Application of Elliptic Curve Cryptography for Broadcasting in Mobile Devices

Solomon SARPONG
Department of Physical and Mathematical Sciences
University of Environment and Sustainable Development, Somanya, Ghana

Abstract- As Internet availability is becoming increasingly ubiquitous and data sharing has become more pervasive, the security and authenticity of data transmitted has become more important. In scenarios like pay-per-view television, distance learning etc. where a subset of audience needs to be reached, broadcast encryption is used. Different algorithms have been added to the broadcast encryption to enhance the security of the data during transmission. Most of these algorithms have high computational cost and communication complexities making them impractical to use on mobile devices. This paper proposes an enhanced security in broadcast encryption based on elliptic curve cryptography. The smallness of the keys and low computational costs of the elliptic curve cryptography makes it ideal for use on mobile devices.

Keywords – communication complexities, computational cost, elliptic curve cryptography, broadcast, mobile devices.

I. INTRODUCTION

The advances in technology have enabled the provision of services to *User*s on social media, Internet, television etc. wherever they are and anytime they want it. Most often than not, these services maybe free or paid. In the later, only persons that have subscribed/paid can access these services. How can a service provider who is transmitting the content of its service be able to reach only the paid-up customers? The need for targeted accessibility to content of services brought to the fore broadcast encryption. The idea of broadcast encryption was proposed by [1]. Since this innovative idea came to light, there have been a lot of applications and modifications to it [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12]. These modifications to a large extent is to make broadcast encryption suitable for particular purpose(s).

A cryptographic scheme such as broadcast encryption enables encryption of some content such that it is only the selected set of targeted *User*s who can decrypt the content. Hence, broadcast encryption scheme provides confidentiality of broadcast messages such that only the privileged *User(s)* can decrypt the encrypted message received.

There are basically two types of broadcast encryption scheme – static or dynamic. In static broadcast encryption schemes, *User*s are fixed – no one joins or leaves when the protocol begins. Whereas in dynamic broadcast schemes, *User*s can leave or join the protocol after it has begun. Even though, there are changes in parameters in the dynamic encryption it is more flexible hence appropriate in practical applications.

Computers have become ubiquitous or pervasive as it has moved beyond personal computers to devices that are used daily. These devices have technology and connectivity embedded in them in such a way that makes the connectivity unobtrusive and always available. These features make them easier for use on social media applications. The main limitations of a mobile device are computational ability and battery life. These two are interlinked – the more computation undertaken, the more battery power is needed. These limitations bring to the fore the usefulness of elliptic curve cryptography (ECC) in mobile devices. As a result of the low-memory and low-computing environment in mobile and wireless devices, ECC finds its applicability.

This paper proposes the application of ECC in broadcast encryption scheme for use in mobile devices. As mobile devices have low memory and low computational capacity, the use of ECC enhances the security features of the protocols and at the same time lower the computational intensity.

The rest of the paper is partitioned as following: the introduction takes a brief look at broadcast encryption and elliptic curve cryptography; the related works in this area of research is reviewed; the next sections consist the proposed protocol, the security of the protocol. The paper is concluded in the next section.

II. RELATED WORKS

[13] proposed an enhanced authentication model, suitable for low-power mobile devices using extended password key exchange protocols and elliptic-curve-cryptosystem based trust delegation mechanism which is effective against denial-of-service attacks. [14] used ECC to secure communication between base stations and nodes hence denying eavesdroppers any information.

The use of multi-curve elliptic curve cryptography (MECC) to increase security in Smart cards has been proposed by [15]. In pervasive computing environment, elliptic curve cryptography (ECC) has been found to be suitable as a result of the limited bandwidth, battery power, less computational resources and less memory of these devices [16]. [17] proposed the implementation of ECC over the finite field GF(p) for transmitting audio files over 3G networks.

