An ECC based Public Key Infrastructure usable for Mobile Applications

Sangram Ray
Department of Computer Science and Engineering
Indian School of Mines, Dhanbad
Dhanbad-826004, Jharkhand, India
+91-8797369171
sangram.ism@gmail.com

G. P. Biswas
Department of Computer Science and Engineering
Indian School of Mines, Dhanbad
Dhanbad-826004, Jharkhand, India
+91-9431124198
gpbiswas@gmail.com

ABSTRACT

The demand of a mobile phone and its various applications are increasing rapidly in recent era and as a result, it becomes vital to design and/or improve the existing PKI (Public Key Infrastructure) useful for mobile phones. Since a mobile phone has small screen, low computing power, small storage capacity etc, the present paper proposes an ECC-based PKI that overcomes all the limitations for the mobile phones as mentioned above. For this, we introduce a Mobile Home Agent (MHA) and Registration Authority (RA) that minimize the major work/processing loads of mobile phone and Certificate Authority (CA), respectively. In addition, the use of ECC reduces the computation cost, message size and transmission overhead over RSA significantly. The security analysis of the proposed PKI against relevant attacks and the comparisons with existing schemes shows overall improved performance.

Categories and Subject Descriptors

D.3.2 [**EC**]: Elliptic Curve – *elliptic curve cryptography.*
H.2.3 [**ECDL**]: Elliptic Curve Discrete Logarithm – *computational Diffie-Hellman Assumption, decisional Diffie-Hellman Assumption*.

General Terms

Design, Security.

Keywords

Elliptic Curve Cryptography (ECC); Public Key Infrastructure (PKI); Wireless Application Protocol (WAP); Certificate Management Protocol (CMP); Certificate Authority (CA); Registration Authority (RA); Mobile Home Agent (MHA); Wireless Certificate Management Protocol (WCMP).

1. INTRODUCTION

The demand of several e-services supporting the mobile phones has been increasing rapidly in recent years, where wireless technology and the supporting infrastructure with tools/software are playing a major role. As a result, even during their mobility, the mobile users can now access the internet through wireless communication using mobile phones for their information, recourses, data etc. very easily. Also different mobile applications like in banking, payment, business, health care, voting are in use and developing. However, the security aspect in wireless/mobile technology as such has not been considered and it becomes a great concern for mobile users to successfully utilize the electronic services through the wireless systems. Generally, five different fundamental security functions like confidentiality, data integrity, entity authentication, availability and non-repudiation are considered for an application to be secured and thus to support these security functions to mobile phone as well as wireless environment, the development of mobile-PKI is needed, where a PKI provides initial security associations among arbitrary entities based on which actual security functions are applied [1-5].

The PKI in wired environment [5, 6, 7] is well established, where a PKI is mainly used to generate a public key certificate for a user containing a valid and authenticate public key with its owner's identity and validity period, and the same is used for secure negotiation of cryptographic parameters. To get a certificate from PKI, the CMP (Certificate Management Protocol) supports the online interactions between PKI user and management entities. It is also used to carry the user's registration information and certificate revocation request. However, it is difficult to use wired PKI in wireless environment since wireless internet [4, 7, 8] has lower transmission bandwidth and mobile phone has fundamental limitations like small screen size, low computing power, low memory capacity, [5, 7, 9, 10, 11] etc. Hence, the requirement of PKI/CMP suitable for mobile environment must be developed for mobile-PKI to guarantee security of different mobile applications.

At present, the mobile phone is supported by WAP (Wireless Application Protocol) to access wireless internet and several extensions of WAP such as WAP 1.x, WAP 2.0 [6] etc. are proposed. Among them, the WAP 1.x [5, 8] is applied in wireless internet, where WTLS (Wireless Transport Layer Protocol), equivalent to SSL (Secure Socket Layer) of the wired internet, is in-charge of security. However, the WTLS can't support end-to-end security since in WAP, data must go through the WAP gateway that decodes the WTLS's encrypted data before it is encrypted using SSL and transmitted to the server and inversely, the data encrypted with SSL is decoded by the WAP gateway before it is encrypted with WTLS and sent to mobile phone. This causes a serious security problem in the gateway [5, 7, 12] since WTLS could not support the confidentiality of certificate request message. Another problem of WAP is that it does not support POP (Proof of Possession) [11, 13, 14] function in its certificate request message.

To eradicate these problems, Y. Lee et al. [15, 16] proposed a wireless CMP suitable for wireless PKI in 2007 and 2008 where a

CCSEIT-12, October 26-28, 2012, Coimbatore [Tamil nadu, India]
Copyright © 2012 ACM 978-1-4503-1310-0/12/10…$10.00.

mobile phone securely requests a public key certificate to CA (Certificate Authority). After receiving the request, CA performs the user authentication and verifies the POP function, which means that the mobile phone has generated the valid private key corresponding to the public key. If the verification passes, CA publishes the certificate on its directory or WEB and should give a certificate or certificate URL to the mobile user. Some of the limitations of this scheme are observed such as it requires an out-of-band user identification at CA to collect an ID and password for certificate request which is not realistic. Also, the mobile phone generates the certificate request message itself, which may not be always possible for a device like mobile phone that has less memory and less powerful CPU.

