Research Journal of Information Technology

ISSN 1815-7432

Academic Journals Inc.

www.academicjournals.com

Research Journal of Information Technology 4 (3): 106-123, 2012
ISSN 1815-7432 / DOI: 10.3923/rjit.2012.106.123

Security Analysis and Enhancement of Authentication in CDMA based on Elliptic Curve Cryptography

[1]Mohammed A. Mahdi, [2]Mohamed M. Abd-Eldayem, [2]Salwa S. Elgamal and [1,3]Tat-Chee Wan

[1]School of Computer Sciences, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia
[2]Faculty of Computers and Information, Cairo University, 12613, Cairo, Egypt
[3]National Advanced IPV6 (NAV6) Centre, Universiti Sains Malaysia, 11800 USM, Penang, Malaysia

Corresponding Author: Mohammed A. Mahdi, School of Computer Sciences, Universiti Sains Malaysia, 11800SM, Penang, Malaysia

ABSTRACT

Now-a-days, CDMA mobile network is used in many countries around the world, therefore, its security system is hot topic. Authentication and encryption are the most important security techniques used by CDMA mobile network. While encryption technique is used to provide confidentiality to the transmitted data and voice over the network, the authentication technique is used to allow only authorized users to access the network. The authentication of CDMA2000 mobile network includes two processes: provisioning the subscriber authentication key and mutual authentication between user and service network. In this study a comparison among some modern public key cryptosystem (RSA, ECC, ECDH, DH, MQV and ECMQV) is implemented, the results show that the superiority of ECC and ECDH public key technique over other public key techniques based on the key length and the implementation speed. For these reason the proposed CDMA2000 authentication technique uses ECDH public key to provide the subscriber authentication key, also ECC is proposed to be used to provide authentication request and mutual authentication between user and service network.

Key words: CDMA, CDMA2000, security, public key, cryptography, mobile network, authentication

INTRODUCTION

CDMA is a abbreviation for Code Division Multiple Access communication, it is a form of spread spectrum communications where voice and data signals are spread over a much wider bandwidth than the original signal, therefore the interference in this bandwidth is minimal.. CDMA use unique codes to differentiate between communicators in the same spectrum. CDMA allows multiple users to share a single radio channel frequency at the same time.

CDMA uses codes to transmit and to distinguish between multiple wireless subscribers. Because of the connections are distinguished by digital codes, multiple users can share the same channel simultaneously. This greatly increases system capacity and allows wireless providers to transmit more information over the same frequency spectrum (Harte *et al*., 1999).

There are many systems of CDMA such as cdmaOne and CDMA2000. cdmaOne is a second generation (2G) wireless technology, it describes a complete wireless system based on the TIA/EIA IS-95 CDMA standard, there are two revisions of cdmaOne which are IS-95A and IS-95B.

CDMA2000 is a third generation (3G) wireless technology that is an evolutionary of cdmaOne, CDMA2000 represents a family of International Telecommunications Union (ITU), International

Mobile Telecommunications (IMT-2000) standards, it includes CDMA2000 1X and CDMA2000 1xEV technologies. CDMA2000 1X can double the voice capacity of cdmaOne networks and delivers peak packet data speeds of 307 kbps in mobile environments. CDMA2000 1xEV includes CDMA2000 1xEV-DO and CDMA2000 1xEV-DV. CDMA2000 1xEV-DO delivers peak data speeds of 2.4 Mbps and supports applications such as MP3 transfers and video conferencing. CDMA2000 1xEV-DV provides integrated voice and simultaneous high-speed packet data multimedia services at speeds of up to 3.09 Mbps (De Vriendt *et al*., 2002).

Now-a-days, CDMA mobile networks are used in many countries around the world, therefore, the security of CDMA mobile network has been raised in the last few years. In this paper an authentication process for CDMA2000 mobile network based on elliptic curve public key technique is proposed.

CDMA2000 MOBILE NETWORK SECURITY

Authentication and encryption are important issues in mobile networks, authentication is used to authenticate user on the network and encryption is used to protect voice and data over the network. CDMA2000 uses a new security technique different from cdmaOne security technique for the following reasons:

- Weakness of the Cellular Authentication and Voice Encryption (CAVE), Cellular Message Encryption Algorithm (CMEA) and ORYX algorithms
- Weakness of the 64-bit key because it is too short
- Lack of mutual authentication (Wingert and Naidu, 2002)

CDMA2000 security architecture: As shown in Fig. 1 CDMA2000 security architecture consist of User Identity Module (UIM), Mobile Station (MS) and network subsystem. The network subsystem consists of two units MSC/VLR and HLR/AC (Rose and Koien, 2004). For the purpose of analysis, in this paper a difference is made between the UIM and the MS. In reality, The UIM is inside the MS and it can either be removable (R-UIM) or not removable (UIM).

The Network Subsystem consists of a Serving Network (SN) and the Home Environment (HE). The HE is the network of a user's personal service operator, it contains of the Home Location Register (HLR) and the Authentication Center (AC). The HLR holds a database containing subscriber data; the AC uses this data to validate users. The SN holds a database consisting of every user currently authenticated on the network. A user can only be in one VLR at any time as he can only visit one SN at a time. The SN will communicate with the HE to exchange data used to authenticate visiting users. The problem for the security architecture is the initial provisioning of the subscriber authentication key K into the UIM and AC. When the subscriber authentication key exists, it can be used in the AKA protocol to establish trust and session keys between the MS and the SN. The session keys can be used to ensure the privacy of the user's communication (whether voice or data) and the integrity of signaling messages (Rose and Koien, 2004; Perez, 2004).

