BULETINUL INSTITUTULUI POLITEHNIC DIN IAŞI
Tomul XLVI (L), Fasc. 1–2, 2000
ELECTROTEHNICĂ. ENERGETICĂ. ELECTRONICĂ

D.C. 621.394.9

SECURE COMMUNICATION BASED ON ELLIPTIC CURVE PUBLIC KEY CRYPTOSYSTEMS (II)

BY
EUGEN PETAC

The cryptographic importance of the *Elliptic Curve Public Key Cryptosystems* (ECPKC) consists of the difficulty to determine discrete logarithms over extensions of finite fields [6]. This is much harder than factorization of integers or calculating discrete logarithms in $\mathcal{F}_q$. Another most important aspect consists in 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 an *Elliptic Curve Authenticated Encryption Scheme* using an *Universal Hash Function* (UHF). The UHF can take an input octet string message M of arbitrary length. The output of the UHF is an octet string H of a 64 bits fixed length.

For computing in finite extensions over finite rings we have used the ZEN-new toolbox [4]: there are some computing routines implementing the *group law* defined for an elliptic curve.

1. Introduction

The key of each user of a *Public Key Cryptosystem* (PKC) is divided into two portions - a private component and a public one. The public component generates a public transformation, E, and the private component generates a private transformation, D. E and D can be called encryption and decryption functions, respectively. In a system we may have $D(E(M)) = M$, $E(D(M)) = M$ or both. There are the following possibilities [2] for the authentication of the send message:

1. The signature, S, is obtained as being $S = D(M) = D_{KdB}(M)$. After reception, $\mathcal{A}$ compiles the result of procedure $E(S) = E_{KeB}(S) = E_{KeB}(D_{KdB}(M))$ and decides that the signature belongs to $\mathcal{B}$, if this result is equal to M. Through the algorithms specific to the PKC, the signature of a message has to depend of all bits of the message, without modifying the text. The procedure to generate the signature is different from the symmetric authentication because the emitter uses data to which the receiver doesn't have access. The receiver can check the required signature by using the emitter public key.

2. We show a system which allows the combination between encipher and authentication operations: the message M is protected against a change or an unauthorized reading. For the beginning we apply to the M message, the E-secret transformation,

A *universal* class *of hash functions* (UHF) is specified [13] by an octet string, S, of length $sLen = 2hLen$ octets. UHF can take an input message, M, of an arbitrary length and the output of UHF is a string, H, of a fixed length, $hLen$ octets: UHF$(M, S) = H$. Following steps are necessary:

1. Break up to message M into n blocks M_k, $(k = \overline{0, n-1})$, each containing $hLen$ octets.

2. Convert the block M_i to an element, m_i, of the field $\mathcal{F}_{2^{8hLen}}$.

3. Let the polynomial $m(x)$ of degree less than n over $\mathcal{F}_{2^{8hLen}}$:

$$m(x) = m_{n-1}x_{n-1} + m_{n-2}x_{n-2} + \ldots + m_1 x + m_0.$$

4. Convert the two $hLen$ octets of S string in the elements a and b of the field $\mathcal{F}_{2^{8hLen}}$.

5. Evaluate $h = m(a) + b$ with h-element of the field $\mathcal{F}_{2^{8hLen}}$.

6. Convert h to an octet string, H, as the hash value of M under S.

In Fig. 1 we present how to use a Universal Hash Function for *Secrecy* and *Authenticity*.

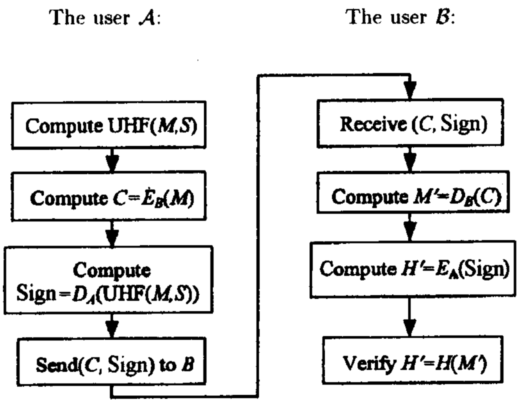

Fig. 1.- UHF for Secrecy and Authenticity.

2.1. The Implementation of Universal Hash Functions

Universal Hash Functions (UHF) are specified by means of a *specification*, S, a sequence of octets of $sLen$ length. Message, M, is of an arbitrary length, divided into m blocks noted M_i, $(i = \overline{0, m-1})$, each of $hLen$ octets. The last block, M_{m-1}, is completed with zeros if its length is less than $hLen$ octets.

MDOPUHF $(M,\ S;\ H)$ is a *message digest obtaining procedure* by using UHF. For the message, M, is obtained the H message digest of the $hlen$ octets.