To overcome these limitations, we propose a MHA (Mobile Home Agent) based Mobile-PKI using elliptic curve cryptography (ECC) in this paper, where MHA performs all cryptographic operations in favor of mobile phone. In our scheme, instead of RSA, ECC [19-21] is used which provides same security of RSA with comparatively less processing time and less key size, for instance, a 1024-bit key in RSA is equivalent with 160-bit in ECC. Moreover, a MHA holds the full responsibility on behalf of mobile phone to store all the cryptographic information of mobile phone and performs all the operations to get a certificate from CA. A Registration Authority (RA) is also considered to minimize the work load of CA where RA performs the user authentication and POP verification procedure instead of CA. If the verification passes, RA approved the certificate request message and sends it to CA with attestation. After receiving, CA checks the attestation of RA of the certificate request message and then creates the ECC-based public key certificate of mobile phone, signs on it, and publishes the certificate in its directory and finally, gives a certificate URL to MHA. After receiving, MHA only forwards the certificate URL to the mobile phone, and the mobile phone user uses the URL for authentication and any secret value negotiation with other entity through open channel.

The remaining parts of this paper are organized as follows: Section 2 introduces the Elliptic Curve Cryptography, its computational problems and the CA/RA for issuing and management of ECC-based public key certificate. The proposed Mobile Home Agent based Mobile-PKI using ECC is given in section 3, whose security analysis is described in section 4. Section 5 provides a comparison results with the existing schemes that shows improved performance, and finally, the paper is concluded in section 6.

2. PRELIMINARIES

To facilitate of our proposed scheme, the following articles are briefly introduced.

2.1 Elliptic Curve Cryptography (ECC)

The elliptic curve cryptosystem [18, 19] was initially proposed by Koblitz [20] and then Miller [21] in 1985 to design public key cryptosystem and presently, it becomes an integral part of the modern cryptography. A brief introduction of ECC is given below.

Let *E/Fp* denotes an elliptic curve *E* over a prime finite field *Fp*, which can be defined by

$$y^2 = x^3 + ax + b \qquad (1)$$

where, $\mathrm{a},\ \mathrm{b} \in \mathrm{F}_p$ and the discriminate $D = 4a^3 + 27b^2 \neq 0$

The points on *E/Fp* together with an extra point *O* called the point at infinity used for additive identity form an additive group *A* as

$$A = \{(x, y) : x, y \in F_p,\ E(x, y) = 0\} \bigcup \{0\} \qquad (2)$$

Let *n*, the order of *A*, is very large and it can be defined as $n \times G$ $mod\ q = O$, where *G* is the generator of *A*. Also *A* be a cyclic additive group under the point addition ‘‘+’’ defined as follows

$$P + O = P, \text{ where } P \in A.$$

The scalar point multiplication over *A* can be defined as

$$tP = P + P + \cdots + P \text{ (t times)} \qquad (3)$$

If $P, Q \in A$, the addition $P + Q$ be a point $-R$ (whose inverse is *R* with only changing the sign of *y* coordinate value and lies on the curve) on the E/F$_P$ such that all the points *P, Q* and *–R* lie on the straight line, i.e., the straight line cuts the curve at *P, Q* and *–R* points. Note that if $P = Q$, it becomes a tangent at *P* or *Q*, which is assumed to intersect the curve at the point *0*.

The security strength of the ECC lies on the difficulty of solving the Elliptic Curve Discrete Logarithm Problem (ECDLP) [19-21] and it provides same level of security of RSA with less bit-size key, which are addressed in the next sub-section.

2.2 Computational Problems

Similar to the DH problem (known as discrete logarithm problem) [22], some computational hard problems on ECC are defined below, which have not any polynomial time algorithm.

- ***Elliptic Curve Discrete Logarithm Problem (ECDLP)***

Given $Q, R \in A$, find an integer $k \in F_p^*$ such that R=k.Q.

- ***Computational Diffie-Hellman Assumption (CDHA)***

Given $P, xP, yP \in A$, it is hard to compute $xyP \in A$.

- ***Decisional Diffie–Hellman Problem (DDHP)***

Given $P, aP, bP, cP \in G$ for any $a, b, c \in F_p^*$, decide whether or not $cP = abP$.