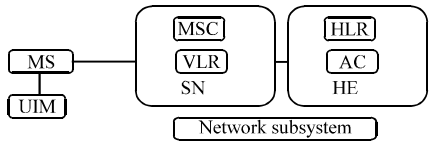

Fig. 1: CDMA2000 security architecture

Authentication of CDMA2000 mobile network: Authentication in CDMA2000 is done between the SN and the MS. Unlike with 2G technologies, the authentication is now mutual. This means that both the MS will be authenticated to the SN and the SN will be authenticated to the MS. This procedure for establishing the security association between the MS and the SN is called AKA.

Provisioning the subscriber authentication key: Before the AKA procedure can be explained we will first discuss the provisioning of the subscriber authentication key (K). K is used as input for almost all of the functions in the AKA procedure, these functions are used in the AC and the MS. If UIM is removable then K is saved in R-UIM during the production phase or by the mobile service provider and K is communicated to the AC along with the batch of R-UIM (Rose and Koien, 2004). There are two standardized methods to provide the key to not removable UIM and the AC. In the first method the customer or salesperson enters a special code of about 26 digits provided by the home service provider to provide the key.

The second, is more preferred uses Over-the-air Service Provisioning (OTASP) method (Rohini, 2004; 3GPP2, 2008), where the customer or salesperson calls a special phone number, during this call a Diffie-Hellman key exchange algorithm (DH) is performed between the MS and the AC. This algorithm allows two parties to agree on a secret key over an insecure channel for more information on the DH algorithm.

To exchange the key K between SN and UIM through Over-the-Air using DH algorithm there are two commands.

MS key request and key generation request as follows:

MS key request: In this step the numbers q and α are sent to the UIM from the network. The MS also generates a random number and sends it to the UIM along with q and α.

The UIM may use this random number for generating its private number X_A, it then calculates:

$$Y_A = \alpha^{X_A} \bmod q$$

and stores the result.

Key generation request: The network sends:

$$Y_B = \alpha^{X_B} \bmod q$$

where, X_B is the network's private number, to the UIM. In response to this, UIM sends Y_A to network. Now, the UIM calculates:

$$K = Y_B^{X_A} \bmod q$$

and the network calculates:

$$K = Y_A^{X_B} \bmod q$$

K is subscriber authentication key.

Authentication and key agreement: The AKA protocol (Rohini, 2004; Blumenthal *et al*., 2002; Mun *et al*., 2009) in CDMA2000 is done between the SN and the MS, this means that the MS will be authenticated to the SN and the SN will be authenticated to the MS. The AKA procedure is executed in two stages: the first stage (authentication request) involves the generation of Authentication Vector (AV) and transfers it from the Home Environment (HE) to the Serving Network (SN), the second stage provides the authentication between MS and SN, this stage called mutual authentication.

Authentication request: Authentication process starts with authentication request by the SN to the HE. When the MS roams into the SN it will make an attempt to register itself by send Mobile Station Identifier (MSID), the SN needs an (AV) to perform the mutual authentication with the MS, the SN can request this AV from the HE, also SN can request more than one AV's at once, so it is possible that the SN will not have to perform this request every time.

Authentication Vector (AV) includes (RAND, XRES, CK, IK, AUTN and UAK) and AUTN includes (SQN, AMF and MAC) (3GPP2, 2008; 3GPP, 2009).

As shown in Fig. 2 AV can be generated as follows: (3GPP, 2009: Blumenthal *et al*., 2002):

- Generates a fresh sequence number (SQN)
- Generates challenge (RAND)
- Computes a message authentication code for authentication MACA = $f1_K$ (SQN || RAND || AMF) where f1 is a message authentication function
- Computes an expected response XRES = $f2_K$ (RAND) where f2 is a message authentication function
- Computes a cipher key CK = $f3_K$ (RAND) where f3 is a key generating function
- Computes an integrity key IK = $f4_K$ (RAND) where f4 is a key generating function
- Computes an anonymity key AK = $f5_K$ (RAND) where f5 is a key generating function and computes the concealed sequence number SQN⊕AK. If SQN is to be concealed
- Compute UIM Authentication Key (UAK) which protects against rogue shell attacks
- Assembles the authentication token:

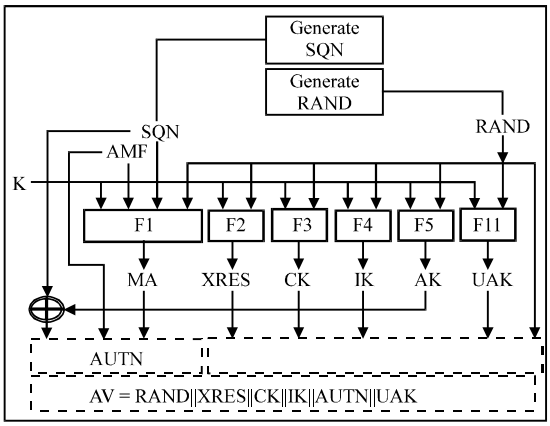

Fig. 2: Generation of authentication vector (AV)

