Lightweight and Secure Elliptical Curve Cryptography (ECC) Key Exchange for Mobile Phones

https://doi.org/10.3991/ijim.v15i23.26337

Muneer Ahmad Dar[1](✉), Aadil Askar[2], Daliya Alyahya[2], Sameer Ahmad Bhat[3]
[1]National Institute of Electronics & Information Technology (NIELIT), Srinagar, India
[2]King Saud University, Riyadh, Saudi Arabia
[3]Gulf University for Science and Technology, Mubarak Al-Abdullah, Kuwait
muneer@nielit.gov.in

Abstract—Open networks enable data communication between different types of mobile devices that showcase the need to enforce elevated security measures. Securing sensitive or confidential data in mobile phones is accomplished by implementing a diverse range of cryptographic techniques. While encryption algorithms, such as Rivest–Shamir–Adleman (RSA) may offer secure solutions that are often difficult to compromise, these in turn prerequisite high speed computational resources for effective operation. Elliptical curve cryptography (ECC) is well thought-out standard that offers a workable and feasible methods of encryption/decryption, whilst being applicable to resource constraint devices. This paper implements a novel key exchange mechanism that helps to secure exchange of data between the communicating mobile devices. The study aims to address the limitation of Elliptic Curve Deffie Hellman, which is susceptible to Man-in-the-Middle attack and proposes an enhanced Elliptic Curve Deffie Hellman (ECDH) technique for secure data communication in open networks. The study results reveal, how the implementation of ECDH allows exchange of keys between the two communicating devices with limited resources.

Keywords—android, cryptography, ECC, ECDH, instructional technology, RSA

1 Introduction

Smart phones' processing capabilities are almost leveling up with the currently available desktop computers, and these carry functions that are equally comparable to functions offered by desktop computers. Although the size of a mobile phone is comparatively much lower than that of a desktop, it presents needs to have –high computational power to perform high speed operations, long battery life for powering the device, storage to handle large amounts of data, perhaps in Terabytes (TBs). Widely used smart phone operating systems–Android and IoS, include the basic and more advanced features targeted to attract a huge customer base, are consistently enriched with new intuitive features in the form of new product release. Innovative products as such offer more user centric features in addition to technological changes that improve the major devices functionalities such as processing capabilities, network bandwidth, storage, and I/O functions.

Despite more and more appealing features, researchers, techno savvy and educated class [24] of smartphone users are highly concerned about the security of smartphone data, and the way how confidentiality is ensured when smart phones exchange data with each other or communicate over a network susceptible to intruders. Over open networks, Android and IOS operating systems use apps that are specifically designed to allow data exchange or communication [17–20].

However, due to open networks being more vulnerable to attacks, these apps pose more security concerns, and must guarantee confidentiality and privacy to users' data during the communication process [21] [22]. Typical scenarios experiencing such challenges, include mobile learning environments, as well as mobile performance support systems that demand security solutions to improve the security of data exchanges, particularly in Mobile Electronic Performance Support Systems (MEPSS). A study by [23], implements MEPSS to benefit participants who aim to complete assigned tasks in a more reliable and timely manner. MEPSS supports delivery of instructions ubiquitously in the moment of need and eliminates the serve to reserve hard copy of the data for learning, or to follow instructions from a web-based information source on the internet. However, to access and allow data exchanges as such, prerequisites the need to secure data communication between the employed mobile devices. This leads us to propose a solution that secures data on resource constraint devices, such as mobile phone with limited processing capability. The field of instructional technology encompasses instructional and non-instructional solutions to improve learning and performance. The ECC based solution, in this context, has been used as a non-instructional solution for mobile communication end users.

1.1 Problem statement

Operating systems referred in the prior section typically run on resource constraint mobile devices, and as such encryption algorithms, like RSA, may not find its place within those operating systems due to their intense computational nature [1]. In symmetric encryption, both the dispatcher or sender, and the beneficiary or receiver use a particular secret key in both encryption/encipher and decryption/decipher operations, whereas in asymmetric encryption, exchanging key during data communication between the phone devices, presents a major challenge [3–7], (see Figure 1).

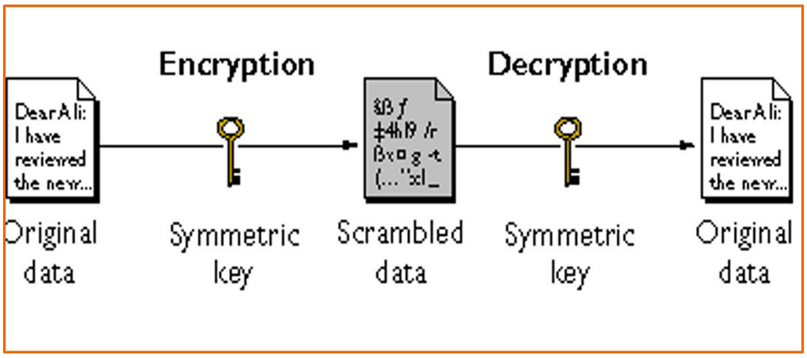


Fig. 1. Symmetric key cryptography

In the case of symmetric encryption, the key used to encrypt data, must be communicated confidentially to the receiver, to allow decryption of data at the receiver's end. Although the performance of such algorithms is better than the asymmetric algorithms, however the major concern is the secret exchange of key in an open network [8–16]. Asymmetric encryption, on the other hand makes use of two keys—public and private key to support encryption and decryption of data. The Figure 2 shows a typical process flow of asymmetric key exchange:

