Indian J. Res. Found., (2015) 1, 35-40. ISSN NO. 2454-6577

INDIAN JOURNAL OF RESEARCH FOUNDATION

www.indianjournal.net

RESEARCH ARTICLE

Roll of elliptic curve in data security

C. Sajeev*, C. Suyambulingom

Department of Computer Science & Engineering, Sathyabama University, Chenna-600119

Tamil Nadu, India

Received 20 May 2015; Accepted 27 June 2015
Available online 27 June 2015.

Abstract

Elliptic Curve Cryptography is gaining wide acceptance as an alternative to the conventional cryptosystems (DES, RSA, AES) which tend to be power hungry. Elliptic Curve ciphers require less computational power, memory and communication bandwidth giving it a clear edge over the traditional Crypto-Algorithms. This paper provides an overview of elliptic curves and their use in cryptography. The focus is on the performance advantages to be obtained in the wireless environment by using elliptic curve cryptography instead of a traditional cryptosystem like RSA. Specific applications to secure messaging and identity-based encryption are discussed.

Keywords

Computational power

Memory

Band width

Cryptography

Encryption

Introduction

Forecasters predict more than a trillion wireless users by Dec. 2012. As the wireless industry explodes, it faces a growing need for security. Applications in sectors of the economy such as healthcare, financial services, and government depend on the underlying security already available in the wired computing environment. Both for secure (authenticated, private) web transactions and for secure (signed, encrypted) messaging, a full and efficient public key infrastructure is needed. Three basic choices for public key systems are available for these applications [1-17].

RSA

Diffie-Hellman (DH) or Digital Signature Algorithm (DSA) modulo a prime p. Elliptic Curve Diffie-Hellman (ECDH) or Elliptic Curve Digital Signature Algorithm (ECDSA). RSA is a system that was published in 1978 by Rivest, Shamir, and Adleman, based on the difficulty of factoring large integers. Whitfield Diffie and Martin Hellman proposed the public key system now called Diffie-Hellman Key Exchange in 1976. DH is key agreement and DSA is signature, and they are not directly interchangeable, although they can be combined to do authenticate key agreement. Both the key exchange and digital signature algorithm are based on the difficulty of solving the discrete logarithm problem in the multiplicative group of integers modulo a prime p. Elliptic curve groups were proposed in 1985 as a substitute for the multiplicative groups modulo *p* in either the DH or DSA protocols. For the same level of security per best currently known attacks, elliptic curve-based systems can be implemented with much smaller parameters, leading to significant performance advantages. Such performance improvements are particularly important in the wireless arena where computing power, memory, and battery life of devices are more constrained. In this article we will highlight

*Corresponding author, Tel : +91 9443607744
E-mail : csajeev@live.com

Published by GM SOFTWARE

comparing their performance with RSA in the context of protocols from different standards. There are various standards bodies guiding the implementation of security protocols for the industry. Some of the organizations involved in standards activities are the Internet Engineering Task Force (IETF), American Bankers Association, International Telecommunications Union (BAITU) IEEE, and National Institute of Standards and Technology (NIST). The IETF has working groups drafting standards for S/MIME, IP Sec, and Transparent Layer Security (TLS). Using the Cryptographic Message Syntax (CMS) format, S/MIME specifies the protocols for exchanging signed encrypted messages or email and is an alternative to PGP. IPSec is needed to establish virtual private network connections, and TLS isused to establish secure browser sessions. X.509 guides the issuing of certificates on parties public keys and their management and revocation. In the last few years, working groups in each of these areas have added specifications for using elliptic curve groups through request for comment drafts. At the level of specifying the mathematical operations underlying these protocols, the X9 organization of the American Bankers Association provides the standards ANSI X9.39 for RSA and Mod *p* signatures, ANSI X9.62 for ECDSA and ANSI X9.63 for ECDH. Even more specific to elliptic curve cryptography is the IEEE P1363 published standard for describing implementation of elliptic curve operations. NIST provides a list of curves to be used, specified in FIPS 186-2, Digital Signature Standard.

The focus of this paper will be to examine the impact of using elliptic curve cryptography in S/MIME instead. We will provide a brief introduction to elliptic curves in cryptography (ECC). We will give some background on S/MIME and summarize the protocols. We will present our conclusions about the performance advantages to be obtained by using elliptic curves in the wireless environment. We will explain a new application of elliptic curves in identity based encryption that may help to launch deployment of a public key infrastructure.