III. TOPOLOGY OF BROADCAST ENCRYPTION

Broadcast encryption scheme is needed when sending exactly the same message to a group of *User*s simultaneously. This is a necessity in some applications like distance learning, television (pay-per-view, etc.), grid computing etc. where only a selected number of persons (*User*s) are the targeted audience at a particular period. For example, in the University settings – there maybe students, lecturers and other auxiliary staff – where management wants to send messages to these different categories of University staff, broadcast encryption can be used.

In broadcast encryption, there is the universal set of *User*s, U; legitimate members (members in good standing), G and revoked members, R. Hence, for n *User*s $U = \{u_i\}_{i=1}^{n}$ and also, $U = G \cup R$. A broadcast encryption scheme consists of four polynomial time algorithms (Setup, Registration, Encryption, Decryption).

Setup – the setup takes as an input a security parameter α and outputs a public key P_k and a secret key S_k. **Registration** – a secret private key is generated for each member. The input is the *User*'s *ID* and outputs a private key $S_{k_{ID}}$. **Encryption** – during encryption, there should be a set of legitimate *User*s G and a public key P_k. The output is a pair (H_{dr}, G_k); where G_k is an encryption key chosen from a finite key space k and H_{dr} is a set of information which is used to retrieve the key G_k by members. In order to send a message M, it is encrypted with a key G_k. Thus $C = Enc_{G_k}(M)$. The message (H_{dr}, C); are broadcast, where H_{dr} is the broadcast header and C is the broadcast body. **Decryption** – to decrypt C, the *User(s)* G (the set of legitimate *User*s) use the key G_k from H_{dr} to decrypt C. Thus $M = Dec_{G_k}(C)$.

3.1 Basic Requirements

The basic requirements needed for a broadcast encryption include; dependability, statelessness, correctness, exclusiveness, collusion resistance and efficiency. **Dependability**: Without the knowledge of $\alpha = \left\{G_i, E_{k_{G_i}}(s_k)\right\}_{G_i \in U}$ from the *Server*, no *User* should be able to infer the current session key, s_k. **Statelessness**: The current session key s_k should not be derived only from the current rekey message and private/public storage. Hence, it must be independent of past rekey message, α. **Correctness**: Any allowed *User(s)* from G should be able to get from rekey message α exactly the same session key s_k that was chosen by the *Server*. **Exclusiveness**: Non-registered or revoked members should not be able to infer current session key s_k from broadcast rekey message, α. **Collusion Resistance**: *User*s who have been revoked must not be able to infer current session key s_k when they cooperate. **Efficiency**: There should be communication, storage and computational efficiency.

As a result of possible malicious attacks on the broadcast encryption protocol, there should be authentication of messages. Traditionally, authentication can be achieved when the sender uses a digital algorithm to sign the message. Some broadcast encryption schemes authenticate their messages by signing [18], [19], [20], and [21]. In this paper, in order to reduce computational costs, digital signature algorithms will not be used for authentication of messages. Elliptic curve Diffie-Hellman protocol will be used to establish secure communication channel between the *Server* and *User(s)*.

IV. ELLIPTIC CURVE

In 1985, Neal Koblitz and Victor Miller independently proposed ECC based on the elliptic curve [22] and [23]. An Elliptic Curve $E = \{(x, y) | y^2 = x^3 + ax + b\}$ is defined over a finite field F_p, where P is a large prime; $a, b \in F_p$; a and b satisfy $4a^3 + 27b^2 \neq 0$. The *Server* publishes Q, G, t, a, b. where; Q, P – the chosen field (mod p); also, P and $Q \in E(Z/Z_p)$; G – the generator point; t – order of G (the size of the subgroups).