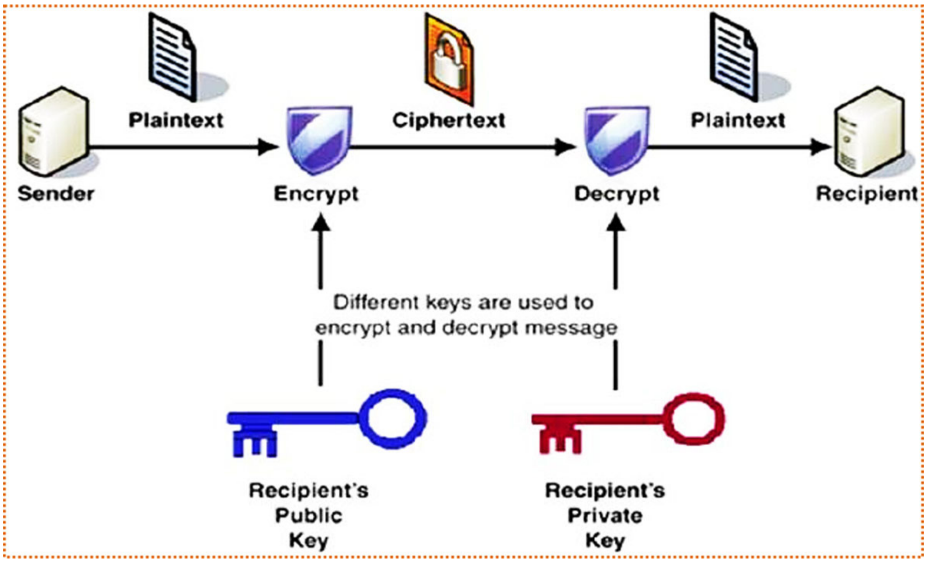


Fig. 2. Asymmetric key cryptography

Prime factorization based RSA and Elliptic Curve Cryptography ECC are the two well-known algorithms that base on the asymmetric encryption technique. These cryptographic techniques comprise of public/asymmetric key cryptographic algorithms along with the symmetric key cryptography, wherein only one key is employed for encryption. Cryptographic techniques as such, are competent enough to ensure security of data in networks with higher availability of and unlimited resources. Apparently, these cryptographic techniques are neglected for use in Android and IOS based smart phones owing to their reserve restriction characteristic.

As explained in the previous section, standard public key cryptographic algorithms, like RSA, is practically hard to implement on resource constraint devices since these type of algorithms demand high speed processors and need sustained supply sources to operate the device. Alternatively, a substitute called as **Elliptical Curve Cryptography (ECC)**, has been devised as a pragmatic solution to the problems demanding application of public key cryptography. Several research studies have been conducted [1–16] and researchers have concluded that ECC based algorithms can proficiently execute on the reserve/restricted equipment like smart phones. ECC offers the same level of encryption/decryption with key size of 210 bits compared to the encryption/decryption offered by RSA that uses a long key size of 2048 bits for encryption. RSA key length offers hardcore cryptographic security to block hackers to crack the algorithm (see Table 1). This concludes that ECC serve as an alternative to the established RSA.

Table 1. Comparison of the length of keys in RSA and ECC

RSA with Key Length in Bits	ECC with Key Length in Bits	Proportion of RSA/ECC
512	106	5:1
768	132	6:1
1024	160	7:1
2048	210	10:1

This study focuses on the need of ECC in Android smart phone operating system.

1.2 Study outline

The study is organized in the following sections. **Section 2.0—**gives an insight into the fundamental concepts of Android based operating system, as the implementation of the proposed key exchange will be demonstrated on this mobile operating system. **Section 3.0**—introduces prior works related to the Elliptic Curve Cryptography with main focus on the generation of public and private keys. **Section 4.0—**presents the implementation of the Enhanced ECC Key Exchange for secure data transmission using mobile phones. **Section 5.0—**recommends an improvement to the Elliptic Curve Diffie Hellman (ECDH) algorithm on Android based operating system to avoid Man-in-the-Middle attack. **Section 6.0—**provides an overall conclusion of the current study.

2 Overview of Android OS

Smart phone adoption has expanded at a rapid pace and mobile computing has seen huge increase over the past decade. This is what Andy Robin director of versatile smartphone frameworks at Google, has envisioned. He envisions that desktops ought to be totally supplanted by these little, intelligent and compact handheld gadgets called the smartphones [2]. The time has arrived, when smart phones have totally supplanted the desktops and personal computers (PCs). Handheld gadgets as such, are now becoming a close companion and part of the daily lives of ICT clients/users across the globe.

Smart phones typically run either Android or IOS operating system. Android operating system (AOS) is a superimposed or layered framework. Various functionalities are incorporated at the application layer, which include apps, like SMS, Email, GPS and other interesting applications with user interfaces. The development of such applications carrying several intuitive features is mainly carried out with Java programing language. Figure 3 shows detailed architecture of the Android Operating System is as under.