2.3 Certificate Authority (CA), Registration Authority (RA) and ECC-based Certificate

A CA [13], which is the base of a PKI, is a trusted agency that issues digital certificates for the verification and validation of users’ public keys with the owners of certificates. The entities individually contact with a CA by providing their identity such as name, address, date of birth, public key etc of each entity and after validation through handshake procedure, the CA performs some level of entity authentication according to Certificate Practices Statement (CPS) [17] and then issues a digital certificate containing the entity’s identity and public key for each entity. The entities can now exchange the certificates among themselves and become authenticated to each other which assure the receiving of the authenticated public keys of the entities.

To minimize the workload of CA, many PKI considers the existence of a Registration Authority (RA) [13] separated from the CA to perform authentication, verification, distribution, revocation reporting etc. where each RA being certified by CA has both authenticated private and public key pairs.

The Public-Key Infrastructure working group of Internet Engineering Task Force (IETF) similar to the X.509 RSA based public key certificate [24-28], specifies ECC based public key certificate standardized as the PKI-X.509. Subsequently, the Elliptic Curve Digital Signature Algorithm (ECDSA), Elliptic Curve Diffie-Hellman (ECDH) public keys and the generation of ECC based certificate with ECDSA signature of PKIX are proposed in [23, 27, 28]. Note that the ECC-based PKIX is easily interoperable with RSA-based PKI (X.509) and the Certificate Authority (CA) issues and certifies ECC-based certificates. A simple ECC-based X.509 certificate format [23] to combine user's identity and the ECC-based public key proposed by ITU (International Telecommunication Union) is described in figure 1.

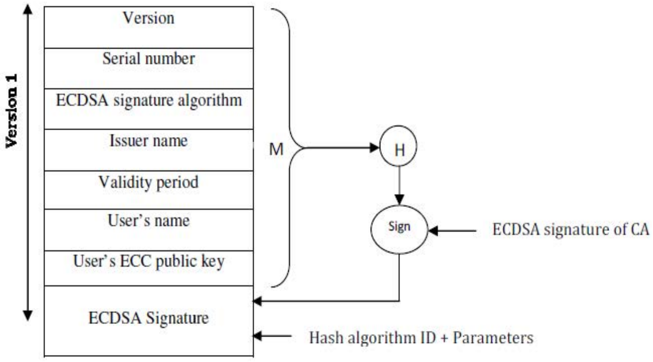

Figure 1. ECC-based X.509 certificate format

2.4 Certificate Management Protocol (CMP)

The Certificate Management Protocol holds the online interactions between PKI user and management entities such as RA/CA. It manages the life cycle of the certificate through initial issue, renewal, suspension and revocation, which are briefly described below [15, 16].

2.4.1 Initial Issue of Certificate

The initial certificate request protocol has five steps as follows:

Step 1: User generates a public-private key pair and sends a certificate request message to CA.
Step 2: CA receives the request and checks the certificate request message.
Step 3: CA performs the user authentication.
Step 4: CA checks the user must possess a private key corresponding to the public key received.
Step 5: CA issues the public key certificate to the user.

2.4.2 Renewal of Key Pair and Certificate

The renewal of the user's key pair is required when the private key has been damaged, lost, leaked or stolen. To get an updated key pair, the user sends a key pair renewal request together with his public key and information for POP to CA, which also updates the certificate during renewal of the key pair. On the other hand, in case of certificate renewal, a user sends his certificate update request to CA before the certificate expiration date for maintaining the validity of the certificate.

2.4.3 Suspension and Revocation of Certificate

The certificate suspension request is sent by the user to CA only when he suspects that his private key has been damaged, lost, leaked or stolen. Later, he checks the security of the certificate and depending upon the outcome he takes any one of the following steps: 1) If the user realizes that his private key has not damaged, lost, leaked or stolen, then he sends the certificate reuse request to CA and to change the status of the certificate as valid which make him able to reuse the same public key as the key pair and the certificate, 2) If user be sure that his private key has been damaged, lost, leaked or stolen, then he sends the certificate revocation request to CA to revoke his certificate and as a result, CA revokes that certificate and includes it in its CRL (Certificate Revocation List) [17].

3. THE PROPOSED ECC-BASED MOBILE-PKI

The proposed ECC-based Mobile-PKI is described in this section, which as stated before, is more efficient than RSA-based PKI in terms of computation and communication cost and key size used with comparable security. Similar to mobile internet, a MHA, which is considered to overcome the fundamental limitations of mobile phone, stores all the information of mobile phone and performs all the operations to get a certificate of mobile phone from CA. Moreover, a trusted RA is also considered to minimize the work load of CA such as user authentication and POP verification procedures.

3.1 Mobile Public Key Certificate Management Procedure

The proposed MHA based Certificate Management Procedure is discussed below, where a workflow diagram is shown in figure 2 for visual illustration.