Elliptic curves in cryptography

History of ECC

Elliptic curves were proposed for use as the basis for discrete logarithm-based cryptosystems almost 20 years ago, independently by Victor Miller of IBM and Neal Koblitz of the University of Washington. At that time, elliptic curves were already being used in various cryptographic contexts, suchas integer factorization and primality proving.

History of attacks

Since then, many of the top mathematicians in algorithmic number theory have tried their hand at attacking elliptic curve discrete log based cryptosystems, so far to little or no avail. Successful attacks have been found only for a few very special families of curves (*e.g.*, the Menezes-Okomoto-Vanstone attack using the Weil pairing on super singular elliptic curves). Versions of index calculus have been tried with no success. (Index calculus yields an attack on traditional mod p discrete log based cryptosystems by creating a factor base of small elements, finding relations between them, and solving a system of linear equations). Weil descent has been proposed for embedding elliptic curves in higher-dimensional Abelian varieties where attacks are known, but has not yielded a good attack on elliptic curves in general. Currently, the best known attacks on elliptic curve discrete log systems run in time proportional to the square root of the group size of the elliptic curve, using Pollard rho, Pollard kangaroo, or Baby Step Giant Step algorithms. By comparison, much more efficient attacks are known for both RSA and mod p discrete log-based cryptosystems. For RSA, the attack goes *via* factoring using the Number Field Sieve, and for mod p systems it is the index calculus attacks mentioned above.

Equivalent security levels

Currently, the system parameters for an elliptic-curve-based system can be chosen to be much smaller than the parameters for RSA or mod p systems. For example, an elliptic curve over a 163-bit field currentlygives the same level of security as a 1024- bit RSA modulus or Diffie-Hellman prime. The differ- rence becomes even more dramatic as the desired security level increases. For example, 571-bit ECC is currently equivalent in security to 15, 360- bit RSA/DH/DSA. Public key protocols are used in combination with symmetric key algorithms. The overall strength of the system is the strength of the weakest link. Recently the new symmetric key predecessor. At key lengths of 128, 192 and 256, AES has made ECC systems even AES was introduced, providing grea- ter security than its more attractive as a key agree- ment alternative **Table 1** is found in a number of the standard documents.

Table 1 Key sizes for equivalent security levels (in bits).

Symmetric	ECC	DH/DSA/RSA
80	163	1024
128	283	3072
192	409	7680
256	571	15,360

This growing difference in key bit length for equivalent security levels accounts for the performance advantages to be obtained from substituting ECC for RSA/DH/DSA in public key cryptographic protocols.

Role of groups in key exchange and signatures

The Diffie-Hellman Key Exchange and Digital Signature Algorithm can be described abstractly in the mathematical language of groups. A group is a set of elements with an operation specifying how to combine two elements to get another element of the set such that the operation satisfies certain technical properties. When we refer to mod groups where p is some large prime number, we mean the set of natural numbers less than p where the operation is multiplication taking the remainder modulo p. A secret key exchange between two parties, A and B, can be achieved publicly if a group G and a fixed group element g are agreed upon, and if each generates a random number which they keep to themselves. If A generates the random number a and broadcasts the group element ga and B generates the random number b and broadcasts the group element gb the common secret will be gab, where the notation ga means to compose the element g with itself a times in the group. The security of this protocol depends on the discrete log problem being hard to solve in the given group. The discrete log problem is given ga and g, find a.

Elliptic curve groups

For the purpose of cryptography, an elliptic curve can be thought of as being given by an affine equation of the form y2 = x3 + ax+b, where a and bare elements of a finite field with pn elements, where p is a prime larger than 3 (The equation over binary and ternary fields looks slightly different) The set of points on the curve is the collection of ordered pairs (x, y) with coordinates in the field and such that x and y satisfy the relation given by the equation defining the curve, plus an extra point that is said to be at infinity. Coordinates in a finite field also form a group, and the operation is as follows: to add two points on the curve Q1 and Q2 together, pass a straight line through them and look for the third point of intersection with the curve, R1. Then reflect the point R1 over the x-axis to get–R1, the sum of Q1 and Q2. Thus, Q1 + Q2 = –R1. The idea behind this group operation is that the three points Q1, Q2, and R1 lie on a common straight line, and the points that form the intersection of a function with the curve are considered to add up to be zero (**Fig 1**).