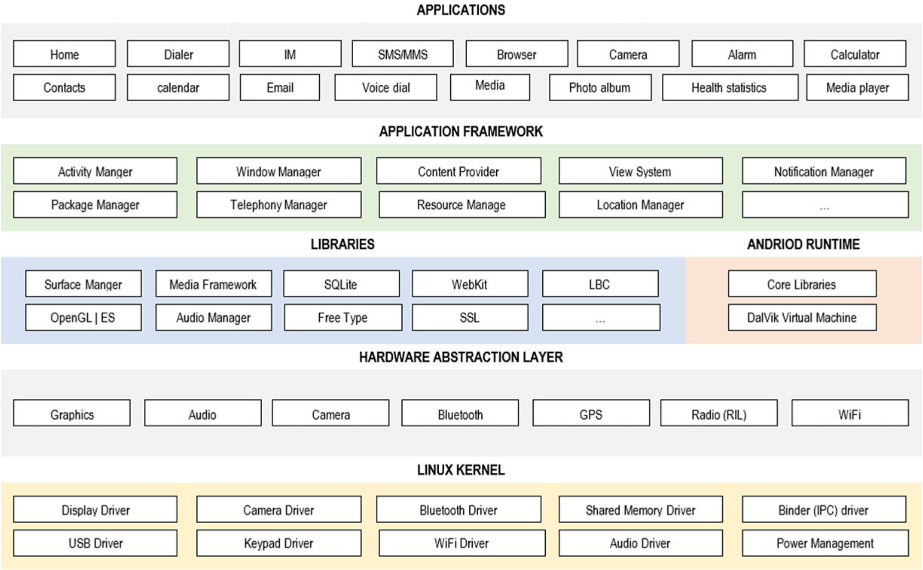


Fig. 3. Architecture of Android OS

2.1 Review of elliptic curve cryptography

Elliptic Curve Cryptography (ECC) is an asymmetric, public key cryptographic technique wherein the communicating devices generate two keys—a public key and a secret key called the private key. The public key is distributed to all the devices, whereas the private key is hidden and kept secret by the client encrypting or decrypting the message [1]. Elliptic curve is represented (see Figure 4) by a sextuple called as domain parameters

$$T = (P, a, b, G, n, h)$$

An Elliptic curve E over $\mathbb{F}_p$ is a set of all solutions $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ to an equation

$$y^2 = x^3 + ax + b$$

where $\mathbb{F}_p$ represents the field of integers modulo p, and $a, b \in \mathbb{F}_p$, satisfy the equation $4a^3 + 27b^2 \not\equiv 0$, together with a special point ∞ called *the point at infinity*.

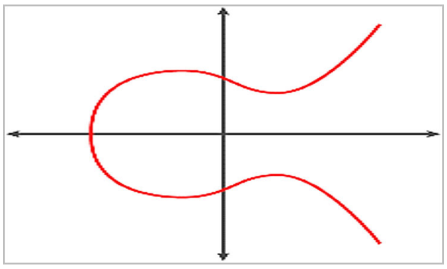


Fig. 4. Elliptic curve

The realization of ECC depends on the following concepts, being imperative in the implementation of ECC.

2.2 ECC as discrete logarithm problem

As depicted in Figure 5, let's suppose two points on the curve are *P* and *Q*, such that $k.P = Q$, where k is a scalar. If by chance an intruder gains access to the values of *P* and *Q*, then it is not easy for the intruder to compute the value of k since computing the value of k is impractical from the curve. Thus, ECC is very hard to be cracked, as computing the value of k basically represents a Discrete Logarithm Problem (DLP) [3–6], that needs to be intractable.

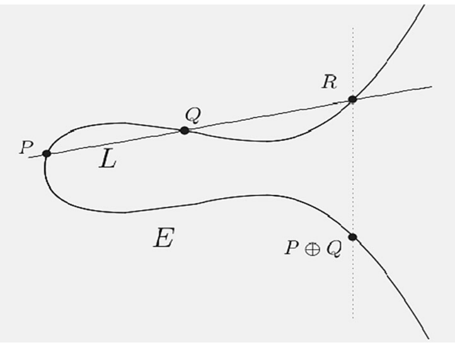


Fig. 5. Point addition in elliptic curve

2.3 ECC public key cryptosystem

In public key elliptic curve cryptosystems, suppose that a mobile device-1 intends to send a message '**Msg**' to mobile device-2, confidentially. Then, the points on the curve can be computed as N, such that,

$$NPo = P_o + P_o + \cdots + \cdots\cdots P_o$$

$$N\ times = 0\ (\infty).$$

2.4 Generate public and private keys

As mentioned in the prior section, the communication between two entities, say mobile device-1 and mobile device-2 is established once both the parties would agree to use the same parameters on the curve. The initial generated point is P, and the N is concurred upon successive additions of P to itself. Now mobile device-1 generates a random number $\boldsymbol{M1\_no} < N$, which is the private key for mobile device-1. The public key is set by the mobile device-1 as $\boldsymbol{pub\_M1} = \boldsymbol{P.M1\_no}$. Accordingly, the mobile device-2 generates a private key as $\boldsymbol{M2\_no} < N$ and a public key by computing $\boldsymbol{pub\_M2} = \boldsymbol{P.M2\_no}$.

2.5 Generate shared keys

As mobile device-1 and mobile device-2 start communication to exchange public keys, as generated in the above step, both the devices then generate a common key by computing a $shared\_Key = M1\_no \times pub\_M2$ and $shared\_Key = M2\_no \times pub\_M1$. The shared keys have a common value, and hence we can have:

$$M1\_no \times pub\_M2 = M1\_no \times \left(M2\_no \times P\right)$$

or

$$M2\_no \times (M1\_no \times P) = M2\_no \times pub\_M1$$

2.6 Encryption

In this step, mobile device-1 wishes to encrypt a message “Msg”, which it intends to send it to mobile device-2. The mobile device-1 randomly selects a number N and a private key $M1\_no$, and the public key is generated by computing $pub\_M1 = P \times M1\_no$. The encryption of the given message then generates a new encrypted text called as cipher text.