(1) MHA generates an ECC-based public-private key pair for the mobile phone and sends it to RA as a *certificate request message* along with others relevant information to get the public key certificate of mobile phone from CA.

(2) After receiving, RA authenticates MHA and verifies the POP function to be sure that the user possess a private key corresponding to the public key for which a certificate being requested. If the verification passes, RA approves the *certificate request* message and sends it to CA with attestation for generating the certificate.

(3) CA checks whether the *certificate request* message is attested by RA, if so, CA creates the public key certificate, signs it, and publishes the certificate in its directory.

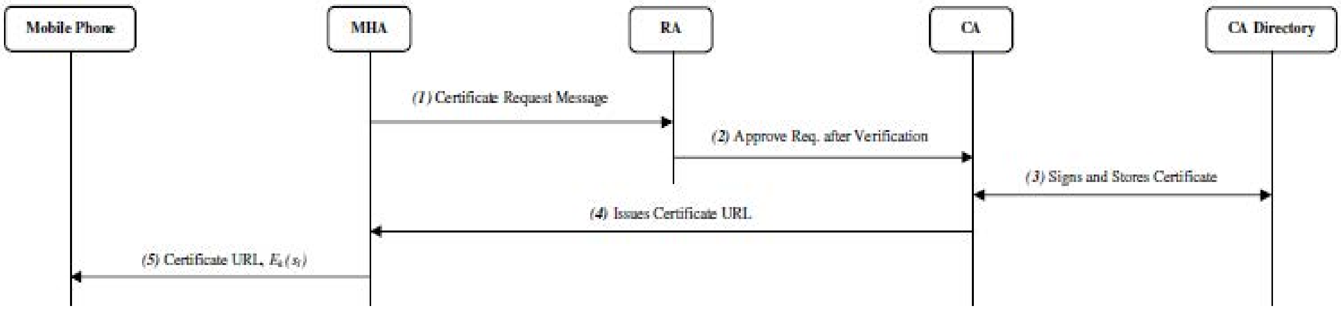
Mobile Phone
MHA
RA
CA
CA Directory
(1) Certificate Request Message
(2) Approve Req. after Verification
(3) Signs and Stores Certificate
(4) Issues Certificate URL
(5) Certificate URL, $E_k(s_1)$

Figure 2. Workflow diagram of proposed certificate management procedure

(4) CA generates the certificate, stores it at CA-directory and sends a *certificate URL* to MHA.

(5) On receiving, MHA forwards the same to the mobile phone along with the mobile phone's private key encrypted using the pre-shared secret key *k*.

Now, the mobile phone user gets the *certificate URL* and decrypts the encrypted private key using the pre-shared secret key *k* and gets his private key.

3.2 Notations Used

$h(.)$	One-way hash function such as SHA1;
ID_{MHA}	Identity of mobile home agent
ID_M	Identity of mobile phone
$\|\|$	Concatenation
p, n	Two large prime numbers;
Fp	A finite field;
E	An elliptic curve defined on Fp with prime order n;
P	A point on elliptic curve E with order n;
(s_1, V_1)	Private/public key pair of MHA, where $V_1 = s_1P$;
(s_2, V_2)	Private/public key pair of RA, where $V_2 = s_2P$;

3.3 Certificate Request Message Generation and Verification

Initially, MHA of a mobile phone user collects the public key of a RA and generates a certificate request message with an ECC-based public-private key pair $(s_1, V_1= s_1P)$ and then, sends it to RA. After receiving, RA verifies the certificate request message after prior mutual authentication and forwards to CA with attestation. The details of the *certificate request message generation and verification* procedure are given below along with a diagram as shown in figure 3.

(1) $MHA \rightarrow RA$: $M \,||\, E_{K_X} (H \,||\, R_1)$

Certificate request message *M* is generated by MHA by concatenating its identity, mobile phone's identity and public key and the hash digest of the same is computed as $M = ID_{MHA}||ID_M||V_1$ and $H=h(M)$. Now it selects a random number r_1 and generates $R_1 = r_1.P$ and also calculates an ECDH session key $K = s_1.V_2 = s_1.s_2.P = (K_X, K_Y)$. Then MHA concatenates the hash digest of *M* with R_1, encrypts the concatenated message using K_X, concatenates *M* with the encrypted message and then sends the concatenated message to RA as a certificate request message.

(2) $RA \rightarrow MHA$: R_1+R_2, $E_{K_X} (ID_{RA}|| h (R_2))$

After receiving the certificate request message, RA gets the identity of MHA and mobile phone and also gets the public key of the mobile phone from *M* for which the certificate is requested. Now it calculates the hash digest *H'* of received *M* as $H' = h(M)$ and the ECDH session key using mobile phone's public key as $K = s_2.V_1 = (K_X, K_Y)$, decrypts the encrypted message using K_X, gets *H* and R_1 and then, compares the received *H* with calculated *H'*. If both match, then RA confirms that the MHA have generated the private key for the corresponding public key i.e. POP function is verified. Now, for authentication purposes, RA selects a random number r_2 and generates $R_2=r_2.P$, calculates the hash digest of R_2 as $h(R_2)$, encrypts its identity and the hash digest using K_X and then sends the encrypted message along with (R_1+R_2) to MHA for authentication.

