation Mobile Applications, Services and Technologies 2007 (NGMAST '07) , pp.235-240,2007.

AUTHOR PROFILES:

Mojtaba Mohammadpoor received the BSc. degree in Electrical Eng. from the Ferdowsi University of Mashhad, Mashhad, Iran, in 1998, the MSc. degree in telecommunication engineering from the Islamic Azad University, south Tehran branch, Iran, in 2006, and the Ph.D. degree in telecommunication and network engineering from University Putra Malaysia (UPM), in 2012. In 1998 he joined the Telecommunication Ministry of Iran and experienced different job titles in different parts like Mobile Company of Iran and Iranian Telecommunication Research Center till 2008. He is currently an Assistant Professor in the Dept. of Electrical and Computer Eng., University of Gonabad, Gonabad, Iran. His research interests are network and network security, Radar systems, signal and image processing.

Abbas Mehdizadeh obtained his M.Sc. in IT & Multimedia Systems, and Ph.D. in Communication and Network Eng., both from Universiti Putra Malaysia (UPM), in 2008 and 2012, respectively. He is currently Principal Lecturer at Nilai University, Malaysia. He served as researcher at MIMOS Bhd Malaysia, in 2008. Abbas was the proud recipient of the Best-of-the-Best and Gold awards of Malaysia Technology Expo 2009 (MTE'09 – The largest Invention and Innovation Expo in Malaysia). Abbas is a Senior Member of IEEE, member of IEICE-Japan, International Engineering Consortium (IEC), and International Association of Engineers. His research interest includes IPv6, network security, wireless communications and networks, and multicast networks.