MDOPUHF $(M, S; H)$

1. The blocks M_i, $(i = \overline{0, n-1})$, are obtained for the message M, each having a length of $hLen$ octets.

2. S_1 and S_2, partial subsequences of $hLen$ octets, are obtained from the octets sequence, S, with a length $sLen = 2\,hLen$.

octets. $\mathcal{A}$ and $\mathcal{B}$ users of the system know SECP (*set of the elliptic curve parameters*) [5], [10], [11], and the input values necessary for the applying of the *procedure* GMSC, of *generating* the mask MSC. Each of the users $\mathcal{A}$ and $\mathcal{B}$ choose the integer numbers, kept secret, d_A, e_A, respectively d_B, e_B, with d_A, e_A, d_B, $e_B \in [2, r-2]$. By computation they obtain the public information $Q_A = d_A \bullet P$ and $Q_B = d_B \bullet P$, inscribed in the *public register* PR. The process of secretly transmitting of the message M, of the length $mLen$ octets, between two users, $\mathcal{A}$ and $\mathcal{B}$, develops in three phases: *of initiation, of generation and broadcasting of the cryptogram, of decrypting of the received cryptogram and of its authentication.*

The *procedure* GMSC(x, x_{ini}, b, t; MSC) proposes, for a binary sequence line, x, of $b = 160$ bits, to generate a sequence of $(mLen + sLen)$ octets, named *mask* and termed with MSC. The fields MSCM and S, of $mLen$, respectively $Llen$ octets correspond to it. The mask is obtained by applying the *procedure* CDBO to a number of $tb = 8(mLen + sLen)$ octets. The generation process takes place under the influence of an initiation binary sequence, termed x_{ini}, of $b = 160$ bits.

```
GMSC($x$, $x_{\mathrm{ini}}$, $b$, $t$; MSC)
1. Read $x$ and $x_{\mathrm{ini}}$, binary sequences of 160 bits, with at least a bit different from zero.
2. Apply the procedure CDBI and obtain the integer numbers $\overline{x}$ and $\overline{x}_{\mathrm{ini}}$, with $\overline{x}$, $\overline{x}_{\mathrm{ini}} \in (0, 2^b)$.
3. The binary sequence $H = H_0||H_1||H_2||H_3||H_4$ is initiated for SHA (Secure Hash Algorithm) [8].
4. For $j = \overline{0, t-1}$ do
        4.1. Set $xv := (\overline{x} + \overline{x}_{\mathrm{ini}}) \bmod 2^b$.
        4.2. Apply one of the procedures OWF-SHA or OWF-DES and obtain the binary
             sequence $y_j := \mathrm{OWF}(H, xv)$.
5. Set $y := y_0||y_1||\ldots||y_{t-1}$.
6. Apply the procedure CDBO for $y$ and obtain the mask MSC = MSCM$||S$.
End.
```

AEPPKECUHF

a) *The initiation phase*

The $\mathcal{A}$ user:

1. Apply the *procedure* CdP and find the points $Q_A := d_A \bullet P$ and $R_A := e_A \bullet P$.

2. Apply one of the *procedures* CPOTC or CPOFTC and obtain the octets sequences Q_A^{**} and R_A^{**}. They contain $l+1$ octets if a compression technique has been used (*procedure* CPOTC) or $2l$ octets, if a compression technique hasn't been used (*procedure* CPOFTC).

3. Inscribe Q_A^{**} in the public register PR.

The user $\mathcal{B}$:

1. Apply the *procedure* CdP and find the points $Q_B := d_B \bullet P$ and $R_B := e_B \bullet P$.

2. Apply one of the *procedures* CPOTC or CPOFTC and obtain the octets sequences Q_B^{**} and R_B^{**}. They contain $l+1$ octets if a compression technique hasn't been used (*procedure* CPOFTC).

3. Inscribe Q_B^{**} in the public register PR.

b) *The generation and broadcasting of the cryptogram phase*

The $\mathcal{A}$ user:

1. Read from PR the octets sequences Q_B^{**}.

2. Apply one of the *procedures* COPTR or COPFTR (depending on the using or not of a compression technique) and obtain the point Q_B that belongs to the elliptic curve.

3. Apply the *procedure* CdP and find the point $S_{AB} := d_A \bullet Q_B$.

4. Apply one of the *procedures* CPOTC or CPOFTC and obtain the sequence of octets S_{AB}^{**}, of $l+1$ or $2l$ octets.

5. Apply the *procedure* SHA and obtain $x = \mathrm{SHA}\,(S_{AB}^{**})$ and $x_{\mathrm{ini}} = \mathrm{SHA}\,(R_A^{**})$.

6. Apply the *procedure* GMSC (x, x_{ini}, b, t, $mLen$; MSC) and obtain the mask MSC of $mLen + sLen$ octets.

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

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

e) *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 ($\boldsymbol{+}$) method.

f) 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$_{2^n}$).

g) 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.

h) Conversion *procedure* of a sequence of octets P^{**} in a point P of the eliptic 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)). For a sequence of octets termed with P^{**} is obtained the point $P = (x_P, y_P)$, $P \in E/K$, depending on the situation in which it was used (*procedure* COPTR) or not (*procedure* COPFTR) a restore technique (TR) of the point (TRPF$_p$ or TRPF$_{2^n}$).

i) The conversion *procedure* of the data in an element of a finite field, in a binary sequence (CDECFB(β; s)). Let $\beta \in \mathcal{F}_q$. 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) [9], [12] then β is obtained as a binary sequence and $s = \beta$.

j) *Procedure* CDECFO (β; S) executes the conversion of the data for an element of a finite field into a sequence of octet⁻.

k) *Procedure* CDOECF (S; β) executes the conversion of the data for a sequence of octets into an element of a finite field.