(3) $MHA \rightarrow RA$: $E_{K_X} (ID_{MHA}|| h (R_2))$

MHA decrypts the encrypted message using K_X and gets the identity of RA and $h(R_2)$. It also retrieves R_2 by subtracting R_1 from $(R_1 + R_2)$, calculates the hash digest of R_2 and compares it with the received hash digest. If both match, RA is authenticated to MHA and then to be authenticated to RA, MHA calculates the hash digest of R_2 as $h(R_2)$, encrypts its identity and the hash digest using K_X and then sends the encrypted message to RA.

(4) $RA \rightarrow MHA$: *Yes/No*

RA decrypts the message using K_X and compares the output with the hash digest of R_2. If both match, MHA is authenticated to RA, which completes the mutual authentication procedure. Now RA sends an acknowledgement message to MHA and forwards the message *M* with ECDSA signature to CA to generate the ECC public key certificate of the mobile phone.

3.4 Certificate Issuance

After receiving the verified and attested certificate request message *M* from RA, CA verifies for confirmation and then generates the ECC-based public key certificate of mobile phone, signs it using ECDSA, publishes the certificate in its directory and finally, sends the *certificate URL* to MHA.

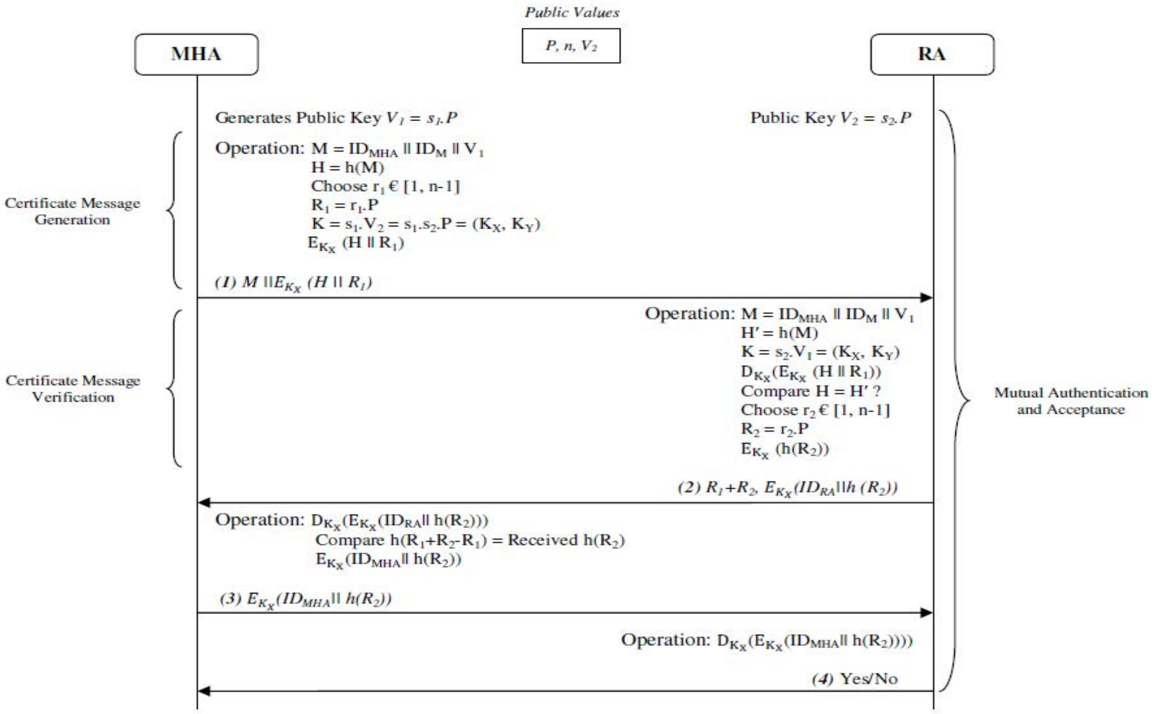
Public Values
P, n, V₂
MHA
RA
Generates Public Key V₁ = s₁.P
Public Key V₂ = s₂.P
Certificate Message Generation
Operation: M = ID_MHA || ID_M || V₁
H = h(M)
Choose r₁ € [1, n-1]
R₁ = r₁.P
K = s₁.V₂ = s₁.s₂.P = (K_X, K_Y)
E_Kx (H || R₁)
(1) M ||E_Kx (H || R₁)
Certificate Message Verification
Operation: M = ID_MHA || ID_M || V₁
H' = h(M)
K = s₂.V₁ = (K_X, K_Y)
D_Kx(E_Kx (H || R₁))
Compare H = H' ?
Choose r₂ € [1, n-1]
R₂ = r₂.P
E_Kx (h(R₂))
Mutual Authentication and Acceptance
(2) R₁+R₂, E_Kx(ID_RA||h (R₂))
Operation: D_Kx(E_Kx(ID_RA|| h(R₂)))
Compare h(R₁+R₂-R₁) = Received h(R₂)
E_Kx(ID_MHA|| h(R₂))
(3) E_Kx(ID_MHA|| h(R₂))
Operation: D_Kx(E_Kx(ID_MHA|| h(R₂))))
(4) Yes/No