2.7 Decryption

For the mobile device-2 to decrypt the cipher text, a reverse process used in the encryption, is applied to isolate the actual intended message.

3 Related work

The research done in the area of security and privacy of users using various IoT devices is summarized in Table 2, as under:

Table 2. Previous studies

Reference	Description	Limitations
Al-Mahmud and Morogan [6]	The Elliptic Curve based digital signature is implemented for the identity and authentication of users who are registered at the base stations who have every control to give access to the authenticated users	This research has contributed in preventing the denial of service attacks (DoS) but the base station is vulnerable to number of attacks while the users are registering or doing any other activity through base station.
Gupta et al. [7]	A cloud based approach is used wherein the IoT devices are directly communicating with the cloud with the help of embedded sensors. The XML based web services are used to enable IOT devices to secure their data and to fast the interaction with the cloud.	The increased numbers of users are directly interacting with the server database which may lead to the delay in the response. No authentication is provided by the researchers and the data may be compromised
Rathee et al. [8]	Smart Healthcare is discussed in this research and a blockchain technology based framework is proposed	The proposed approach has a 86% success against different attacks in a smart city
Wang et al. [9]	The authentication of users is executed through a central repository called the key distribution centre. The KDC is responsible for the authentication and the privileges are provided to different users through KDC	The mutual authentication of sensors and users is not supported by the proposed framework
Kavitha et al. [10]	The cryptographic techniques are discussed and an ECC based enhanced technique is implemented to secure the privacy of users while communicating.	The performance of the proposed technique is not computed.
Wazid et al. [11]	The detailed comparison of authentication schemes is presented and the issues pertaining to IoT devices in terms of computational capabilities are discussed. The use of cloud and big data is discussed in this research	Only the theoretical description os provided.
Ummer Iqbal et al [12]	The researchers have made use of hash function and a modified ECC to design a novel access control scheme which is able to execute with low computational facility. The realistic testing of the proposed framework is conceded using the TinyOS operating system on MICAz motes. The security validation is done using AVISPA.	The proposed protocol is restricted and is not permitted for the cross-domain.

4 Key exchange via Android OS

The study employs two mobile devices with insufficient hardware resources, limiting the power to execute the computationally expensive algorithms on the employed devices. The key objective of this study is to securely communicate confidential data, in this study a simple message, between two smart phones using the ECC. As explained in the prior sections, ECC is capable to perform well on low power devices.

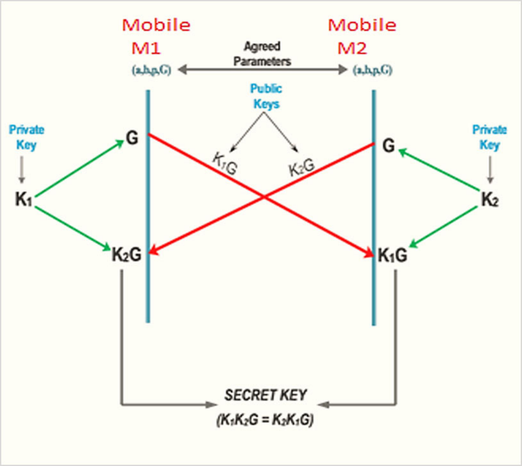


Fig. 6. Elliptic curve Diffie-Hellman exchange

We also reviewed the functionality of Android so as to implement ECC on Android platform. Before implanting the ECC on the two Smartphone devices, the concept of Elliptic Curve Diffie-Hellman Exchange (ECDH) is illustrated in the Figure 6. The generator point *G* is used by the two communicating smart phones, and the *public_key* and *private_key* is generated. Public keys are shared by the two devices so as to generate a common shared key.

The javax.Crypto library is used for the implementation of ECC on the Android platform. The class libraries used are described as in the Figure 7.

