INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

INFORMATICS CONTROL

STUDIES IN INFORMATICS AND CONTROL

With Emphasis on Useful Applications of Advanced Technology

September 1999 **Volume 8 Number 3**

Informatics and Control Publications

STUDIES IN INFORMATICS AND CONTROL

With Emphasis on Useful Applications of Advanced Technology

A Quarterly Journal
An International Editorial Board
Publication Commencement Date: March 1992

CORE SUBJECTS

Papers on current topics of Information Technology: integration of IT with control; IT use in control and management systems; advanced automatic control; processing technologies in applied informatics.

PUBLISHING DETAILS

Usually the format of published papers is a two-column one, with abstracts, keywords and short biographies preceding the main text.

Author index and instructions to authors, together with an order form, are to be found at the back of the journal.

CONTENTS REMARKS

Currently, the journal invites original articles, studies and technical and research reports which describe work-in-progress and/or discuss new results in and even conjectures on Information Technology and Computer Based Control.

Publishing intervals would exceptionally be longer than 3 months.

French-written text and abstracts will also be accepted for publication if they make significant contributions and are submitted as such.

With its quality-abiding and contents flexibility, the journal also accepts special issues and doctoral dissertations.

The journal will feature book reviews, letters and scientific events, or reflect history of the field through inserting outstanding communications.

The journal circulation is worldwide.

Where to spot recent articles of the journal: INSPEC database.

How to access the published issues:

http://www.ici.ro/revista/sic.html

This publication is supported by National Agency for Research ,Technology and Innovation
Advisory Board for Scientific Research and Technological Development

STUDIES IN INFORMATICS AND CONTROL

With Emphasis on Useful Applications of Advanced Technology

September 1999 **Volume 8 Number 3**

Informatics and Control Publications

Published under the auspices of the Romanian Academy

EDITORIAL ORGANIZATION

EDITORIAL BOARD

CONTENTS

BOOK REVIEWS

AUTHOR INDEX

Methods for Distribution Of the Cryptographic Keys in A Computer Network

Eugen Petac

"Ovidius" University of Constanta
Faculty of Mathematics and Informatics
124 Mamaia Str.
8700 Constanta
ROMANIA
e-mail: epetac@ovidius.ct.ro

Abstract: The paper presents some new methods for the implementation of privacy enhancement in a packet-switched local -area network, using *Elliptic Curve Public-Keys Cryptography* for the distribution of cryptographic keys. The cryptographic importance of the *Elliptic Curve Public Keys Cryptosystems* (**ECPKC**) is justified by the difficulty in finding discrete logarithms over the finite fields. The different forms that the secret keys may have, if compared to the public keys, recommend the use of **ECPKC** especially for their using smart cards in the distributed systems as well, where there are limits as far as the computation and integration power is concerned. For computing in finite extensions over finite rings we have used the ZEN-new toolbox [1]: there are some computing routines implementing the *group law* defined for an elliptic curve.

Keywords: Public Keys Cryptosystems, Elliptic Curve Public Keys Cryptosystems, Communication Systems, Certification Authorities, Elliptic Curve, Communication Protocols, Distributed Systems

Eugen Petac was born in Constanta, Romania, on January 26, 1961. He received the MSc. degree in Electronics-Data Communications, and the Ph. D degree from the "Gh. Asachi" Technical University, Jassy, Romania, in 1985 and 1998 respectively. In 1991 he became an Assistant Professor at the Department of Electronic Engineering and Computer Science , the "Mircea cel Batran" Naval Academy, Constanta, Romania. Since 1995 he has been Lecturer and further Senior Lecturer at the Department of Mathematics and Informatics, "Ovidius" University of Constanta. He has worked in the area of data security and privacy and published numerous papers in the field of cryptography. His current research interests include systems and software in data security, cryptography, coding theory and computer network. Dr. Eugen Petac is a member of IEEE Computer Society, American Mathematical Society and Society for Industrial and Applied Mathematics.

Alexandru D. Sotir was born in Constanta, Romania, in 1943. He took the degree in Electrotechnical Engineering from the Polytechnical Institute of Jassy in 1966, and respectively, the doctoral degree in the same domain, from the Polytechnical Institute of Bucharest in 1989.
From 1972 to 1991 he worked at the Territorial Computing Centre in Constanta, as technical manager.
Since 1991 he has been a lecturer at the Electrotechnics Department of the "Mircea cel Batran" Naval Academy in Constanta.
He co-authored the following books: "Practice of Databases", Vol. I and Vol. II, Technical Publishing House, Bucharest, 1989, "Principles of Electromagnetic Compatibility", Vol. I and Vol. II, the Navy Publishing House, Constanta, 1994, and "Electromagnetic Compatibility", Military Publishing House, Bucharest, 1995 (all in Romanian).
His main research interest is in electromagnetic compatibility of computers with application in computer control systems.

Alexandru D. Sotir

"Mircea cel Batran" Naval Academy of Constanta
1 Fulgerului Str.
8700 Constanta
ROMANIA

1. Introduction

One of the most important applications of *Public Keys Cryptosystems* (**PKC**)[2] is the *distribution of the secret key*, necessary for secure communication. By means of the **PKC**, both the *integrity* of the data and the *authentication* (of users and data) can be simultaneously obtained in a distributed process. For a **PKC**, we note with E and D, the encryption and decryption procedures, reversed one against the other. There may be used only a **PKC**, or a hybrid system, made up of two **PKC** that ensure the security and the authentication separately.

