ANALELE UNIVERSITĂŢII "DUNĂREA DE JOS" DIN GALAŢI
MATEMATICĂ, FIZICĂ, MECANICĂ TEORETICĂ
Fascicula II, supliment la tomul XVI(XXI) 1998
Volumul I, Matematică

AN ENCRYPTION PROTOCOL WITH PUBLIC KEYS THAT USES ELLIPTIC CURVES

Eugen PETAC, "Ovidius" University, Constanta, România.

Abstract:

The cryptographic importance of the new *elliptic curve public-key cryptosystem* (ECPKC) consists of the difficulty to determine discrete logarithms over finite fields [3]. This is much harder than factorization of integers or calculating discrete logarithms in F_q. Another most important aspect consists of the forms for the private keys and for the public keys. The private keys are ordinary integers and the public keys are points on an elliptic curve. Elliptic curve systems are very good for applications with smart cards and in distributed systems, where computational power and integrated circuit space are limited, because computations are easily performed and bandwidth requirements are minimal.

The paper presents a proposal for the implementation of privacy enhancement in a packet-switched local area network, using *elliptic curve public-key cryptography* for key management and authentication. For computing in finite extensions over finite rings we have used the ZEN-new toolbox [2]: there are some computing routine implementing the *group law* defined for an elliptic curve. We have implemented in ZEN the conversions between bit string, integer, point-to-octet string, octet string-to-point, field element and point of the elliptic curves.

Key words: elliptic curve; public-key cryptography; network protocols.

1. INTRODUCTION

There are two modes of implementation of encryption in a network: *link* and *end to end*.

a. Link encryption is easy to incorporate into network protocols. This encryption mode provides good protection against external threats such as traffic analysis: all data flowing on links can be encrypted including addresses. There are two disadvantages:

1. If a node is compromised, all traffic flowing through that node is also compromised.
2. An individual user lasses control over algorithms used.

b. In end to end encryption a message is encrypted and decrypted only at endpoints. Some address information (data link leaders) must be left unencrypted to allow nodes to route packets. High-level network protocols must be augmented with a separate set of cryptographic protocols.

In terms of **OSI** (Open System Interconnection) model encryption can occur at various levels: application, presentation, network, transport. Integration at the application layer gives the individual user complete control over the algorithms used. The security services of our paper are authentication; secrecy; integrity; nonrepudiation.

a. Authentication refers to verification of the identity of the sender or receiver of a communication.

b. Secrecy refers to protection against interception of data.

c. Integrity refers to protection against manipulation of data.

d. Nonrepudiation refers to protection against denial of sending (or possibly receipt) of a message.

For the ECPKC [1], [3] each user I is assumed to have a secret key and a public key, noted by d_I, nonnegative integer number, and $Q_I = d_I \bullet P$, where P and $Q_I \in \boldsymbol{E/F_q}$, with $Q_I = (x_{QI}, y_{QI})$. There are some parameters for elliptic curves, which consist of the following [5], [6]:

1. A field of size q, which defines the underlying finite field $\mathbb{F}_q$, where q shall either be a prime number p, or a power of two.
2. An indication of the type of basis used to represent the elements of $\mathbb{F}_q$: *polynomial basis representation* or *optimal normal basis representation.*
3. Two elements of the field $\mathbb{F}_q$, which define the equation of the elliptic curve $\boldsymbol{E}$.
4. Two elements x_P and y_P of the field $\mathbb{F}_q$ which define a point $P=(x_P, \acute{y}_P)$ on $\boldsymbol{E}$ of prime order.
5. The *order* of the point P.

These form the *set of the elliptic curve parameters* (**SECP**).

For the point $P(x_P, y_P)$ which is on the elliptic curve $\boldsymbol{E/F_q}$, $q=2^m$, $\boldsymbol{E/F_q}$: $y^2+xy = x^3+a_2x^2+a_6$, is possible to define the $\tilde{y}_P$: $\tilde{y}_P = 0$, for $x_P = 0$ and $RM(y_P\ x_P^{-1})$, for $x_P \neq 0$. $RM(z)$ offers the rightmost bit of the field element z. Over the $\mathbb{F}_q$, $q=2^m$, with an *optimal normal basis* representation, a *point compression technique* is used [1], [4]: the point $P = (x_P, y_P)$ is represented by storing only the x-coordinate x_P and the $\tilde{y}_P$.

