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

AN AUTHENTICATED ENCRYPTION PROTOCOL WITH PUBLIC KEYS THAT USES ELLIPTIC CURVES AND UNIVERSAL HASH FUNCTIONS

Eugen PETAC, "Ovidius" University, Constanta, România
Dorina PETAC, "Mircea cel Bătrân" National College, Constanta, România

Abstract:
Public-key systems (two-key or asymmetric) differ from conventional systems in that there is no longer a single secret key shared by a pair of users. Each user has his proper cryptographic key. The key of each user 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 termed encryption and decryption functions, respectively. In a system we may have *D(E(M)) = M*, *E(D(M)) = M* or both.
The novelty of our proposed method is the use of the *elliptic curve chord tangent group law*. We developed an *Elliptic Curve Authenticated Encryption Scheme* using a *universal hash function*. The hash function can take an input octet string message *M* of arbitrary length. The output of the hash function is an octet string *H* of a *64* bits fixed length.

Key words: hash function; message digest; public-key system; digital signature; computer network.

1. INTRODUCTION

A *hash function H* accepts a variable-size message *M* as input and outputs a fixed-size representation *H(M)* of *M* (called a *message digest*).
The hash function *H* has[3] the following properties:
a. *H* can be applied to an argument of any size.
b. *H* produces a fixed-size output.
c. *H(x)* is relatively easy to compute for any given *x*.
d. For any given *y* it is computationally infeasible to find *x* with *H(x) = y*.
e. For any fixed *x* it is computationally infeasible to find $x' \neq x$ with *H(x') = H(x)*.
H(M) will be much smaller than *M*: *H(M)* might be *64* or *128* bits, whereas *M* might be a megabyte or more.
Diffie and Hellman [1], [2] introduced *digital signature*. A digital signature is the electronic analogue of a handwritten signature and has the following properties:
a. A receiver must be able to validate the sender's signature.
b. A signature must not be forgeable.
c. The sender of a signed message must not be able to repudiate it later.

A digital signature may be applied to $H(M)$, because $H(M)$ is signed rather than M. Both M and the signed $H(M)$ may be encapsulated in another message, which may be encrypted for secrecy. The receiver may validate the signature on $H(M)$ and apply the public function H directly to M and check to see that it coincides with the forwarded signed version of $H(M)$. This validates both the authenticity and integrity of M simultaneously. If $H(M)$ were unsigned only integrity would be assured.

An *universal* class *of hash functions* (*UHF*) is specified [9] by an octet string S of length $sLen = 2\ hLen$ 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 the 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 $F_{2^{8hLen}}$.
3. Let the polynomial $m(x)$ of degree less than n over $F_{2^{8hLen}}$:
   $m(x) = m_{n-1}x_{n-1} + m_{n-2}x_{n-2} + \ldots + m_1 x + m_0$.
4. Convert the two *hLen* octets of S string in the elements a and b of the field $F_{2^{8hLen}}$.
5. Evaluate $h = m(a) + b$ with h, element of the field $F_{2^{8hLen}}$.
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*:

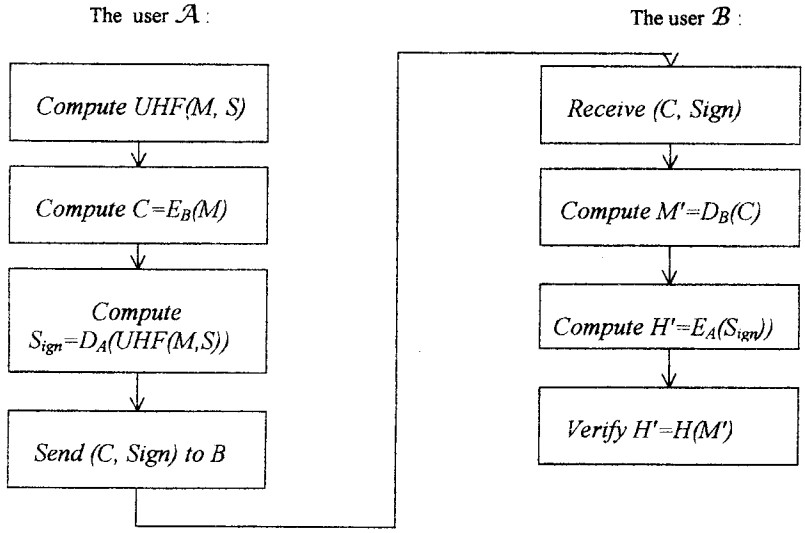

Fig.1 UHF for Secrecy and Authenticity

2. 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 note M_i, $i = \overline{0, m-1}$, each oh *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 $sLen = 2\ hLen$.
3. Set $n := 8\ hlen$.
4. *Procedure* **CDOECF** is applied and there are obtained m_i, as elements of $\mathbb{F}_{2^n}$ field (following the conversion of each sequence octets M_i, $i = \overline{0, m-1}$) and the elements α and $\beta \in \mathbb{F}_{2^n}$, following the conversion of the partial subsequences S_1 and S_2 , of *hLen* octets.
5. There asserts the polynomial $f(x)$, whose degree is less than m:
   $f(x) = f_{m-1}x^{m-1} + f_{m-2}x^{m-2} + \ldots + f_1 x + f_0$.
6. There computes $h := (f(\alpha)) \cdot \alpha + \beta$ with $h \in \mathbb{F}_{2^n}$.
7. There applies the *procedure* **CDECFO** for h, an element of the finite field $\mathbb{F}_{2^n}$ and the message digest H, as a sequence of *hLen* octets is obtained.

End.

