Internet of Things and Cloud Computing
2015; 3(3): 33-41
Published online October 11, 2015 (http://www.sciencepublishinggroup.com/j/iotcc)
doi: 10.11648/j.iotcc.s.2015030601.14
ISSN: 2376-7715 (Print); ISSN: 2376-7731 (Online)

Research on Innovating and Applying Cryptography Algorithms for Security Routing in Service Based Routing

Nguyen Thanh Long[1], Nguyen Duc Thuy[2], Pham Huy Hoang[3]

[1]Informatic Center of HaNoi Telecommunications, Hoan Kiem, HaNoi, VietNam
[2]Post and Telecommunications Institute, Nghia Tan, Cau Giay, HaNoi, VietNam
[3]Ha Noi University of Science Technology, HaNoi, VietNam

Email address:
ntlptpm1@yahoo.com (N. T. Long), nguyenducthuy07@gmail.com (N. D. Thuy), hoangph@soict.hut.edu.vn (P. H. Hoang)

To cite this article:
Nguyen Thanh Long, Nguyen Duc Thuy, Pham Huy Hoang. Research on Innovating and Applying Cryptography Algorithms for Security Routing in Service Based Routing. *Internet of Things and Cloud Computing*. Vol. 3, No. 3, 2015, pp. 33-41.
doi: 10.11648/j.iotcc.s.2015030601.14

Abstract: In Cryptography, there are two kinds of methods for encryption and decryption. They are symmetric cryptography and asymmetric cryptography. In symmetric cryptography, it uses a single key for encryption and decryption. In asymmetric cryptography, it uses both public key and private key for encryption and decryption. In MANET (mobile ad-hoc network) every node usually moves and has limited energy, so network link is not stable and has low bandwidth. Therefore it is very carefully to choose the cryptography for ad-hoc network or MANET. If the transaction is very important, for example online banking, can use an asymmetric algorithm. So Elliptic Curve Cryptography is right choice. For data and control traffic, the symmetric cryptography can be the good selection, especially stream ciphers are very comfortable. We use Elliptic Curve Cryptography (ECC) with its arithmetic operations on finite fields to build public - key cryptographic schemes consisting of: i) signature schemes; ii) encryption schemes; and iii) key agreement schemes. In key management, public key cryptography is used to distribute the secret keys used in other cryptographic algorithms (for example DES). For digital signatures, public key cryptography is used to authenticate the origin of data and protect the integrity of that data. Early public key systems are secure based on difficulty of factoring a large integer composed by two or more large prime integers. With Elliptic Curve based protocol, its security is based on assuming that is difficult to find the discrete logarithm of a random point on Elliptic Curve with respect to a publicly known base point. The size of EC determines the level of difficulty of the problem. If comparing to RSA, with the same level of security, RSA has to use larger public key, for example ECC of 256 bits public key is with the same level of security as 3072 bits public key RSA. To use RSA or Diffie-Hellman to protect 128-bit AES keys one should use 3072-bit parameters: three times the size in use throughout the Internet today. The equivalent key size for elliptic curves is only 256 bits.

Keywords: Elliptic Curve, Cryptography, Signature, Scheme, Key Agreement

1. Overview

1.1. Overview of Some Encryption Methods

Public-key cryptography is based on the intractability of certain mathematical problems. Early public-key systems are secure assuming that it is difficult to factor a large integer composed of two or more large prime factors. For elliptic-curve-based protocols, it is assumed that finding the discrete logarithm of a random elliptic curve element with respect to a publicly known base point is infeasible. The size of the elliptic curve determines the difficulty of the problem. The primary benefit promised by ECC is a smaller key size, reducing storage and transmission requirements, that an elliptic curve group could provide the same level of security afforded by an RSA-based system with a large modulus and correspondingly larger key, a 256-bit ECC public key should provide comparable security to a 3072-bit RSA public key.

1.2. The Group Law

By adding a “point at infinity”, we obtain the projective version of this curve. If P and Q are two points on the curve, then we can uniquely describe a third point which is the intersection of the curve with the line through P and Q. If the line is tangent to the curve at a point, then that point is counted twice; and if the line is parallel to the y-axis, we define the third point as the point "at infinity". Exactly one of these conditions then holds for any pair of points on an elliptic curve.

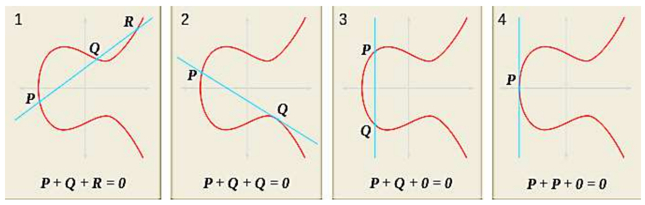


Fig. 1. *Examples of intersection of a pair of points on Elliptic curve.*

1.3. Group Operation

It is then possible to introduce a group operation, +, on the curve with the following properties: i) consider the point at infinity to be 0, the identity of the group; and ii) if a straight line intersects the curve at the points P, Q and R, then we require that:

$$P + Q + R = 0 \text{ , in the group.} \tag{1}$$

One can check that this turns the curve into an abelian group, and thus into an abelian variety. It can be shown that the set of K-rational points (including the point at infinity) forms a subgroup of this group. If the curve is denoted by E, then this subgroup is often written as E(K).