For a **PKC**, a secret key K is a particular form of a *message*. In a system used for the distribution of the public components of users, users can fix secret keys and mean the system for the encryption and the authentication of these secret keys. The secret keys may be changed later without any difficulty. This is opposite to a system in which secret keys are fixed with the help of a messenger from a central authority (symmetrical cryptographic systems).

The use of the **PKC** makes the question of the distribution keys consist only in how to connect the identity of the system users with the public keys of the users. No data communication on a secret channel is required. Users can generate pairs of *public keys/ private keys* and there is no need for exposing the private keys to any other users. If one takes into account only the data integrity and authentication services, users should be able to record their public compositions without using a secret channel.

In **WAN** (*Wide- area Network*) type networks, this requires a distributed system [3], [5], [9], [10] of the *Certification Authorities* (**CA**) ordered at hierarchical levels. *Central Certification Authorities* (**CCA**) must certify a second level, the one of the certification agents

that finally certify the public components of the users. There may be added other levels based on the same principle. A user 's certification is done on the basis of a certification protocol.

In the authors' opinion a user should register himself with *Local Certification Authorities* (**LCA**). The **LCA** must be affiliated to an organization and has to be able to use an identifier of the user, proper to the organization concerned, in order to certify the public key of the user. If **LCA** are able to safely communicate with **CCA**, the user's public key must be acknowledged to **CCA** for getting known.

As the public components of a user may be certified and distributed, either user must possess other public components in order to fix common keys for the encryption, without resorting to a safety channel.

There are systems organized such as to allow that **CA** set up digital certificates [9]. **CA** master the procedures E_{CA} (public) and D_{CA} (secret). We appreciate that **CA** have set up the $CERT_A$ and $CERT_B$ certificates for user $\mathcal{A}$ and user $\mathcal{B}$. They define the public components of user $\mathcal{A}$ and user $\mathcal{B}$, corresponding to the procedures E_A and E_B .

The users' secret components are adequate for the inverse procedures D_A and D_B. Both $\mathcal{A}$ and $\mathcal{B}$ keep secret the $CERT_A$ and $CERT_B$ certificates . If $\mathcal{A}$ has previously exchanged certificates with $\mathcal{B}$, each of $\mathcal{A}$ and $\mathcal{B}$ has other stored certificates. If not, there are two ways thereby $\mathcal{A}$ can get $\mathcal{B}$'s certificate:

- $\mathcal{A}$ requests directly $CERT_B$ from $\mathcal{B}$. The advantage of this method consists in avoiding **CA**. Instead there appear questions as to the security and the integrity of data. There may happen that $CERT_B$ has recently been invalidated.
- $\mathcal{A}$ may get hold of $CERT_B$ by means of **CA**. Each of the *CERT* certificates has the form $D_{CA}(M)$, where M contains information about the E procedure. $\mathcal{A}$ gets hold of the required $CERT_A$ and $CERT_B$ certificates. The $E_{CA}(CERT_A)$ computation validates $CERT_A$, thus allowing that the E_A procedure runs, and obtains the E_B procedure, necessary for the communication with $\mathcal{B}$, by computing $E_{CA}(CERT_B)$. This phase implicitly validates the **CA** source of the certificates, too. Secret storing of the certificates means the impossibility for **CA** of providing certification services on a regular basis. Such services would impose if two of the users communicated for the first time, if certain components were compromised, or if the certificates were invalidated. The $\mathcal{A}$ user must be sure that the $CERT_A$ and $CERT_B$ certificates are obtained in real- time. A *handshake* type protocol [12], [13] may be used for real-time guaranteeing the authenticity and the integrity of the transmitted certificates. The advantages are:

- information is transmitted via uncertain channels under proper security conditions;
- distribution of the certificates by **CA** sees that the users are in possession of some certificates that can be used as soon as they are received. The system must benefit from the services specialized in the construction of public keys of their users and identifiers [9], [11].

But there are also drawbacks of such a method:

- **CA** have to cope with a big traffic, a fact that sometimes results in a throttle effect on the traffic.
- An intruder who succeeds in getting hold of the secret components of **CA**, may forge the user's certificates.

More and more attention is being claimed by the decentralized administrating method of the certificates: each of the users of the cryptographic system becomes responsible for the administration of its own certificate. When A intends to initiate the communication process of a secret key, it will convey a message to B. that contains $CERT_A$ (A's certificate), ID_A (A's identifier), as well as other information that is specific to the used protocol: information concerning the moment of time when they performed the transmit, the lapse of time during which the certificate is validated, sequences of data randomly generated, as well as a request for the certificate of $\mathcal{B}$, $CERT_B$. Use of an authentication protocol completes the changing phase of the certificates. There will result authenticated certificates for either user. User $\mathcal{A}$ validates the $CERT_B$ certificate by computing $E_{CA}(CERT_B)$. **CA** must periodically submit lists of certificates that got invalidated due to the expiry of the data, the compromising of the key or on account of administrative reasons.

