AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

A Comparative Study on Digital Signatures Based on Elliptic Curves in High Speed Ad Hoc Networks

[1]S. Prabhadevi and [2]Dr. A.M. Natarajan

[1]*Nandha Engineering College, Anna University, Department of Computer Science and Engineering, Associate Professor, Box. 638052. Erode. India.*
[2]*Bannari Amman Institute of Technology, Anna University, Department of Computer Science and Engineering, Chief Executive, Box.638503. Sathyamangalam. India*

ARTICLE INFO

Article history:
Received 23 December 2013
Received in revised form 25 February 2014
Accepted 26 February 2014
Available online 15 March 2014

Keywords:
Digital signatures, Elliptic Curve Cryptography, Elliptic Curve Digital Signature Algorithm, Elliptic Curve Pinstov Vanstone Signature, high speed networks

ABSTRACT

Background: Cryptographic primitives need to be applied to networks which aim security and protection of data in its first place. Digital signature is one such method that can be efficiently developed for providing authenticity and integrity of the message. The progression of elliptic curve in mathematics has complemented the cryptographic field which led to the development of Elliptic Curve Cryptography in public key cryptosystems. **Results:** Although a number of elliptic curve authentication algorithms are present in literature, we discuss the elliptic curve digital signature algorithm and elliptic curve Pinstov and Vanstone signature schemes in this paper. The analysis of these signatures shows that the computational overhead is very low compared to the traditional asymmetric key algorithms**. Conclusion:** A very small signature payload is an interesting factor for their use in high speed networks.

To Cite This Article: S. Prabhadevi and Dr. A.M. Natarajan., A Comparative Study on Digital Signatures Based on Elliptic Curves in High Speed Ad Hoc Networks. ***Aust. J. Basic & Appl. Sci.,*** *8(2): 1-6, 2014*

INTRODUCTION

In 1976, Diffie-Hellman (Brown and Johnson, 2001) first described the notion of a digital signature scheme. A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document. It also ensures that the original content of the message sent remains unchanged. Digital signatures can be used to authenticate the source of messages. The importance of high confidence in sender authenticity is especially obvious in a financial context. Digital signatures are easily transportable, cannot be imitated by someone else, and can be automatically time-stamped. The ability to ensure that the original signed message arrived means that the sender cannot easily repudiate it later. A digital signature scheme typically consists of three algorithms:

- A key generation algorithm to generate *private key* and a corresponding *public key*.
- A signing algorithm that, given a message and a private key, produces a *signature*.
- A signature verifying algorithm that, given a message, public key and a signature, either accepts or rejects the message's claim to *authenticity*.

Two main properties are required. First, a signature generated from a fixed message and fixed private key should verify the authenticity of that message by using the corresponding public key. Secondly, it should be computationally infeasible to generate a valid signature without knowing the private key.

The digital signature schemes in use today can be classified according to the hard underlying mathematical problem which provides the basis for their security:

1. Integer Factorization (IF) schemes, which base their security on the intractability of the integer factorization problem. Examples of these include the RSA (Rivest *et al*., 1978) and Rabin (Rabin, 1979) signature schemes.
2. Discrete Logarithm (DL) schemes, which base their security on the intractability of the (ordinary) discrete logarithm problem in a finite field. Examples of these include the ElGamal (ElGamal, 1985), Schnorr (Schnorr, 1991), DSA (NIST, 1994), and Nyberg-Rueppel (Rueppel, 1993 and Nyberg-Ruppel, 1996) signature schemes.
3. Elliptic Curve (EC) schemes, which base their security on the intractability of the elliptic curve discrete logarithm problem.

Background:

Corresponding Author: S. Prabhadevi, Nandha Engineering College, Anna University, Department of Computer Science and Engineering, Associate Professor, Box. 638052. Erode. India.

Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 1-6

In this section we brief overview of prime field, Elliptic Curve over that field and Elliptic Curve Discrete Logarithm Problem (ECDLP).

The finite field F_p:

Let p be a prime number. The finite field F_p is comprised of the set of integers $0, 1, 2, \ldots, p - 1$ with the following arithmetic operations (Koblit, 1994, Rosen, 1986 and Menezes *et al*., 1997):

- *Addition:* If $a, b \in F_p$, then $a + b = r$, where r is the remainder when $a + b$ is divided by p and $0 \le r \le p - 1$. This is known as addition modulo p .
- *Multiplication:* If $a, b \in F_p$, then $a.b = s$, where s is the remainder when $a.b$ is divided by p and $0 \le s \le p - 1$. This is known as multiplication modulo p.
- *Inversion:* If a is a non-zero element in p, the inverse of a modulo p, denoted a^{-1}, is the unique integer $c \in F_p$ for which $a.c = 1$.

