A Matricial Public Key Cryptosystem with Digital Signature

RAFAEL ALVAREZ[1], FRANCISCO-MIGUEL MARTINEZ[2],
JOSE-FRANCISCO VICENT[3], and ANTONIO ZAMORA[4]
Dpt. of Computer Science and Artificial Intelligence
University of Alicante
Campus de San Vicente,
Ap. Correos 99, 03080 Alicante
SPAIN
{ralvarez[1], fmartine[2], jvicent[3], zamora[4]}@dccia.ua.es

This research was partially supported by the Spanish grant GV06/018

Abstract: - We describe a new public key cryptosystem using block upper triangular matrices with elements in $\mathbb{Z}_p$, based on a generalization of the discrete logarithm problem over a finite group. The proposed cryptosystem is very efficient, requiring very few operations and also allows an ElGamal based digital signature scheme. The main benefit is that the security level is higher than other algorithms for the same key size.

Key-Words: - Cryptography, Security, Public-Key, DLP, Finite Fields, Diffie-Hellman, Polynomial Matrices, ElGamal, Digital Signature.

1 Introduction

Information is recognized by many organizations as an important asset. Few businesses could function effectively without the ability to rely, to some extent, on information as a resource: banks need to know the details of each account, and hospitals need to access patient medical records. Information security is concerned with providing assurances about data.

Broadly speaking, information security is frequently classified as the provision of the following services: *confidentiality* (the assurance that data is not disclosed to unauthorized parties), *integrity* (the assurance that data is genuine) and *availability* (the assurance that data is readily accessible).

Communication over open networks is very cheap, but represents easy pickings for an adversary who wants to intercept, modify, or inject data; data stored on networked computers faces similar threats. If society is to benefit from the advantages offered by electronic data storage and open networks, information security must therefore provide techniques capable of supplying confidentiality, integrity, and availability in this new environment.

In order to establish a confidential channel between two users of such a network, classical single-key cryptography requires them to exchange a common secret key over a secure channel. This may work if the network is small and local, but it is infeasible in non-local or large networks.

To simplify the key exchange problem, modern public-key cryptography provides a mechanism in which the keys to be exchanged do not need to be secret. In such a framework, every user possesses a key pair consisting of a (non-secret) public key and a (secret) private key; only public keys are published.

They are used to encrypt the messages to be sent to the owner of the key or to verify digital signatures issued by the owner of the key. Before using someone else's public key to encrypt a message or verify a signature, one should make sure that the key really belongs to the intended recipient or the indicated issuer of the signature.

Achieving authenticity of public keys can be done in several ways. Public key cryptosystems are essential for electronic commerce or electronic banking transactions; they assure privacy as well as integrity of the transactions between two parties. Digital signatures are used to sign electronic documents and they are also mostly based on public-key techniques.

A lot of popular public-key encryption systems are based on number-theoretic problems such as factoring integers or finding discrete logarithms. The underlying algebraic structures are, very often, abelian groups; this is especially true in the case of the Diffie-Hellman method (DH, see [20]), that was the first practical public key technique and introduced in 1976. In such a system, when two parties want to communicate with each other, the sender encrypts the message with the recipient's public key and then transmits the cipher text to the recipient. Upon

receiving the encrypted information, the recipient can decrypt the message with his private key.

The Discrete Logarithm Problem (DLP, see [19, 27, 29]) is, together with the Integer Factoring Problem (IFP, see [28]) and the Elliptic Curve DLP (ECDLP, see [18]), one of the main problems upon which public-key cryptosystems are built. Thus, efficiently computable groups where the DLP is hard to break are very important in cryptography. In recent years, cryptographic research has become more and more important due to the increasing number of application areas related to the field, requiring data confidentiality, authentication and integrity.

The method presented in this paper, generalises the DH approach to a group based on the powers of a block upper triangular matrix, which is a very flexible and practical technique.

The usual sizes for the keys in the IFP or DLP are around 1024 binary digits, existing well known algorithms of sub-exponential order that solve these problems (see [22, 24, 25]).

The so called square root algorithms (see [23, 31, 32, 34]) reach an order of complexity $\sqrt{\overline{p}}$ where $\overline{p}$ is the greatest prime factor of the order of the group. This is not enough to be used in big and arbitrary finite groups, but if this order does not have great prime factors, these algorithms can be practical. Therefore it is necessary that the order of the group in which we are working has great prime factors.

Our system is capable of increasing the computational cost required for a successful attack on the generated DLP for equivalent key sizes.