2. ECC Concepts and Definitions

Elliptical curve cryptography (ECC) is a public key encryption technique based on elliptic curve theory that can be used to create faster, smaller, and more efficient cryptographic keys. Because ECC helps to establish equivalent security with lower computing power and battery resource usage, it is becoming widely used for mobile applications [9]. Multiplying a point on the curve by a number will produce another point on the curve, but it is very difficult to find what number was used, even if you know the original point and the result. Equations are based on elliptic curves that have a characteristic which is very valuable for cryptography purposes: they are relatively easy to perform, and extremely difficult to reverse. So it is very important to apply ECC for securing data transmission in MANET.

Elliptic Curve (EC) is based on the formula:

$$y^2 = x^3 + ax + b \tag{2}$$

With its arithmetic operations on finite fields and condition $4a^3 + 27b^2 \neq 0$ in its determined domain. In particular, ECC can be based on prime number field $F_p$with value domain of results of modulo by p operator which are less than for polynomial field F_{2^m} with value domain of results of polynomial modulo based on an prime polynomial f(x) has max degree of x is less than 2^m. In order to use ECC We have to determine its domain parameters, these parameters define the value field to which values of x, y coordinates of any point on Elliptic Curve belong. Some other parameters define some initial conditions to build ECC.

The most important operations in any ECC based techniques such as key exchange or encryption is the scalar multiplication. This point scalar multiplication is achieved by repeated point addition and doubling [2]. [2] proposed a novel technique which can reduce the time required in scalar multiplication.

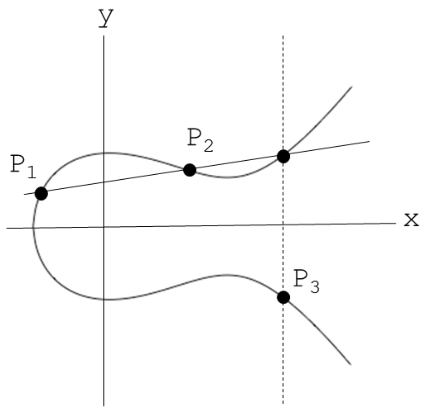


Fig. 2. *Example of Elliptic Curve Point addition: $P_3 = P_1 + P_2$.*

As Fig 2, that shows the principle to establish the set of points on ECC graph. From any two points$P_1(x_1, y_1)$ and $P_2(x_2, y_2)$ of ECC, assume the line that is formed by these point cuts ECC at P_3', then the point $P_3(x_3, y_3)$ is symmetric with P_3' through X axis is called the sum of these two points that is calculated by following formulas:

$$x_3 = \alpha^2 - x_1 - x_2 \tag{3}$$

$$y_3 = \alpha(x_1 - x_3) - y_1 \tag{4}$$

$$\alpha = \frac{y_2 - y_1}{x_2 - x_1}, \text{ if } P_1 \neq P_2 \tag{5}$$

$$\alpha = \frac{3x^2 + a}{2y_1}, \text{ if } P_1 = P_2 \tag{6}$$

2.1. ECC Based on Prime F_p Fields

In prime F_p fields, the formula (2) can be expressed as:

$$y^2 \equiv x^3 + ax + b \pmod{p} \tag{7}$$

with p is a large prime number.

To define this kind of ECC, We have to determine its domain parameters in a six tuple T:

T = {p, a, b, G, n, h}, with p is a prime number specifying finite field F_p, two parameters a and b $\in F_p$ specifying an elliptic curve $E(F_p)$defined by equation (7), a base point $G=(x_G,y_G)$ on $E(F_p)$, a prime n is order of G and h is cofactor defined by $h=\frac{\#E(F_p)}{n}$.

2.2. ECC Based on Polynomial F_{2^m} Field

Domain parameters of Elliptic Curve on F_{2^m} fieldcontain7 items in a septuple T:

T=(m, f(x), a, b, G, n, h). This septuplet consists of an integer m to determine finite field F_{2^m}, f(x) is an irreducible binary polynomial with degree of m, other parameters as defined in (2.1).