AUTN =[SQN⊕AK] || AMF || MAC and the
AV = RAND || XRES || CK || IK || UAK ||AUTN

- Send AV to SN

Mutual authentication: As shown Fig. 3 and 4 the mutual authentication process is as follows (Perez, 2004; Blumenthal *et al.*, 2002; Mun *et al.*, 2009):

- The SN sends RAND and AUTN to MS
- Upon receipt of RAND and AUTN the UIM first computes the anonymity key AK = $f5_K$ (RAND) and retrieves the sequence number SQN = (SQN⊕AK)⊕AK
- The UIM then computes XMAC = $f1_K$ (SQN || RAND || AMF) and compares XMAC with MAC included in AUTN
- If they are different, the UIM triggers the MS to send back a user authentication response with indication of integrity failure to the VLR and abandons the procedure. The next stages are for the case where XMAC and MAC are equal
- Next the UIM verifies that the received sequence number SQN is acceptable. The MS has some flexibility in the management of sequence numbers
- The UIM then computes the response RES = $f2_K$ (RAND) and triggers the MS to send back RES to the SN
- SN validates the MS by comparing RES to XRES

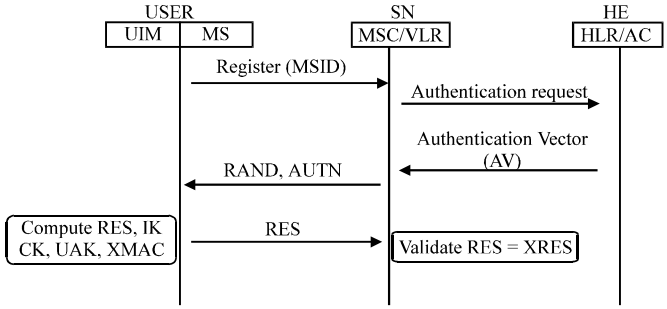

Fig. 3: Full authentication procedures

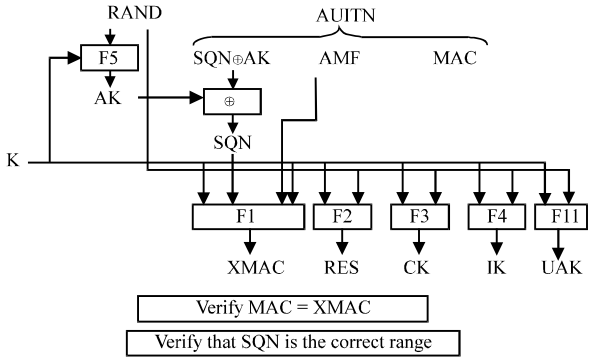

Fig. 4: Authentication function in the MS

Parameters used in the AKA procedure: There are many parameters used in AKA procedure, these parameters will describe as follows (3GPP, 2009; Blumenthal *et al*., 2002):

SQN : This is an increasing unique number. This number is used to prevent replay attacks.
RAND : This is a random number for the challenge and response mechanism.
AMF : The purpose of this parameter is left up to the service provider. An example for its purpose is to allow multiple authentication algorithms and keys.
K : This is a key used as input for almost all functions used by the AKA procedure. The K is secret and known only to the AC and the UIM.
MAC : This value is used to protect and validate the integrity and authenticity of the SQN and AMF
XRES : This parameter is the expected response and will be used by the SN to validate the MS
CK : This key can be used to encrypt user and signaling data after the AKA procedure has finished
IK : This key can be used to generate a MAC for signaling messages
AK : This key is optionally used to conceal the SQN
UAK : This is optional extension to AKA, UAK, unlike the ciphering key and integrity key (CK and IK), is not passed out of the UIM, so this transformation ensures that the UIM is present. This mechanism was introduced to solve the "rogue shell problem," where a mobile does not delete CK and IK when the R-UIM is removed or somehow sends CK and IK to another mobile
AUTN : This is a token made up by a concatenation of parameters. It contains all parameters the MS needs to authenticate itself with the SN:

$$AUTN = [SQN \oplus AK] \| AMF \| MAC$$

AV : This is a vector made up by concatenated parameters containing data used to authenticate a user:

$$AV = RAND \| XRES \| CK \| IK \| AUTN \| UAK$$

These parameters are computed by the HE and sent to the SN. The SN can use them to perform a mutual authentication with the MS (Perez, 2004).

Functions used by the AKA procedure: The following functions are used by the AKA procedure, there are implemented exclusively in the UIM and AC (3GPP2, 2010; Blumenthal *et al*., 2002):

f0 : Random challenge generation function. This function generates random numbers which are used as input for the other functions
f1 : Network authentication function. This function is used by the MS to compute XMAC and by the HE to compute MAC:

$$MAC = f1\ (K, SQN, RAND, AMF)$$

f2 : User authentication function. This function is used by the MS to compute RES and by the HE to compute XRES:

$$RES = f2 (K, RAND)$$

f3 : Cipher key derivation function. This function derivates the session key CK from the K:

$$CK = f3 (K, RAND)$$

f4 : Integrity key derivation function. This function derivates the integrity key IK from the K:

$$IK = f4 (K, RAND)$$

f5 : Anonymity key derivation function. This function derivates AK from the K:

$$AK = f5 (K, RAND)$$

f11 : UIM authentication key derivation function. This function derives UAK from K:

$$UAK = (K, RAND)$$

PUBLIC KEY INFRASTRUCTURE

Public key cryptography, also known as asymmetric cryptography, is a form of cryptography in which a user has a pair of cryptographic keys-a public key and a private key. The private key is kept secret, while the public key may be widely distributed. The keys are related mathematically but the private key cannot be practically derived from the public key. A message encrypted with the public key can only be decrypted with the corresponding private key.