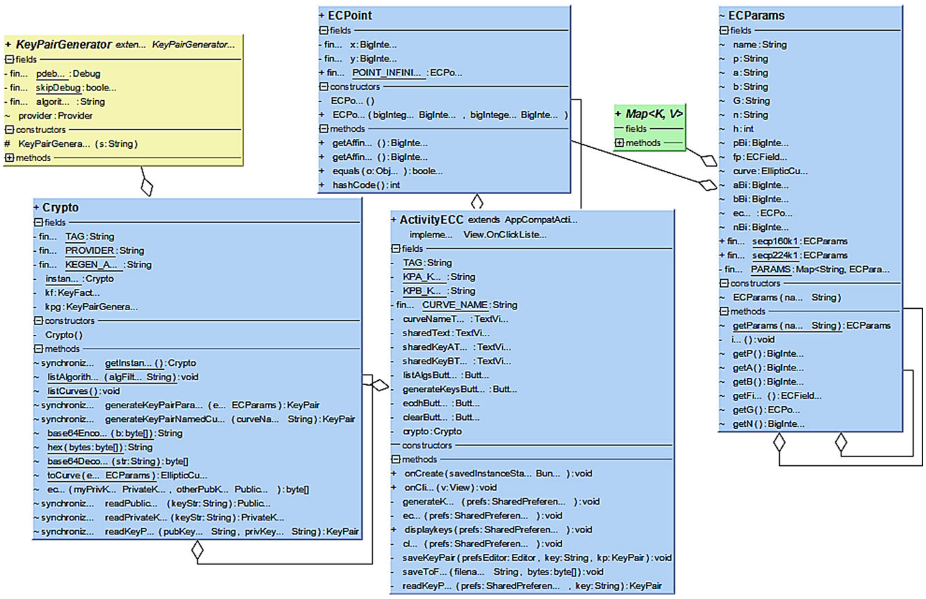
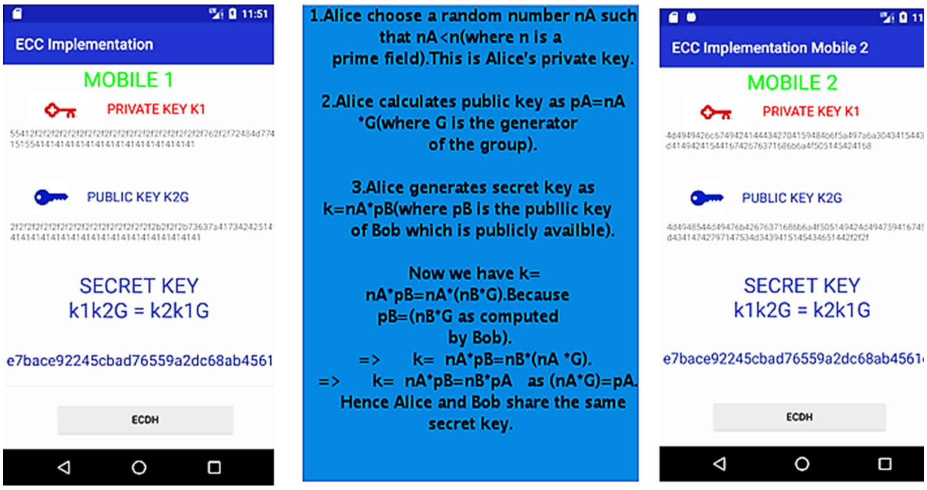

+ KeyPairGenerator exten... KeyPairGenerator...
fields
- fin... pdeb... : Debug
- fin... skipDebug : boole...
- fin... algorit... : String
~ provider : Provider
constructors
# KeyPairGenera... (s: String)
methods
+ ECPoint
fields
- fin... x: BigInte...
- fin... y: BigInte...
+ fin... POINT_INFINI... : ECPo...
constructors
- ECPo... ()
+ ECPo... (bigInteg... BigInte... , bigIntege... BigInte...)
methods
+ getAffin... () : BigInte...
+ getAffin... () : BigInte...
+ equals (o: Obj...) : boole...
+ hashCode () : int
+ Map<K, V>
fields
methods
~ ECParams
fields
~ name : String
~ p : String
~ a : String
~ b : String
~ G : String
~ n : String
~ h : int
~ pBi : BigInte...
~ fp : ECField...
~ curve : EllipticCu...
~ aBi : BigInte...
~ bBi : BigInte...
~ ec... : ECPo...
~ nBi : BigInte...
+ fin... secp160k1 : ECParams
+ fin... secp224k1 : ECParams
- fin... PARAMS : Map<String, ECPara...
constructors
~ ECParams (na... String)
methods
~ getParams (na... String) : ECParams
- i... () : void
~ getP () : BigInte...
~ getA () : BigInte...
~ getB () : BigInte...
~ getFi... () : ECField...
~ getG () : ECPo...
~ getN () : BigInte...
+ Crypto
fields
- fin... TAG : String
- fin... PROVIDER : String
- fin... KEGEN_A... : String
- instan... : Crypto
- kf : KeyFact...
- kpg : KeyPairGenera...
constructors
- Crypto ()
methods
~ synchroniz... getInstan... () : Crypto
~ listAlgorith... (algFilt... String) : void
~ listCurves () : void
~ synchroniz... generateKeyPairPara... (e... ECParams) : KeyPair
~ synchroniz... generateKeyPairNamedCu... (curveNa... String) : KeyPair
~ base64Enco... (b: byte[]) : String
~ hex (bytes: byte[]) : String
~ base64Deco... (str: String) : byte[]
~ toCurve (e... ECParams) : EllipticCu...
~ ec... (myPrivK... PrivateK... , otherPubK... Public...) : byte[]
~ synchroniz... readPublic... (keyStr: String) : Public...
~ synchroniz... readPrivateK... (keyStr: String) : PrivateK...
~ synchroniz... readKeyP... (pubKey... String , privKey... String) : KeyPair
+ ActivityECC extends AppCompatActi...
impleme... View.OnClickListe...
fields
- TAG : String
- KPA_K... : String
- KPB_K... : String
- fin... CURVE_NAME : String
- curveNameT... : TextVi...
- sharedText : TextVi...
- sharedKeyAT... : TextVi...
- sharedKeyBT... : TextVi...
- listAlgsButt... : Butt...
- generateKeysButt... : Butt...
- ecdhButt... : Butt...
- clearButt... : Butt...
- crypto : Crypto
constructors
methods
+ onCreate (savedInstanceSta... Bun...) : void
+ onCli... (v: View) : void
- generateK... (prefs: SharedPreferen...) : void
- ec... (prefs: SharedPreferen...) : void
+ displaykeys (prefs: SharedPreferen...) : void
- cl... (prefs: SharedPreferen...) : void
- saveKeyPair (prefsEditor: Editor , key: String , kp: KeyPair) : void
- saveToF... (filena... String , bytes: byte[]) : void
- readKeyP... (prefs: SharedPreferen... , key: String) : KeyPair

Fig. 7. Various classes used in Android for the implementation of ECC


11:51

ECC Implementation

MOBILE 1

PRIVATE KEY K1

55412f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f762f2f72484d774
1515541414141414141414141414141414141414

PUBLIC KEY K2G

2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2f2b2f2f2b73637a4173424251 4
41414141414141414141414141414141414141414141

SECRET KEY
k1k2G = k2k1G

e7bace92245cbad76559a2dc68ab4561

ECDH

1.Alice choose a random number nA such that nA<n(where n is a prime field).This is Alice's private key.