The rest of the paper is divided as follows: section 2 shows some properties necessary for the proposed cryptosystem. Section 3 is divided in several subsections: a key exchange protocol, an encryption scheme and a digital signature scheme. Finally, several conclusions about the system are given in section 4.

2 Preliminaries

Some basic linear algebra properties (see [26, 35]), necessary for the purpose of the paper, are presented in this section.

Given p a prime number and $r, s \in \mathbb{N}$, we denote by $Mat_{r\times s}(\mathbb{Z}_p)$ the matrices of size $r\times s$, with elements in $\mathbb{Z}_p$, and by $GL_r(\mathbb{Z}_p)$ and $GL_s(\mathbb{Z}_p)$ the invertible matrices of size $r\times r$ and $s\times s$.

We define

$$\theta=\left\{\begin{bmatrix} A & X \\ \mathbf{0} & B \end{bmatrix}, A\in GL_r(\mathbb{Z}_p), B\in GL_s(\mathbb{Z}_p), X\in Mat_{r\times s}(\mathbb{Z}_p)\right\}.$$

Theorem 1. *The set θ has a structure of a non abelian group for the product of matrices.*

Proof: Given the definition of θ, it is obvious that the product operation is closed.

The identity element is

$$I=\begin{bmatrix} I_r & \mathbf{0} \\ \mathbf{0} & I_s \end{bmatrix},$$

where I_r and I_s, are respectively the identity matrices $r\times r$ and $s\times s$.

The inverse of any element $M=\begin{bmatrix} A & X \\ \mathbf{0} & B \end{bmatrix}$, is

$$M^{-1}=\begin{bmatrix} A^{-1} & -A^{-1}XB^{-1} \\ \mathbf{0} & B^{-1} \end{bmatrix}.$$

The associative property is obvious since they are square matrices. □

Theorem 2. *Let $M=\begin{bmatrix} A & X \\ \mathbf{0} & B \end{bmatrix}\in\theta$, we consider the subgroup generated by the different powers of M.*

Taking *h* as a non negative integer then

$$M^h=\begin{bmatrix} A^h & X^{(h)} \\ \mathbf{0} & B^h \end{bmatrix}, \qquad (1)$$

where

$$X^{(h)}=\begin{cases} \mathbf{0} & if \quad h=0, \\ \sum_{i=1}^{h} A^{h-i}XB^{i-1} & if \quad h\geq 1. \end{cases} \qquad (2)$$

Also, if $0\leq t\leq h$, then

$$X^{(h)}=A^tX^{(h-t)}+X^{(t)}B^{h-t}, \qquad (3)$$
$$X^{(h)}=A^{h-t}X^{(h)}+X^{(h-t)}B^t. \qquad (4)$$

Proof: The equation (1) is proven using induction on *h*. For *h = 0* and *h = 1*, the result is obvious. It is supposed to be true for *h-1* and will be demonstrated true for *h*.

We have

$$\begin{aligned} M^h &= MM^{h-1} \\ &= \begin{bmatrix} A^{h-1} & X^{(h-1)} \\ \mathbf{0} & B^{h-1} \end{bmatrix} \begin{bmatrix} A^h & X^{(h)} \\ \mathbf{0} & B^h \end{bmatrix} \\ &= \begin{bmatrix} A^h & AX^{(h-1)} + XB^{h-1} \\ \mathbf{0} & B^h \end{bmatrix}, \end{aligned}$$

from the induction hypothesis, applying (2) we have

$$\begin{aligned} X^{(h)} &= AX^{(h-1)} + XB^{h-1} \\ &= A\sum_{i=1}^{h-1} A^{h-1-i} XB^{i-1} + XB^{h-1} \\ &= \sum_{i=1}^{h-1} A^{h-i} XB^{i-1} + XB^{h-1} \\ &= \sum_{i=1}^{h} A^{h-i} XB^{i-1}, \end{aligned}$$

obtaining the same expression as in (2).

Also, if $0 \le t \le h$, we have

$$\begin{aligned} M^h &= M^t M^{h-t} \\ &= \begin{bmatrix} A^t & X^{(t)} \\ \mathbf{0} & B^t \end{bmatrix} \begin{bmatrix} A^{h-t} & X^{(h-t)} \\ \mathbf{0} & B^{h-t} \end{bmatrix} \\ &= \begin{bmatrix} A^h & A^t X^{(h-t)} + X^{(t)} B^{h-t} \\ \mathbf{0} & B^h \end{bmatrix}. \end{aligned}$$

Comparing this result to (1) we obtain (3). Expression (4) is proven in the same way.□

As a consequence, in the case *t = 1* we have

$$\begin{aligned} X^{(h)} &= AX^{(h-1)} + XB^{h-1}, \\ X^{(h)} &= A^{h-1} X + X^{(h-1)} B, \end{aligned}$$

and, taking *a*, *b* integers such as $a + b \ge 0$, we have

$$X^{(a+b)} = A^a X^{(b)} + X^{(a)} B^b. \qquad (5)$$

In this scheme, the key space is bound to the order of the group generated by the *M* matrices. For this reason, we present the way to guarantee that this order is big enough.