2.2 Elliptic Curve over F_p

Let p ¸ 3 be a prime number. Let $a, b \in F_p$ be such that $4a^3 + 27b^2 \neq 0$ in F_p. An elliptic curve E over F_p defined by the parameters, a and b are the set of all solutions (x, y), $x, y \in F_p$ to the equation $y^2 = x^3 + ax + b$ together with an extra point o, the point at infinity. The set of points $E (F_p)$ forms an abelian group with the following addition rules (Certicom):

- Identity: $P + O = O + P = P$, for all $P \in (E)F_p$
- Negative: If $P(x, y) \in (E) F_p$ then $(x, y) + (x, -y) = 0$. The point $(x, -y)$ is denoted as $–P$ called negative of P.
- Point addition: Let $P(x_1, y_1)$, $Q(x_2, y_2) \in (E) F_p$ then $P + Q = R \in (E) F_p$ and coordinate (x_3, y_3) of R is given by $x_3 = \lambda^2 – x_1 – x_2$ and $y_3 = \lambda(x_1 – x_3) – y_1$ where $\lambda = (y_2-y_1)/(x_2-x_1)$
- Point doubling: Let $P (x_1,y_1) \in E (F_p)$ where $P \neq -P$ then $2P = (x_3, y_3)$

where $x_3 = ((3x_1^2 + a)/2y_1)^2 – 2x_1$ and $y_3 = (3x_1^2 + a)/(2y_1 (x_1 – x_3) –y_1)$

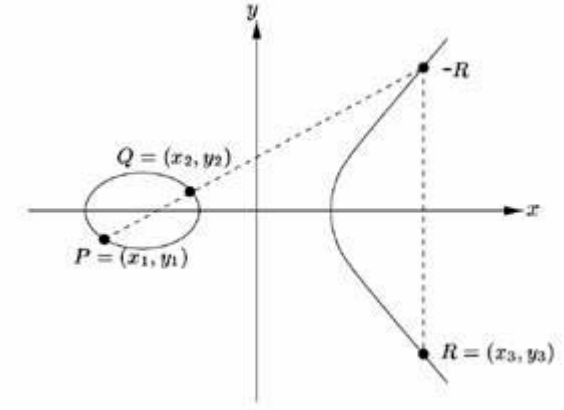

Fig. 1: Addition of two distinct elliptic curve points: $P + Q = R$.

Elliptic Curve Discrete Logarithm Problem (ECDLP):

Given an elliptic curve E over a finite field $F p$, a point $P \in (E)F_p$ of order n, and a point $Q \in <P>$, find the integer $l \in [0, n-1]$ such that $Q = lP$. The integer l is called discrete logarithm of Q to base P, denoted $l = log_P Q$ (Certicom).

Elliptic Curve Cryptography:

In 1985, Neal Koblitz (Koblitz, 1985) and Victor Miller (Miller, 1986) independently proposed using elliptic curves to design public key cryptographic systems. In the late 1990`s, ECC was standardized by a number of organizations and it started receiving commercial acceptance. Nowadays, it is mainly used in the resource constrained environments, such as ad-hoc wireless networks and mobile networks. Elliptic Curve Cryptography (ECC) can be used as an alternative mechanism for implementing public-key cryptography. The mathematical basis for the security of elliptic curve cryptosystems is the computational intractability of the elliptic curve discrete logarithm problem (ECDLP).

Since the ECDLP appears to be significantly harder than the discrete logarithm problem, the strength-per key bit is substantially greater in elliptic curve systems than in conventional discrete logarithm systems. Thus, smaller parameters can be used in ECC than with discrete logarithm systems but with equivalent levels of security. The advantages that can be gained from smaller parameters include speed (faster computations) and smaller keys and certificates. These advantages are especially important in environments where processing power, storage space, bandwidth, or power consumption is constrained.

Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 1-6

The length of a key, in bits, for a conventional encryption algorithm is a common measure of security. To attack an algorithm with a *k*-bit key it will generally require roughly (*2k - 1*) operations. Hence, to secure a public key system one would generally want to use parameters that require at least (*2k – 1)* operations to attack. The following table gives the key sizes recommended by the National Institute of Standards and Technology to protect keys used in conventional encryption algorithms like the Data Encryption Standard (DES) (DES, 1977) and Advanced Encryption Standard (AES) (AES, 2001) together with the key sizes for RSA, Diffie-Hellman and elliptic curves that are needed to provide equivalent security.

Table 1: NIST Recommended Keysizes.