The possible number of points on an elliptic curve is the anchor of its security. The number of points on the elliptic curve can be computed using the Hasse's theorem [20]. By the Hasse's theorem, the number of points on an

elliptic curve E over the finite field F_P satisfies the inequality $|E-(P+1)| \leq 2\sqrt{P}$. Hence, the number of points on the elliptic curve E is in the range $(P+1)-2\sqrt{P} \leq E \leq (P+1)+2\sqrt{P}$. If the elliptic curve is defined over any finite field F_Q, where $Q = P^t$, Hasse's theorem becomes $|E-(Q+1)| \leq 2\sqrt{Q}$. Hence, the number of points on that elliptic curve is given by $(P^t+1)-2\sqrt{P^t} \leq E \leq (P^t+1)+2\sqrt{P^t}$.

Table 1: Prime number and the number of points on the elliptic curve

Prime Number (P)	Minimum Number of Points	Maximum Number of Points
809	753	866
3067	753	866
3917	3792	4034
4993	4852	5135
5077	4936	5220
5987	5833	6142
11587	11372	11803
25253	24936	25571
79999	79434	80565
89989	89390	90589
119993	119301	120686

From Table 1 it can be observed that with moderately large prime number, a lot of points on the elliptic curve can be generated. This is a security feature of the elliptic curve that makes it very appropriate for this protocol.

The advantage of the Elliptic curve over other protocols is that, it has shorter encryption key hence, uses small memory and CPU resources. The strength of the elliptic curve is that, it is impossible to compute K such that, $Q = KP$. Another strength of the ECC lies in the random number. The random numbers (r) should be generated such that the secret number d should be secured. Assuming there is an elliptic curve with points P and Q having an initial point S_0. The random numbers can be; $S_0 \xrightarrow{S_0P} S_1 \xrightarrow{S_1P} S_2 \xrightarrow{S_2P} S_3 \xrightarrow{S_3P}$. Multiplying each $S_1, S_2, S_3, \ldots$ by Q gives $S_1Q, S_2Q, S_3Q, \ldots$. Let $r_1 = S_1Q$, $r_2 = S_2Q$, $r_3 = S_3Q, \ldots$ However, if the secret number d is known by the adversary the random numbers can be acquired by computing $S_1P = S_1(dQ) = d(S_1Q) = dr_1$. On the other hand, the random number can also be generated from true random number generator (TRNG). On a mobile device, an entropy for generating TRNG can be the accelerometer, compass or the touch patterns on the keypad by the *User*.

The use of the Internet and mobile devices are on the rise, data transfer with less computational cost and more security is becoming the foremost priority of cryptographers hence, the use of ECC is on the rise. The small keys and security of the algorithm in the ECC makes it suitable for use in devices with small computational power e.g., mobile devices.

ECC has become the encryption standard for devices that have limited bandwidth, battery power, less computational resources and less memory. As compared to other PKC, ECC has achieved this remarkable status as it offers comparably high security with smaller key sizes, faster computation, lower power consumption, as well as memory and bandwidth savings.

V. ELLIPTIC CURVE DIFFIE-HELLMAN KEY EXCHANGE (ECDH)

In order to create a secure communication channel between two parties, elliptic curve Diffie-Hellman key exchange protocol can used. Assume there is an agreed upon elliptic curve, E and a base point p by the parties (Alice and Bob). Alice chooses a secret large random number, A_R; Bob also chooses a secret large random number, B_R. Alice computes pA_R and sends to Bob. Bob also computes pB_R and sends to Alice. Both Alice and Bob compute the secret key $pA_R \cdot pB_R$ and $pB_R \cdot pA_R$ respectively. It can be observed that, $pA_R \cdot pB_R = pB_R \cdot pA_R$. This the symmetric key generated by Alice and Bob for communication.

5.1 Proposed Protocol

The *Server* has publicly known elliptic curve. In order to form the private and public keys for the protocol, the *Server* and *User(s)* execute the elliptic curve Diffie-Hellman key exchange protocol as follows:

Step 1: Choice of Private Key. The *Server* chooses a random number β (a large prime) such that, $1 \leq \beta \leq t-1$ and computes $B = \beta G$.

Step 2: The *Server* sends B to the *User(s)*.

Step 3: Computation of Private key(s).