Let $f(x) = a_0 + a_1 x + \cdots + a_{r-1} x^{r-1} + x^r$ a monic polynomial in $\mathbb{Z}_p[x]$ and

$$\overline{A} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \dots & -a_{n-2} & -a_{n-1} \end{bmatrix}$$

its companion matrix. If $f(x)$ is a primitive polynomial then the order of $\overline{A}$ is exactly $p^n - 1$. Consequently, if we work in $\mathbb{Z}_p[x]$, it is possible to easily construct matrices whose order is maximum.

Odoni, Varadharajan and Sanders (see [30]) propose an extended scheme based on the construction of the block matrix

$$\overline{A} = \begin{bmatrix} \overline{A}_1 & 0 & \cdots & 0 \\ 0 & \overline{A}_2 & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \overline{A}_k \end{bmatrix},$$

where $\overline{A}_i$ is the companion matrix of f_i, being f_i for $i = 1, 2, \dots, k$, primitive polynomials of degree n_i for $i = 1, 2, \dots, k$ respectively. The order of $\overline{A}_i$ is $p^{n_i} - 1$ for $i = 1, 2, \dots, k$, therefore the order of the matrix $\overline{A}$ is $lcm(p^{n_1} - 1, p^{n_2} - 1, \dots, p^{n_k} - 1)$.

With the intention of using a matrix of this type in public key cryptography, the mentioned authors use an invertible matrix *P* obtaining a new matrix $A = P\overline{A}P^{-1}$ with the same properties.

Constructing the matrix *M* using primitive polynomials we can guarantee a certain order. Let

$$\begin{aligned} f(x) &= a_0 + a_1 x + a_2 x^2 + \cdots + a_{r-1} x^{r-1} + x^r, \\ g(x) &= b_0 + b_1 x + b_2 x^2 + \cdots + b_{s-1} x^{s-1} + x^s, \end{aligned}$$

p	*r*	*s*	*o(M)*	*p*	*r*	*s*	*o(M)*
3	32	31	30	19	16	19	39
	48	47	39		32	31	63
	64	63	47		64	63	98
	130	131	145		130	131	298
5	32	31	38	29	31	32	82
	30	33	39		47	48	97
	64	63	61		60	61	103
	130	131	184		130	131	311
7	24	27	39	31	16	15	40
	32	31	43		32	31	87
	64	63	70		64	63	111
	130	131	213		131	131	342
11	22	21	39	251	12	13	46
	32	31	50		32	31	276
	64	63	77		64	63	457
	130	131	239		130	131	1379
13	31	32	53	257	9	10	40
	47	48	63		32	31	287
	60	61	81		64	63	479
	130	131	247		130	131	1479

Table 1. Order of M, for different values of p, r and s

be two primitive polynomials in $\mathbb{Z}_p[x]$, and $\overline{A}, \overline{B}$ the corresponding associated matrices; let *P, Q* be two invertible matrices, $A = P\overline{A}P^{-1}$ and $B = Q\overline{B}Q^{-1}$.

With this construction, the order of *M* is

$$o(M) = lcm(p^r - 1,\ p^s - 1),$$

this number will be maximum if we take $p^r - 1$ and $p^s - 1$ relatively prime (see [28]).

In table 1, where the value that appears in the column *o(M)* represents the number of decimal digits (the integer 2^{128} has 39 digits), it can be observed that the values of *r* and *s* do not need to be very big to optimise the order.

It is easy to reduce a generic DLP in a cyclic group (with order $o(M)$) whose factorization is known. It is very important in the election of the group that the order is prime or at least with very big prime factors. So if $o(M)$ is a prime number, it will require on the order of $\sqrt{m}$ operations to compute the discrete logarithm in group θ.

3 The algorithms

3.1 Key exchange protocol

We will see now the proposed system of block matrices applied to the DH key exchange protocol.

Let *U* and *V* be two interlocutors who wish to exchange a key, then

1. *U* and *V* agree on $p \in \mathbb{Z}$,
   $M_1 = \begin{bmatrix} A_1 & X_1 \\ \mathbf{0} & B_1 \end{bmatrix} \in \theta$, with order m_1
   and
   $M_2 = \begin{bmatrix} A_2 & X_2 \\ \mathbf{0} & B_2 \end{bmatrix} \in \theta$, with order m_2.