3. Some Strategies to Improve ECC Performance

So in order to reduce time and cost to calculate k*P, convert k into binary form, that is a string with length n of '1' and '0' digits: $k = x_1 x_2 \dots x_n$. Can use following algorithm to get result point Q:

```
Function Point Calculate_ECC_Point(integer k, Point P) {
String k_binary = Convert_to_Binary(k); /*k=(k_{m-1},k_{m-2},…,k_0)_2, k_{m-1}=1*/
Point[] Q_arr = new Point[k_binary.Length-1];
Point Q = null, Q_arr [0]=P;
For(integer i=0; i<k_binary.Length; i++){
If(substr(k_binary,i,1)==1)
If(Q!=null) Q=Q+Q_arr[i+1]; else Q=Q_arr[i];
If(i!=(k_binary.Length-1)) Q_arr[i+1]=2tmp_p;
}
Return Q;
}
```

K=7=(111)2, Q=7P=2(2(P)+P)+P, 6=(110)2, Q=6P=2(2(P)+P)

This algorithm has complexity depended on the length of k in binary form that is approximate O(|k_binary_form|) in bad situation. This algorithm can be executed parallel by two processes of multi processors architecture as in one process for calculating array Q_arr and other process for calculating Q momentarily.

According to [2] the cost to calculate an addition of two points is higher than doubling a point, it has proposed some methods for increasing the rate to execute and reducing the cost:

3.1. Hamming Weight Reduction Strategy

Hamming weight concept [2]: The Hamming weight of a binary string is the number of symbols that are diverse from the "0" symbol. So in general Hamming weight is the number of "1" bits presence in the binary sequence.

Purpose: Change the way to calculate a series of some addition operations to reduce cost, assume P is a point on ECC, k is an integer, calculate point Q on ECC satisfies the formula: Q = k*P. Based on some principles:

$$k*P = P+P+\dots+P;$$

$$P+P = 2P;$$

Point doubling operation cost is less than point addition cost: cost(P+P)>cost(2P);

$$P-Q = P+(-Q);$$

For example: k=1111111111112=10000000000002-1, if use k in the form: 10000000000002-1, it requires only one addition operation. So, in some situation, the complexity is constant.

3.2. Pairing Strategy

In this strategy, assume k can be converted to binary form: $k=(k_{m-1}, k_{m-2}, \dots, k_0)2$,

$k_{m-1}=1$, consider pairing size (Psize) from 2 to m, there are $2^{(P_{size}-1)}$-1pre-computations that are: P, 2P, …, $[2^{(P_{size}-1)}$-1]P. Calculate Q by the following algorithm:

```
Function Point Calculate K P(integer k, Point P, P_size){
String k_binary_form = Convert_to_Binary(k);
String tmp_str;
Point Q;
Q=Convert_to_integer(substr(k_binary_form,0, P_size))*P;
For(integer i=0; i<k_binary_form. Length/ P_size; i++){
tmp_str = substr(k_binary_form,i*P_size, P_size);
For(integer j=0; j<tmp_str.length; j++)
Q=2Q;
Q=Q+ Convert_to_integer(tmp_str)*P;
}
If(k_binary_form. Length> i*P_size){
Integer cs = i* P_size;
While(cs< k_binary_form. Length){
Q=2Q;
cs=cs+1;
}
Q=Q+ Convert_to_integer(substr(k_binary_form, i* P_size))*P;
}
Return Q;
}
```

3.3. Complement Based Arithmetic Strategy of 1 Digit

Assume Q=kP, Q, P are two points on ECC, k is a integer, in the case k has hamming weight is high, so if applying strategy (a), it requires many addition and doubling operations. So for reducing the operations, convert k into another form based on the following formula:

With k in binary form: $k = (k_{m-1}, k_{m-2}, \dots, k_0)_2$

Calculate the complement of k by the formula:

$$\mathrm{Comp_k}=(2^m - 1) - k \quad (8)$$

Calculate k based on complement of

$$k\text{: } k = (2^m - 1) - \mathrm{Comp_k} \quad (9)$$

3.4. Soft Computing Based Arithmetic Strategy

In this strategy use some rules for calculating multiplication of two integer P and Q, calculate result R modulo m:

$$R = (PQ) \bmod R \quad (10)$$

instead of directly computing, R can be calculated by following rules:

- Convert P and Q into binary form: P=(Pm-1, Pm-2, …, P0)2, Q=(Qm-1, Qm-2, …, Q0)2
- Scan each digit d in the binary form of P, calculate multiplication of d and Q by the following rules:
- If d='0' then apply the rule: callp_d,w_d is bit position and bit position weight of d,

$$\begin{cases} w_d = -2^{p_d} \text{ if } p_d = \overline{0..2} \\ w_d = 2^{p_d} \text{ if } p_d = \overline{3..\infty} \end{cases} \tag{11}$$

then multiply the result with multiples of m to reduce common result.

- If d != '0' normal calculate.