2.Alice calculates public key as pA=nA *G(where G is the generator of the group).

3.Alice generates secret key as k=nA*pB(where pB is the publlic key of Bob which is publicly availble).

Now we have k= nA*pB=nA*(nB*G).Because pB=(nB*G as computed by Bob).
=> k= nA*pB=nB*(nA *G).
=> k= nA*pB=nB*pA as (nA*G)=pA.
Hence Alice and Bob share the same secret key.

11

ECC Implementation Mobile 2

MOBILE 2

PRIVATE KEY K1

4d4949426c6749424144434270415948 4b6f5a497a6a3043415443
d414942415441674267637168 6b6a4f505145424168

PUBLIC KEY K2G

4d4948544d49476b4267637168 6b6a4f5051494 24d494759416745
d43414742797147534d3439415145434651442f2f2f

SECRET KEY
k1k2G = k2k1G

e7bace92245cbad76559a2dc68ab4561

ECDH

Fig. 8. Screen shot of secure and confidential key exchange on Android platform

The screen shot shown in Figure 8 of the implementation of ECC on Android based operating system with two communicating smart phones is shown in the Figure 8. The two applications developed are—Mobile-1 and Mobile-2, to exchange the data. The validation of secure data exchange between mobile phones is completed by generating two keys- the *public_key* and the *private_key*. Once the *public_key* is sent via an open network, the two communicating mobile phones are proficient to generate a universal *secret_key*. After sharing the keys, the two communicating mobile phones were capable to establish a communication link and could now proficiently exchange confidential data on an insecure communication channel.

5 Proposed enhancement and results

In the previous section, we implemented Elliptic Curve Dephie Helman (ECDH) algorithm to secretly communicate the shared keys between the two smart phone devices. While communicating the shared keys for the encryption and decryption of the confidential message, researchers identify Man-in-the-Middle attack as a security loophole in such type of communication, and this attack is enforced as a result of weak validation methods used in the user authentication process [11–12]. We can further enhance the above implemented algorithm on mobile phones by incorporating more authentication procedures so as to get rid of the Man in the Middle Attacks. The proposed enhanced algorithm is presented in this section and the secret codes utilized in the algorithm are described in the Table 3.

Table 3. Symbol table used by the algorithm

Symbol	Description
A	Random Number selected through Smartphone-1
B	Random Number selected through Smartphone-2
G	Generator Point on the Curve
A.G	Undisclosed at Smartphone-1
B.G	Undisclosed at Smartphone-2
Ka	Private_Key of Smartphone-1
Kb	Private_Key of Smartphone-2
Pub_a=(Ka.G)	Public_ Key of Smartphone-1
Pub_b=(Kb.G)	Public_Key of Smartphone-2

The proposed algorithm works with the objective to create a common shared key that is not susceptible to the Man-In-the-Middle Attack. The two algorithms which are getting executed at two smart phones are as under:

```
Algorithm Smartphone 1
{
  Chooses A.G as a Secret key
  Compute A.PUb.Ka and sends it to the Smartphone 2
  Wait for the Smartphone 1 to receive the public key
  Compute A.G.Ka.B.Ka-1= A.G.B and send it Smartphone 2
}
Algorithm Smartphone 2
{
  Compute: A.PUb.Ka.Kb1=A.Kb.G.Ka .Kb-1 = A.G.Ka
  Compute: A.G.Ka.B and sends it to Smartphone 1
  Waits for the Smartphone 1 to receive the public key
  Computes the Secret: A.G.B.B-1= X.G }
}
```

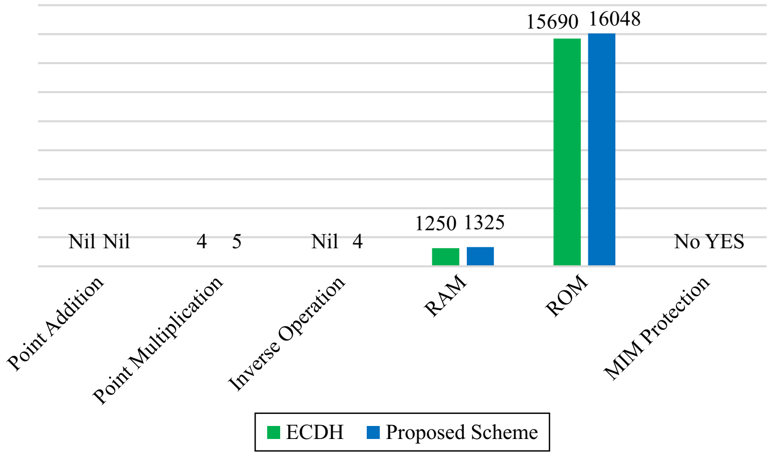


Fig. 9. Performance comparison of ECDH and proposed scheme

The developed protocol aims to make ECDH authentication more secure and safeguard data exchanges from the Man-in-the-Middle attack. The above algorithm relatively consumes more memory as compared to the original ECDH algorithm, nonetheless it simultaneously prevents Man-in-the-Middle attack. The proposed protocol was evaluated and tested on an emulator in the Android Studio and the results of our proposed protocol over the traditional ECDH are highlighted in the Figure 9.