2. *U* randomly generates two private keys r, s with $1 \le r \le m_1$, $1 \le s \le m_2$, computes $C = M_1^r M_2^s$ and publishes this value.

3. *V* randomly generates two private keys v, w with $1 \le v \le m_1$, $1 \le w \le m_2$, computes
$$F = M_1^{\,v} M_2^{\,w},$$
$$\begin{aligned} D &= M_1^{\,v} C M_2^{\,w} \\ &= M_1^{\,v} M_1^{\,r} M_2^{\,s} M_2^{\,w} \\ &= M_1^{\,v+r} M_2^{\,s+w} \\ &= M_1^{\,r+v} M_2^{\,w+s} \\ &= M_1^{\,r} M_1^{\,v} M_2^{\,w} M_2^{\,s}, \end{aligned}$$
and publishes this matrix.

4. *U* calculates $M_1^{\,-r} M_2^{\,-s}$ and
$$\begin{aligned} F &= M_1^{\,-r} D M_2^{\,-s} \\ &= M_1^{\,-r} M_1^{\,r} M_1^{\,v} M_2^{\,w} M_2^{\,s} M_2^{\,-s} \\ &= M_1^{\,v} M_2^{\,w}. \end{aligned}$$

5. The public key of *U* and *V* are respectively *C* and *D*.

In this way, the key shared by *U* and *V* is *F*, now both interlocutors, share a common and secret element.

An attacker could know *p* and *M*, but to obtain the shared secret would have to face a problem with a complexity similar to that of the DLP (see [19]).

3.2 Data encryption

We have to start from the same public and private elements seen previously in the key exchange protocol (which we suppose already done).

The interlocutor *U* wishes to, privately, send a message to *V*. The message must be coded as a matrix $\Delta \in Mat_{r\times s}(\mathbb{Z}_p)$.

Encryption:

1. U builds the matrices $T = \begin{bmatrix} A_1 & \Delta \\ \mathbf{0} & B_1 \end{bmatrix}$ and *F*, that are invertible since A_1, A_2, B_1 and B_2 are invertible too.
2. *U* computes matrix $C = TF$ and sends this matrix to *V*.

Decryption:

1. *V* computes the inverse of the matrix *F*.
2. *V* obtains T carrying out the product CF^{-1}.
3. *V* recovers the message Δ selecting, the respective block of T.

With this, the functions of encryption and decryption of the interlocutor *V* would be respectively

1. $E_{k_2}(\Delta) = TF$.
2. $D_{k_2}(C) = CF^{-1} = T$.

With the appropriate quick exponentiation algorithms (see [18]), the powers of the matrices can be computed efficiently.

The complexity of the problem that an attacker would face is in the order of that of the DLP, acting, in effect, as a deterrent for a possible attack.

3.3 Signature scheme

We propose a digital signature scheme that requires the original message in order to verify the signature.

The scheme, that follows, is based on the ElGamal (see [21]) digital signature scheme.

We suppose that the users *U* and *V* have exchanged the key *F*, and *U* has sent the message Δ to *V*, according to the previous protocol. If the transmitter *U* wishes to digitally sign the message Δ proceeds in the following way

1. *U* generates a random number *r*.
2. *U* computes F^r.
3. With T computes $Q = T - F^r$.
4. The digital signature is (r, Q).

If the receiver wishes to verify the digital signature of *U*, he proceeds in the following way

1. *V* computes F^r and then $Q + F^r = T$
2. *V* extracts the corresponding block of T named *Y* and compares Δ and Y, turning out to be an authentic signature if $\Delta = Y$ and false if $\Delta \neq Y$.

4 Integral Kernel

Most protocols in digital business employ symmetric cryptography to transfer large quantities of data, while asymmetric cryptography is used to swap session keys, digital signatures, etc. Additionally, hash functions can be used in order to improve efficiency and data integrity.

Our proposal can be integrated with these basic components, obtaining a security kernel which can be the basis of many protocols. Since cryptographic algorithms are extremely diverse in nature, scope and requirements, this integration is highly beneficial since it allows for cheap mass production, and ease of design of new secure systems which could use the kernel as a black box.

The cryptographic kernel is based on the powers of a block upper triangular matrix, which is a very flexible technique. It can be adjusted to satisfy memory and speed requirements and be implemented successfully either in hardware or software. Another advantage is that the same basic mathematical scheme can be used to build private key cryptosystems, public key cryptosystems and hash functions. Therefore, we only require implementing this technique once in order to provide these three types of algorithms, integrating a full cryptographic kernel in a single low cost device. This is a remarkable new concept that shows how useful this technique can be in cryptography.