3.5. Soft Computing Based Controller

Based on Pairing strategy, for estimating value of Window size (as in (3.2) it is pairing size P_{size}) based on fuzzy logic controller with three input parameters that are:

- Storage capacity, it has three states: i) Low; ii) Rather low; iii) Average; iv) Rather high; v) High.
- Pre-Computing: the second parameter is pre-computing that is pre-computing working load for calculating result. This parameter has three states: i) Min; ii) Average; iii) Max;
- Doubling: the third parameter is doubling that are doubling operations for calculating result. This parameter has three states: i) Min; ii) Average; iii) Max;

The fuzzy controller estimates based on these input parameters that returns one output that is window size.

Following is diagram of this fuzzy logic controller:

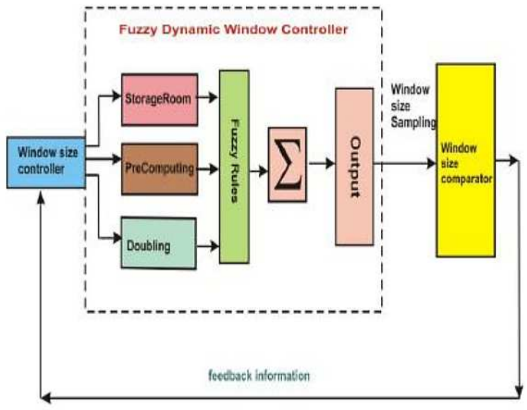


Fig. 3. *Fuzzy logic controller for calculating optimal pairing size (P_{size}) [2].*

4. ECC Based Authenticated Key Agreement Schemes

4.1. Elliptic Curve Based on Diffie – Hellman Scheme

Use Elliptic Curve based on Diffie – Hellman (ECDH) scheme to exchange shared secret key between two nodes in each data communication session by using a symmetric key encryption such as AES (Advanced Encryption Standard). This scheme is called key establishment protocol.

4.2. Assumptions

Assume that A and B are communicating shared secret key with each other. A and B can use ECC, in case of prime integer with domain parameters are:(p, a, b, G, n, h) or binary case with domain parameters are: (m, f(x), a, b, G, n, h).Each node has a key pair suitable for ECC that consists of a private key d is chosen randomly in the range [1, n-1], and a public key Q is calculated by formula: Q=d*G. Therefore A and B have key pair (d_A, Q_A) and (d_B, Q_B) respectively.

4.2.2. Protocol Structure

Each node must store other node's public key, this means that they have communicated their public keys before. Thus A, B will receive other public key Q_B and Q_A respectively. In each node, it will calculate secret key K by following formula:

Nodes A, B have to compute a pair of two items of a point of ECC graph: (x_K, y_K), in which $x_K$is shared secret key of A and B, it is calculated by A based on: $x_K = d_A{*}Q_B$ and by B: $x_K = d_B{*}Q_A$. We have $d_A{*}Q_B = d_B{*}Q_A = d_A{*}d_B{*}G$.

4.2.3. Public key Communicating Methods

Public key can be static or temporary. In static case, it is packaged in a certificate that is authenticated by a known certificate authorizer, so it is not attacked by Man - in - middle attacker.

4.2.4. Evaluation

In case of temporary public key, if a node C is an attacker, it can impersonate node A or B then communicate public key with another node. Thus A or B will transfer data over network, but only C can receive data. In case of static public key by using certificate, it is still attacked by attacker which can solve the Elliptic Curve problem. For more security we can use following key exchange protocols.

4.3. MQV (Menezes-Qu-Vanstone) Scheme

This protocol is also based on Elliptic Curve algorithm, but it use more complex formula to calculate shared key.

4.3.1. Assumptions

Assume two node$N_1$and N_2 need to exchange shared key, N_1 has a key pair (A, a) with A is public key and a is private key. N_2 has a key pair (B, b) in which B is public key and b is private key. Assume R=(x, y) is a point on an Elliptic Curve, we denote $\overline{R}$ by formula: $\overline{R} = (x \bmod 2^L) + 2^L$, in which $L = \left\lceil \frac{\lfloor \log_2 n \rfloor + 1}{2} \right\rceil$ andn is the order of the use generator point P. We can find that L is first L bit of the x coordinate of R.

4.3.2. Protocol Structure

Use 6 steps algorithm below to create shared key in each node N_1 and N_2:

Table 1. *The steps for protocol structure.*

Step	Operation
1	N_1 generates a key pair (X, x) by choosing private key x randomly in range [1, n-1], public key X = x*P with P is a point on an EC.
2	N_2 generates a key pair (Y, y) by the same way as N_1.
3	N_1 calculates $S_{N_1} = (x + \overline{X}a)$and sends its public key X to N_2.
4	N_2 calculates $S_{N_2} = (y + \overline{Y}b)$ and sends its public key Y to N_1.
5	N_1 calculates shared key K by formula $K{=}h.S_{N_1}(Y+\overline{Y}B)$, N_2 calculates $K{=}h.S_{N_2}(X+\overline{X}A)$ in which h is the cofactor.
6	K is used to communicate secretly by $N_1$and N_2.