2. Elliptic Curve Public Keys Cryptosystems (ECPKC)

By the points of a fixed field $_K$ we understand the points of the plane $\mathcal{P}(x, y)$ whose co-ordinates belong to $_K$. An algebraic curve [4], [8], [14] is the multitude of all the points (x, y) from the affine plane whose co-ordinates meet the equation $f(x, y)=0$, where $f(x,y)=0$ is a polynomial with coefficients from $_K$. Let F_q be a finite field containing q elements, with q the prime number. For $_K = F_q$ we term with $\overline{K}$ its algebraic closure $\overline{K} = \bigcup_{m \geq 1} F_{q^m}$.

Definition 1. An elliptic curve E (also named an algebraic curve of the first genus) is a set of solutions for the equation $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$, corresponding to a smooth curve from the affine plane $P^2(\overline{K}) = \overline{K} \times \overline{K}$, together with the point O (from the infinite) asserted in affine coordinates. The multitude of points of the elliptic curve E whose co-ordinates belong to K, together with point O, represent the multitude of points named the rational points of E [14].

Definition 2. If $a_1, a_2, a_3, a_4, a_6 \in _K$, the elliptic curve is defined over K and we term it with E/K. We term its multitude of rational points with $E(K)$, and the number of points with $\#E(K)$, representing the order of the elliptic curve. If two points are given on E, there is a way of associating them with a third point, termed with R.

Definition 3. Let the points $P,Q \in E$ be. We define $PQ=R'$ as being the third cross point of the straight line determined by P and Q with E. By convention $OO = O$ [4],[14].

Definition 4. Let the points $P=(x_1,y_1)$, $Q=(x_2, y_2)$, $P,Q \in E$ be. We define the algebraic operation $\boxplus: E \times E \rightarrow E$, $P \boxplus Q = OR' = R = (x_3, y_3)$. [4],[14].

Sentence 5. [4],[6],[14] The algebraic operation $\boxplus$ given by *Definition 4*, endows E with an Abelian group structure so that the following are carried out [4],[6],[14]:

a) $P \boxplus Q \in E$ for $(\forall) P,Q \in E$.

b) $(P \boxplus Q) \boxplus R = P \boxplus (Q \boxplus R)$ for $(\forall) P,Q,R \in E$.

c) $(\exists) O \in E$ so that $O \boxplus P = P \boxplus O = P$ for $(\forall) P \in E$.

d) $(\exists) Q \in E$ so that $P \boxplus Q = Q \boxplus P = O$, for $(\forall) P \in E$. In particular $Q = -P$ and $-O = O$.

e) $P \boxplus Q = Q \boxplus P$ for $(\forall) P,Q \in E$.

Sentence 6 [8] $(E(K), \boxplus)$ is a subgroup of the group $(E, \boxplus)$.

For the elliptic curves:

a) Supersingular $E_1/K : y^2 = x^3 + ax + b$

b) Non-supersingular

$E_2/K: y^2 + xy = x^3 + a^2x + b$

with $K = F_q$ where $q = p$ = prime number or $q = 2^n$, $n \in N^*$, we define a set of parameters of the elliptic curve (**SPEC**) that consists of:

1. q, where $q=p$ is a prime number or $q=2^n$, $n \in N^*$.
2. The base [5],[7],[11] in which are represented the elements of the field F_q, which may be polynomial (**BP**), normal (**BN**) or normal-optimal (**BNO I** or **BNO II**).
3. $a, b \in F_q$, that defines the type of the elliptic curve [5]: supersingular elliptic curve (**SEC**) or non-supersingular elliptic curve (**NSEC**).
4. $x_P, y_P \in F_q$, co-ordinates of the point $P = (x_P, y_P)$ that belongs to the elliptic curve.
5. The order of the point P that belongs to the elliptic curve.

For **ECPKC**, **SPEC** is public. As a rule of principle, the operation of generating a key K, necessary for the communication in such a system, consists of:

1. Its randomly selecting an integer number $d \in [2,r-2]$
2. Computing the point $Q \in E/K$, $Q = (x_Q, y_Q) := d \bullet P = P \boxplus P \boxplus P \boxplus \ldots \boxplus P$ (d times).
3. The co-ordinates of the point Q representing the public key, and the integer number d representing the secret key.

Cryptographic importance for **ECPKC** is justified by the difficulty in finding discrete

logarithms over the finite fields[5],[6],[7]. The different forms that the secret keys may have, if compared to the public keys, recommend the use of **ECPKC** especially for their using smart cards in the distributed systems as well, where there are limits as far as the computation and integration power is concerned.

3. Techniques of Distribution of the Cryptographic Keys

DPCKECOWF is a *distribution protocol of the cryptographic keys using elliptic curves and one-way functions* –**OWF** [2]. The user $\mathcal{A}$ and the user $\mathcal{B}$ of the system are in possession of all the information about **SECP** and the one -way function *w(.)*, written in the public register PR . Each of them chooses the integer numbers e_A, d_A, respectively e_B, d_B with $e_A, d_A, e_B, d_B \in$ [*2,r-2*], which they kept secret. The process of transmitting a key *K* between the two users $\mathcal{A}$ and $\mathcal{B}$ develops in two phases:

a. Initiation Phase

User $\mathcal{A}$:

1. Applies the **CdP** *procedure* and determines the points $Q_A := d_A \bullet P$ and $R_A := e_A \bullet P$.
2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequences Q_A^* and R_A^*, corresponding to the points Q_A and R_A from the elliptic curve.
3. Puts down Q_A^* in **PR**.
4. Transmits R_A^* to the user $\mathcal{B}$, via a safe channel.