Each *User* chooses a random number α_i (a large prime) such that $1 \leq \alpha_i \leq t-1$ and $i = 1, \ldots, n$.

Step 4: Computation of Public key(s)

Each *User* computes and sends $U_i = \alpha_i G$ to the *Server*.

The *Server* chooses a point $S_p = (X_s, Y_s)$ on the curve and sends to each *User*.

Each *User* chooses a point $U_{p_i} = (X_{U_i}, Y_{U_i})$ on the curve and sends to the *Server*.

Both the *Server* and each *User* computes the public key, $\beta G \alpha_i$.

The private and public key pairs of the *Server* and *User(s)* are $(\beta, \beta G \alpha_i)$ and $(\alpha_i, \beta G \alpha_i)$ respectively. The *Server* has an identity, ID_{Server} and a collision resistant one-way hash function. The *Server* also hashes its ID_{Server} to obtain $H(ID_{Server})$. Each *User* chooses an identity, ID_i computes $User_{ID_i} = [U_i || \{ID_i\}^{\alpha_i}]$ and sends to the *Server* using the public key from the ECDH protocol. The *Server* computes the hash of $User_{ID_i}$, encrypts it with the public keys of the *User(s)* from the symmetric key encryption algorithm and ECDH protocol and sends to the *User(s)*. Hence, *Server* sends $E_{User_{i_{pub}}}\{E_{\beta G \alpha_i}\{User_{ID_i} || H(User_{ID_i}) || H(ID_{Server})\}\}$ to the *User(s)*. A *User* is therefore recognized by the *Server* with $H(User_{ID_i})$.

5.2 Encryption

To broadcast a message, the *Server* uses the public key from the ECDH protocol to encrypt the message, *M*. The *Server* further uses the public key of the *User*(s') symmetric key to encrypt the encrypted message to obtain $Enc_{User_{i_{pub}}}\{Enc_{\beta G \alpha_i}(M || H(ID_{Server}))\}$. The *Server* broadcasts the encrypted messages concatenated with the hash of its *ID*. The *Server* broadcasts $\left[H(ID_{Server}) || Enc_{User_{i_{pub}}}\{Enc_{\beta G \alpha_i}(M || H(ID_{Server}))\}\right]$.

5.3 Decryption

The *User(s)* receive $\left[H(ID_{Server}) || Enc_{User_{i_{pub}}}\{Enc_{\beta G \alpha_i}(M || H(ID_{Server}))\}\right]$ from the *Server*. The *User(s)* first use the private key of the symmetric encryption to decrypt the message received. The *User(s)* further uses his/her private key of the ECDH protocol to decrypt the rest of the message. The *User(s)* do the following decryption $Dec_{User_{i_{pri}}}\left[Enc_{User_{i_{pub}}}\{Enc_{\beta G \alpha_i}(M || H(ID_{Server}))\}\right] \rightarrow Enc_{\beta G \alpha_i}(M || H(ID_{Server}))$. The *User(s)* decrypt the message further $Dec_{\alpha_i}[Enc_{\beta G \alpha_i}(M || H(ID_{Server}))] \rightarrow M || ID_{Server}$. The retrieval of $M || ID_{Server}$ helps the *User(s)* to know the message being broadcast and confirm it is coming from the *Server*.

5.4 *User(s)*

User(s) cannot directly send message to each other. When a legitimate $User_p$, wants to send message to the other *User(s)*, the message should be sent to the *Server* first. The *Server* then broadcasts the message to other *User(s)*. In order to send message to the *Server* to broadcast, the $User_p$ encrypts the message using the public key of the symmetric keypair of the *Server*. The $User_p$ concatenates the encrypted message and the hash his/her ID and encrypts with the private key from the ECDH protocol. The $User_p$ sends to the *Server*, $Enc_{Server_{pub}}\left\{Enc_{\alpha_{p\beta G}}\left(M || User_{ID_p} || H\left(User_{ID_p}\right)\right)\right\}$. The *Server* decrypts the message received $Enc_{Server_{pub}}\left\{Enc_{\alpha_{p\beta G}}\left(M || User_{ID_p} || H\left(User_{ID_p}\right)\right)\right\}$ using the private key of the symmetric encryption. The *Server* further uses the private key of the ECDH protocol to decrypt the rest of the file received. That is the *Server* does the following;