4.3.3. Correctness and Evaluation

We have at N_1:

$K=h.S_{N_1}(Y+\bar{Y}B)=h.S_{N_1}(yP+\bar{Y}bP)=h.S_{N_1}(y+\bar{Y}b)P=h.S_{N_1}S_{N_2}P$,

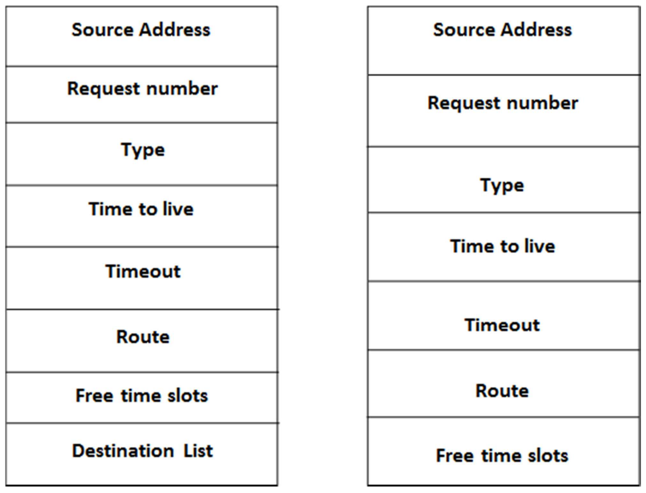
and at N_2 : $K=h.S_{N_2}(X+\bar{X}A)=h.S_{N_2}(xP+\bar{X}aP)=h.S_{N_2}(x+\bar{X}a)P=h.S_{N_2}S_{N_1}P$.

Thus the same key is created at N_1 and N_2.

This protocol is similar to the above protocol in that each node has to exchange its public key with each other. But shared key is calculated by more complex formula. So it is more secure for using in symmetric key encryption algorithm. All key – agreement messages are signed by ECDSA as presented below for data integrity, data origin authentication, and non-repudiation.

5. Key Renewing Scheme Based on Message Pair of RREQ and RREP

Use ECC based Key – Agreement schemes above require large processing capability for generating and calculating big prime number or polynomial with high degree. So it is not suitable for MANET nodes, in MANET every node usually has limited power and load, processing capability. With a pair of Route Request (RREQ) and Route Reply (RREP) messages:



a) Route request message b) Route reply message

Fig. 4. Structure of message pair RREQ and RREP [1].

We can send a request by broadcasting a RREQ message to agree a private shared key between the sender and receiver nodes by following method:

Create a new type of RREQ: type = 3, for both functions: route request and key request, create this kind of RREQ message to send to receiver node by broadcast protocol;

After regulated time interval, this RREQ message will reach the receiver through multiple different paths. By each path which receiver receives this RREQ message:

The receiver will create its own key which is encrypted by Advanced Encryption Standard (AES) algorithm or stream cipher algorithm;

Make a RREP message to contain all fields as specified by [1] in addition with an encrypted key and send it back to the sender by unicast protocol through the reverse path;

The sender will receive all RREP messages in the regulated time interval (T), it extracts all keys, decrypt to get their original keys. From these keys to make new private key K for encrypt data in communication. Receiver also makes private key K depending on all these keys that has sent to sender. Assume that sender receives correctly all keys and if any key is not received by the receiver it will request to get it again.

Use message pair of RREQ and RREP to collect information to create a new key for a data communication session until it's regulated valid time is expired.

Formula to create a shared key K based on n key received from receiver $k_1, k_2, \ldots, k_n$:

$$K = HASH_{(SHA-algorithm)}(k_1||k_2|| \ldots ||k_n) \qquad (12)$$

In which || is string concatenation operator.

All messages exchange for renewing shared key will be signed by Elliptic Curve Digital Signature Algorithm (ECDSA) as presented below for protecting against impersonating.

6. Elliptic Curve Digitall Signature Algorithm

When two nodes want to exchange data with each other, they have to exchange their public keys though above key – agreement scheme.

Each node has two keys, one is public key and other is private key.

When the valid duration of private key is expire, the sender has to make RREQ message to get private key by calculating based on formula with information from RREP messages received in time interval regulated by sender.

6.1. Review Digital Signature Scheme

The Digital Signature Algorithm (DSA) is a United States Federal Government standard or FIPS for digital signatures. Key generation has two phases. The first phase is a choice of algorithm parameters which may be shared between different users of the system, while the second phase computes public and private keys for a single user.

Parameter generation

Choose an approved cryptographic hash function H. In the original DSS, H was always SHA-1, but the stronger SHA-2 hash functions are approved for use in the current DSS. The hash output may be truncated to the size of a key pair.