User $\mathcal{B}$:

1. Applies the **CdP** *procedure* and determines the points $Q_B := d_B \bullet P$ and $R_B := e_B \bullet P$.
2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequences Q_B^* and R_B^*, corresponding to the points Q_B and R_B from the elliptic curve.
3. Puts down Q_B^* in **PR**.
4. Transmits R_B^* to user $\mathcal{A}$, via a safe channel.

b. Generation Phase of the Common Key

User $\mathcal{A}$:

1. Reads Q_B^* from **PR**.
2. Receives R_B^* .
3. Applies one of the **CBPTC** and **CBPFTC** *procedures* and obtains the points Q_B and R_B that belong to the elliptic curve.
4. Applies the **CdP** *procedure* and determines the points K_{AB1}, K_{AB2} $K_{AB1} := d_A \bullet Q_B$, $K_{AB2} := e_A \bullet R_B$.
5. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequences K_{AB1}^* and K_{AB2}^*.
6. Applies the one-way function *w(.)* and obtains the common key $K=w(K_{AB1}^*, K_{AB2}^*)$.

User $\mathcal{B}$:

1. Reads Q_A^* from **PR.**
2. Receives R_A^* .
3. Applies one of the **CBPTC** and **CBPFTC** *procedures* and obtains the points Q_A and R_A that belong to the elliptic curve.
4. Applies the **CdP** *procedure* and determines the points K_{BA1}, K_{BA2} $K_{BA1} := d_B \bullet Q_A$, $K_{BA2} := e_B \bullet R_A$.
5. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequences K_{BA1}^* and K_{BA2}^*.
6. Applies the one-way function *w(.)* and obtains the common key $K=w(K_{BA1}^*, K_{BA2}^*)$.

End

Remark:

As $K_{AB1} = d_A \bullet Q_B = d_A \bullet (d_B \bullet P) = d_B \bullet (d_A \bullet P) = d_B \bullet Q_A = K_{BA1}$

$K_{AB2} = e_A \bullet Q_B = e_A \bullet (e_B \bullet P) = e_B \bullet (e_A \bullet P) = e_B \bullet Q_A = K_{BA2}$

it results that by applying the function $w(.)$ the users $\mathcal{A}$ and $\mathcal{B}$ are in possession of the common key K.

DPCKTEC is a *distribution protocol of a cryptographic key of the token type, using the elliptic curve*. The users $\mathcal{A}$ and $\mathcal{B}$ of the system know **SPEC** and choose each of the integer numbers e_A, d_A, respectively e_B, d_B, with e_A, d_A, e_B, $d_B \in [2, r-2]$, which they keep secret. The process of transmitting a key K between two users $\mathcal{A}$ and $\mathcal{B}$ runs in three phases:

a. Initiation Phase

User $\mathcal{A}$:

1. Applies the **CdP** *procedure* and determines the points $Q_A := d_A \bullet P$ and $R_A := e_A \bullet P$.
2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequence Q_A^*, corresponding to the point Q_A.
3. Puts down Q_A^* in **PR**.

User $\mathcal{B}$:

1. Applies the **CdP** *procedure* and determines the points $Q_B := d_B \bullet P$ and $R_B := e_B \bullet P$.
2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequence Q_B^*, corresponding to the point Q_B.
3. Puts down Q_B^* in **PR**.

b. Generation Phase of the Token Message (*TM*)

User $\mathcal{A}$:

1. Reads from **PR** the binary sequence Q_B^*.
2. Applies the **CdP** *procedure* and defines the points $S_{AB} := e_A \bullet Q_B$ and $R_A := e_A \bullet P$, that belong to the elliptic curve.
3. Applies successively one of the **CPOTC** or **CPOFTC** and **CDOI** *procedures* and obtains the integer number $\overline{S}_{AB} \in [2, r-2]$, corresponding to the point S_{AB} of the elliptic curve.
4. Computes $TM_1 = (K \cdot \overline{S}_{AB}) \bmod r$.
5. Applies the **CDIO** *procedure* and obtains the octets sequence TM_1^{**}, corresponding to TM_1.
6. Applies one of the **CPOTC** or **CPOFTC** *procedures* and obtains the octets sequence R_A^{**}, corresponding to the point R_A.
7. Generates the token message $TM = TM_1^{**} \| R_A^{**}$ that is transmitted to user $\mathcal{B}$.

c. The Phase of Obtaining the Key *K*

User $\mathcal{B}$:

1. Extracts the sequences of octets TM_1^{**} and R_A^{**} from *TM*.
2. Applies one of the **COPTR** and **COPFTR** *procedures* and obtains the point R_A of the elliptic curve that corresponds to the octets sequence R_A^{**}.
3. Applies the **CdP** *procedure* and computes $d_B \bullet R_A = d_B \bullet (e_A \bullet P) = e_A \bullet (d_B \bullet P) = e_A \bullet Q_B = S_{AB}$.
4. Applies successively one of the **CPOTC** or **CPOFTC** and **CDOI** *procedures* and obtains the integer number $\overline{S}_{AB} \in [2, r-2]$, corresponding to the point S_{AB}.
5. Computes the inverse element $\overline{S}_{AB}^{-1} \bmod r$.