4.1 The Symmetric Component

To cipher large amounts of information efficiently, we need a private key cryptosystem. For that purpose, we can build a stream cipher using the mathematical base of the kernel by taking advantage of its great randomness properties as shown previously. We first create a good pseudorandom generator and, once we have that, we use it as the keystream generator in a Vernam cipher scheme, taking the seed of the generator as the key of the stream cipher. This pseudorandom generator can also be used to generate general purpose random numbers such as session keys, challenge values, etc.

For each matrix $X^{(h)}$, for $h = 2, 3, \ldots,$ we establish a bit extraction operation which can be as simple as adding all the elements of $X^{(h)}$ obtaining a new element $x^{(h)}$, for $h = 2, 3, \ldots,$ in Z_p from which we take the least significant bit, $b^{(h)}$, of its binary expression; or as complex as required and taking as many bits per iteration as needed. In this way, we have the sequence of bits

$$b^2, b^3, b^4, \ldots$$

This sequence is then filtered by the following process, improving security and bias:

$$c^{(i)} = b^{(i)} \oplus c^{(i-1)},\ i = 2, 3, 4, \ldots; c^{(1)} = 0.$$

Once we have a proper keystream, ciphering the plaintext is as simple as XORing the keystream with it bit by bit. In order to decipher we XOR the keystream again with the ciphertext and retrieve the original plaintext. The seed of the generator is shared by both parties so that they can reproduce correctly the keystream.

The algorithm has been compared with the BBS pseudorandom generator (see [1]), achieving comparable results in terms of the randomness of the keystream and being a lot faster (in the order of 10^3 times). A comparison with the RC4 stream cipher has also yielded comparable results in randomness and similar speed in software. Further optimizations are being studied and could make the algorithm even faster.

4.2 The Asymmetric Component

Defining the operator $\otimes$ as

$$X^{(a)} \otimes X^{(b)} = X^{(a+b)}, \quad (9)$$

set $G = \{X^{(0)}, X^{(1)}, X^{(2)}, X^{(3)}, \ldots\}$ has a finite group structure and its order can be taken as large as needed to make our scheme secure.

The key exchange scheme between two users *U* and *V*, proposed for our kernel, is:

1. *U* and *V* accord values for *p*, *n*, *A*, *B* and *X*
2. *U* generates a random number *k* and computes A^k, B^k and $X^{(k)}$
3. *V* generates a random number *m* and computes A^m, B^m and $X^{(m)}$
4. The numbers *k* and *m* are respectively the private keys of *U* and *V*
5. The pairs $(X^{(k)}, B^k)$ and $(X^{(m)}, B^m)$ are respectively the public keys of *U* and *V*
6. *U* computes $X^{(k+m)} = A^k X^{(m)} + X^{(k)} B^m$
7. *V* computes $X^{(m+k)} = A^m X^{(k)} + X^{(m)} B^k$