6.2. Sign Message Using ECDSA

The ECDSA algorithm is used everywhere and has not been cracked and it is a vital part of most of today's security. The signature is 2A bytes and is represented by two values of A bytes each, the first one is called R and the second one is called S. So the pair (R, S) together is your ECDSA signature. First must generate a random value k (of A bytes), and use point multiplication to calculate the point P=k*G. That point's x value will represent R. Since the point on the curve P is represented by its (x, y) coordinates (each being A bytes long), we only need the x value (A bytes) for the signature, and that value will be called R. Now calculate the S value.

To calculate S, must make a hash (for example SHA1) of the message, this gives a A bytes value that will be considered as a very huge integer number and we'll call it z. Now can calculate S using the equation:

$$S = k^{-1} (z + d_A * R) \bmod p \qquad (13)$$

Note here the k-1 which is the modular multiplicative inverse of k. It's basically the inverse of k, it's a number such that $(k^{-1} * k) \bmod p = 1$. And k is the random number used to generate R, z is the hash of the message to sign, d_A is the private key and R is the x coordinate of k*G (where G is the point of origin of the curve parameters).

6.3. Correctness and Evaluation

Now that We have the signature, have to verify it, only need the public key (with curve parameters) to do that. Use this equation to calculate a point P:

$$P= S^{-1}*z*G + S^{-1} * R * Q_a \qquad (14)$$

If the x coordinate of the point P is equal to R, that means that the signature is valid, otherwise it's not.

As in section (2.1.2): $Q_a = d_A * G$, Eq. 5 $\Leftrightarrow P = S^{-1}*(z + R * d_A) * G$

$\Leftrightarrow k * G = S^{-1} * (z + R * d_A) * G \Leftrightarrow S = k^{-1} * (z + R * d_A) \bmod p$, this is Eq. 13 to calculate the S component of the signature.

7. Security Routing Is Done by Symmetric Cryptography Method

7.1. Some Concepts

Symmetric cryptography [4] uses a single private key to both encrypt and decrypt data. Symmetric cryptography algorithms consist of block ciphers, stream ciphers. Block ciphers convert each fixed-length block of plane text into cipher text of the same length. In stream ciphers, plain text digits $b_1b_2..b_n$ are combined with key digit stream $k_1k_2..k_n$ using XOR binary operator to get cipher-text stream $r_1r_2..r_n$:

$$b_i \oplus k_i = r_i \qquad (15)$$

In decryption process, the cipher-text stream $r_1r_2..r_n$ are also combined with key digit stream $k_1k_2..k_n$ to get original plane text digits$r_1r_2..r_n$, each digit for a time:

$$r_i \oplus k_i = b_i \qquad (16)$$

Stream ciphers execute very fast and are suitable for processing large streams of data. But they have disadvantage that both sender and receiver have to agree secure key before transmission.

In MANET routing, the mobile nodes usually have limited energy and network topology is dynamically changed. So network link is not stable and has low bandwidth. The asymmetric encryption algorithm (AE) requires much computing load more than symmetric encryption (SE) with the same security level. Key length of AE is much larger than SE. Therefore use SE for encryption data in MANET routing is better choice. In this paper will introduce model of key management, encryption and decryption schema for security routing of service oriented routing. There are some block ciphers, for example DES – data encryption standard, triple DES or 3DES, AES – advanced encryption standard. In which, DES uses key of 56 bits length, 3DES uses key of168 bits length. AES uses key sizes: 128, 192, 256 bits with block size 128 bits. The key length decides the security level of algorithm.

Table 2. *Technical specification of some encryption algorithms.*

Encryption algorithm	Key sizes (bit)	Block sizes (bit)
DES	56	64
Triple DES	168, 112, 56	64
AES	128, 192, 256	128

These block cipher algorithms work on a block of data for each data encryption session. The block size is based on their key sizes. In mobile network, mobile node usually uses stream ciphers to encrypt and decrypt data for very fast processing.

7.2. Key Management Model

In cluster based routing for service oriented routing, the entire network is divided into several clusters for saving energy and reducing control traffic transferring. In each cluster, there is a cluster head (CH) is a node in this cluster that manages configuration of its cluster. The cluster head in each cluster exchanges its configuration information with the cluster head in another cluster. So use the cluster head for managing key is very effective and consistency. The key management schema is processed by some following steps:

- Periodically, CH will generate a new key by a predetermined algorithm, this key will be informed to all cluster members by inner cluster routing protocol. In key transferring session, the new key will be encrypted by using the old key. Assume that at first each node will be provided a same secure key by an predetermined algorithm.
- Key table in each cluster will be set up and exchanged with each other. The key table has three fields as the following table:

Table 3. *Key table format.*

Field name	Field description
Cluster ID	Cluster identification
Key	The value of key
Key valid time	The begin valid time of the key
Key valid duration	The valid time duration of the key