6. Applies the **CDIO** *procedure* and obtains the integer number TM_1 corresponding to the octets sequence TM_1^{**}.

7. Finds the key K, following the computations:

$(TM_1 \cdot \overline{S}_{AB}^{-1}) \bmod r = (K \cdot \overline{S}_{AB} \cdot \overline{S}_{AB}^{-1}) \bmod r = K.$

End

DPCKTCEIDTV is a *distribution protocol of the cryptographic key by a token type message , using elliptic curves*. The token message contains data about the *identifiers* of the users: ID_A, ID_B as well as data about the moment of time when the transmission takes place, put down in the *time vector* (*TV*). Octets by two or by four correspond to the fields *TV* and *ID*, depending on the size of the network. Users $\mathcal{A}$ and $\mathcal{B}$ of the system know **SECP** and choose each of the integer numbers e_A, d_A respectively e_B, d_B, with e_A, d_A, e_B, $d_B \in [2, r-2]$, which they keep secret. The process of transmitting a key K between two users, $\mathcal{A}$ and $\mathcal{B}$, runs in three phases:

a. Initiation Phase

User $\mathcal{A}$:

1. Applies the **CdP** *procedure* and determines $Q_A := d_A \bullet P$.

2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequence Q_A^*, corresponding to the point Q_A of the elliptic curve.

3. Puts down Q_A^* in the public register **PR**.

User $\mathcal{B}$:

1. Applies the **CdP** *procedure* and determines $Q_B := d_B \bullet P$.

2. Applies one of the **CPBTC** and **CPBFTC** *procedures* and obtains the binary sequence Q_B^*, corresponding to the point Q_B of the elliptic curve.

3. Puts down Q_B^* in the public register **PR**.

b. Generation Phase of the Token Message (*TM*)

User $\mathcal{A}$:

1. Reads the binary sequence Q_B^* from **PR**.

2. Applies the **CdP** *procedure* and determines the points $S_{AB} := e_A \bullet Q_B$ and $R_A := e_A \bullet P$, which belong to the elliptic curve.

3. Applies successively one of the **CPOTC** or **CPOFTC** and **CDOI** *procedures* and obtains the integer number $\overline{S}_{AB} \in [2, r-2]$, corresponding to the point S_{AB} of the elliptic curve.

4. Applies the **CDIO** *procedure* and obtains the octets sequence K^{**}, that corresponds to the key K represented by the integer number $K \in [2, r-2]$.

5. Obtains $(ID_A \parallel K^{**})$ by concatenation.

6. Applies the **CDOI** *procedure* and obtains the integer number $\overline{(ID_A \parallel K^{**})}$.

7. Computes $TM_1 := (\overline{(ID_A \parallel K^{**})} \cdot (\overline{S}_{AB})) \bmod r$.

8. Applies the **CDIO** *procedure* and obtains the octets sequence TM_1^{**}, corresponding to TM_1.

9. Applies one of the **CPOTC** and **CPOFTC** *procedures* and obtains the octets sequence R_A^{**} that corresponds to the point R_A.

10. Generates the token message $TM = (TM_1^{**} \parallel ID_B \parallel TV \parallel R_A^{**})$.

c. The Phase of Obtaining the Key *K*

User $\mathcal{B}$:

1. Extracts the octets sequences TM_1^{**}, ID_B, TV and R_A^{**}.
Applies one of the **COPTR** and **COPFTR** *procedures* and obtains the point R_A of the

elliptic curve, that corresponds to the octets sequence R_A^{**}.

2. Applies the ***CdP procedure*** and computes $d_B \bullet R_A = d_B \bullet (e_A \bullet P) = e_A \bullet (d_B \bullet P) = e_A \bullet Q_B = S_{AB}$.

3. Applies successively one of the **CPOTC** or **CPOFTC** and **CDOI** *procedures* and obtains the integer number $\overline{S}_{AB} \in [2, r-2]$, that corresponds to the point S_{AB}.

4. Computes the inverse element $\overline{S}_{AB}^{-1}$ mod. r.

5. Applies the **CDIO** *procedure* and obtains the integer number TM_1, corresponding to the octets sequence TM_1^{**}.

6. Identifies the integer number $\overline{(ID_A \| K^{**})}$, following the computations
$(TM_1 \cdot (\overline{S}_{AB}^{-1}))\text{mod}.r =$
$((\overline{(ID_A \| K^{**})}) \cdot (\overline{S}_{AB}) \cdot (\overline{S}_{AB}^{-1}))\text{mod}.r =$
$\overline{(ID_A \| K^{**})}$.

7. Applies the **CDIO** *procedure* and obtains the octets sequence $ID_A \| K^{**}$.

8. With ID_A known results K^{**}.

9. Applies the **CDOI**(K^{**};K) *procedure* and obtains the key K.

End.

4. Conclusions

Cryptography has for long been used for keeping military and diplomatic communications secret. At present, there is a continuous development of cryptography, due to an unprecedented evolution of the communication means and techniques, mainly of those oriented on digital transmissions, with applications to almost any field of activity (economic, social, diplomatic, military).

The implementation of the procedures used in this paper was done with ZEN– toolbox, specialized in the number theory [1]. This toolbox is highly suitable for the building of a cryptographic library, as it benefits from the facilities offered by the BigNum package, that contains work routines with numbers of degree of hundreds of decimal digits. The main procedures having been implemented are:

CdP(*d*, (x_P, y_P); *d* •*P*) : for *d*, a positive integer number and *P*, point of an elliptic curve, $d \bullet P$ is computed by means of *repeated square-and-multiply method* [5][6].