Symmetric Keysize (Bits)	RSA and Diffie-Hellman Keysize (Bits)	ECC Keysize (Bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

The reason is that there exist sub-exponential time algorithms for factoring and discrete logarithm problem, whilst only exponential-time algorithms for ECDLP.

Security is not the only attractive feature of elliptic curve cryptography. Elliptic curve cryptosystems also are more computationally efficient than the first generation public key systems, RSA and Diffie-Hellman. The following table shows the ratio of DH computation versus EC computation for each of the key sizes listed in Table 1. In channel-constrained environments, elliptic curves offer a much better solution than first generation public key systems like Diffie-Hellman.

Table 2: Relative Computation Costs of Diffie-Hellman and Elliptic Curves.

Security Level (Bits)	Ratio of DH Cost : ECC Cost
80	3:1
112	6:1
128	10:1
192	32:1
256	64:1

Elliptic Curve Digital Signature Algorithm (ECDSA):

The ECDSA signature scheme is a variant of the Digital Signature Algorithm (DSA) which uses elliptic curve cryptography. The domain parameters for ECDSA consist of a suitably chosen elliptic curve *E* defined over a finite field F_p of characteristic *P*, and a *base point* $G \in E(F_p)$. Domain parameters may either be shared by a group of entities, or specific to a single user. Also choose two field elements *a* and *b* in which define the equation of the elliptic curve *E* over F_p (i.e., $y^2 = x^3 + ax + b$ in the case $p>3$)

An ECDSA key pair is associated with a particular set of EC domain parameters. The public key is a random multiple of the base point, while the private key is the integer used to generate the multiple.

ECDSA Key Pair Generation- Each entity in the network does the following:

1. Select a random or pseudorandom integer, in the interval *[1, n-1]*
2. Compute $Q = dG$
3. A's public key is *Q*. A's private key is *d*

ECDSA Signature Generation- To sign a message, an entity A with associated key pair *(d, G)* does the following:

1. Select a random or pseudorandom integer *k*, $1 \leq k \leq n-1$
2. Compute $kG = (x_1, y_1)$ and convert x_1 to an integer X_1
3. Compute $x_1 \bmod n$. If $r = 0$, then go to step 1.
4. Compute $k^{-1} \bmod n$
5. Compute SHA-1 *(m)* and convert this bit string to an integer *e*
6. Compute $s = k^{-1} (e + dr) \bmod n$. If $s = 0$ then go to step 1.

A's signature for the message *m* is *(r, s)*.

ECDSA Signature Verification- To verify A's signature *(r, s)* on *m*, B obtains an authentic copy of A's domain parameters and associated public key *Q*. It is recommended that B also validates the domain parameters and associated public key *Q*. B then does the following:

1. Verify that *r* and *s* are integers in the interval *[1, n-1]*.
2. Compute SHA-1*(m)* and convert this bit string to an integer *e*
3. Compute $w = s^{-1} \bmod n$
4. Compute $u_1 = ew \bmod n$ and $u_2 = rw \bmod n$
5. Compute $X = u_1G + u_2Q$

Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 1-6

6. If $X = 0$, then reject the signature. Otherwise, convert the x-coordinate x_1 of X to an integer X_1 and compute $v = X_1 \bmod n$
7. Accept the signature if and only if $v = r$

Elliptic Curve Pinstov Vanstone Signature (ECPVS):

ECPVS is an elliptic curve variant of Nyberg-Rueppel signatures. ECPVS is standardized in ANSI, 2009 and IEEE, 2004. It has three distinct advantages over ECDSA when it comes to constrained environments.

The first being that it allows for smaller signature sizes by the incorporation of part of the message into a signature field.

The second is a simplification of the integer arithmetic. The signing transformation does not require a modular inverse, improving both code size and computational performance. The verification transformation requires no integer arithmetic and so also removes a modular inverse and modular multiplies in the verification transformation.

The third is that it is a Schnorr signature scheme which loosens the collision resistance requirement on the underlying hash function. A second consequence is a performance increase in the signature verification, where a scalar multiply with a 256-bit integer (ECDSA) is replaced by a 128-bit integer (ECPVS).

We assume all parties possess the domain parameters for the elliptic curve and that the public keys of signers are validated. ECPVS uses encoding and decoding routines to process signatures.

Table 3 describes the key size reduction with ECPVS signatures. The use of ECPVS scheme in sensor networks has been shown in Wang and Li, 2009 and Amir *et al*., 2008. In our description we assume both parties have a common set of domain parameters, i.e., an elliptic curve group of order n, generated by a point G, a suitable key derivation function, denoted *KDF*, and hash function, denoted *HASH*, and a symmetric key encryption function, denoted ENC_K , with associated decryption function DEC_K. Complete details are left to the standards referenced above. The signer generates a key pair by choosing at random a private key d from $[1, n-1]$, and computing the public key as $Q = dG$.