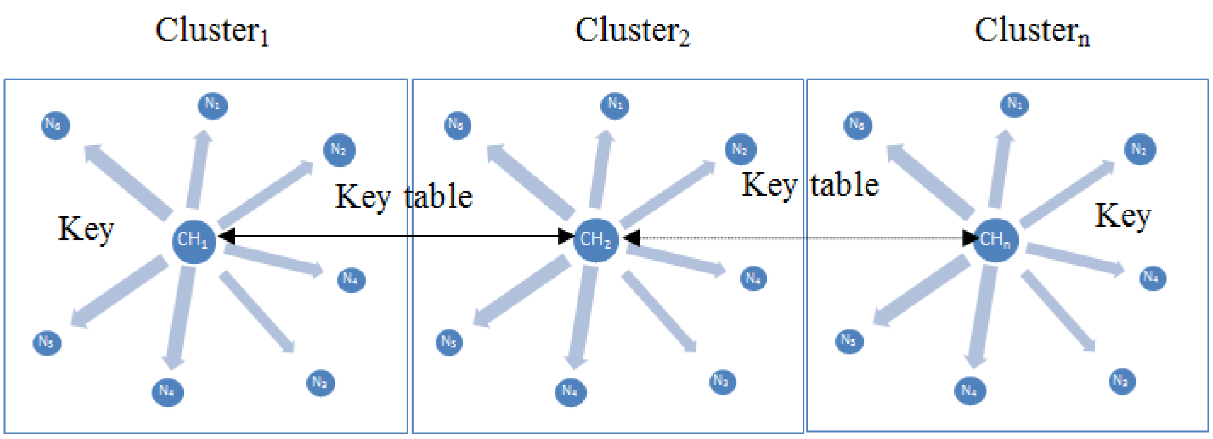


Fig. 5. *Key exchange model in cluster based Network.*

7.3. Encryption and Decryption Processes

7.3.1. Use Block Ciphers

- Encryption: Assume node N_1 need to send package D to node N_2, they also have the same secure key that is provided by the cluster head. Split data into sized blocks according to key size of EA. Assume D is data that is encrypted.

$$D = \{B_1, B_2, \ldots, B_n\} \tag{15}$$

where: $|D| = m$ (bits), B_i is a block of D that has the same length for $i \in \overline{1..n-1}$:

$$|B_i| = |B_j| = |Key|, i, j \in \overline{1..n-1}. \tag{16}$$

$|B_n| \leq |B_i|$, add parity bits to B_n for equal size with B_i.
Apply EA to each Bi to get:

$$Ei = EA(Bi) \tag{17}$$

Combine E_i ($i \in \overline{1..n-1}$) to get E is encryption data of D:

$$E = \cup_{i=1}^{n} E_i \tag{18}$$

Node N_1 send E to N_2.

- Decryption: Use same key as encryption algorithm to decrypt data. When N_2 receives E, E will be decrypt each block at a time to get original plane text block These blocks will be combined to get plain text data D.

7.3.2. Use Stream Ciphers

As describe above, plain text data D is converted to binary stream. The binary stream is split into fixed length binary streams that length is equal to key stream length. Each binary sub-stream is combined with key stream digits using XOR binary operator, each digit at a time to get cipher text stream. These cipher text streams are assembled to get result cipher stream E. In receiving node, this cipher stream E is decrypted by the same algorithm to get plane text stream that is converted to plain text data D.

7.4. Complexity Evaluation

It is easy to evaluate stream cipher algorithms have complexity is O(n), n is size of data.

Block cipher algorithms have complexity is O(m*n), where m is number of rounds to process data and n is the size of data.

7.5. Security and Cryptanalysis

For any cipher, Brute force is the most basic attach, it tries every possible key. So the length of key decides the security of the cipher.

8. Security Model to Protect the Routing Protocol in MANET

8.1. Some Concepts

In each routing protocol, there are two separated processes: i) routing process; ii) forwarding process. In forwarding process, there are three kinds of misbehavior: i) packet drop that means malicious node drops packets that are supposed to forward to its neighbors; ii) packet replication, that malicious node replicates packets that has already forwarded; iii) network layer packet jamming that a node sends too much packets that occupies a significant part of the network bandwidth. In routing process, an intruder node can attack by advertising wrong routing information or wrong network metrics for example wrong distance metric, wrong sequence number, wrong source or destination node address, … For protecting MANET routing from these attacks, it uses self-organized security mechanism. It uses some techniques to monitor its data flows to and from its neighbors packet by packet and statistic these packets to get the decisions.

8.2. Collaboratively Self Organizing to Protect MANET Routing

8.2.1. Message Structure

To authenticate each node when it takes part in network routing, assign a token for each node when it joins network. The token is encrypted by the one of the above mentioned algorithms. The token certifies the membership of any node. If one node is detected the malicious node, it will be revoked the token collaboratively by its neighbors. The token of a node is renewed by sending a token request message (TREQ) to local group manager or any other member of its neighbors. TREQ contains some information: i) request node ID; ii) valid time duration; iii) beginning time to activate. Use public key cryptography primitives to protect the tokens. When TREQ is received, it is executed by one of the neighbors that have received this message, it makes TREP to send to sender of request.

