Elliptic Curve Cryptography-Overview for Recent Cloud Architecture

[I]Hemanth Chakravarthy M, [II]Kannan E
[I]Research Scholar, [II]Professor
[I,II]Dept. of CSE, Veltech Technical University, Chennai, India.

Abstract

Elliptical Curve cryptography has become popular because of its advanced security features and its ability to provide security with short keys which are much efficient than RSA (for example, a 160 bit ECC has roughly the same security strength as 1024 bit RSA). So its security features and advanced developments will be discussed in this paper

Keywords

Elliptic curve cryptography Public Key Cryptography, Cloud Computing, RSA

I. Introduction

Elliptic curve cryptography was introduced in the mid-1980s independently by Koblitz and Miller [1] as an alternative for cryptographic protocols based on the discrete logarithm problem in the multiplicative group of a finite field.

RSA is based on factoring but ECC is based on logarithms. Elliptic curves are known in mathematics over hundred years but their applications to cryptography were found twenty years ago by Diffie, Hellman and ElGamal

Elliptical curve cryptography (ECC) is a public key encryption technique that can be used to create faster, smaller, and more efficient cryptographic keys. ECC generates keys through the properties of the elliptic curve equation instead of the traditional method of generation as the product of very large prime numbers. The technology can be used in conjunction with most public key encryption methods, such as RSA, and Diffie-Hellman.

Till date, the ECC has the highest strength-per-bit compared to other public key cryptosystems. Because of its keys sizes being smaller bandwidth and memory are easily reduced so, many are preferring this than RSA

In current public key cryptography, factors decomposition problems based on large numbers are commonly used, for example, RSA. With the development of computer hardware and high-performance computing technology, RSA has encountered some difficulties.

ECC has the desired properties from overcoming lot of existing problems. This comes from the fact, that there are no sub exponential algorithms for the ECDLP (elliptic curve discrete logarithm problem) known today. This means that we can use shorter keys (compared to other cryptosystems) for high security levels

II. Elliptical curve cryptography

ECC is an emerging public key which has numerous advantages compared to other public key algorithms.

The security due to ECC relies on the difficulty of Elliptic Curve Discrete Logarithm Problem. Let P and Q be two points on an elliptic curve such that kP = Q, where k is a scalar. Given P and Q, it is computationally infeasible to obtain k. If k is sufficiently large, k is the discrete logarithm of Q to the base P. Hence the main operation involved in ECC is related to the point multiplication i.e. multiplication of a scalar k with any point P on the curve to obtain another point Q on the curve.

The efficiency of an algorithm is measured by the limited resources it uses. The basic measure is time but even memory plays a very important role. It is reasonable to expect that an algorithm consumes greater resources for larger inputs and the efficiency of an algorithm is therefore described as a function of the input size.

An overview of EC cryptographic algorithms for key agreement and digital signature are explained below.

A. ECDH – Elliptic curve Diffie Hellman

ECDH, a variant of DH, is a key agreement algorithm. For generating a shared secret between A and B using ECDH, both have to agree up on Elliptic Curve domain parameters. An overview of ECDH is given below

1. Key Agreement Algorithm

For establishing shared secret between two device A and B

Step1. If dA and dB are the private key of device A and B respectively, Private keys are random number less than n, where n is a domain parameter.

Step2. If QA = dA*G and QB = dB*G are the public key of device A and B respectively, G is a domain parameter

Step3. A and B exchange their public keys

Step4. The end A computes K = (xK, yK) = dA*QB

Step5. The end B computes L = (xL, yL) = dB*QA

Step6. Since K=L, shared secret is chosen as xK

B. ECDSA - Elliptic curve Digital Signature Algorithm

Digital Signature Standard (DSA) is a public key algorithm that is used for Digital Signature. The DSA standard is specified FIPS 186-2, Digital Signature Standard . ECDSA is a variant of the Digital Signature Algorithm (DSA). For sending a signed message from A to B, both have to agree up on Elliptic Curve domain parameters. Sender A have a key pair consisting of a private key dA (a randomly selected integer less than n, where n is the order of the curve, an elliptic curve domain parameter) and a public key QA = dA*G (G is the generator point, an elliptic curve domain parameter).

1. Signing

Consider the device A that signs the data M that it sends to B.

Step7. Let dA be A's private key

Step8. Calculate m = HASH (M), where HASH is a hash function, such as SHA-1

Step9. Select a random integer k such that 0<k<n
Step10. Calculate r = x1 mod n, where (x1, y1) = k*G
Step11. Calculate s = k -1(m + dA*r) mod n
Step12. The signature is the pair (r, s)

2. Verification

Step13. Let M be the message and (r, s) be the signature received from A.
Step14. Let QA be A's public key. Since QA is public, B has access to it.
Step15. Calculate m = HASH (M)
Step16. Calculate w = s $_{-1}$ mod n
Step17. Calculate u_1= m*w mod n and u_2 = r*w modn
Step18. Calculate (x1, y1) = u1*G + u2*QA
Step19. The signature is valid if x1 = r mod n, invalid Otherwise .

III. Comparison of ECC over RSA

The most important difference between ECC and other conventional cryptosystems is that for a well-chosen curve, the best method currently known for solving the ECDLP is fully exponential.
The fundamental operation underlying ECC is point multiplication, which is defined over finite field operations. All standardized elliptic curves are defined over either prime integer fields GF(p) or binary polynomial fields GF(2m).The contrast in key lengths of RSA, DSA and ECC are shown in the graph (Fig1) below. Clearly, ECC keys take much more effort to break compared to RSA and DSA keys. Due to this, many people believe that ECDLP is intrinsically harder than the other two problems.