CPBTC((x_P, y_P), *E/K*; *P)** implements the conversion of a point *P* of the elliptic curve in a binary sequence *P**, using a compression technique.

CPBFTC((x_P, y_P), *E/K*; *P)** implements the conversion of a point *P* of the elliptic curve in a binary sequence *P**, without using a compression technique.

CBP(*P, *E/K*; (x_P, y_P))** implements the conversion of a binary sequence *P** in a point $P=(x_P, y_P)$ of the elliptic curve, using a reconstruction technique of the point (**CBPTR**) or without using a reconstruction technique of the point (**CBPFTR**).

CPOTC((x_P, y_P), *E/K*; *P*)** implements the conversion of a point *P* of the elliptic curve in an octets sequence *P***, using a compression technique.

CPOFTC((x_P, y_P), *E/K*; *P*)** implements the conversion of a point *P* of the elliptic curve in an octets sequence *P***, without using a compression technique.

COP(*P*, *E/K*; (x_P,y_P))** implements the conversion of an octets sequence *P*** in a point $P = (x_P, y_P)$ of the elliptic curve, using a reconstruction technique of the point (**COPTR**) or without using a reconstruction technique of the point (**COPFTR**).

CDOI(*x*,*x*)** implements the conversion of the data from an octets sequence to an integer number.

CDIO(*x*; *x*)** implements the conversion of the data from an integer number to an octets sequence.

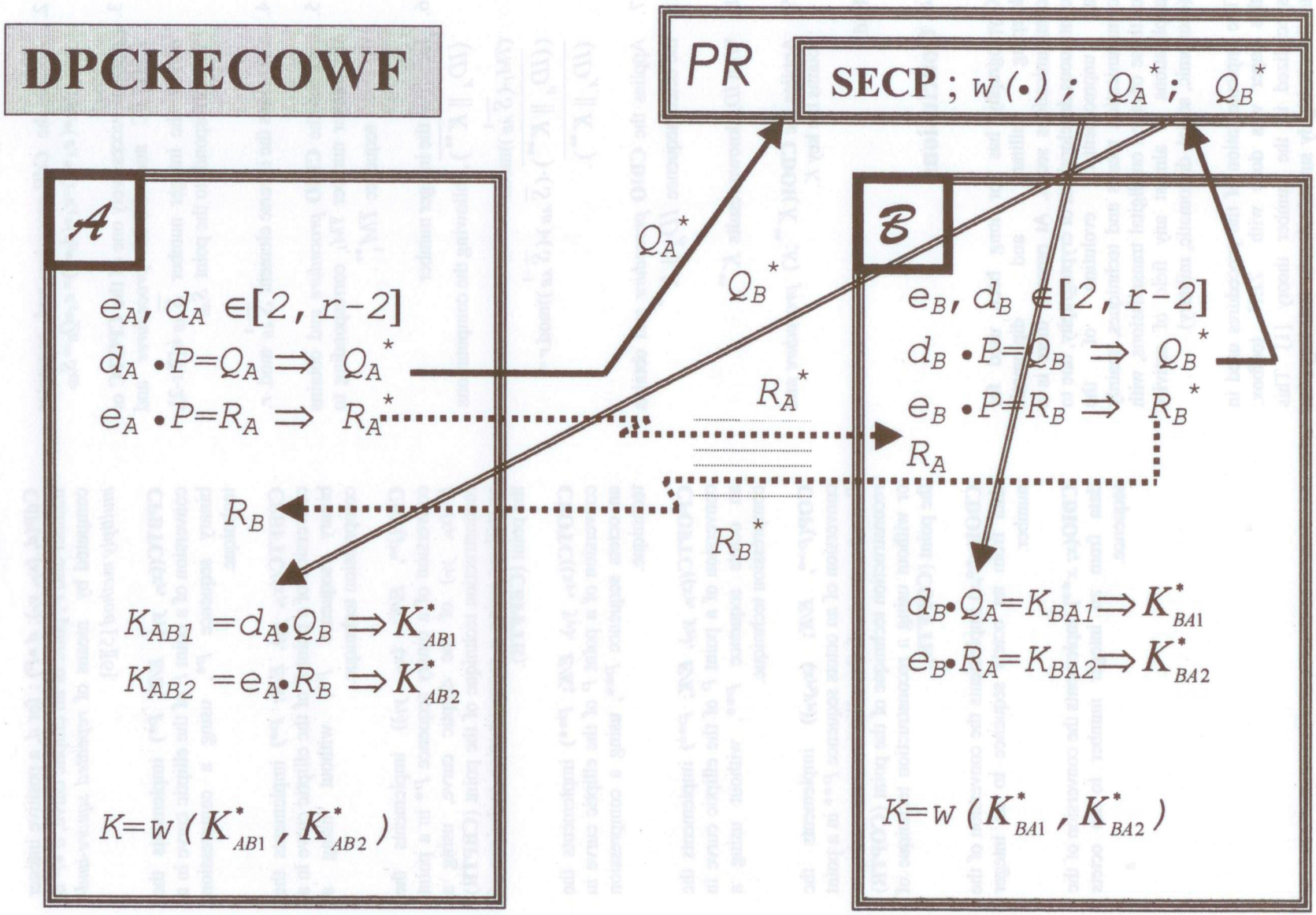

Figure 1. Method for Distribution of the Cryptographic Keys (DPCKECOWF)