Elliptic curve cryptography

To implement the Diffie-Hellman Key Exchange with an

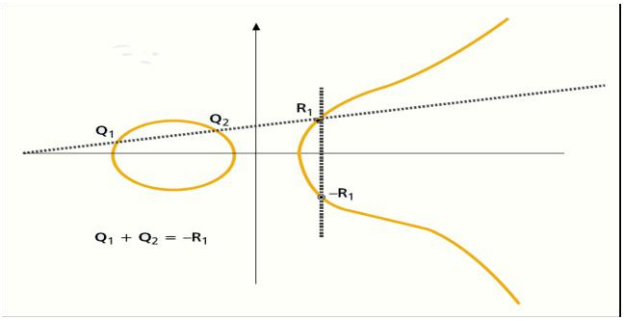

Fig 1 Group law on an elliptic curve.

elliptic curve group, many iterations of the group operation must be performed. Therefore, it is important to optimize p the implementation of the group operation. Many approaches have been explored, but choices about how to optimize the elliptic curve group operation often depend on the relative costs of operations such as multiplication and division of elements in the underlying field.

Affine *vs* projective coordinates

Suppose we represent points on an elliptic curve with affine coordinates as described above. Then to add two points Q1 = (x1, y1) and Q2 =(x2, y2), where x1 2, we first compute the slope of the line passing through them as (2 – y1)/ (x2– x1). This requires one division in the underlying finite field. Then solving for the third point of intersection of the line with the curve, we find that –R1 = (x3, y3), where x3 = 2 – x1 – x2 and y3 = (x1 – x3) – y1. So forming the sum requires 1 division, 1 squaring, and 1 multiplication in the underlying finite field (*p* when adding two affine points with distinct x-coordinates, and ignoring the cost of addition or subtraction in the field. Alternative representations for an elliptic curve and the points on it are also available. Projective and weighted projective (also called Jacobian) coordinates are some times used, especially in cases where division in the underlying field is costly. Weighted projective coordinates work with triples of coordinates (x, y, z), corresponding to the affine coordinates (x/z2, y/z3) whenever z projective coordinates is that point addition. On the elliptic curve can be done in 16 field multiplications, avoiding all field divisions.

Inversion/Multiplication ratio

Field divisions in prime fields are often reported to be roughly 80 times as costly computationally as multiplications in the field. Such a ratio would clearly indicate the use of weighted projective coordinates instead of affine coordinates. However, this ratio is obtained when taking advantage of special modular

reduction routines that can be used when the size of the underlying prime field has a particular form, called generalized Mersenne primes. For arbitrary primes, where special modular reduction is not available, field multiplication costs are higher. Taking advantage of Lehmer's method can significantly cut the cost of doing field divisions. Together these two considerations lead to the fact that for random primes, a ratio of 5 or 10 to 1 is reasonable. Under these circumstances, the use of affine coordinates is warranted.

Standard techniques for fast exponentiation

Many different fast exponentiation techniques are used to perform group exponentiations. For example, to perform binary exponentiation, express the exponent as a binary string; then for each bit in the expansion, either perform a squaring or a squaring and a multiplication with the base, depending on whether bit 0 or 1 occurs in the expansion. More sophisticated and efficient versions of the binary method have been developed. Other methods available involve:

1. Windowing: using some precomputed values of the base and different "window" sizes to break up the binary expansion of the exponent into chunks to be processed iteratively.
2. NAF: nonadjacent form of the exponent so that no two adjacent bits are both nonzero.
3. Compressible exponents.

For elliptic curves, the group operation is written as addition instead of multiplication, and in that case exponentiation is more appropriately referred to as scalar multiplication, but the same techniques apply. For elliptic curves, the "square-and multiply" technique described above is referred to as *double-and-add*. When using affine coordinates, a field multiplication can be saved each time a double and add operation is performed in a scalar multiplication, leading to a more efficient implementation of elliptic curve cryptographic protocols for general elliptic curves. When field inversions are more costly than 6 field multiplications, another technique given in is beneficial. It allows one to trade an inversion for 6 multiplications, and leads to an efficient algorithm for tripling a point on general elliptic curves. Finding further ways to improve elliptic curve scalar multiplication is an active area of research.