2. CRYPTOSYSTEM WITH PUBLIC KEYS

EPPKEC is an *Encryption Protocol* with *Public Keys* that uses *Elliptic Curves*. It generates the cryptogram C for the message M, both of them considered as sequences of octets. Users $\mathcal{A}$ and $\mathcal{B}$ of the system know **SECP** and the format mode of the message M, operation by which it is obtained $m^* := F(M) : \{\mathcal{M}\} \rightarrow \Sigma_{\{0,1\}}$. $\{\mathcal{M}\}$ is the set of the messages M and $\Sigma_{\{0,1\}}$ is the set of binary sequences. The users $\mathcal{A}$ and $\mathcal{B}$ choose at random and each of them keep secretly the integer number d_A, respectively d_B, with d_A, $d_B \in [2, r-2]$. They apply the *procedure* **CdP** and each of them obtain by computation the points $Q_A = d_A \bullet P = (x_{QA}, y_{QA})$ and $Q_B = d_{AB} \bullet P = (x_{QB}, y_{QB})$ of the elliptic curve. The binary representations Q_A^* and Q_B^* obtained following the application of one of the *procedures* **CPBTC** or **CPBFTC**, depending on the situation in which it is used or not a compression technique, are registered in a public register **PR**. We note with t the number of bits corresponding to the binary transformation of an element of the field $\mathbb{F}_p$ and with l the number of octets, $l = [[t / 8]]$. We note with [[x]] the smallest integer great or equal with x. The message M, that is to be sent secretly from $\mathcal{A}$ to $\mathcal{B}$, contains at least $l-2$ octets. We note the number of octets of the message M with $\|M\|$. **EPPKEC** contains three phases: *of format of the message M, of encryption and of transmitting of the cryptogram C, of decryption of the received cryptogram.*

EPPKEC

a. the format phase of the message

1. To message M a number of $l - 2 - \|M\|$ octets, that have alternatively the values FF and 00, is associated on the left. A sequences of octets noted with M, of length l-1 octets, of the size $M' = (00 / FF) \| 00 \| M$ is obtained.
2. The user $\mathcal{A}$:

2.1. Chooses at random an integer number $e_A \in [2, r-2]$.

2.2. Read Q_B* from **PR** and apply one of the *procedures* **CBPTC** or **CBPFTC**, for the obtaining of the point Q_B of the elliptic curve.

2.3. Apply the *procedure* **CdP** and compute the points R_A and S_A : $R_A = (x_{RA}, y_{RA}) := e_A \bullet P$; $S_A = (x_{SA}, y_{SA}) := e_A \bullet Q_B$.

2.4. Apply one of the *procedures* **CPOTC** or **CPOFTC** and receive the sequence of octets R_A** that corresponds to R_A.

2.5. Apply the *procedure* **CDECFB** and obtain the binary representation x_{SA}* of x_{SA}.

2.6. Obtain in two steps a binary sequence m* of t bits:

2.6.1. Apply the *procedure* **CDECFB** and receive the binary representation (M')* of M'.

2.6.2. Complete (M')* with $(8$-$8l+t)$ bits of 0 on the left.

b. the encryption and transmitting of the cryptogram C phase

1. Compute $CR^* = (m^* + x^*_{SA}) \bmod . 2$.
2. Apply the *procedure* **CDBO** and obtain the sequence of octets CR^{**}.
3. Find the cryptogram C by a concatenation operation: $C = R^{**}_A \| CR^{**}$.
4. The user $\mathcal{A}$ transmits the cryptogram C to $\mathcal{B}$ user.

If compression techniques $\mathbf{TCPF}_p$ or $\mathbf{TCPF}_{2^m}$ are used, the cryptogram C is represented on $2l+1$ octets and, to the contrary, on $3l+1$ octets.

c. the decryption phase of the received cryptogram

The user $\mathcal{B}$:

1. If the most in the left bit of the cryptogram C, termed with $BS(C)$, it is 1, then the cryptogram C corresponds to $l+1$ octets in the left (note $S_{l+1}(C)$), and to the contrary to the $2l+1$ octets in the left (note $S_{2l+1}(C)$). And this is depending on the using or not of some compression techniques in the encryption operation.
2. Apply the *procedure* **COP** and from R^{**}_A is received the point $R_A = (x_{RA}, y_{RA})$, that belongs to the elliptic curve.
3. Apply the *procedure* **CDOB** and from CR^{**} it is received the binary sequence CR^*.
4. Apply the *procedure* **CdP** and compute $d_B \bullet R_A = S_A = (x_{SA}, y_{SA})$.
   It is checked if: $d_B \bullet R_A = d_B \bullet (e_A \bullet P) = e_A \bullet (d_B \bullet P) = e_A \bullet Q_B = S_A = (x_{SA}, y_{SA})$.
5. Apply the *procedure* **CDECFB** and obtain the binary representation x_{SA}* for x_{SA} .
6. From $(x^*_{SA} + CR^*) \bmod . 2 = m^*$ it results (M')* by doing away with the $(8$-$8l+t)$ most bits that are in the left.
7. Apply the *procedure* **CDBO** for (M')* and obtain M' as a sequence of l-1 octets.
8. Being given the known structure of M', it is obtained the message M, that contains at the most l-2 octets.