UHF works with a constant number of bits. It is used experimentally *hLen=64* and m of hundreds order. *UHF* allows the obtaining of the message digest for an arbitrary length message, by means of an efficient computation. Moreover, they allow every user of a cryptosystem to be able to particularize them by means of the specification S.

3. THE ELLIPTIC CURVE AUTHENTICATED ENCRYPTION ALGORITHM

AEPPKECUHF is an *authenticated encryption protocol* with *public keys* that uses *elliptic curves* and *universal hash functions*. It allows the generation of the cryptogram C for the message M, both of them considered as sequences of octets. $\mathcal{A}$ and $\mathcal{B}$ users of the system know **SECP** (*set of the elliptic curve parameters*)[6] 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 $t \cdot b = 8(mlen + slen)$ octets. The generation process takes place under the influence of an initiation binary sequence, termed x_{ini}, of $b = 160$ bits.

GMSC(x, x_{ini}, b, t **;** ***MSC*** **)**

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

End.

PCCPCEUHF

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 has been used (*procedure* **CPOTC**) or $2l$ 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 user $\mathcal{A}$:

1. Read from **PR** the octets sequence 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 $x = SHA(S_{AB}$**) and $x_{ini} = SHA(R_A$**).
6. Apply the *procedure* **GMSC(** $x, x_{ini}, b, t, mlen$; *MSC*) and obtain the mask *MSC* of *mlen*+*slen* octets.
7. Extract *MSCM* from *MSC* of length *mLen* (mask for message *M*) and *S*, of length *sLen* (the specification for *UHF*).

Fig 2 Authenticated Encryption Protocol with Public Keys that uses Elliptic Curves and UHF

8. Set $C^{**} := MSCM \oplus M$.
9. Set $M_{UH} := M \| ID_A \| ID_B \| VT$.
10. Apply the *procedure* **MDOPUHF**(M_{UH}, S; H) and obtain the message digest, of length *hLen* octets.
11. Obtain the cryptogram $C := R_A^{**} \| C^{**} \| H$, that is transmitted to $\mathcal{B}$. This contains ((*l*+*1*) + *mlen* + *hlen*) or (*2l* + *mlen* + *hlen*) octets, depending on the using or not of a compression technique.

b. The decryption phase of the cryptogram received and its authentication

The user $\mathcal{B}$:

1. Extract the sequences of octets RA^{**}, C^{**} and H from the received cryptogram C.
2. Read Q_A^{**} from **PR**.
3. Apply one of the *procedures* **COPTR** or **COPFTR** and obtain the point Q_A of the elliptic curve, that corresponds the sequence of octets Q_A^{**}.
4. Apply the *procedure* **CdP** and compute
   $d_B \bullet Q_A = d_B \bullet (d_A \bullet P) = d_A \bullet (d_B \bullet P) = d_A \bullet Q_B = S_{AB}$.
5. Apply one of the *procedures* **CPOTC** or **CPOFTC** and obtain the sequence of octets S_{AB}^{**}, of *l*+*1* or *2l* octets.
6. Apply the *procedure* **SHA** and obtain $x = SHA(S_{AB}^{**})$ and $x_{im} = SHA(R_A^{**})$.
7. Extract *MSCM* from *MSC* of length *mlen* (the mask for message M) and S, of length *sLen* (the specification for *UHF*).
8. Set $M' := C^{**} \oplus MSCM$.
9. Set $M_{UH} := M' \| ID_A \| ID_B \| VT$.
10. Apply the *procedure* **MDOPUHF**(M_{UH}, S; H) and obtain the message digest H', of length *hlen* octets.
11. If $H' = H$, then M' is the message sent that belongs to the user $\mathcal{A}$.

End.

4. IMPLEMENTATION

For computing in finite extensions over finite rings we have used the ZEN-new toolbox [3]: there are some computing routine implementing the *group law* defined for an elliptic curve. We present in the following the procedures which have used for *authenticated encryption protocol* with *public keys* that uses *elliptic curves* and *universal hash functions* (**AEPPKECUHF**).

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

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

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

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

Procedure **CdP(*d*, (x_P, y_P); *d* •*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 **TCP F**$_{2^n}$).

Conversion *procedures* of a sequence of octets into a point of the elliptic curve (**COP(*P***; *P*, *E/K*)**: 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 **TRP F**$_{2^n}$).

REFERENCES

[1] W. Diffie, M. Hellman, New directions in cryptography, IEEE Transaction on Information Theory, 1976, No. 22, pp. 644-654.

[2] W. Diffie, M. Hellman, Privacy and authentication: an introduction to cryptography, IEEE Transaction on Information Theory, 1979, No.67, pp. 397-427.

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

[4] R. Merkle, A certified digital signature, Advances in Cryptology - CRYPTO '89, Lecture Notes in Computer Sciences, 1990, No. 435, Springer-Verlag, pp. 218-238.

[5] National Institute of Standards and Technology, Secure Hash Standard, Computer Security, 1995, FIPS PUB 180-1.

[6] E. Petac, Security Elements of Communications Using Elliptic Curve Cryptosystems ,Proceedings of The Third Asia-Pacific Conference on Communications (APCC'97), Sydney, Australia, 7-10 December, 1997, pp. 1362-1365.

[7] J. Silverman, The Arithmetic of Elliptic Curves, Springer-Verlag, New York, 1986.

[8] J. Silverman, J. Tate, Rational Points on Elliptic Curves, Springer-Verlag, New York,1992.

[9] V. Shoup, On fast and provably secure message authentication based on unuversal hashing, Advances in Cryptology-Crypto'96, Lecture Notes in Computer Sciences, 1996, No.1109, Springer-Verlag, pp.313-328.