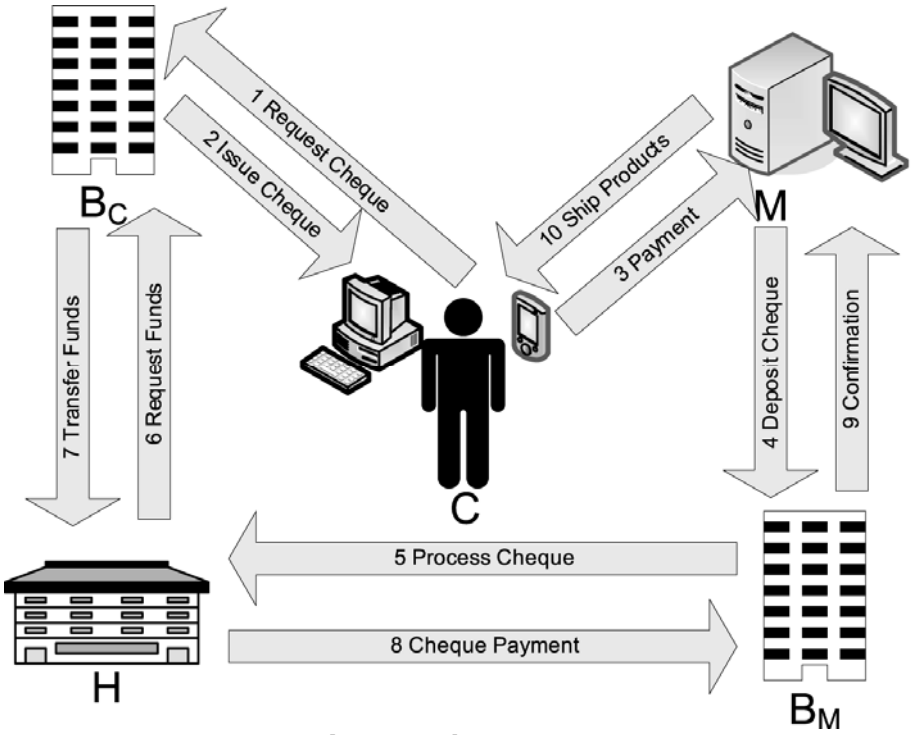

Figure 1. Electronic cheque payment system

With this scheme users U and V share matrix $X^{(k+m)}$ in G.

The computation of A^k, A^m, B^k, B^m, $X^{(k)}$ and $X^{(m)}$ can be done efficiently adapting the existing quick exponentiation algorithm in Z_p.

It is computationally infeasible, for an attacker, to know the shared key $X^{(m+k)}$ without the previous knowledge of k and m, because the problem the attacker would be facing is in the order of complexity of the discrete logarithm problem.

The scheme described previously can be adapted to perform digital signature using a similar technique to the ElGamal cryptosystem. Since the kernel requires little resources, it is suitable for low power or low cost environments.

4.3 The Hash Component

Taking the mathematical base of the kernel we can also build a hash function. We can use the pseudorandom generator as a diffusion and compression mechanism accumulating its results over a fixed length register. The stream cipher proposed can be also adapted to perform a hash function in the way shown in [29, 35].

4.4 Applications

This integral security kernel can be used by any digital business protocol requiring security at any level, like A/V content distribution systems, anonymous peer to peer systems, certified email systems, online payment systems, etc. It can be implemented on any platform (PC, dedicated hardware, PDA, latest generation of cell phones, smart cards) and data transport system (Internet, wireless networks, satellites, terrestrial digital transmissions, etc.), being capable of adapting to the technological evolutions in the communications sector. Application examples can be seen in [2-9, 11, 14].

It is efficient and easy to implement either in hardware or software and requires very little resources, making possible its implementation in a wide spectrum of devices, especially those of low cost. In this way, confidentiality (ciphered information), integrity (no alteration warranty) and authentication (identity verification using digital signature) are assured in the communications.

As an application example of the kernel, we can take an electronic cheque payment system involving five parties: the client (C) and his bank (BC), the merchant (M) and his bank (BM), and a clearing

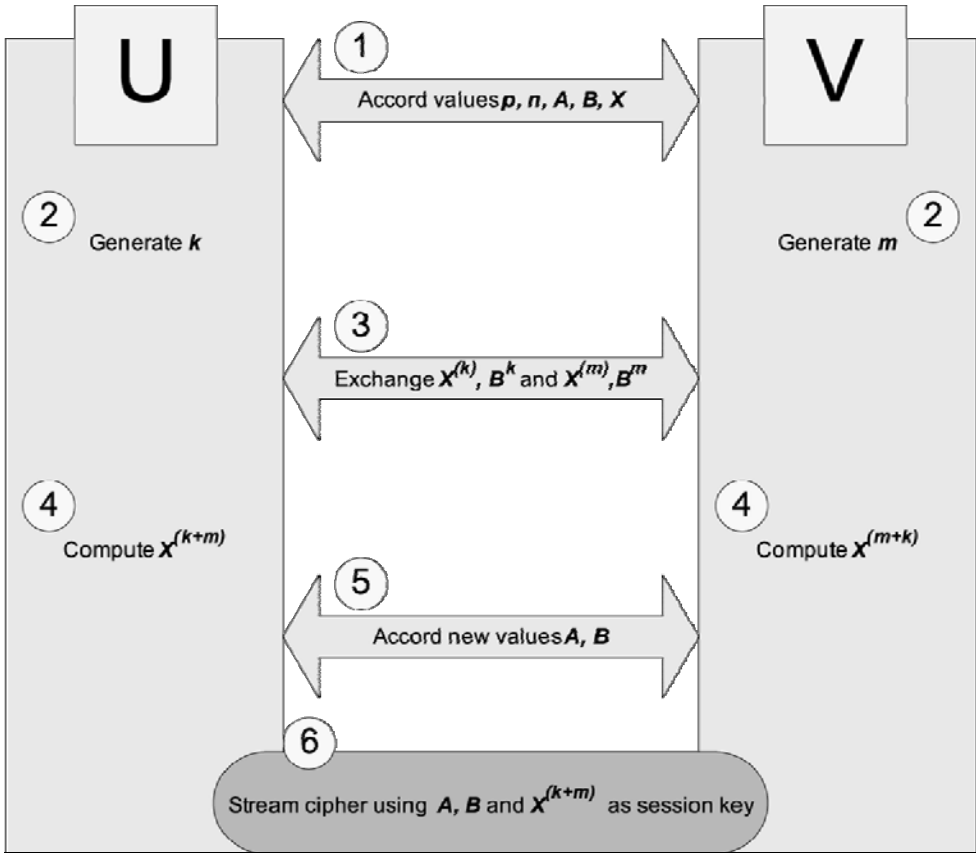

Figure 2. Secure communication scheme.