$Dec_{Server_{pri}} Dec_{Server_{pri}} \left[Enc_{Server_{pub}} \left\{ Enc_{\alpha_{p\beta G}} \left(M||User_{ID_p}||H\left(User_{ID_p}\right)\right)\right\}\right] \rightarrow$
$Enc_{\alpha_{p\beta G}} \left(M||User_{ID_p}||H\left(User_{ID_p}\right)\right)$

The *Server* does further decryption $Dec_{\beta}\left[Enc_{\alpha_{p\beta G}}\left(M||User_{ID_p}||H\left(User_{ID_p}\right)\right)\right] \rightarrow M||User_{ID_p}||H\left(User_{ID_p}\right)$. The retrieval of $M||User_{ID_p}||H\left(User_{ID_p}\right)$ enables the *Server* know the message to broadcast, the identity of the *User* that sent it and the hash of the *User*'s ID. The *Server* hashes the $User_{ID_p}$ received and compares with the $H\left(User_{ID_p}\right)$ in its possession. If they are the same, the *Server* knows the message has not been tempered with and the identity of the *User (a legitimate User)* who sent it. The *Server* can then broadcast the message received from the *User(s)*.

VI. SECURITY

In order to achieve high security and low computational cost, the elliptic curve cryptography was used. The elliptic curve used should be secure hence, it is ensured that the elliptic curve is neither supersingular nor anomalous curve as these types of elliptic curves are susceptible to MOV and other known attacks. Elliptic curve cryptography uses shorter encryption keys hence, needs less CPU memory.

*User*s do not broadcast messages directly but routes them through the *Server* to broadcast. This is to prevent honest but curious attacks from malicious *User*s. A malicious *User* can pose as the *Server* to tricking other *User*s into disclosing their private keys. When there are revoked members, *r*, the *Server* and the non-revoked *User(s)* need not undertake the ECDH protocol again. This also reduces the computational cost of this protocol. Only the *Server* re-computes and shares a new public key with the $(n-r)$ *User(s)*. Hence, reducing the computational cost making it appropriate to operate on mobile devices.

The inclusion of $H(ID_{Server})$ in the message $\left[H(ID_{Server})||Enc_{User_{i_{pub}}}\left\{Enc_{\beta G\alpha_i}\left(M||H(ID_{Server})\right)\right\}\right]$ broadcast by the *Server* is a security check. Even though it is not possible, but when *User(s)* receive message that does not have the $H(ID_{Server})$ it indicates it is not coming from the *Server* hence, the message can be discarded. A *User* is therefore recognized by the *Server* by $H\left(User_{ID_i}\right)$. Also, the inclusion of $User_{ID_p}||H\left(User_{ID_p}\right)$ in the message $Enc_{Server_{pub}}\left\{Enc_{\alpha_{p\beta G}}\left(M||User_{ID_p}||H\left(User_{ID_p}\right)\right)\right\}$ sent by a *User* to the *Server* helps the *Server* know the *User* that sent it. After decryption, the *Server* can check if the $H\left(User_{ID_p}\right)$ included in the message is the same as what it computed at the beginning of the protocol. The *Server* processes the message if $H\left(User_{ID_p}\right)$ received is the same as $H\left(User_{ID_p}\right)$ computed but it is discarded if they are not.

This is to help the *Server* identify the *User* who sent the message. This also prevents persons who are not part of the protocol (malicious persons or revoked members) from taking part. It also prevents impersonation attacks. The hash of the *User* ID, $H\left(User_{ID_i}\right)$ serves as a security token that the *Server* uses to identify a *User*. This helps the *Server* to prevent impersonation attacks (both from adversaries and revoked members).