End.

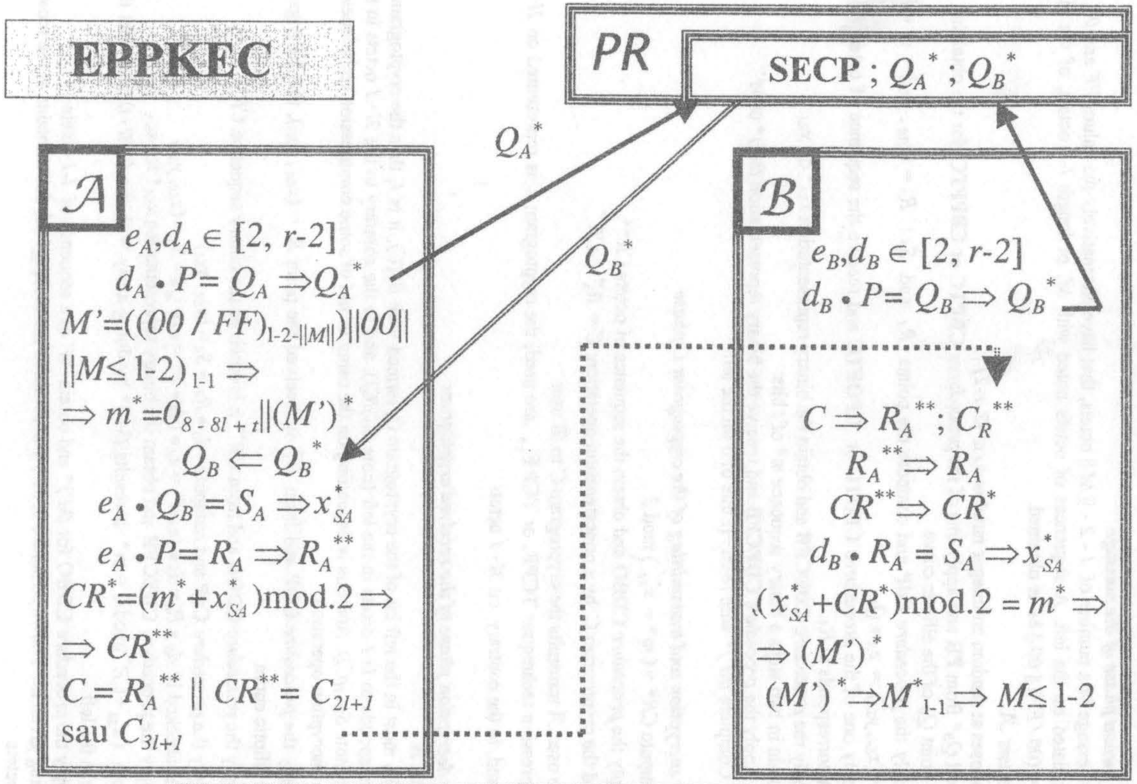
EPPKEC
PR
SECP ; Q_A^* ; Q_B^*
Q_A^*
Q_B^*
$\mathcal{A}$
$e_A, d_A \in [2, r\text{-}2]$
$d_A \bullet P = Q_A \Rightarrow Q_A^*$
$M' = ((00 / FF)_{1\text{-}2\text{-}\|M\|}) \| 00 \|$
$\|M \leq 1\text{-}2)_{1\text{-}1} \Rightarrow$
$\Rightarrow m^* = 0_{8\text{-}8l+t} \| (M')^*$
$Q_B \Leftarrow Q_B^*$
$e_A \bullet Q_B = S_A \Rightarrow x_{SA}^*$
$e_A \bullet P = R_A \Rightarrow R_A^{**}$
$CR^* = (m^* + x_{SA}^*)\text{mod}.2 \Rightarrow$
$\Rightarrow CR^{**}$
$C = R_A^{**} \| CR^{**} = C_{2l+1}$
sau C_{3l+1}
$\mathcal{B}$
$e_B, d_B \in [2, r\text{-}2]$
$d_B \bullet P = Q_B \Rightarrow Q_B^*$
$C \Rightarrow R_A^{**}; C_R^{**}$
$R_A^{**} \Rightarrow R_A$
$CR^{**} \Rightarrow CR^*$
$d_B \bullet R_A = S_A \Rightarrow x_{SA}^*$
$(x_{SA}^* + CR^*)\text{mod}.2 = m^* \Rightarrow$
$\Rightarrow (M')^*$
$(M')^* \Rightarrow M_{1\text{-}1}^* \Rightarrow M \leq 1\text{-}2$

Fig.1 Encryption Protocol with Public Keys that uses Elliptic Curves

3. PROCEDURES USED

Procedure **CDIB**$(x;x^*)$ executes the conversion of the data from an integer into a binary sequence.