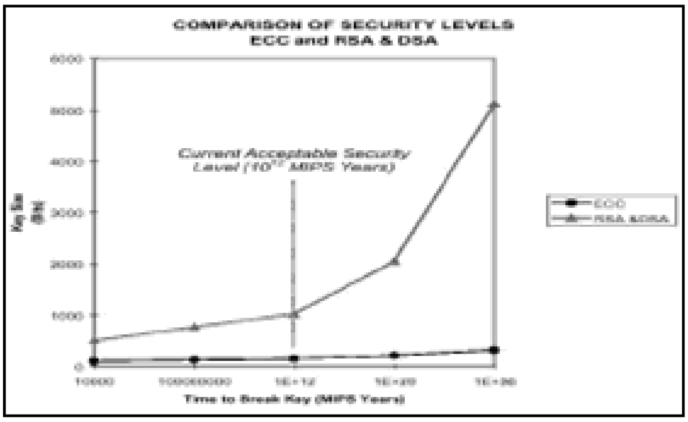

Fig. 1: Comparitive study of ECC over RSA/DSA

A simple example to compare between the Elliptic Curve Cryptography and RSA, A 160 bit key in ECC is considered to be secured as 1024 bit key in RSA. And also When securing a 128-bit symmetric key, it is prudent to use 256-bit Elliptic Curve Cryptography
The majority of public key systems in use today use 1024-bit parameters for RSA and Diffie-Hellman.
The U.S. National Institute for Standards and Technology (NIST) has recommended that these 1024 – bit systems are sufficient for use until 2010.
All NIST and some SECG curves further allow for optimization based on the curve parameters.

Table1: Comparision of ECC over RSA(Source:NIST)

S no	Symmetric Key Size (In bits)	Elliptical Curve Cryptography key size (In bits)	RSA and Diffie-Hellman key size (In bits)
1	80	160-223	1024
2	112	224-255	2048
3	128	256-383	3072
4	192	384-511	7680
5	256	512+	15360

The key size relationship between the ECC and the RSA, and the appropriate choice of the AES key size is as following table:

Table-2: ECC and RSA Key Comparison (Key size in bits – source: Certicom, NIST)

S No	ECC Key Size	RSA Key Size	Key-Size Ratio	AES Key size
1	163	1024	1:6	NA
2	256	3072	1:12	128
3	384	7680	1:20	192
4	582	15360	1:30	256

IV. Equations of Elliptical Curve

The equation of an elliptic curve can be written in the form:

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

Where A and B are integers. This condition ensures the curve does not have a cusp or double point on the real axis. This is also called a Weierstrass equation. It is a special type of elliptic curve which is the ideal one required here.
Some examples of elliptical curves can be
$y^2 = x^3 - 3x + b$
$y^2 = x^3 - 6x + b$
The very important property of elliptic curves is that we are able to define an operation + of addition for points which are on the curve, to produce a third point which is also on the curve. This operation makes the curve and various subsets of points on the curve, into a finitely generated abelian group.

V. Conclusion

ECC is truly an obvious choice for doing asymmetric Cryptology in portable, necessarily constrained device right now. As an example, a popular, recommended RSA key size for most application is 2,048 bits. For equivalent security using ECC, key size of only 224 bits is needed. The difference becomes more and more pronounced as security levels increase and as hardware gets faster, and the recommended key size must be increased. A 384 bit ECC key matches a 7860-bit RSA key for security.
In this paper, we demonstrated the Elliptic Curve Cryptography (ECC). Theoretical and practical aspects of ECC are mentioned. It is evident that ECC can be used for saving time, cost less memory usage.
ECC's advantages and some application of ECC like ECDSA.
A detailed study of ECDSA is done for verification.
The performance, security and future enhancement of ECC is

discussed which gives scope for future work for advanced ECC methods

References

[1] *Neal Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48:203-209, 1987..*

[2] *Athavale, A.; Singh, K.; Sood, S., "Design of a Private Credentials Scheme Based on Elliptic Curve Cryptography", First International Conference on Computational Intelligence, Communication Systems and Networks, 2009. CICSYN '09, pp. 332 - 335*

[3] *FIPS PUB 186-2,Digital Signature standard(DSS), January 2000.*

[4] *Chandramohan, B. and Baskaran, R., "Reliable transmission in network centric military network", European Journal of Scientific Research, Vol. 50, No. 4, pp. 564-574, 2011a*

[5] *Mr. S.K. Pathan, Mr. S.N. Deshmukh, and Dr. R.R. Deshmukh Kerberos Authentication System – A Public Key Extension, IJRTE, Vol 1, No.2, May 2009*

[6] *http://en.wikipedia.org/wiki/Elliptic_curve_crypto graphy.*

[7] *Eman El-Emam, Magdy Koutb, Hamdy Kelash, and Osama Farag Allah, A Network Authentication Protocol Based on Kerberos, IJCSNS vol.9, no.8, Aug 2009*

[8] *Sufyan T. Faraj Al-Janabi and Mayada Abdul-salam Rasheed, " Public-Key Cryptography Enabled Kerberos Authentication", - 2011Developments in E-Systems Engineering, IEEE Computer Society*

[9] *Darrel Hankerson. Guide to Elliptic Curve Cryptography. Springer, 2004*

[10] *Certicom, Standards for Efficient Cryptography, SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0, September 2000*

[11] *Certicom, Standards for Efficient Cryptography,SEC 1: Elliptic Curve Cryptography, Version 1.0, September 2000*

[12] *Wang You-Bo, Dong Xiang-Jun, Tian Zhi-Guang, "FPGA Based Design of Elliptic Curve Cryptography Coprocessor", Third International Conference on Natural Computation (ICNC 2007), 2007, pp. 185 – 189*