house (H) that reconciles bank transfers and processes cheques.

- The client purchases some goods and sends the corresponding electronic cheque to the merchant.
- The merchant sends the check to his bank to validate and deposit it.
- The merchant's bank sends the check to the clearing house in order to receive payment.
- The clearing house requests the required funds from the client's bank. Then, both banks update the corresponding accounts.
- Once the electronic cheque has been validated and correctly processed, the merchant proceeds to send the goods to the client.

For each of the communication channels established between the different parties (see figure 1), we need to guarantee confidentiality, authentication, and information integrity. For that purpose we require the usage of symmetric and asymmetric cryptography, a random generator and a hash function (operations offered by the proposed kernel).

In this way, the kernel provides all the means for a secure communication between two parties, as shown in figure 2:

- First, both parties must establish values for p, n, A, B and X.
- Then, U generates a random k of sufficient length, V generates m in the same way.
- U sends V values $X^{(k)}$ and B^k, V does the same sending U the values $X^{(m)}$ and B^m.
- U computes $X^{(k+m)}$ and V computes $X^{(m+k)}$, since both parties reach the same result they now share this secret key.
- U and V can agree on new values for A and B.
- Taking the new A and B, along with $X^{(k+m)}$, we have the session key for our secure channel, using the kernel's stream cipher.

5 Conclusions

With the aim of creating systems that allow increasing the computational cost required to break certain well known problems, we have presented a public key cryptosystem based on a generalization of the DLP for block upper triangular matrices with elements in $\mathbb{Z}_p$, which presents the advantage of reducing the required key length for a given level of security, this is achieved as a consequence of the usage of the quick exponentiation and the algebraic properties of θ.

This cryptosystem provides an efficient protection against common attacks without the need for bigger key sizes.

For the development of this cryptosystem we have defined a set of matrices θ constructed using primitive polynomials. Therefore, we can work with big groups, requiring neither enormous matrices nor high numbers.

Given two parties, the key exchange protocol guarantees that both parties share a secret element of set *G*; the public key cryptosystem defined assures data confidentiality and the digital signature scheme guarantees authentication and integrity.

References:

[1] Aguirre, J-V., Alvarez, R., Tortosa, L., Zamora, A. Fast Pseudorandom Generator based on Packed Matrices. WSEAS Information Security and Privacy (2007) 98-101

[2] Aguirre, J-V., Alvarez, R., Tortosa, L., Zamora, A. Secure Lightweight P2P Multiconferencing. WSEAS Transactions on Communications, vol. 6-1 (2007) 195-200

[3] Aguirre, J-V., Alvarez, R., Sanchez, J., Zamora, A. Broadcast Multiplexing and Subchanneling for Secure P2P Multiconferencing. WSEAS Transactions on Computers, vol. 6-3 (2007) 522-527

[4] Aguirre, J-V., Alvarez, R., Noguera, J-V., Zamora, A. A Secure Remote Database Backup System. WSEAS Artificial Intelligence, Knowledge Engineering and Databases (2006) 43-46

[5] Aguirre, J-V., Alvarez, R., Noguera, J-V., Zamora, A. A Database Backup System with Secure Remote Data Transmission. WSEAS Transactions on Information Science Applications, vol. 4-3 (2006) 796-801

[6] Aguirre, J-V., Alvarez, R., Noguera, J., Tortosa, L., Zamora, A. Secure VoIP and Instant Messaging on Small PDA Devices. WSEAS Transactions on Computers, vol.5-1 (2006) 171-176

[7] Aguirre, J-V., Alvarez, R., Sanchez, J., Zamora, A. Silence Detection in Secure P2P VoIP Multiconferencing. WSEAS Information Security and Privacy (2006) 11-14

[8] Aguirre, J-V., Alvarez, R., Tortosa, L., Zamora, A. Lightweight Peer-to-Peer Secure Multi-Party VoIP Protocol. WSEAS Information Security and Privacy (2006) 7-10