Table 3: Bit Strength to primitive sizes (in bytes).

Cryptographic Strength	ECPVS	ECDSA	RSA
64	14	28	64
80	20	40	128
112	28	56	256
128	32	64	384
192	48	96	960
256	64	128	1920

Unlike other types of signature schemes, ECPVS relies on certain characteristics of the recoverable message to determine if the signature is valid. These characteristics, called redundancy, can be inherent in the message (for example, ASCII or all numeric), or can be added in the form of padding. For example, if the recoverable message resembles a random string, then there is no way for the verifier to tell if the signature is valid. Padding can be added to the recoverable message to increase the redundancy so the verifier can validate a signature with high confidence. IEEE 1363a (IEEE, 2004) specifies the padding to be between 1 to 255 bytes. In general, the total redundancy should be half the subgroup order of the elliptic curve, or half the hash function output length. If you are unsure of the redundancy in the recoverable message, use a pad length that equals the total required redundancy.

ECPVS Signature Generation

Input:
1. Private key of the signer: d
2. The visible message part: V
3. The recoverable message part M (with intrinsic redundancy)
4. Optional padding added to M

Action:
1. Generate a random value k in $[1, n-1]$
2. Compute $R = kG$
3. Compute $K = KDF(R)$
4. Compute $r = ENC_K(PAD(M))$
5. Compute $s = k + HASH(r \,||\, V)d \bmod n$

Output:
1. Recovery part r
2. Signature part s
3. Visible part V

Australian Journal of Basic and Applied Sciences, 8(2) February 2014, Pages: 1-6

ECPVS Verify and Recover
Input:
1. Public key of the signer: *Q*
2. The visible message part: *V*
3. The recovery part: *r*
4. The signature part: *s*
5. Optional amount of padding

Action:
1. Compute $R = sG - HASH(r \| V)Q$
2. Compute $K = KDF(R)$
3. Compute $M = UNPAD(DEC_K(R))$

{Includes check for padding correctness}
4. Check intrinsic redundancy of *M*

Output:
1. Recovered message part: *M*

Authenticity of the signature:

The padding and redundancy checks in the verification step are required for security, since it should be difficult to create a cipher text that decrypts to a chosen message, as an attacker can use this to create a forgery. By requiring the padded message to have sufficient redundancy, it should be infeasible to find such a cipher text.

The amount of padding depends on the message. For instance, if we know the message is a URL that starts with "http://www.", where each character is encoded with 8 bits, we have 11*8 = 88 bits of redundancy. This is 8 more bits than enough at the 80-bit security level, but 40 bits short at the 128-bit level (so we'd need to pad with 40 bits if 128 bits of security is desired). In general the number of bits of redundancy should be equal to the security level *b*. So if the message has *r* bits of redundancy we need to pad the message with *b* - *r* bits.

Brown and Johnson, 2001 enunciates a complete security analysis of the PV signature scheme, including details of the redundancy requirement

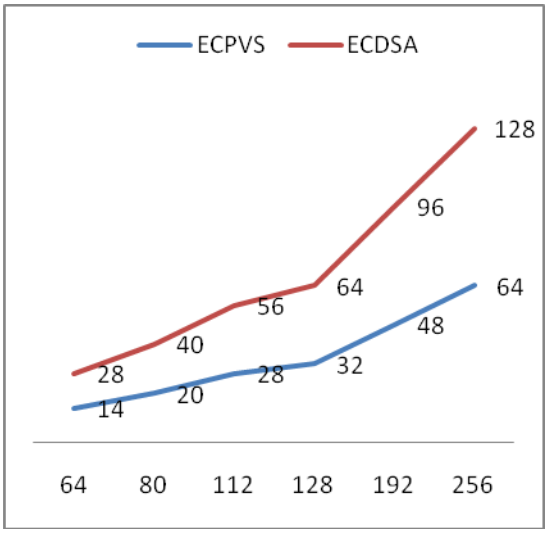

Fig. 2: Cryptographic Bit Strength Comparison.

The figure 2 shows the comparison of ECPVS and ECDSA key sizes with the standard cryptographic bit strength required. The size of an ECPVS signature is a function of the padded recoverable message part r plus the key size (since the signature value s is as long as the key). The signature expands depending on the size of the recoverable part of the message *M*. For example at ECPVS P192 with a padded recoverable message part of 50 bytes has total signature length of 74 bytes.