Figure 3. Certificate request message generation and verification protocol

3.5 Certificate Life Cycle

The mobile-PKI supports not only the initial issue of mobile phone's certificate but also includes update of key pair, certificate, certificate suspension and revocation. For these, a common message with a *request for type-update* is used to define the type of the certificate request made to CA. The different message formats for certificate life cycle are given in table 1.

Table 1. Message formats of certificate life cycle.

Life cycle	Message *M* with update-type
Initial issue	M = initial issue || ID_{MHA} || ID_M || V_1
Key pair update	M = key pair update || ID_{MHA} || ID_M || $V_{1\ New}$
Certificate update	M = certificate update || certificate serial No. || ID_{MHA} || ID_M || V_1
Certificate suspension	M = certificate suspension || certificate serial No. || ID_{MHA} || ID_M || V_1
Certificate revocation	M = certificate revocation || certificate serial No. || ID_{MHA} || ID_M || V_1

3.6 Certificate Validation and Applications

After receiving the *certificate URL* and private key from MHA, the mobile phone user utilizes the certificate URL for different mobile applications and generic workflow diagram is shown in figure 4. Initially, a mobile phone user sends a *transaction request type* to MHA, which then identifies the server-URL on searching and sends the certificate URL of the mobile-phone user to the server for making an authenticated and secure channel for transactions to be made between them. However, before any transaction, the application server retrieves the certificate of mobile phone from the CA directory using the certificate URL and acquires the CRL for validation of the certificate. Similarly, the server sends its certificate URL and CA's certificate together to MHA for the validation of the server. After completion of the both side validation procedure, MHA informs mobile phone user to connect the server and starts the required transactions.

4. SECURITY ANALYSIS OF PROPOSED MOBILE-PKI

In our proposed ECC-based Mobile-PKI, initially, MHA should be authenticated to RA to ensure that the mobile user possess the valid private key of corresponding public key. And once the certificate is issued by CA through RA and used by the mobile phone user, the mutual authentication of both server and mobile user must also be established. Thus the proposed ECC-based PKI involves the following cryptographic operations, where shown briefly that all are well protected.

Moreover, the proposed ECC-based scheme provides low computation and communication cost as well as less key-size to provide same level of security as of RSA, and thus it is efficient as explained below.

4.1 Mutual Authentication

After receiving the certificate request message, RA verifies the message, completes the mutual authentication and then forwards the message to CA with attestation. The detail procedure of mutual authentication is discussed in step 1 to 4 of section 3.3 which assure that before issuing a certificate, the RA must authenticates the MHA/user.

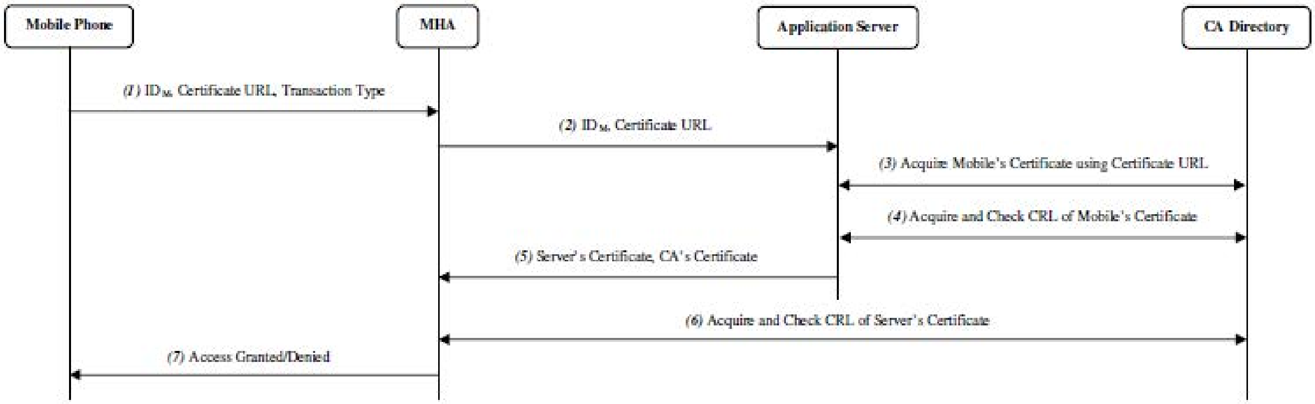
Mobile Phone
MHA
Application Server
CA Directory
(1) ID_M, Certificate URL, Transaction Type
(2) ID_M, Certificate URL
(3) Acquire Mobile's Certificate using Certificate URL
(4) Acquire and Check CRL of Mobile's Certificate
(5) Server's Certificate, CA's Certificate
(6) Acquire and Check CRL of Server's Certificate
(7) Access Granted/Denied