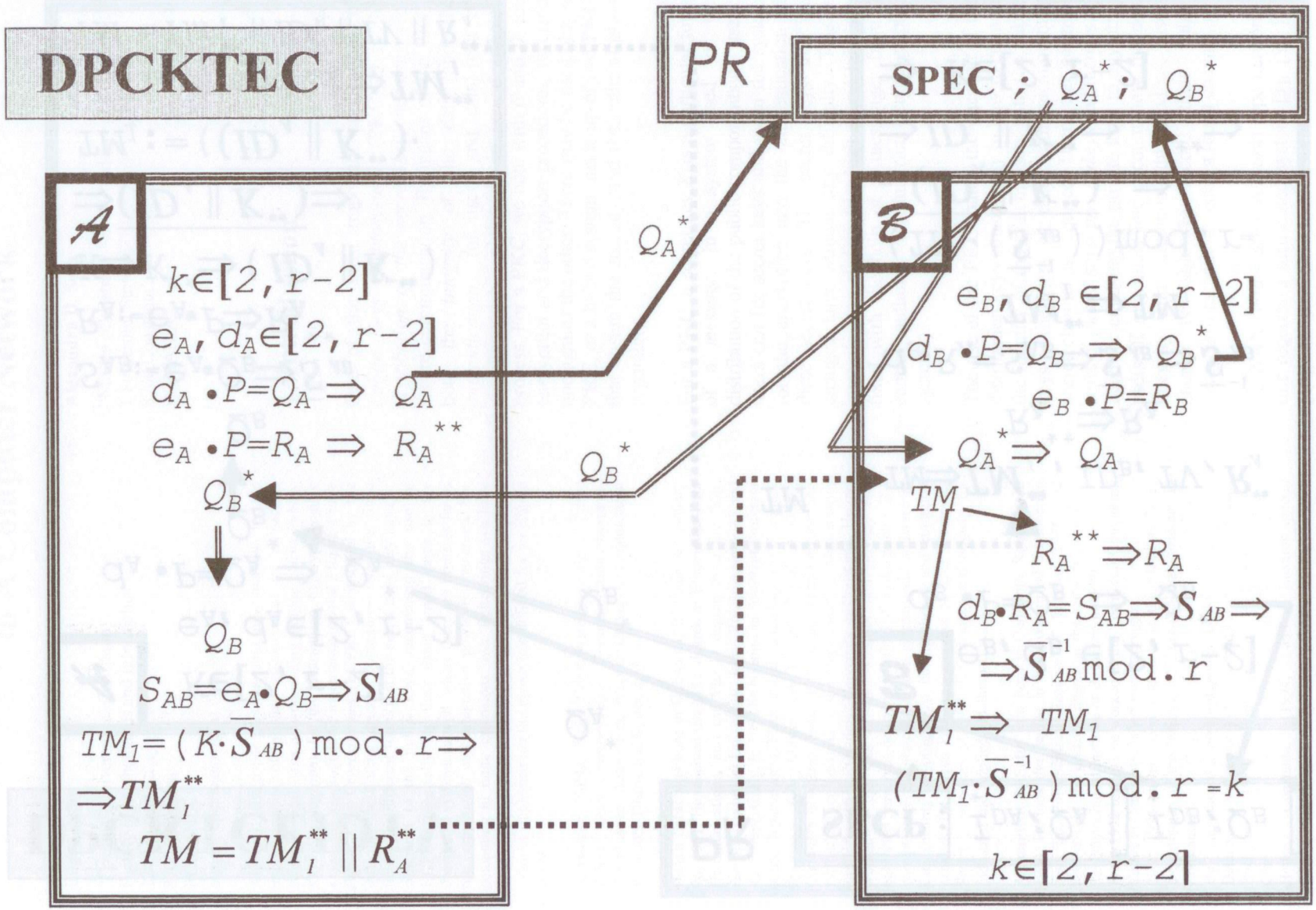

Figure 2. Method for Distribution of the Cryptographic Keys (DPCKTEC)