8.2.2. Use Threshold Secret Sharing Scheme

With the threshold secret sharing scheme [15] we divide the message into n parts (message share), then send these parts on n separated routes or multipath routes on two nodes that are communicating. So on receiver node, that needs at least T (threshold) parts of n sent parts to recover the original message based on Lagrange polynomial or Shamir's Secret Sharing.

8.3. Some Techniques to Monitor

The some techniques for monitoring the routing process and purpose of network monitoring:

- Monitor that one node executes the routing process compliances the routing specification. The technique is based on overhearing packets from the monitored nodes. So all communications of its neighbor nodes will be captured to check for routing execution regulations. By this technique, the monitor node will compare the real packets received from one neighbor node (N) and the overheard packets from this node.
- Choose a node N of network that monitors forwarding process of each its neighbor node. In cluster network, on each cluster, N is usually the cluster head of the cluster. The technique for performing this task on the basic of the common principle: the traffic that gets in a node is approximate equal to the traffic that gets out this node in the same time duration.

$$\sum_{Flow_i(t)\in S_{in},\ t\in[t_1,t_2]} |Flow_i(t)| \cong \sum_{Flow_j(t)\in S_{out},\ t\in[t_1,t_2]} |Flow_j(t)| \qquad (19)$$

9. Conclusion and Future Development

In ECC-based authentication algorithm, Elliptic Curve scalar multiplication is core operation, but this operation is the most time consuming operation. This operation takes 80% of executing time [2].

Following is the result of ECC algorithm for encrypting data file with size changes in case of using Soft Computing based Controller (SCC) and not using it.

File size (KB)	ECC without Soft Computing based Arithmetic (Time in ms)			ECC with without Soft Computing based Arithmetic (Time in ms)		
	E	*D*	*T*	*E*	*D*	*T*
1	1490	1370	2860	1320	1340	2660
3	2470	2410	4880	1920	1980	3900
5	3510	3460	6970	2820	2890	5710
7	4573	4512	9085	3916	3996	7912
8	5685	5642	11325	4613	4663	9276

Fig. 6. *The time for encrypting and decrypting data with and without SCC.*

Elliptic Curve Cryptography provides greater security and more efficient performance than the first generation public key techniques (RSA and Diffie-Hellman) now in use. Seriously consider the elliptic curve alternative for the computational and bandwidth advantages they offer at comparable security [11]. In the future research some other techniques to increase performance of ECC based on some new algorithms in addition with other cryptography techniques for MANET security.

References

[1] Nguyen Thanh Long, Nguyen Duc Thuy, Pham Huy Hoang, "Research on Innovating, Evaluating and Applying Multicast Routing Technique for Routing messages in Service-oriented Routing", Springer, ISBN: 978-1-936968-65-7, Volume Number 109, 2012.

[2] Arindam Sarkar, Department of Computer Science & Engineering, University of Kalyani, J. K. Mandal, Department of Computer Science & Engineering, University of Kalyani, "Secured Wireless Communication using Fuzzy Logic based High Speed Public-Key Cryptography (FLHSPKC)".

[3] Debdeep Mukhopadhyay, Dept of Computer Sc and Engg, IIT Madras, "Elliptic Curve Cryptography".

[4] Maya Mohan, S.Mary Saira Bhanu, Department of CSE, NSS College of Engineering, National Institute of Technology, "Secured and QoS based multicast routing in MANETs", (IJCSIS) International Journal of Computer Science and Information Security, Vol. 8, No. 4, July 2010.

[5] Http://en.wikipedia.org/wiki/Elliptic_Curve_Diffie%E2%80%93Hellman.

[6] Http://en.wikipedia.org/wiki/NSA_Suite_B.

[7] Http://en.wikipedia.org/wiki/MQV.

[8] Http://en.wikipedia.org/wiki/Elliptic_curve_cryptography.

[9] Http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf.

[10] Http://kakaroto.homelinux.net/2012/01/how-the-ecdsa-algorithm-works/.

[11] Http://en.wikipedia.org/wiki/Digital_Signature_Algorithm.

[12] Http://searchsecurity.techtarget.com/definition/elliptical-curve-cryptography.

[13] Http://en.wikipedia.org/wiki/Advanced_Encryption_Standard.

[14] Http://www.nsa.gov/business/programs/elliptic_curve.shtml.

[15] W. Lou, W. Liu and Y. Fang, SPREAD: Enhancing data confidentiality in mobile ad hoc networks, in: IEEE INFOCOM 2004 (Hong Kong, China, Mar 2004).

[16] https://en.wikipedia.org/wiki/Lagrange_polynomial.