Procedure **CDBI**$(x^{**};x^*)$ executes the conversion of the data from a binary sequence into an integer.

Procedure **CDOB**$(x^{**};x^*)$ executes the conversion of the data from a sequence of octets into a binary sequence. To the sequence of octets $x^{**} = X_1, X_2, \dots, X_d$ it corresponds sequence of bits $x^* = x_1, x_2, \dots, x_{8d}$.

Procedure **CDBO**$(x^*;x^{**})$ executes the conversion of the data from a binary sequence into a sequence of octets. To the binary, sequence $x^* = x_1, x_2, \dots, x_{8d}$ it corresponds sequence of octets $x^{**} = X_1, X_2, \dots, X_d$, $d = [[k/8]]$. The first $8d\text{-}k$ bits of X_1 will be zero.

Procedure **CdP**$(d, (x_P, y_P); d \bullet P)$: for d, a positive integer number and with $P = (x_P, y_P)$, point of an elliptic curve, is computed $d \bullet P$ by a raising to power and addition (✚) method.

Conversion *procedures* of a point of the elliptic curve into a sequence of octets (**CPO**$(P, E/K; P^{**})$: for the point $P = (x_P, y_P)$, $P \in E/K$, is obtained a sequence of octets termed with P^{**}, depending on the situation in which it was used (*procedure* **CPOTC**) or not (*procedure* **CPOFTC**) a compression technique (**TC**) of the point (**TCPF**$_p$ or **TCPF**$_{2^n}$).

Conversion *procedures* of a point P of the elliptic curve in a binary sequence (**CPB**$(P, E/K ; P^*)$: for the point $P = (x_P, y_P)$, $P \in E/K$, it is received a binary sequence noted with P^*, depending on the situation in which it was used (*procedure* **CPBTC**) or not (*procedure* **CPBFTC**) a compression technique (**TC**) of the point (**TCPF**$_p$ or **TCPF**$_{2^n}$). The *procedures* **CPBTC** or **CPBFTC** are obtained following the successive applying of the *procedures* **CPOTC** and **CDOB**, respectively **CPOFTC** and **CDOB**.

Conversion *procedure* of a sequence of octets P^{**} in a point P of the elliptic curve (procedure **COP**): for a sequence of octets P^{**} it is received the point $P = (x_P, y_P)$, $P \in E/K$, by applying the procedure **COP**$(P^{**}, E/K; (x_P, y_P))$.

The conversion *procedure* of the data in an element of a finite field, in a binary sequence (**CDECFB**$(\beta; s)$). Let $\beta \in F_q$. The binary sequence s of length $t = [[\log_2 q]]$ bits is received:

1. if $q = p$ is a prime number different from 2, β is an integer number, $\beta \in [0,p-1]$ and it is applied the *procedure* **CDIB** (β, s) in order to obtain a binary sequence s.
2. If $q = 2^n$ and it is used a representation in *optimal normal base* (**ONB**) [5] , then β is obtained as a binary sequence and $s = \beta$.

REFERENCES

[1] G. Agnew, R. Mullin, S. Vanstone , An Implementation of Elliptic Curve Cryptosystems Over $F_{2^{155}}$, IEEE Journal on Selected Areas in Communication, 1993, Vol.11, No.5, pp. 804-813.

[2] F. Chabaud, R. Lercier, ZEN – A new toolbox for computing in finite extensions over finite rings, 1996, INRIA-ftp.

[3] A. Menezes, S. Vastone, The implementation of elliptic curve cryptosystem, Advances in Cryptology – AUSCRYPT '90, Lecture Notes in Computer Sciences, 1990, No. 453, Springer-Verlag, pp. 2-13.

[4] V. Miller, Uses of elliptic curves in cryptography, Advances in Cryptology–CRYPTO '85, Lecture Notes in Computer Sciences, 1986, No. 218, Springer-Verlag, pp. 417-426.

[5] R. Mullin, I. Onyszchuk, S. Vanstone, R. Wilson, Optimal normal bases in *GF*(p^n), Discrete Applied Mathematics, 1988/1989, No.22, pp. 149-161.

[6] E. Petac, On the Construction of Elliptic Curve CryptoSystems, Proceedings of the Development and application systems, “Stefan cel Mare” University, 1996 , pp.93-96.

[7] E. Petac, Public Key Encryption Schemes Using Elliptic Curves, Proceedings of the Eighth Congress of the International Maritime Association of Mediterranean, Istanbul Technical University, 1997,pp.16.2-9-16.2-13.