MORE SECURE ANALOGUE OF DSA OVER ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

Priyank Sharma[1], Manoj Nagar[2], Anunay Inani[3]
[1]Gurukul Institute of Engineering and Technology, Kota,
[2]Gurukul Institute of Engineering and Technology, Kota
[3]Gurukul Institute of Engineering and Technology, Kota ,

ABSTRACT

The Elliptic Curve Digital Signature Algorithm (ECDSA) is the elliptic curve analogue of the Digital Signature Algorithm (DSA). It was accepted in 1999 as an ANSI standard, and was accepted in 2000 as IEEE and NIST standards. It was also accepted in 1998 as an ISO standard, and is under consideration for inclusion in some other ISO standards. Unlike the ordinary discrete logarithm problem and the integer factorization problem, no sub exponential-time algorithm is known for the elliptic curve discrete logarithm problem. For this reason, the strength-per-key bit is substantially greater in an algorithm that uses elliptic curves. This paper describes the implementation of more secure digital signature concept over Elliptic Curve Digital Signature Algorithm.

Keywords: Integer Factorization, Discrete Logarithm Problem, Elliptic Curve Cryptography, DSA, ECDSA.

1. INTRODUCTION

Cryptography is the branch of cryptology dealing with the design of algorithms for encryption and decryption, intended to ensure the secrecy and/or authenticity of message. The DSA was proposed in August 1991 by the U.S. National Institute of Standards and Technology (NIST) and was specified in a U.S. Government Federal Information Processing Standard (FIPS 186) called the Digital Signature Standard (DSS). Its security is based on the computational intractability of the discrete logarithm

problem (DLP) in prime-order subgroups of Zp*. Digital signature schemes are designed to provide the digital counterpart to handwritten signatures (and more). Ideally, a digital signature scheme should be existentially non-forgeable under chosen message attack. The ECDSA have a smaller key size, which leads to faster computation time and reduction in processing power, storage space and bandwidth. This makes the ECDSA ideal for constrained devices such as pagers, cellular phones and

smart cards.

Digital signature schemes can be used to provide the following basic cryptographic services:

- data integrity (the assurance that data has not been altered by unauthorized or unknown means)
- data origin authentication (the assurance that the source of data is as claimed)
- non-repudiation (the assurance that an entity cannot deny previous actions or commitments)

2. CRYPTOGRAPHIC SCHEMES

2.1 Integer Factorization

In Integer factorization, given an integer n which is the product of two large primes p and q such that:

$$n = p * q$$

It is easy to calculate n for given p and q but it is computationally infeasible to determine p and q given n for large values of n.

One of the famous algorithms is RSA. The RSA Algorithm is shown below:

1. Choose two large prime numbers, p and q (1024 bits)
2. Compute n = p * q and z = (p-1) * (q-1).
3. Choose a number, e, less than n, which has no common factors (other than 1) with z.
4. Find a number, d, such that e * d -1 is exactly divisible (i.e., with no remainder) by z.

The public key is the pair of numbers (n, e), private key is the pair of numbers (n, d).

The encryption is done as follows:

$c = m^e \bmod n$

To decrypt the received cipher text message, c

$m = c^d \bmod n$

which requires the use of the private key, (n, d).

Its security depends on the difficulty of factoring the large prime numbers. The best known method for solving Integer Factorization problem is Number Field Sieve which is a sub exponential algorithm and having a running time of $\exp[1.923*(\log n)^{1/3}*(\log \log n)^{2/3}]$ [2].

2.2 Discrete Logarithm

Discrete logarithms are ordinary logarithms involving group theory. An ordinary logarithm $\log_a(b)$ is a solution of the equation $a^x = b$ over the real or complex numbers. Similarly, if g and h are elements of a finite cyclic group G then a solution x of the equation $g^x = h$ is called a discrete logarithm to the base g of h in the group G, i.e. $\log_g(h)$. A group with an operation * is defined on pairs of elements of G. The operations satisfy the following properties:

1. Closure: a * b ε G for all a, b ε G.

2. Associativity: a * (b * c) = (a * b) * c for all a, b ε G.

3. Existence of Identity: There exists an element e ε G, called the identity, such that e * a = a * e = a for all a ε G.

4. Existence of inverse: For each a ε G there is an element b ε G such that a * b = b * a = e. The element b is called the inverse of a.