Public key crypto system: secrecy: As shown in Fig. 5 the source A encrypts the message X using B's public key PUb and send encrypted the message Y to B. ($Y = E_{PUb} (X)$), the destination B decrypt the message Y using its private key PRb. Because the message was encrypted using B's public key, only B could have prepared the message ($X = D_{PRb} (Y)$).

There are many modern public key techniques such as (RSA, DH (Stallings, 2010), ECC (Chou, 2003; Lauter, 2004; Kalra and Sood, 2011), ECDH (Stallings, 2010), MQV (Law *et al.*, 1998) and ECMQV (Zheng *et al.*, 2006)) these techniques described in the following subsection.

Diffie-Hellman key exchange (DH): The purpose of the algorithm is to exchange session key between two parties A and B (Stallings, 2010).

For this scheme, there are two publicly known numbers: A prime number q and an integer α that is a primitive root of q.

Suppose the users A and B wish to exchange a key.

User A selects a random integer $X_A<q$ and computes $Y_A = \alpha^{X_A} \bmod q$. Similarly, user B independently selects a random integer $X_B<q$ and computes $Y_B = \alpha^{X_B} \bmod q$.

Each side keeps the X value private and makes the Y value available publicly to the other side. User A computes the key as $K=(Y_B)^{X_A} \bmod q$ and user B computes the key as $K = (Y_A)^{X_B} \bmod q$.

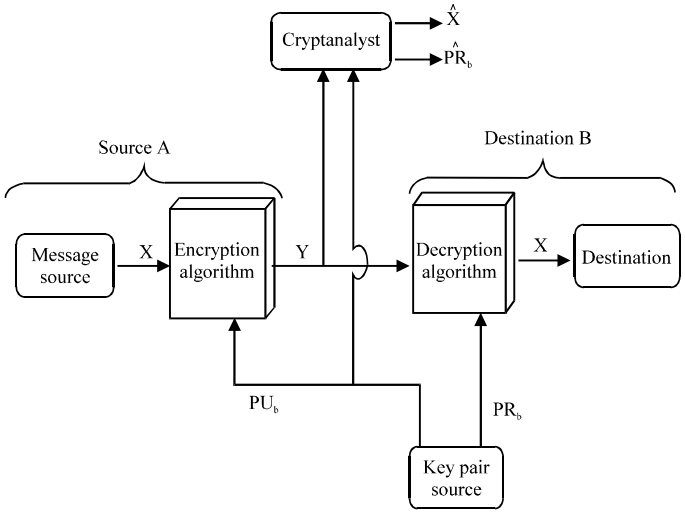

Fig. 5: Public key crypto system: secrecy

These two calculations produce identical key. DH is summarized as follows:

Public elements:

q : Prime number
α : α<q and α a primitive root of q

User A key generation:

- Select private X_A X_A<q
- Calculate public Y_A:

$$Y_A = \alpha^{X_A} \bmod q$$

User B key generation:

- Select private X_B X_B<q
- Calculate publicY_B:

$$Y_B = \alpha^{X_B} \bmod q$$

Generation of secrete key by user A:

$$K = (Y_B)^{X_A} \bmod q$$

Generation of secrete key by user B:

$$K = (Y_A)^{X_B} \bmod q$$

Elliptic Curve Diffie-Hellman key exchange (ECDH): Key exchange using elliptic curves can be done in the following manner, first pick a large integer *q* which is either a prime number p or an integer of the form 2^m and elliptic curve parameters a and b This defines the elliptic group of points Eq (a, b). Next pick a base point G = (x1, y1) in Ep (a, b) whose order is a very large value n.

The order n of a point G on an elliptic curve is the smallest positive integer n such that nG = O. Eq (a, b) and G are parameters of the cryptosystem known to all participants.

A key exchange between users A and B can be accomplished as follows:

- A selects an integer n_A less than n. This is A's private key. A then generates a public key $P_A = n_A$ x G. P_A is a point in Eq (a, b)
- B similarly selects a private key n_B and computes a public key $P_B = n_B$ x G. P_B is a point in Eq (a, b)
- A generates the secret key K = n_A x P_B, B generates the secret key K = n_B x P_A

The two calculations in step 3 produce the same result because n_A x P_B = n_A x (n_B x G) = n_B x (n_A x G) = n_B x P_A To break this scheme, an attacker would need to be able to compute k given G and kG which is assumed hard.

These two calculations produce identical key. ECDH is summarized as follows:

Public elements:

- Eq(a,b) elliptic curve with parameters a, b and q
- where q is a prime or an integer of the form 2^m
- G point on elliptic curve whose order is large value n

User A key generation:

- Select private n_A n_A<n
- Calculate public P_A $P_A = n_A$ x G
- User B key generation
- Select private n_B n_B<n
- Calculate public P_B $P_B = n_B$ x G
- Generation of secrete key by user A:

$$K = n_A \times P_B$$

Generation of secrete key by user B:

$$K = n_B \times P_A$$

Elliptic curve encryption/decryption: As with the key exchange system, an ncryption/decryption system requires a point G and an elliptic group Eq (a, b) as parameters.