S/MIME overview

History

MIME stands for Multipurpose Internet Mail Extensions, a specification for formatting messages so that they can be sent over the Internet. MIME was initiated in 1992 by the IETF. S/MIME stands for Secure MIME, and provides the following security services for electronic messages:authentication, message integrity and non-repudiation of origin (using digital signatures) and privacy and data security (using encryption).

Protocol

Suppose two parties A and B wish to exchange signed encrypted messages. Assume that A and B already have their own public key/private key. pairs, and have certificates on their public keys from some common trusted certificate authority (CA). If A wants to send a message to B, A can obtain the certificate on B's public key and check its validity. Then using B's public key, A encrypts a message to B (usually just a symmetric key that subsequently acts as the content encryption key). A may include data such as a message encrypted with a symmetric key algorithm using the content encryption key. A then signs the whole message using its own private key. When B receives the data from A, B first checks the certificate on A's public key for validity. B then uses A's public key to verify A's signature on the message. B uses its own public key to decrypt the content encryption key and uses that key to decrypt the received data. To analyze the work load on each party, note that the sender and receiver must each perform three public key operations.

The sender, A, must:

* Verify 1 signature (on B's certificate)
* Perform 1 encryption (using B's public key)
* Sign message

The receiver, B, must:

* Verify 1 signature (on A's certificate)
* Verify 1 signature (on A's message)
* Perform 1 decryption (using its own private key)

Currently RSA certificates are often issued even on elliptic curve public keys. Certificates are signed only once but verified many times, and since only the CA must perform the expensive RSA signing operation using its private key, the individual parties do not incur much computational burden in performing the RSA signature verifications of the certificates. However, using an elliptic curve cryptosystem to perform steps can significantly decrease the computational burden on the individual parties.

Performance advantages

Comparison of ECC with RSA

We will start by giving some sample timings for RSA

and ECC on different platforms. In **Table 2** rows 1, 2 are taken from 1, and do not claim to be optimized, but show two different platforms and are directly comparable to RSA numbers for the same platforms. Rows 3, 4 in **Table 2** are taken from 7.

Comparison

An elliptic curve exponentiation for general curves over arbitrary prime fields is roughly 5-15 times as fast as an RSA private key operation, depending on the platform and optimizations. At the 256-bit ECC/3072-bit RSA security level the ratio has already increased to between 20 and 60, depending on optimizations. To secure a 256-bit AES key, ECC-521 can be expected to be on average 400 times faster than 15,360-bit RSA.

Impact on S/MIME

For example, if elliptic curve cryptosystems are used in the S/MIME protocol for signing and encryption, the sender will perform an ECDSA signature and an ECDH key agreement operation.

Table 2 Sample elliptic curve exponentiation timings over prime fields (in milliseconds).

Processor	M	16	19	25	38	52
	Hz	3-bit	2-bit	6-bit	4-bit	1-bit
Ultra SPARC II	450	6.1	8.7	--	--	--
Strong ARM	200	22.9	37.7	--	--	--
Pentium II	400	--	18.3	42.4	136.4	310.4
Pentium II	400	--	2.1	5.1	16.4	27.8

Table 3 Sample RSA encrypt/decrypt timings (in milliseconds).

Process or	MHz	1024-RSA_d	1024-RSAe	2048-RSA_d	2048-RSAe
Ultra SPARC II	450	32.1	1.7	205.5	6.1
StrongARM	200	188.7	10.8	1273.8	39.1
Pentium II	1	12,070	1180	--	--

instead of an RSA signature and an RSA encryption. For the purpose of mobile communication, a strong ARM 200 MHz processor can be thought of as a typical device. Without assuming an optimized version of ECC, use row 2 of the ECC table instead of row 4. On such a device, at the current minimum 1024-bit security level, the difference in the computation time for the sender per message is roughly 47 ms instead of 200 ms. For the recipient of a message, it would mean performing an ECDSA signature verification and an ECDH operation instead of an RSA signature verification and an RSA ference would be about 79 ms instead of 200 ms/message. Using the optimized versions of ECC quoted in row 4, the costs for the sender and receiver per message can be cut to roughly 8 and 12 ms. At higher security levels (e.g., for keying AES-256), RSA operations will be far too costly computationally for such a small device (roughly 45 s for sender and receiver per message), whereas ECC will cost instead roughly 56 ms for the sender per message and 84 ms for the receiver per message.