Figure 4. Proposed certificate validation and application procedure in mobile-PKI

4.2 POP Verification

The POP verification procedure is considered in our scheme which ensures that the mobile user possess the private key of corresponding public key. The verification procedure is done in our scheme by calculating *K* and comparing hash digests at step 2 of *section 3.3*.

4.3 Confidentiality

MHA sends the mobile phone's private key encrypted using the pre-shared secret key *k*. Mobile phone gets his private key after decrypting it using the pre-shared secret key *k*. So, the private key of the mobile phone is securely transmitted to mobile phone from MHA as explained in step 5 of *section 3.1*.

4.4 Replay Attack Prevention

To prevent replay attack in our scheme, MHA sends the public key to RA only once and each mobile phone has different public key. If the same public key is applied twice, RA should realize the possibility of replay attack and as a result, terminates the process.

4.5 Efficiency

The proposed ECC-based Mobile-PKI is more efficient than the existing RSA-based schemes due to the following reasons:

(1) **Provides comparable security with small key-length:** In general, it is seen that 160-bit key in ECC is equivalent in security with 1024-bit key in RSA [19, 20, 21]. This is because the existing RSA based mobile-PKI uses Diffie–Hellman key exchange protocol [22] in which the public challenges generated with key-size is at least 1024 bits, otherwise it is assumed that RSA is compromisable. On the other hand, in ECC, the public challenges are of 160 bits key length, which is not easily compromisable due to the unique properties of ECC.

(2) **Requires less computation cost:** Since the main computation carried out in ECC is the scalar point multiplication, thus it requires much lesser computation cost than RSA, which uses the most costly modular exponentiation operation. In addition, ECC uses all 160-bit operation, but RSA requires 1024-bit manipulation for comparable security [19, 20, 21]. Also the proposed scheme uses cryptographic hash function, elliptic curve multiplication/addition and symmetric encryption, which further reduces the processing time over the RSA based scheme that follows public key encryption technique (as it is known that the symmetric approach is faster in processing than the public-key one). Therefore, the proposed ECC based mobile-PKI requires less computation cost than the existing RSA based mobile-PKI.

(3) **Requires less communication cost:** Due to the use of less key-size in ECC, each message-size in the proposed scheme is reduced and also due to use of MHA, the total number of messages used with respect to the mobile phone is reduced to a minimum as possible. Thus the proposed scheme is communication efficient.

5. COMPARISON

Finally, the proposed scheme is compared with other two existing schemes and the outputs are shown in table 2, which shows that the proposed scheme is only ECC based and achieves all six parameters considered.

Table 2. **Comparison of proposed scheme with existing schemes**

Parameters	WAP [5,8]	Y. Lee et al.[15,16]	Proposed Protocol
Cryptosystem	RSA	RSA	**ECC**
End-to-end security	No	Yes	**Yes**
Confidentiality	No	Yes	**Yes**
Absence of out-of-band user identification	Yes	No	**Yes**
POP verification	No	Yes	**Yes**
Considers the fundamental limitations of mobile phone	No	No	**Yes**
Supports less computation and communication overhead	No	No	**Yes**

6. CONCLUSION

An ECC-based Mobile-PKI is introduced in this paper that quite fits for small screen, low computation, and small storage devices like mobile phones. In our implementation, a MHA similar to the home-agent in mobile-IP is used to carry out most of the processing cost required for mobile phone. On the other hand, a RA is used to take the burden of CA before issuing a digital public-key certificate. Instead of RSA, we also use ECC-based private-public-key pair for generating, issuing and using digital certificate for reducing key-size, computation and communication costs appreciably and practically useful for mobile devices. In our design, we consider a URL for the certificate that further enhances the use and verification of certificate through directory and solves the requiring of memory space used to store the certificate. Finally, the security analysis, efficiency and comparison of the proposed schemes have been done and it has been found that our proposed Mobile-PKI overcomes the limitations for mobile phones and suitable for practical applications.