Each user A selects a private key n_A and generates a public key $P_A = n_A \times G$.

To encrypt and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext Cm consisting of the pair of points: $C_m = \{kG, P_m+kP_B\}$

Note that A has used B's public key P_B.

To decrypt the ciphertext, B multiphes the first point in the pair by B's secret key and subtracts the result from the second point:

$$P_m+kP_B-n_B (kG) = P_m+k (n_BG)-n_B (kG) = P_m$$

RSA algorithm: RSA is an algorithm for public-key encryption. It was the first algorithm known to be suitable for signing as well as encryption.

RSA involves a public and private key, the public key can be known to everyone and is used for encrypting messages. Messages encrypted with the public key can only be decrypted using the corresponding private key. The keys for the RSA algorithm are generated the follows:

Key generation:

- Select p, q p and q both prime, $p \neq q$
- Calculate $n = p \times q$
- Calculate $\phi (n) = (p-1)*(q-1)$

Select an integer e such that $1<e<\phi (n)$ and gcd $(e, \phi (n)) = 1$.

- Calculate d $d \equiv e^{-1} \bmod \phi (n)$
- Public key $KU = \{e, n\}$
- Private key $KR = \{d, n\}$

Encryption:

- Plaintext $M<n$
- Ciphertext $C = M^e \bmod n$

Decryption:

- Ciphertext C
- Plaintext $M = C^d \bmod n$

THE PROPOSED CDMA2000 AUTHENTICATION TECHNIQUE

The proposed technique enhances the authentication of cdma2000 network by using the best public key algorithm.

A comparison among some modern public key cryptosystem (RSA, ECC, ECDH, ECMQV, DH and MQV) is implemented, the results show that the superiority of ECC and ECDH public key technique over other public key techniques based on the key length and the implementation speed. For these reason the proposed CDMA2000 authentication technique uses ECDH public key to provide the subscriber authentication key, also ECC is proposed to be used to provide authentication request and mutual authentication between user and service network.

The proposed CDMA authentication technique uses ECC to encrypt AV between HE and SN. During the set up of CDMA network infrastructure or during adding a new SN, the HE and each of SN generate their public (P_{HE}, P_{SN}) and private keys (n_{HE}, n_{SN}) based on the ECC public key technique, each SN sends its public key P_{SN} to the HE, also HE sends its public key P_{HE} to all SNs networks. In the following subsections, we describe how to implement the proposed authentication processes: provisioning the subscriber authentication key, authentication request and mutual authentication.

Provisioning the subscriber authentication key: In the previous work The DH algorithm is used to generate K but this algorithm is vulnerable to a man-in-the middle attack, for this reason the ECDH is proposed to be used to provide the subscriber authentication key K.

To exchange the K between HE and UIM through OTASP method and using ECDH algorithm, the customer or salesperson calls a special phone number; during this call the ECDH key exchange algorithm is performed between the MS and the AC. For more information about ECDH algorithm.

To exchange the key K between HE and UIM through OTASP method and using ECDH algorithm there are two commands: mskey request and key generation request. Assume that Eq(a,b) is an elliptic curve with parameters a, b and q, where q is a prime number or an integer of the form 2^m, G is a point on elliptic curve whose order is large value n.

Key exchange using ECDH can be done in the following manner:

- **MS key request:** In this step the numbers n and G are sent to the UIM from the network, the MS also generates a random number and sends it to the UIM along with n and G. The UIM may use this random number for generating its private key n_{MS}. It then calculates MS's public key $P_{MS} = n_{MS} \times G$ and stores the result
- **Key generation request:** The network sends its public key $P_{HE} = n_{HE} \times G$, where n_{HE} is the network's private number, to the UIM. In response to this, UIM sends P_{MS} to the network. Now, the UIM calculates $K = n_{MS} \times P_{HE}$ and the network calculates $K = n_{HE} \times P_{MS}$

Generation of secret key by MS (A):

$$K = n_{MS} \times P_{HE}$$

Generation of secret key by HE (B):

$$K = n_{HE} \times P_{MS}$$

- After provisioning the subscriber authentication key, HE sends its public key (P_{HE}) to MS, then MS store it, after that MS generate its public and private keys (P_{MS}, n_{MS}) and sends its public key (P_{MS}) to HE to store it

Authentication request: Authentication process starts with an authentication request by the SN to the HE. When the MS roams into the SN it will make an attempt to register itself by send Mobile Station Identifier (MSID) to recognize on the MS. The SN needs an Authentication Vector (AV) to perform the mutual authentication between the MS and the SN. Authentication Vector (AV) includes RAND, XRES, CK, IK and AUTN, also AUTN includes SQN, AMF and MAC. HE generates