6.1 Revoked Members

When members in a broadcast scheme are no longer part of the scheme, they are removed. Hence, the revoked members *(r)* can no longer take part in the protocol. When some of the members are revoked, the *Server* broadcasts information to only the $i^* = (n-r)$ *User*s. The revoked members cannot take part in any of the protocols as they do not have the necessary information to enable them to.

6.2 Impersonation Attack

An adversary cannot take part in this protocol. Also, a semi-honest *User* cannot compromise any member of the protocol. For a *User* to broadcast a message, *M*, the message should first be sent to the *Server*. In order to send message to the *Server*, the *User*, *p*, must send $Enc_{Server_{pub}}\left\{Enc_{\alpha_{p\beta G}}\left(M||User_{ID_p}||H\left(User_{ID_p}\right)\right)\right\}$ to the *Server*. On receiving the message, the *Server* will know the *ID* is from a revoked member and the message will be discarded. If the adversary guesses the *Server*'s public key or uses an old public key, the *Server* cannot decrypt the

information received. The inability of the *Server* to decrypt the information received implies the message cannot be extracted for broadcast. When an adversary intercepts the message $Enc_{User_{i_{pub}}}\left\{Enc_{\beta G\alpha_i}\left(M||User_{ID_i}||H(ID_{Server})\right)\right\}$ the adversary will not be able to know the message *M* being sent. To decrypt $\mathrm{Enc}_{User_{i_{pub}}}\left\{\mathrm{Enc}_{\beta G\alpha_i}\left(\mathrm{M}||User_{\mathrm{ID_i}}||\mathrm{H}(\mathrm{ID}_{Server})\right)\right\}$ and know the message, the adversary should know the private key $User_{i_{pri}}$ of the symmetric encryption of the *User* and the private key of the *User* from the ECDH protocol. The adversary will not know the private keys of the *User(s)* from of the symmetric encryption; the $User_{i_{pri}}$ and α_i from the ECDH protocol. The lack of knowledge of this information will prevent the adversary from knowing *M*. Hence, impersonation attacks is not possible in this protocol.

VII. CONCLUSION

As opposed to desktop computers and other non-movable devices, the acquisition and use of mobile devices is increasing. Mobile devices like smartphones have Internet and other features hence making it more appealing for use. This has necessitated the need for light weight protocols that can be used on them. This paper has proposed a broadcast encryption scheme based on the elliptic curve cryptography. As mobile devices have low memory and low computational capability, the use of elliptic curve cryptography is very appropriate. Hence, the protocol in this paper will be appropriate for use on mobile devices.

REFERENCES

[1] A. Fiat and M. Naor, "Broadcast Encryption," in *Advances in Cryptology - CRYPTO '93, 13th Annual International Cryptology Conference, Santa Barbara, California, USA, August 22-26, 1993, Proceedings*, in Lecture Notes in Computer Science, vol. 773. Springer, 1993, pp. 480–491. doi: 10.1007/3-540-48329-2_40.

[2] D. Halevy and A. Shamir, "The LSD broadcast encryption scheme," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2002, pp. 47–60. doi: 10.1007/3-540-45708-9_4.

[3] F. Li, X. Xin, and Y. Hu, "Indentity-based broadcast signcryption," *Computer Standards and Interfaces*, vol. 30, no. 1–2, pp. 89–94, Jan. 2008, doi: 10.1016/j.csi.2007.08.005.

[4] Yi MuWilly SusiloYan-Xia LinChun Ruan, *Identity-Based Authenticated Broadcast Encryption and Distributed Authenticated Encryption.* Springer, Berlin, Heidelberg, 2004. doi: doi.org/10.1007/978-3-540-30502-6_12.

[5] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3621 LNCS, pp. 258–275, 2006, doi: 10.1007/11535218_16.