4. Conclusions

For the *elliptic curve public keys cryptographic* algorithms the cryptographic importance consist in the difficulty to determine discrete logarithms over finite fields. Another most important aspect consists in the forms for the private keys and the public ones. The private keys are ordinary integers and the public ones are points on an elliptic curve. Elliptic curve systems are very useful for applications with smart cards and in distributed systems, where computational power and integrated circuit space are limited.

Acknowledgements. I wish to thank Professor Dr. Eng. Valeriu Munteanu at the "Gh. Asachi" Technical University of Jassy, Faculty of Electronics and Telecommunications, for their careful reading of the manuscript, helpful comments and for valuable discussions on this work.

Received, November 25, 1998

"Ovidius" University, Constantza

REFERENCES

1. Diffie W., Hellman M., *New Directions in Cryptography.* IEEE Trans. on Information Theory, *22*, 644-654 (1976).
2. Diffie W., Hellman M., *Privacy and Authentication: an Introduction to Cryptography.* IEEE Trans. on Information Theory, *67*, 397-427 (1979).
3. Diffie W., *The First Ten Years of Public Key Cryptography, Contemporary Cryptology.* The Science of Information Integrity, IEEE Press, New York, 1991, 135-175.
4. Chabaud F., Lercier R., *ZEN – A New Toolbox for Computing in Finite Extensions Over Finite Rings.* INRIA-ftp, July, 1996.
5. Koeller J., Menezes A., Qu M., Vanstone A., *Standard for RSA, Diffie-Hellman and Related Public-key Cryptography.* Certicom Corp., Ontario, Canada, 1996.
6. Merkle R., *A Certified Digital Signature.* Advances in Cryptology, CRYPTO '89, Lecture Notes in Computer Sciences, 1990, *435*, Springer-Verlag, New York, 218-238.
7. Miller V., *Uses of Elliptic Curves in Cryptography.* Advances in Cryptology, CRYPTO'85, Lecture Notes in Computer Sciences, 1986, *218*, Springer-Verlag, New York, 417-426.
8. * * * National Institute of Standards and Technology, Secure Hash Standard, Computer Security, 1995, FIPS PUB 180-1.
9. Mullin R., Onyszchuk I., Vanstone S., Wilson R., *Optimal Normal Bases in $GF(p^n)$.* Discrete Appl. Mathem., *22*, 149-161 (1988/1989).
10. Petac E., *Public Key Encryption Schemes Using Elliptic Curves.* Proc. of the Eighth Congress of the Internat. Maritime Assoc. of Mediterranean, Istanbul Technical University, 1997, 16.2 - 9-16.2 - 13.
11. Petac E., *About a Method of Distribution Keys of a Computer Network Using Elliptic Curves.* IEEE Internat. Symp. on Consumer Electronics, ISCE'97, Singapore, December 2-4, 1997, 309-312.
12. Silverman J., Tate J., *Rational Points on Elliptic Curves.* Springer-Verlag, New York, 1992.
13. Shoup V., *On Fast and Provably Secure Message Authentication Based on Universal Hashing.* Advances in Cryptology, Crypto'96, Lecture Notes in Computer Science, 1996, *1109*, Springer-Verlag, New York, 313-328.

COMUNICAŢIE SIGURĂ BAZATĂ PE CRIPTOSISTEME CU CHEI PUBLICE CONSTRUITE PE CURBE ELIPTICE (II)

(Rezumat)

Importanţa criptografică a *criptosistemelor cu chei publice construite pe curbe eliptice* constă în evitarea dificultăţii de a calcula logaritmi discreţi peste extensii ale unor câmpuri finite [6]. Această operaţie este mult mai complexă decât factorizarea unor numere întregi sau calculul logaritmilor discreţi în $\mathcal{F}_q$. Un alt avantaj rezultă din structura cheilor, publică şi privată: cheia privată este un simplu număr întreg în timp ce cheia publică este un punct al unei curbe eliptice. Sistemele criptografice construite pe curbe eliptice sunt indicate pentru aplicaţii care folosesc cartele inteligente şi în sistemele distribuite, unde sunt limitate puterea de calcul şi spaţiul fizic necesar implementării.

Se prezintă un sistem de criptare şi autentificare construit pe curbe eliptice, care foloseşte o funcţie de dispersie universală (UHF). Această funcţie poate prelua la intrare un şir de octeţi de lungime arbitrară, care corespunde mesajului M. Rezultatul aplicării acestei funcţii este un şir de octeţi cu o lungime fixată la 64 de biţi.

Pentru calculul în extensii finite, construite peste inele finite, s-a folosit toolbox-ul ZEN [4]: acesta prezintă rutine de calcul care implementează grupul finit de tip *law* pentru o curbă eliptică.