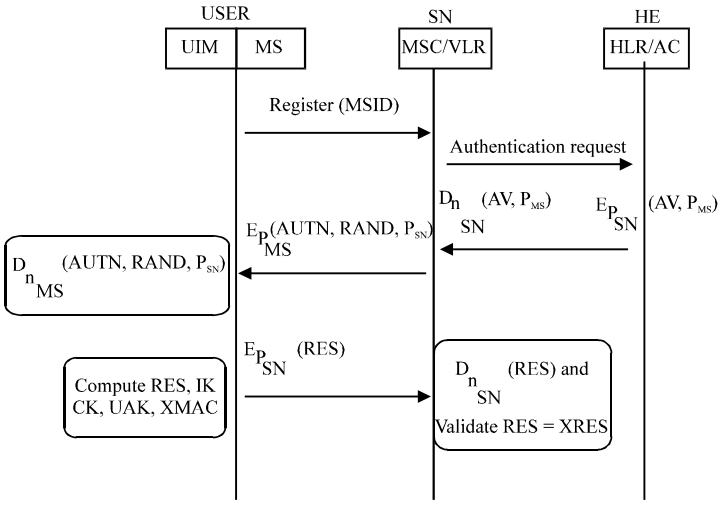

Fig. 6: Proposed CDMA2000 authentications

AV based on subscriber authentication key K (Fig. 2). HE encrypts AV and P_{MS} using public key of SN and then send it to SN ($E_{P_{SN}}(AV, P_{MS})$).

After receiving encrypted AV and $P_{MS,}$ the SN decrypts them using its private key ($D_{n_{SN}}(AV, P_{MS})$).

Mutual authentication: The proposed system enhanced the mutual authentication by securing transfer parameters such as AUTN, RAND, RES, P_{SN} between MS and SN, because of these parameters are transferred over insecure channels.

As shown Fig. 6 the mutual authentication process is as follows:

- The SN sends encrypted RAND, AUTN and P_{SN} ($E_{P_{SN}}(AUTN, RAND, P_{SN})$) using ECC algorithm to MS
- Upon receipt of RAND and AUTN the UIM decrypt RAND, AUTN and P_{SN} ($D_{n_{SN}}(AUTN, RAND, P_{SN})$) then computes the anonymity key $AK = f5_K$ (RAND) and retrieves the sequence number $SQN = (SQN \oplus AK) \oplus AK$
- The UIM then computes $XMAC = f1_K$ (SQN || RAND || AMF) and compares XMAC with MAC included in AUTN
- If they are different, the UIM triggers the MS to send back a user authentication response with indication of integrity failure to the VLR and abandons the procedure. The next stages are for the case where XMAC and MAC are equal
- Next the UIM verifies that the received sequence number SQN is acceptable. The MS has some flexibility in the management of sequence numbers
- The UIM then computes the response $RES = f2_K$ (RAND) and triggers the MS to send back encrypted RES using ECC algorithm to the SN
- SN decrypts RES using its private key (n_{SN}) ($D_{n_{SN}}(RES)$) and validates the MS by comparing RES to XRES

RESULTS AND IMPLEMENTATION

To test the proposed CDMA authentication technique some of the famous public key cryptographic algorithms are implemented using VC6 programming language on a computer with processor Intel Core 3, 3.06 GHz and 2 GB of RAM.

The encryption and decryption time using each of the previous public key technique is measured in milhisecond (ms) and then a comparison is made among them. After a comparison we find that ECDH is better than DH and ECC is the best public key technique, therefore, we propose to use ECDH during the provisioning of the subscriber authentication key. Also we propose ECC to encrypt the authentication parameters (AV, AUTN, RAND RES, P_{MS} and P_{SN}) during their transmission between MS and network.

The implementation results are described in the following subsections.

Comparison of public key techniques: To compare the performance of the public key techniques, we independently calculated the time of key generation, encryption and decryption processes; the performances were tested according to the Table 1 (Barker *et al.*, 2007).

Table 1 show that public key algorithms based on elliptic curve provide the same level of security that other public key provide, however the key size used within elliptic curve public key is the smallest key size. For example ECC with key size equals 160-bits is equivalent in security level to other public key algorithms with key size equals 1024-bits. This difference becomes even more dramatic when security level increases. For example ECC with key size equals 521-bits is equivalent in security level to other public key algorithms with key size equals 15360-bits.

Table 2 show that, the Key generation of public key based on elliptic curve (ECC, ECDH and ECMQV) is faster than other public key algorithms with the smallest key length, while the key size of public key algorithm based on elliptic curve grows linearly with increasing the level of security, the key size of other public key algorithms growth exponentially.

Table 3 describe the comparison between RSA and ECC based on the encryption time of message with size equals 64 bytes. It is clear that the encryption using ECC is slower than using RSA for the same level of security. Also with increasing the level of security the difference of encryption speed between RSA and ECC is increased.

Table 4 describe the comparison between RSA and ECC based on the decryption time of message with size equals 64 bytes. It is clear that the decryption using ECC is faster than using RSA for the same level of security. Also with increasing the level of security the difference of decryption speed between RSA and ECC is increased.

Table 1: Key size for equivalent security levels (in bits)

	Key length	
Security level	ECC/ECDH/ECMQV	RSA/DH/MQV
I	160	1024
II	224	2048
III	384	7680
IV	521	15360

Table 2: Key generation time (ms)

Security level	ECC	DH	MQV	ECDH	ECMQV	RSA
I	0.54	0.89	0.98	0.87	0.95	3.600
II	1.13	3.24	3.94	1.37	1.56	1.483
III	3.58	8.53	11.23	4.25	4.93	25.437
IV	8.75	32.15	43.28	12.65	20.47	243.730

Table 3: ECC and RSA Encryption time (ms)

Security level	ECC	RSA
I	0.31	0.16
II	0.47	0.31
III	1.72	0.35
IV	6.41	0.51

Table 4: ECC and RSA Decryption time (ms)

Security level	ECC	RSA
I	0.17	0.29
II	0.31	1.87
III	0.94	8.75
IV	3.59	54.06

From the Table 2, it is clear that the speed of key generation using ECDH public key technique is faster than the speed of key generation using DH public key technique and this is true for all level of security, therefore, ECDH is used in the proposed system to provide subscriber authentication key.