Moreover, a group G is said to be abelian if a * b = b * a for all a, b ε G. The order of a group G is the number of elements in G.

The discrete logarithm problem is to find an integer x, $0 \leq x \leq n-1$, such that $g^x \equiv h \pmod p$, for given g ε Z*p of order n and given h ε Z*p. The integer x is called the discrete logarithm of h to the base g.

Digital Signature Algorithm (DSA), Diffie Hellman (DH) and ElGamal are based on discrete logarithms.

The best known method for solving Discrete Logarithm problem is Number Field Sieve which is a sub-exponential algorithm, having a running time of $\exp[1.923*(\log n)^{1/3}*(\log \log n)^{2/3}]$ [2].

2.2.1 Comparison with Integer Factorization

While the problem of computing discrete logarithms and the problem of integer factorization are distinct problems they share some properties:

- both problems are difficult (no efficient algorithms are known for non-quantum computers),
- for both problems efficient algorithms on quantum computers are known,
- algorithms from one problem are often adapted to the other, and
- difficulty of both problems has been exploited to construct various cryptographic systems.

2.2.2 Elliptic Curve Discrete Logarithm

An elliptic curve E_k, [3] defined over a field K of characteristic $\neq$ 2 or 3 is the set of solutions (x, y) ε K' to the equation $y^2 = x^3 + ax + b$

a, b ε K (where the cubic on the right has no multiple roots).

Two nonnegative integers, a and b, less than p that satisfy:

$$4a^3 + 27b^2 \pmod p \neq 0$$

Then Ep (a, b) denotes the elliptic group mod p whose elements (x, y) are pairs of nonnegative integers less than p satisfying:

$$y^2 \equiv x^3 + ax + b \pmod p$$

together with the point at infinity O.

The elliptic curve discrete logarithm problem can be stated as follows. Fix a prime p and an elliptic curve.

$$Q= xP$$

where xP represents the point P on elliptic curve added to itself x times. Then the elliptic curve discrete logarithm problem is to determine x given P and Q. It is relatively easy to calculate Q given x and P, but it is very hard to determine x given Q and P. ECC is based on ECDLP. ECDH and ECDSA are cryptographic schemes based on ECC. The best known algorithm for solving ECDLP is Pollard-Rho algorithm which is fully exponential having a running time of $\sqrt{(\Pi*n/2)}$ [2].

2.3. ELLIPTIC CURVE CRYPTOGRAPHY

Elliptic curve cryptosystems (ECC) were invented by Neal Koblitz [3] and Victor Miller [4] in 1985. They can be viewed as elliptic curve analogues of the older discrete logarithm (DL) cryptosystems in which the subgroup of Zp* is replaced by the group of points on an elliptic curve over a finite field. The mathematical basis for the security of elliptic curve cryptosystems is the computational intractability of the elliptic curve discrete logarithm problem (ECDLP) [5].

ECC is a relative of discrete logarithm cryptography. An elliptic curve E over Zp as in Figure 1 is defined in the Cartesian coordinate system by an equation of the form:

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

where a, b ε Zp, and $4a^3 + 27b^2 \neq 0 \pmod p$, together with a special point O, called the point at infinity. The set E(Zp) consists of all points (x, y), x ε Zp, y ε Zp, which satisfy the defining equation, together with O.

Each value of a and b gives a different elliptic curve. The public key is a point on the curve and the private key is a random number. The public key is obtained by multiplying the private key with a generator point G in the curve. The definition of groups and finite fields, which are fundamental for the construction of elliptic curve cryptosystem are discussed in next subsections.