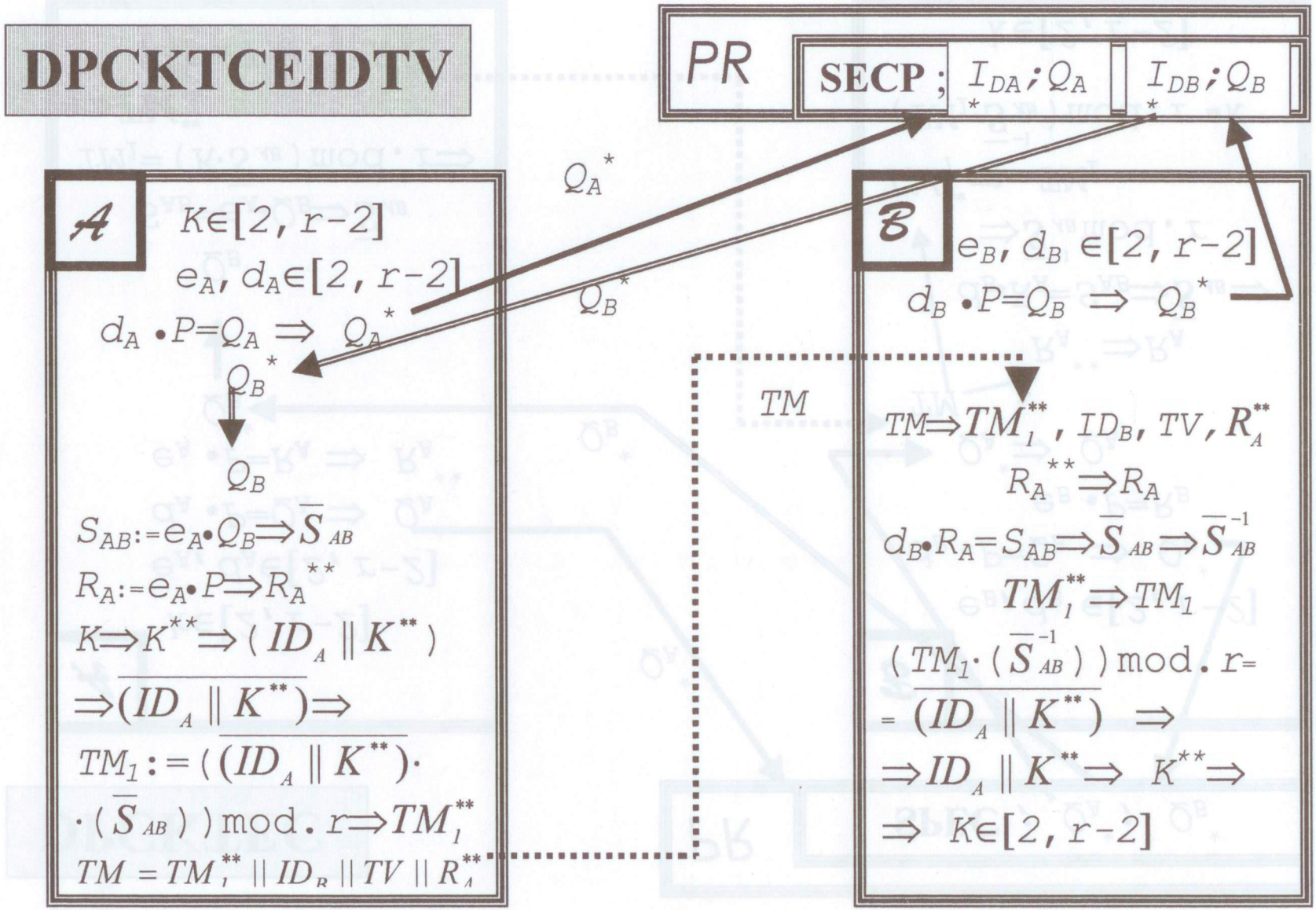

Figure 3. Method for Distribution of the Cryptographic Keys (DPCKTCEIDTV)

REFERENCES

1. CHABAUD, F. and LERCIER, R., ***ZEN* - A New Toolbox for Computing in Finite Extensions Over Finite Rings**, INRIA-ftp, France, July 16, 1996.

2. DIFFIE, W. and HELLMAN, M. E., **New Directions in Cryptography,** IEEE TRANSACTIONS ON INFORMATION THEORY, Vol. IT-22, No. 6, 1976, pp. 644-654.

3. FUMMY, W. and MUNZERT, M., **A Modular Approach to Key Distribution,** Proceedings of Crypto '90, LNCS 537,1991, pp. 151-159.

4. HARTSHORNE, R., **Algebraic Geometry**, SPRINGER, New York, 1977.

5. KOELLER, J., MENEZES, A., QU, M. and VANSTONE, A., **Standard for RSA, Diffie-Hellman and Related Public-key Cryptography**, Certicom Corp., Ontario, Canada, 1996.

6. MENEZES, A. and VANSTONE, S., **Elliptic Curve Cryptosystems and Their Implementation,** JOURNAL OF CRYPTOLOGY, No. 6, 1993, pp. 209-224.

7. MENEZES, A. and VANSTONE, S., **The Implementation on Elliptic Curve Cryptosystems**, Advances in Cryptology – AUSCRYPT '90, Lecture Notes in Computer Science, SPRINGER – VERLAG, 1990, pp. 2-13.

8. MORENO, C., **Algebraic Curves Over Finite Fields**, CAMBRIDGE UNIVERSITY PRESS, 1991.

9. PATRICIU, V. V., **Cryptography and Network Security**, TECHNICAL PUBLISHING HOUSE, Bucharest, Romania,1994 (in Romanian).

10. PETAC, E., **About A Method of Distribution Keys of A Computer Network Using Elliptic Curves**, Proceedings of ISCE'97, Singapore, December 2-4, 1997, pp. 309-312.

11. PETAC, E., **Security Elements of Communications Using Elliptic Curve Cryptosystems**, the 3rd Asia-Pacific Conference on Communications (APCC'97), Sydney, Australia, December 7-10, 1997.

12. PETAC, E. and PETAC, D., **Principles and Techniques for Information Protection in the Computer Network**, MATRIXROM, Bucharest, Romania, 1998.

13. SCHNEIER, B., **Applied Cryptography,** JOHN WILEY & SONS, New York, 1994.

14. SILVERMAN, J. H., **The Arithmetic of Elliptic Curves,** Graduate Texts in Mathematics, Vol.106, SPRINGER, New York, 1986.

ISSN 1220-1766