From all of the previous results and compared to RSA public key technique, ECC public key technique provides the same level of security using much smaller key size, this leads to faster computations and lower usage of memory. For there reasons ECC is used in the proposed system to encrypt and decrypt the authentication parameters (AV, AUTN, RAND, RES, P_{MS} and P_{SN}).

Implementation of proposed CDMA2000 authentication technique: To test the performance of the proposed CDMA2000 authentication technique, we built a simulator using VC6 programming language. This simulator consists of four parts; the first and second part simulates the process of provisioning the subscriber authentication key using DH and ECDH public key cryptosystems. The third and fourth parts simulates the process of mutual authentication using RSA and ECC public key cryptosystems.

The results of implementing the first two parts of the simulator show that the generation time of the subscriber authentication key using ECDH (= 0.87 ms) is less than the generation time of this key using DH (= 0.89 ms).

The results of the third and fourth parts of simulator include the encryption and decryption time of the secure parameters (AV, P_{MS}, AUTN, RAND, P_{SN} and RES) using RSA and using ECC public key cryptography for the same level of security (Table 1).

Table 5 describe the comparison between RSA and ECC based on the encryption time of ($AV+P_{MS}$) with size equals 800-bits using ECC and 1664-bits using RSA for the security level (I), also encryption time of (AV, P_{MS}) with size equals 864-bits using ECC and 2688-bits using RSA with security level (II).

AV = 640-bits and P_{MS} = 160-bits with using ECC for security level (I), P_{MS} = 224-bits using ECC for security level (II), also P_{MS} = 1024-bits using RSA for security level (I) and P_{MS} = 2048-bits using RSA for security level (II).

Table 5 show that the encryption time of ($AV+P_{MS}$) using ECC is slower than using RSA for the same level of security. Table 6 describe the comparison between RSA and ECC based on the decryption time of ($AV+P_{MS}$) in the SN.

Table 5: Encryption time of (AV+ P_{MS}) (ms)

Security level	ECC	RSA
I	37	25
II	55	41

Table 6: Decryption time of (AV+P_{MS}) (ms)

Security level	ECC	RSA
I	0.22	0.32
II	0.39	1.93

Table 7: Encryption time of (AUTN+RAND+P_{SN}) (ms)

Security level	ECC	RSA
I	0.32	0.22
II	0.51	0.37

Table 8: Decryption time of (AUTN+RAND+P_{SN}) (ms)

Security level	ECC	RSA
I	0.19	0.30
II	0.34	1.89

Table 6 show that the decryption time of (AV, P_{MS}) using ECC is faster than using RSA for the same level of security.

Table 7 describe the comparison between RSA and ECC based on the encryption time of (AUTN+RAND+P_{SN}) with size equals 416-bits using ECC and 1280-bits using RSA for the security level (I), also encryption time of (AUTN+RAND+P_{SN}) with size equals 480-bits using ECC and 2304-bits using RSA with security level (II) in the SN.

AUTN = 128-bits, RAND = 128-bits and (P_{SN} = 160-bits with using ECC for security level (I), P_{SN} = 224-bits using ECC for security level (II)) also (P_{SN} = 1024-bits using RSA for security level (I) and P_{SN} = 2048-bits using RSA for security level (II).

Table 7 show that the encryption time of (AUTN + RAND+ P_{SN}) using ECC is slower than RSA for the same level of security.

Table 8 describe the comparison between RSA and ECC based on the decryption time of (AUTN+ RAND+ P_{SN}) in the SN. Table 8 show that the decryption time of the (AUTN+RAND+P_{SN}) using ECC is faster than RSA for the same level of security.

Table 9 and describe the comparison between RSA and ECC based on the encryption time of (RES) knowing that the size of RES equals 128 bit in MS. Table 9 show that the encryption of (RES) with size equals 128 bit using ECC is slower than using RSA for the same level of security.

Table 10 describe the comparison between RSA and ECC based on the decryption time of (RES) in the SN. Table 10 show that the decryption time of (RES) using ECC is faster than using RSA for the same level of security.

Table 11 describes the comparison between RSA and ECC based on the encryption and the decryption of the authentication parameters in MS for the level (I) of security.

Table 9: Encryption time of (RES) (ms)

Security level	ECC	RSA
I	0.27	0.14
II	0.39	0.27

Table 10: Decryption time of (RES) (ms)

Security level	ECC	RSA
I	0.16	0.21
II	0.29	1.78

Table 11: Time of encryption and decryption parameters in the MS

	Technique (ms)	
MS	ECC	RSA
Time to DEC (AUTN, RAND, P_{SN})	0.19	0.30
Time to Enc (RES)	0.27	0.14
Total time	0.46	0.44

Table 12: Time of encryption and decryption parameters in network

	Technique (ms)	
Network	ECC	RSA
Time to Enc (AV, P_{MS})	0.37	0.25
Time to Dec (AV, P_{MS})	0.22	0.32
Time to Enc (AUTN, RAND, P_{SN})	0.32	0.22
Time to Dec (RES)	0.16	0.21
Total time	1.07	1.00

Table 11 shows that the time of encryption and decryption of the authentication parameters in MS using RSA is less than the time of encryption and decryption of the authentication parameters in MS using ECC for the level (I) of security.