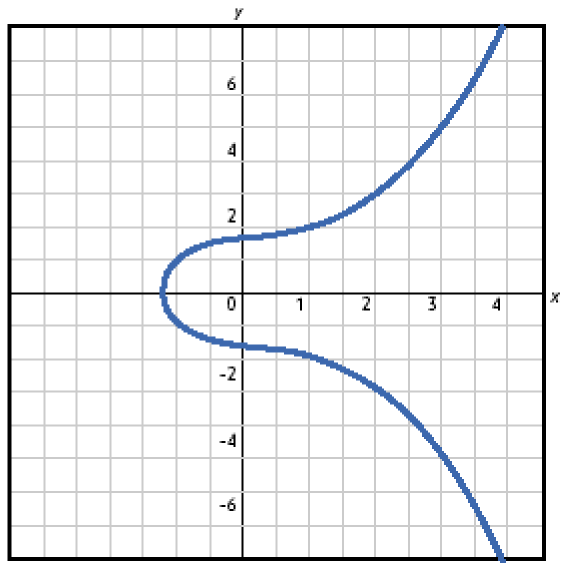

Figure 1. An Elliptic Curve[2]

3. IMPLEMENTAION AND RESULTS

The Project contains necessary modules for domain parameters generation, key generation, signature generation, and signature verification over the elliptic curve.

ECDSA has three phases, key generation, signature generation, and signature verification

Key Generation

Let W be the signatory for a message *M*. Entity W performs the following steps to generate a public and private key:

1. Select an elliptic curve *E* defined over a finite field F_p such that the number of points in $E(F_p)$ is divisible by a large prime j.
2. Select a base point, *P*, of order n such that $P \in E(F_p)$
3. Select a unique and unpredictable integer, *d*, in the interval [1, j-1]
4. Compute $Q = dP + rand(0,1)$
5. Sender W's private key is *d*
6. Sender W's public key is the combination (*E*, *P*, j, *Q*)

Signature Generation

Using W's private key, W generates the signature for message *M* using the following steps:

1. Select a unique and unpredictable integer *k* in the interval [1,j-1]
2. Compute $kP = (x_1,y_1)$, where x_1 is an integer
3. Compute $r = x_1 \bmod n$; If $r = 0$, then go to step 1
4. Compute $h = H(M)$, where *H* is the Secure Hash Algorithm (SHA-1)
5. Compute $s = k^{-1}\{h + dr\} \bmod j$; If $s = 0$, then go to step1

6. The signature of W for message M is the integer pair (r,s)

Signature Verification

The receiver Z can verify the authenticity of W's signature (r,s) for message M by performing the following:

1. Obtain signatory W's public key (E, P, j, Q)
2. Verify that values r and s are in the interval [1,j-1]
3. Compute $w = s^{-1} \bmod p$
4. Compute $h = H(M)$, where H is the same secure hash algorithm used by W
5. Compute $f_1 = hw \bmod \text{j}$
6. Compute $f_2 = rw \bmod \text{j}$
7. Compute $f_1P + f_2Q = (x_0,y_0)$
8. Compute $v = x_0 \bmod \text{j}$
9. The signature for message M is verified only if $v = r$.

Table 1: Average function runtimes by key size

Key Size	Signature Generation	Signature Verification
533	328	2753
399	230	1512
260	120	797
240	92	691
195	80	578