6 Conclusion

This study attempted to address the security needs of the smart phones with limited computational capabilities. Since traditional cryptographic techniques like RSA need sufficient hardware resources, ECC as an alternative is introduced to handle security and privacy necessities in smart phones. An improved key exchange procedure has been suggested to overcome the shortcomings of ECDH in terms of man in the middle attack.

With an immaterial operating outlay of Random Access Memory and slightly higher Read Only Memory, the significant advantage of the proposed protocol is that it protects itself of well-known Man-in-the-Middle attack; Such an active attack is a serious threat to users of smart phones, novice users in particular.

As ECC encryption requires lower computational processing, ECC is practically more viable and suitable for resource constrained mobile devices. ECC can wisely put into practice, the preliminary concepts of authentication, confidentiality and Integrity in smart, and ECC can be exploited more closely in future studies to develop conventional protocols, as explained in this study.

7 References

[1] A. H. Moon and Ummer, "Authentication protocols for WSN using ECC and hidden generator," Int. J. Comput. Appl., vol. 133, no. 13, pp. 42–47, 2016. https://doi.org/10.5120/ijca2016908175

[2] M. A. Dar and J. Parvez, "Novel techniques to enhance the security of smartphone applications," International Journal of Interactive Mobile Technologies (iJIM), vol. 10, no. 4, pp. 32–36, 2016. https://doi.org/10.3991/ijim.v10i4.5869

[3] N. Druml et al., "A flexible and lightweight ECC-based authentication solution for resource constrained systems," Proc.–2014 17th Euromicro Conf. Digit. Syst. Des. DSD 2014, pp. 372–378, 2014. https://doi.org/10.1109/DSD.2014.77

[4] J. W. Bos, J. A. Halderman, N. Heninger, J. Moore, M. Naehrig, and E. Wustrow, "Elliptic curve cryptography in practice," IACR Cryptol. …, pp. 157–175, 2014. https://doi.org/10.1007/978-3-662-45472-5_11

[5] X. Wu, O. Dandash, and P. D. Le, "The design and implementation of a smartphone payment system based on limited-used key generation scheme," Proc.–Third Int. Conf. onInformation Technol. New Gener. ITNG 2006, vol. 2006, pp. 458–463, 2006.

[6] A. Al-Mahmud and M. C. Morogan, "Identity-based authentication and access control in wireless sensor networks,'' Int. J. Comput. Appl., vol. 41, no. 13, pp. 18–24, 2012. https://doi.org/10.5120/5602-7858

[7] P. K. Gupta, B. T. Maharaj, and R. Malekian, "A novel and secure IoT based cloud centric architecture to perform predictive analysis of user's activities in sustainable health centres,'' Multimedia Tools Appl., vol. 76, no. 18, pp. 18489–18512, Sep. 2017. https://doi.org/10.1007/s11042-016-4050-6

[8] G. Rathee, A. Sharma, H. Saini, R. Kumar, and R. Iqbal, "A hybrid framework for multimedia data processing in IoT-healthcare using blockchain technology,'' Multimedia Tools Appl., 2019. https://doi.org/10.1007/s11042-019-07835-3

[9] H. Wang, B. Sheng, and Q. Li, "Elliptic curve cryptography-based access control in sensor networks,'' Int. J. Secur. Netw., vol. 1, nos. 3–4, pp. 127–137, 2006. https://doi.org/10.1504/IJSN.2006.011772

[10] S. Kavitha, P. J. A. Alphonse, and Y. V. Reddy, "An improved authentication and security on ef_cient generalized group key agreement using hyper elliptic curve based public key cryptography for IoT health care system,'' J. Med. Syst., vol. 43, Jul. 2019, Art. no. 260. https://doi.org/10.1007/s10916-019-1378-2

[11] M. Wazid, A. K. Das, R. Hussain, G. Succi, and J. J. P. C. Rodrigues, "Authentication in cloud-driven IoT-based big data environment: Survey and outlook,'' J. Syst. Archit., vol. 97, pp. 185–196, Aug. 2019. https://doi.org/10.1016/j.sysarc.2018.12.005

[12] U. Iqbal, A. H. Mir, "Secure and practical access control mechanism for WSN with node privacy," Journal of King Saud University–Computer and Information Sciences, 2020, ISSN 1319–1578. https://doi.org/10.1016/j.jksuci.2020.05.010

[13] A. H. Moon and Ummer, "Authentication protocols for WSN using ECC and hidden generator," Int. J. Comput. Appl., vol. 133, no. 13, pp. 42–47, 2016. https://doi.org/10.5120/ijca2016908175

[14] N. Druml et al., "A flexible and lightweight ECC-based authentication solution for resource constrained systems," Proc.–2014 17th Euromicro Conf. Digit. Syst. Des. DSD 2014, pp. 372–378, 2014. https://doi.org/10.1109/DSD.2014.77

[15] P. Ragunathan, K. Sambath, and V. Karthik. L, "Accessing a network using a secure android application," Int. J. Adv. Netw. Appl., vol. 4, no. 1, pp. 1503–1508, 2012.