[6] C. Delerablée, "Identity-Based Broadcast Encryption with Constant Size Ciphertexts and Private Keys," in *Advances in Cryptology -- ASIACRYPT 2007*, K. Kurosawa, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 200–215.

[7] Y. Dodis and N. Fazio, "Public Key Broadcast Encryption for Stateless Receivers," in *Digital Rights Management*, J. Feigenbaum, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 61–80.

[8] D. Phan, D. Pointcheval, and M. Strefler, "Adaptively Secure Broadcast Encryption with Forward Secrecy.," *IACR Cryptology ePrint Archive*, vol. 2011, p. 463, 2011.

[9] D. H. Phan, D. Pointcheval, and M. Strefler, "Security Notions for Broadcast Encryption," in *Applied Cryptography and Network Security*, J. Lopez and G. Tsudik, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 377–394.

[10] D.-H. Phan, D. Pointcheval, S. F. Shahandashti, and M. Strefler, "Adaptive CCA broadcast encryption with constant-size secret keys and ciphertexts," *International Journal of Information Security*, vol. 12, no. 4, pp. 251–265, 2013, doi: 10.1007/s10207-013-0190-0.

[11] C. C. Chang, Y. W. Lat, and J. H. Yang, "An efficient authenticated encryption scheme based on elliptic curvecryptosystem for broadcast environment," *ICIC Express Letters*, vol. 4, no. 1, pp. 95–99, Feb. 2010.

[12] R. Kumar and A. Anil, "Implementation of Elliptical Curve Cryptography," *International Journal of Computer Science Issues*, vol. 8, 2011.

[13] M. Prabhakar, "Elliptic Curve Cryptography in Securing Networks by Mobile Authentication," *International Journal on Cryptography and Information Security*, vol. 3, pp. 31–46, 2013, doi: 10.5121/ijcis.2013.3304.

[14] P. Rajeswari and K. Thilagavathi, "An Efficient Authentication Protocol Based on Elliptic Curve Cryptography for Mobile Networks," 2009.

[15] P. Gopalakrishnan and s Sakthivel, "Improving the Security of Smart Cards through Multi- Curve ECC," *International Journal of Applied Engineering Research*, vol. 9, pp. 17601–17611, 2014.

[16] V. Katiyar, K. Dutta, and S. Gupta, "A Survey on Elliptic Curve Cryptography for Pervasive Computing Environment," *International Journal of Computer Applications*, vol. 11, pp. 24–28, 2010.

[17] R. Singh, R. Chauhan, V. K. Gunjan, and P. Singh, "Implementation of Elliptic Curve Cryptography for Audio Based Application," *International Journal of Engineering*, vol. 3, no. 1. 2014.

[18] T. ElGamal, "A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms," in *Advances in Cryptology*, G. R. Blakley and D. Chaum, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 10–18.

[19] Kuo-Feng Hwang and Chin-Chen Chang, "A self-encryption mechanism for authentication of roaming and teleconference services," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 400–407, Mar. 2003, doi: 10.1109/TWC.2003.809452.

[20] R. J. Hwang, C. H. Lai, and F. F. Su, "An efficient signcryption scheme with forward secrecy based on elliptic curve," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 870–881, Aug. 2005, doi: 10.1016/j.amc.2004.06.124.
[21] Y. Mu, W. Susilo, Y.-X. Lin, and C. Ruan, "Identity-Based Authenticated Broadcast Encryption and Distributed Authenticated Encryption," in *Advances in Computer Science - ASIAN 2004. Higher-Level Decision Making*, M. J. Maher, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 169–181.
[22] N. Koblitz, "Elliptic Curve Cryptosystems," vol. 20, pp. 1–209.
[23] V. S. Miller, "Use of Elliptic Curves in Cryptography," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 218 LNCS, pp. 417–426, 1986, doi: 10.1007/3-540-39799-X_31/COVER.