[9] Aguirre, J-V., Alvarez, R., Noguera, J-V., Tortosa, L., Zamora, A. A Viability Analysis of a Secure VoIP and Instant Messaging System on a Pocket PC. WSEAS Information Security and Privacy (2005) 218-223

[10] Alvarez, R., Martinez, F-M., Vicent, J.F., Zamora, A. A New Public Key Cryptosystem based on Matrices. WSEAS Information Security and Privacy (2007) 36-39

[11] Alvarez, R., Oliver, J., Vicent, J., Zamora, A. Improving GSM Security for Voice and Text Data Transmission. WSEAS Transactions on Computers, vol. 5-1 (2006) 165-170

[12] Alvarez, R., Tortosa, L., Vicent, J-V., Zamora, A. Block Upper Triangular Matrices for Authentication and Integrity. WSEAS Transactions on Mathematics, vol.4-4 (2005) 339-346

[13] Alvarez, R., Tortosa, L., Vicent, J-F., Zamora, A. A Public Key Cryptosystem based on Block Upper Triangular Matrices. WSEAS Information Security and Privacy (2005) 163-168

[14] Alvarez, R., Oliver, J., Vicent, J-F., Zamora, A. Secure Communication System over a GSM Network. WSEAS Transactions on Computers, vol.5-1 (2005) 171-176

[15] Alvarez, R., Tortosa, L., Vicent, J-F., Zamora, A. An Integral Security Kernel. WSEAS Transactions on Business and Economics, vol. 1-3 (2004) 241-246

[16] Álvarez, R., Climent, J.J., Tortosa, L., Zamora, A. Un generador matricial de claves frente a Blum Blum Shub, RECSI'04 (2004) 113-123

[17] Álvarez, R., Climent, J.J., Tortosa, L., Zamora, A. A Pseudorandom Bit Generator Based on Block Upper Triangular Matrices. LNCS Web Engineering, vol. 2722 (2003) 299-300

[18] Blake, I., Seroussi, G. Smart, N. Elliptic Curves in Cryptography. London Mathematical Society Lecture Notes Series 265. Cambridge University Press. 1999.

[19] Coppersmith, D., Odlyzko, A., and Schroeppel, R. Discrete logarithms in GF(p). Algorithmica 1-15. 1986.

[20] Diffie, W., Hellman, M. New directions In Cryptography. IEEE Trans. Information Theory. 22: 644-654. 1976.

[21] Elgamal, T. A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Trans. Inform. Theory. 31: 469-472. 1985.

[22] Gordon, D. M. A Survey of Fast Exponentiation Methods. Journal of Algorithms. 27: 129–146. 1998.

[23] Hellman, M.E., Reyneri, J.M. Fast computation of discrete logarithm in GF(p). Advances in cryptology: Proceedings of GRYPTO'82 Plenum Press. 3-13. 1983.

[24] Hoffman, K., Kunze, R. Linear Algebra. Prentice-Hall. New Jersey. 1971.

[25] Koblitz, N. A Course in Number Theory and Cryptography. Springer-Verlag. 1987.

[26] Lidl, R., Niederreiter, H. Introduction to Finite Fields and Their Applications. Cambridge University Press. 1994.

[27] McCurley K. The discret logarithm problem. Crytology and Computational Number Theory, Proceedings of Symposia in Applied Mathematics. 42: 49–74. 1990.

[28] Menezes, A., van Oorschot, P., Vanstone, S. Handbook of Applied Cryptography. CRC Press. Florida. 2001.

[29] Menezes, A., Wu, Y-H. The Discrete Logarithm Problem in GL(n,q). Ars Combinatoria. 47: 22-32. 1997.

[30] Odoni, R. W. K., Varadharajan, V., Sanders, P. W. Public Key Distribution in Matrix Rings. Electronic Letters. 20: 386-387. 1984.

[31] Pohlig. S, Hellman, M. An improved algorithm for computing logarithms over GF(p) and its cryptographic significance. IEEE Trans. 24: 106-110. 1979.

[32] Pollard, J.M. Monte Carlo methods for index computation (mod p). Math. Computation. 32: 918-924. 1978.

[33] Rivest, R., Shamir, A., Adleman, L. A Method for Obtaining Digital Signatures and Public Key Cryptosystems. ACM Communications. 21: 120-126. 1978.

[34] Shanks, D. Class number, a theorey of factorization and generation. Number Theory Institute. Proc. Symposium pure Mathematics. American Mathematics Society. 20: 415-440. 1981.

[35] Stallings, W. Cryptography and Network Security. Principles and Practice. Third Edition. Prentice Hall. New Jersey. 2003