REFERENCES

[1] Critchlow, D. and Zhang, N. 2004. Security enhanced accountable anonymous PKI certificates for mobile e-commerce. *Computer Networks*. 45 (2004), 483–503.

[2] Schwingenschlogl, C., Eichler, S. and Muller-Rathgeber, B. 2006. Performance of PKI-based security mechanisms in mobile ad hoc networks. *International Journal of Electronics and Communications*. 60 (2006), 20–24.

[3] Lam, K. Y., Chung, S. L., Gu, M. and Sun, J. G. 2003. Lightweight security for mobile commerce transactions. *Computer Communications. Elsevier*. 26 (2003), 2052-2060.

[4] Lee, J., Lee Y. and Song, J. 2001. Wireless PKI Technology in Korea. In *Proc. of the First International Workshop for Asian PKI*. 1 (Korea, 2001), 145-158.

[5] OMA. 2001. *Wireless Application Protocol - Wireless Public Key Infrastructure*. WAP-217-WPKI (Apr. 2001).

[6] OMA. 2001. *Wireless Application Protocol Architecture Specification*. WAP-210-WAPArch (Jul. 2001).

[7] OMA. 2001. *Wireless Transport Layer Security*. WAP-261-WTLS (Apr. 2001).

[8] OMA. 2001. *Wireless Application Protocol WAP2.0 Technical White Paper*. (Apr. 2001).

[9] OMA. 2000. *WAP Certificate and CRL*. WAP-211-X.509 (Mar. 2000).

[10] RSA Laboratories. 2000. *PKCS#10: Certification Request Syntax Standard*. (2000).

[11] Myers, M., Ankney, R., Malpani, A., Galperin, S. and Adams, C. 1999. *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP: IETF RFC2560*. IETF Network Working Group (June, 1999).

[12] Frier, A., Karlton, P. and Kocher, P. 1996. *The SSL 3.0 Protocol. Netscape Communications Corp*. (Nov. 1996).

[13] Admas, C., Farrell, S., Kause, T. and Mononen, T. 2005. *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP): IETF RFC 2510*. IETF Network working Group (Sep, 2005).

[14] Myers, M., Adams, C., Solo, D. and Kemp, D. 1999. *Internet X.509 Certificate Request Message Format: IETF RFC2511*. IETF Network Working Group (March, 1999).

[15] Lee, Y., Lee, J. and Lee, G. Y. 2008. Wireless Certificate Management Protocol Supporting Mobile Phones. In *Proc. of IEEE Congress on Services – Part 1*(2008), 353-359.

[16] Lee, Y., Lee, J. and Song, J. S. 2007. Design and implementation of wireless PKI technology suitable for mobile phone in mobile-commerce. *Computer Communications. Elsevier*. 30 (2007), 893-903.

[17] Weise, J. 2001. *Public Key Infrastructure Overview*. Sun PSSM Global Security Practice. Sun Blue Prints™. (2001).

[18] Stallings, W. 2009. *Cryptography and Network Security: Principles and Practices*. Prentice Hall. 4th Edition, 420-430 (2009).

[19] Hankerson, D., Menezes, A. and Vanstone, S. 2004. *Guide to elliptic curve cryptography*. Springer-Verlag, New York, USA (2004).

[20] Koblitz, N. 1987. Elliptic Curve Cryptosystem. *Journal of mathematics computation*. 48, 177 (Janaury, 1987), 203-2009.

[21] Miller, V. 1985. Use of elliptic curves in cryptography. In *Proc. of Advances in Cryptology-CRYPTO*, 85, LNCS 218 (1985), 417–426.

[22] Diffe, W. and Hellman, M. 1976. New directions in cryptology. *IEEE Transaction on Information Theory*. 22 (1976), 644–654.

[23] Dang, Q., Santesson, S., Moriarty, K., Brown, D. and Polk, T. 2010. *Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA*. RFC 5758 (January, 2010).

[24] Islam, S. H. and Biswas, G. P. 2011. Design of improved password authentication and update scheme based on elliptic curve cryptography. *Mathematical and Computer Modeling* (2011), doi:10.1016/j.mcm.2011.07.001, in press.

[25] Ray S. and Biswas, G. P. 2011. Design of a Mobile-PKI for using mobile phones in various applications. In *Proc. of 1st International Conference on Recent Trends in Information Systems (ReTIS-11)*, (Jadavpur University, Kolkata, 2011), IEEE Xplore. 297-302, 2011, DOI: 10.1109/ReTIS.2011.6146885.

[26] 2001. *Introduction to Public Key Technology and the Federal PKI Infrastructure*. National Institute of Standards and Technology (26th Feb, 2001).

[27] Polk, W., Housley, R. and Bassham, L. 2002. *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 3279 (April 2002).

[28] Schaad, J., Kaliski, B. and Housley, R. 2005. *Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. RFC 4055.