[16] L. Wang, H. Wang, M. K. Khan, and D. He, "Lightweight anonymous key distribution scheme for smart grid using elliptic curve cryptography," IET Commun., vol. 10, no. 14, pp. 1795–1802, 2016. https://doi.org/10.1049/iet-com.2016.0091

[17] H. Sarhan, A. a Hafez, and A. Safwat, "Secure android-based mobile banking scheme," Int. J. Comput. Appl., vol. 118, no. 12, pp. 21–26, 2015. https://doi.org/10.5120/20797-3460

[18] S. Rangarajan, N. S. Ram, and N. V. Krishna, "Securing SMS using cryptography," vol. 4, no. 2, pp. 285–288, 2013.

[19] M. A. Dar and J. Parvez, "Security enhancement in android using ellipic curve cryptography," Int. J. Secur. its Appl., vol. 11, no. 6, pp. 27–34, 2017. https://doi.org/10.14257/ijsia.2017.11.6.03

[20] L. Simon and R. Anderson, "Security analysis of android factory resets," 4th Mob. Secur. Technol. Work., p. 10, 2015.

[21] M. A. Dar and J. Parvez, "Enhancing security of Android & IOS by implementing need-based security (NBS)," in 2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014, 2014, pp. 728–733. https://doi.org/10.1109/ICCICCT.2014.6993055

[22] M. A. Dar and J. Parvez, "A live-tracking framework for Smartphones," in ICIIECS 2015–2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems, 2015, pp. 3–6. https://doi.org/10.1109/ICIIECS.2015.7193066

[23] Askar, Aadil, "Mobile electronic performance support system as a learning and performance solution: a qualitative study examining usage, performance, and attitudes," Turkish Online Journal of Educational Technology-TOJET, vol. 17, no. 2, pp. 76–88, 2018.

[24] Dar, Muneer Ahmad, and Sameer Ahmad Bhat. "Evaluation of mobile learning in workplace training." 2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI). IEEE, 2016. https://doi.org/10.1109/ICACCI.2016.7732255

8 Authors

Muneer Ahmad Dar, has received his M.Phil degree (2009) in Computer Science from Madurai Kamaraj University, and M.S. (2004) and Ph.D. degree in Computer Science (2019) from the University of Kashmir, India. He is an active member of Institution of Electronics and Telecommunication Engineers (IETE), and International Association of Engineers (IAENG) group. He is the Senior Researcher (Scientist-C) at National Institute of Electronics and Information Technology (NIELIT–Jammu and Kashmir), Ministry of Electronics and Information Technology, (Govt. of India), and also serves as Head of Department–Computer Science at NIELIT. His research interests relate to multiple areas of computer science and engineering. Dr. Muneer has published research in various international journals and conferences, and he has added his contribution to several book chapters in computer science. Currently, his research activities focus on Security of Smartphone Applications, Network Security, Programming Languages, Design & Analysis of algorithms, Data Structures and Optimization Techniques. (email: muneer@nielit.gov.in)

Aadil Askar, received the Ph.D. degree from the University of Northern Colorado (UNC), USA. He is an Associate Professor at King Saud University (KSU), Saudi Arabia. He is specialized in the field of Educational Technology. He has served as Assistant Vice Dean at the Preparatory Year, KSU and as a consultant to the Ministry of Education, Saudi Arabia. Dr. Aadil has also undertaken several challenging roles, which includes the role an instructor at King Fahad Airbase, Ministry of Defense from 2000 to 2002. Currently, he works with the Ministry of Education Saudi Arabia, and his main research interests relate to Human Computer Interaction (HCI), Human Performance Technology (HPT), Life-long learning, and Educational Technology. (email: aadil@ksu.edu.sa)

Dalia Alyahya, received the Ph.D. degree (2011) from the University of Northern Colorado (UNC), USA. Besides her M.A degree in Educational Technology from the UNC, she has also earned the M.A degree in Arts Education (2005) from King Saud University (KSU), Saudi Arabia. Currently, she works as Associate Professor in the Dept. of Instructional Technology at KSU. She is also the Advisor and Consultant to the Heritage Commission at the Ministry of Culture, Saudi Arabia. Her research interests relate to Information Visualization, Eye Tracking Technology, Instructional Design and Technology, and Human Computer Interaction. (email: dmalyahya@ksu.edu.sa)

Sameer Ahmad Bhat, received the B.S. degree in Information Technology (2008) and M.S. degree in Computer Science (2011) from the University of Kashmir, India in 2008 and 2011, respectively. From 2011 to 2016, he was affiliated with the Dept. of Self Development Skills at P.Y., King Saud University in Saudi Arabia, and from 2016 to 2021, he worked with the Dept. of Electrical and Electronics engineering at Kuwait College of Science and Technology in Kuwait. Currently, he is affiliated with Gulf University for Science and Technology, Kuwait. His current research interests relate to–AI and Machine learning, Human Motion Activity and Recognition Systems, Embedded Systems and Internet-of-Things, Wireless Power Transfer, Multi-level Inverter Circuits, and Educational Technology. (email: bhat.sameerahmad@yahoo.com)

Article submitted 2021-08-17. Resubmitted 2021-10-07. Final acceptance 2021-10-09. Final version published as submitted by the authors.