In the keyed version of ECPVS, recall that there is a third part to the message, which may only be recovered by a chosen party, say Bob. The signer derives two symmetric keys. The first is computed in the same way above, in unkeyed ECPVS. The second key is computed $K_2 = KDF(kB)$, where *k* is the ephemeral random value chosen by the signer, and *B* is Bob's public key. This is essentially a non-interactive Diffie-Hellman key agreement between Bob and the signer, with the signer using the ephemeral key pair *(k, R)*. The second key k_2 is used to encrypt the confidential part of the message.

The size of a keyed ECPVS signature is a function of the padded recoverable message part plus the encrypted part plus the key size. The signature expands depending on the size of the recoverable part of the message *M* and the encrypted part. For example, ECPVS P192 with a padded recoverable message part of 50 bytes and an additional 20 bytes for the encrypted part. The total signature is 94 bytes.

The ECPV signature schemes are straightforward to implement. They leverage the same primitives used in ECDSA.

Conclusion:

The article has outlined two digital signature schemes based on the elliptic curve over finite fields. The public key cryptography based on the traditional Digital Signature algorithm can be replaced with the much efficient smaller key size ECDSA scheme. With smaller signatures it is possible to implement this authentication mechanism for high speed networks. For the same amount of cryptographic strength, the key size was much smaller in ECPVS. Hence ECPVS can be used for bandwidth constraint environments. Since ECPVS has confidential and visible part of the message they can be employed in financial applications over wireless ad hoc networks.

REFERENCES

Amin, F., A.H. Jahangir and H. Rasifard, 2008. Analysis of Public-Key Cryptography for Wireless Sensor Networks Security. World Academy of Science, Engineering and Technology, 17.

Announcing the Advanced Encryption Standard (AES), 2001. Federal Information Processing Standards Publication 197. National Institute of Standards and Technology (NIST).

ANSI X9.92.1, 2009. Public Key Cryptography for the Financial Services Industry – Digital Signature Algorithms giving Partial Message Recovery Part I: Elliptic Curve Pinstov Vanstone Signature (ECPVS).

Brown, D.L. and D.B. Johnson, 2001. Formal Security Proofs for a Signature Scheme with Partial Message Recovery. Topics in Cryptology – RSA. DOI /10.1007/3-540-45353-9_11.

Certicom. ECC Challenge and the Elliptic Curve Cryptosystem, available http://www.certicom.com/index.php.

Data Encryption Standard (DES), 1977. Federal Information Processing Standards Publication.

Diffie, W. and M. Hellman, 1976. New Directions in Cryptography. IEEE Transactions on Information Theory, 22: 644-654.

El-Gamal, T., 1985. A Public Key Cryptosystem and A Signature Scheme Based On Discrete Logarithms. IEEE Transactions on Information Theory, 31: 469-472.

IEEE 1363-A, 2004. Draft available at http://tools.ietf.org/html/draft-campagna-Suitee-02#ref-IEEE1363-A.

Koblit, N., 1994. A course in Number Theory and Cryptography. 2nd edition Springer-Verlag.

Koblitz, N., 1987. Elliptic Curve Cryptosystems. Mathematics of Computation, 48: 203-209.

Menezes, A., P.C. Van Oorschot and S.A. Vanstone, 1997. Handbook of Applied Cryptography. CRC Press.

Miller, V., 1986. Use of Elliptic Curves in Cryptography. Advances in Cryptology. CRYPTO ’85, LNCS, 218(483): 417-426.

National Institute of Standards and Technology, 1994. Digital Signature Standard. FIPS Publication, (186).

Nyberg, K. and R. Rueppel, 1996. Message Recovery For Signature Schemes Based on the Discrete Logarithm Problem. Designs, Codes and Cryptography, 7: 61-81.

Rabin, M.O., 1979. Digitalized Signatures and Public-key Functions as Intractable as Factorization. MIT/LCS/TR-212, MIT Laboratory for Computer Science.

Rivest, R., A. Shamir and L. Adleman, 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Communications of the ACM, 21: 120-126.

Rosen, K.H., 1986. Elementary Number Theory in Science and Communication. 2nd edition, Springer-Verlag.

Rueppel, R., 1993. A New Signature Scheme Based on the DSA Giving Message Recovery. ACM Conference on Computer and Communications Security, 58-61.

Schnorr, C., 1991. Efficient Signature Generation by Smart Cards. Journal of Cryptology, 4: 161-174.

Wang, H. and Q. Li, 2009. Achieving Robust Message Authentication in Sensor Networks: A Public-Key Based Approach Wireless Networks. Springer Science + Business Media, LLC. DOI 10.1007/s11276-009-0184-z.