Conclusions

Over the last five years, elliptic curve cryptography has moved from being an interesting theoretical alternative to being a cutting edge technology adopted by an increasing number of companies. There are two reasons for this new development one is that ECC is no longer new and has withstood a generation of attacks;second, in the growing wireless industry, its advantages over RSA have made it an attractive security alternative.Wireless Internet mail industry leaders such as Qualcomm have embraced ECC, as well as other major companies in the wireless industry such as Motorola, Docomo, and RIM. Major computer companies such as IBM, Sun Microsystems, Microsoft, and Hewlett-Packard are all investing in ECC. The U.S. government is backing the use of ECC as well, with NSA creating the security requirements for wireless devices connecting companies such as Gemplus are also using ECC to improve their products' security. Wireless devices are rapidly becoming more dependent on security features. Such as the ability to do secure email, secure Web browsing, and virtual private networking to corporate networks, and ECC allows more efficient implementation of all of these features.

References

[1] V. Gupta, S. Gupta, S. Chang, "Performance analysis of elliptic curve cryptography for SSL, ACM Wksp. Wireless Security", Mobicom, Atlanta, GA, (2002).

[2] V. S. Miller,"Use of Elliptic Curves in Cryptography", H. C. Williams, Ed., Advances in Cryptology-CRYPTO, LNCS 18, (1985), Springer-Verlag, (1986), 417-26.

[3] N. Koblitz, "Elliptic Curve Cryptosystems,

Mathematics of Computation" , 48, (1987), 203-9.

[4] I. F. Blake, G. Seroussi, N. P. Smart, "Elliptic Curves in Cryptography", London Math. Soc., 265, Cambridge Univ. Press, (2000).

[5] J. Cowie et al., "World Wide Number Field Sieve Factoring Record: on to 512 Bits, Advances in Cryptology-ASIACRYPT '96", Kyongju, Korea, LNCS, Springer-Verlag, 1163**, (**1996), 382-94.

[6] M. Brown et al, "Software Implementation of the NIST Elliptic Curves over Prime Fields", D. Naccache, Ed., Topics in Cryptology-CT-RSA 2001, LNCS, Springer-Verlag, 2020, (2001), 250-65.

[7] J. Solinas, "Generalized Mersenne Numbers, Tech. Rep", (1999).

[8] A. J. Menezes, P. C. Van Oorschot, S. A. Vanstone, "Handbook of Applied Cryptography, RC Press Series on Discrete Mathematics and its Applications", CRC Press, (1997).

[9] D. M. Gordon, *J. Algorithms*, 27(1), (1998), 129-46.

[10] Y. Yacobi, "Discrete-log with Compressible Exponents, Advances in Cryptology CRYPTO '90", LNCS, Springer-Verlag, 537, 639-43.

[11] K. Eisentraeger et al., "Fast Elliptic Curve Weil Pairing Evaluation, M. Joye, Ed., Topics in Cryptology-CT-RSA", Springer-Verlag, 2612, (2003), 343-54.

[12] M. Ciet et al., "Trading Inversions for Multiplications in Elliptic Curve Cryptography", (2003).

[13] J. Guajardo, C. Paar, "Efficient Algorithms for Elliptic Curve Cryptosystems, B. S. Kaliski Jr., Ed., Advancesin Cryptology –CRYPTO", Springer-Verlag, (1997), 342-56.

[14] C. H. Lim, P. J. Lee,"More Flexible Exponentiation with Precomputation, Y. G. Desmedt, Ed., Advances in Cryptology CRYPTO", Springer-Verlag, (1994), 95-107.

[15] A. Shamir, "Identity-Based Crypto systems and Signature Schemes, Advances in Cryptology: Proc. CRYPTO'84", LNCS , 7, (1984), 47-53.

[16] U. Maurer, Y. Yacobi, "Non-interactive Public Key Cryptography, Advances in Cryptography -Eurocrypt' 91", Springer-Verlag, (1991), 498-507.

[17] D. Boneh, M. Franklin, "Identity-Based Encryption from the Weil Pairing, J. Kilian, Ed., Advances in Cryptology - CRYPTO", 2001, LNCS, 2139, Springer-Verlag, (2001), 213-29.