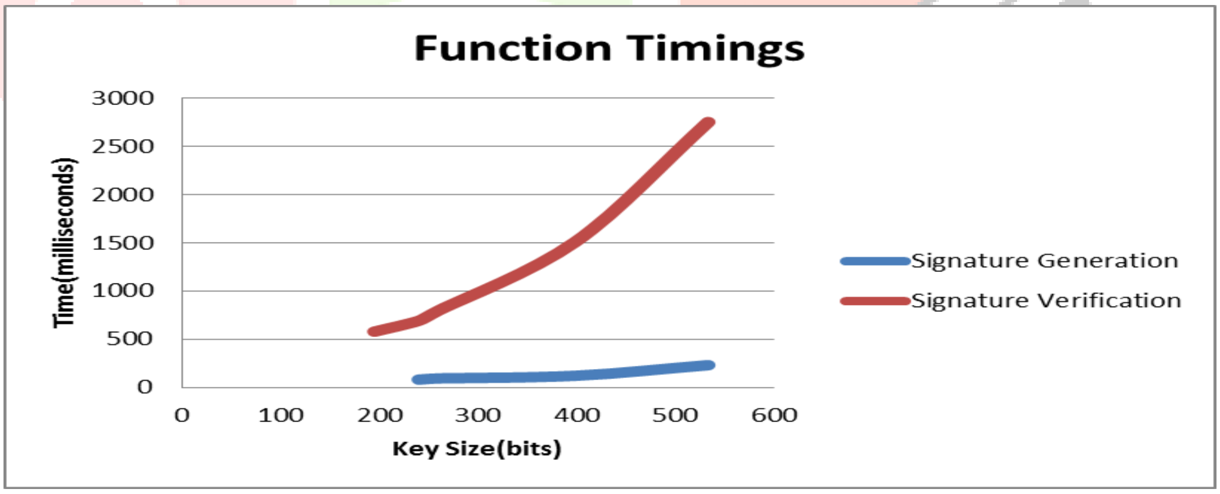

Figure 2: Depiction of Proposed method runtimes as a function of the key size

Thus, the proposed system offered remarkable advantages over other cryptographic system.

1. It provides greater security for a given key size.

2. It provides effective and compact implementations for cryptographic operations requiring smaller chips.

3. Due to smaller chips less heat generation and less power consumption.

4. It is mostly suitable for machines having low bandwidth, low computing power, less memory.

5. It has easier hardware implementations.

4. CONCLUSION

The main reason for the attractiveness of proposed system is the fact that there is no sub exponential algorithm known to solve the elliptic curve discrete logarithm problem on a properly chosen elliptic curve. Hence, it takes full exponential time to solve while the best algorithm known for solving the underlying integer factorization for RSA and discrete logarithm problem in DSA both take sub exponential time. The key generated by the implementation is highly secured and it consumes lesser
band width because of small key size used by the elliptic curves.

Significantly smaller parameters can be used in proposed system than in other competitive systems such as RSA and DSA but with equivalent levels of security.

Some benefits of having smaller key size include faster computation time and reduction in processing power, storage space and bandwidth. This makes proposed system ideal for constrained environments such as pagers, PDAs, cellular phones and smart cards. These advantages are especially important in other environments where processing power, storage space, bandwidth, or power consumption are lacking.

5. REFERENCES

[1] Vanstone, S. A., 1992. Responses to NIST's Proposal Communications of the ACM, 35, 50-52.

[2] Vanstone, S. A., 2003. Next generation security for wireless: elliptic curve cryptography. Computers and Security, vol.22, No. 5.

[3] Koblitz, N., 1987. Elliptic curve cryptosystems.Mathematics of Computation 48, 203-209.

[4] Miller, V., 1985. Use of elliptic curves in cryptography. CRYPTO 85.

[5] Certicom ECC Challenge. 2009. Certicom Research

[6] Hankerson, D., Menezes, A., Vanstone, S., 2004. Guide to Elliptic Curve Cryptography. Springer.

[7] Botes, J.J., Penzhorn, W.T., 1994. An implementation of an elliptic curve cryptosystem. Communications and Signal Processing. COMSIG-94. In Proceedings of the 1994 IEEE South African Symposium, 85 -90.

[8] An intro to Elliptical Curve Cryptography[On-Line]. Available:http://www.deviceforge.com/articles/AT4234154468.html [2010].

[9] Gupta, V., Stebila, D., Fung, S., Shantz, S.C., Gura, N., Eberle, H., 2004. Speeding up Secure Web Transactions Using Elliptic Curve Cryptography. In Proceedings of the 11th Annual Network and Distributed System Security Symposium (NDSS 2004). The Internet Society, 231-239.

[10] Raju, G.V.S., Akbani, R., 2003. Elliptic Curve Cryptosystem And Its Application. In Proceedings of the 2003 IEEE International Conference on Systems Man and Cybernetics (IEEE-SMC), 1540-1543.

[11] Johnson, D.B., Menezes, A.J., 2007. Elliptic Curve DSA (ECDSA): An Enhanced DSA. Scientific Commons.

[12] Aqeel Khalique, Kuldip Singh, Sandeep Sood " Implementation of Elliptic Curve Digital Signature Algorithm" International Journal of Computer Applications (0975 – 8887) Volume 2 – No.2, May 2010