Table 12 describes the comparison between RSA and ECC based on the encryption and the decryption of the authentication parameters in the network using RSA for the level (I) of security. Table 12 shows that the time of encryption and the time of decryption of the authentication parameters in the network using RSA is less than the time of encryption and decryption for the authentication parameters in MS using ECC for the level (I) of security.

CONCLUSION AND FUTURE WORK

CDMA mobile network are used in many countries around the world. Authentication and encryption are the most important security techniques used by CDMA mobile network. While encryption technique is used to provide confidentiality to the transmitted data and voice over the network, the authentication technique is used to allow only authorized users to access the network. The authentication of CDMA2000 mobile network includes two processes: provisioning the subscriber authentication key and mutual authentication between user and service network.

In this study a comparison among some modern public key cryptosystem (RSA, ECC, DH and MQV) is implemented, the proposed CDMA authentication technique is used ECC public key to provide the subscriber authentication key and to provide authentication request and mutual authentication between user and service network.

The results show that the speed of key generation using ECDH is faster than the speed of key generation using DH public key technique and this is true for all level of security. Also the results show that ECC public key technique provides the same level of security using much smaller key size compared with other public key techniques and this lead to faster computations and lower usage of memory.

For there reasons, ECDH is proposed to be used for provisioning subscriber authentication key and ECC is proposed to be used to encrypt the authentication parameters AV, P_{MS}, AUTN, RAND, P_{SN} and RES during the mutual authentication.

The implementation results show that proposed provisioning subscriber authentication key using ECDH is faster than using DH with smaller key size.

Also the results show that the mutual authentication time using RSA is faster than using ECC but ECC provides the same level of security using much smaller key.

As a future work, the proposed technique can be used for providing authentication for GSM mobile network. Also as a future work, public key certificates from certificate authority can be used for world wide authentication for the mobile networks.

REFERENCES

3GPP, 2009. Security architecture. 3GPP TS 33.102, Version 9.1.0.

3GPP2, 2008. Enhanced cryptographic algorithms. 3GPP2 S.S0055-A, Version 4.0. Qualcomm Incorporated, USA.

3GPP2, 2010. Over-the-air service provisioning of mobile stations in spread spectrum standards. 3GPP2 C.S0016-D, Version 1.0.

Barker, E., W. Barker, W. Burr, W. Polk and M. Smid, 2007. Recommendation for key management-part 1: General (Revised). NIST, Special Publication 800-57, March, 2007.

Blumenthal, U., M. Marcovici, S. Mizikovsky, S. Patel, G.S. Sundaram and M. Wong, 2002. Wireless network security architecture. Bell Labs Technical J., 7: 19-36.

Chou, W., 2003. Elliptic curve cryptography and its applications to mobile devices. University of Maryland, College Park. http://www.cs.umd.edu/Honors/reports/ECCpaper.pdf

De Vriendt, J., P. Laine, C. Lerouge and X. Xu, 2002. Mobile network evolution: A revolution on the move. Wireless Commun., 40: 104-111.

Harte, L., M. Hoenig, D. McLaughlin and R. Kta, 1999. CDMA IS-95 for Cellular and PCS. 1st Edn., McGraw-Hill Publisher, New York, ISBN: 978-0070270701, Pages: 300.

Kalra, S. and S.K. Sood, 2011. Elliptic curve cryptography: Survey and its security applications. Proceedings of the International Conference on Advances in Computing and Artificial Intelligence, Rajpura/Punjab, India, July 21-22, 2011, ACM, New York, USA., pp: 102-106.

Lauter, K., 2004. The advantages of elliptic curve cryptography for wireless security. IEEE Wireless Commun., 11: 62-67.

Law, L., A. Menezes, M. Qu, J. Solinas and S. Vanstone, 1998. An efficient protocol for authenticated key agreement. Technical Report, CORR 98-05, Department of CO., University of Waterloo.

Mun, H., K. Han and K. Kim, 2009. 3G-WLAN interworking: Security analysis and new authentication and key agreement based on EAP-AKA. Proceedings of the Wireless Telecommunications Symposium, April 22-24, 2009, Prague, Taiwan, pp: 1-8.

Perez, F.A., 2004. Security in current commercial wireless networks: A survey. School of Electrical and Computer Engineering, Purdue University West Lafayette.

Rohini, P.P., 2004. Over-the-air provisioning in CDMA. Gemplus Technologies, October 2004.

Rose, G. and G.M. Koien, 2004. Access security in CDMA2000 including a comparison with UMTS, wireless communications. IEEE, 11: 19-25.

Stallings, W., 2010. Cryptography and Network Security. 4th Edn., Pearson Education, Inc., New Jersey.

Wingert, C. and M. Naidu, 2002. CDMA 1xRTT security. Over View August, 2002, Qualcomm Inc.

Zheng, S., D. Manz, J. Alves-Foss and Y. Chen, 2006. Security and performance of group key agreement protocols. Proceedings of the IASTED International Conference on Networks and Communication Systems, March 29-31, 2006, Chiang Mai, Thailand, pp: 321-327.