A Three-Pass Protocol for Cryptography Based on Padding for Wireless Networks

André Gustavo Degraf Uchôa, Marcelo Eduardo Pellenz, Altair Olivo Santin, Carlos Alberto Maziero
Graduate Program on Computer Science – PPGIa / Pontifical Catholic University of Paraná – PUCPR
Curitiba – Paraná – Brazil
{agdu, marcelo, santin, maziero}@ppgia.pucpr.br

Abstract — **This paper proposes an alternative cryptography protocol based on padding for wireless networks. It uses an orthogonal set of rotation matrices and a three-pass exchanging protocol to reach the encryption. The communicating parties do not need to know cryptographically nothing from each other in order to guarantee the communication privacy. The security of the algorithm is based on the continuous changing of the orthogonal matrix set used on the encryption process. The proposed protocol does not require any kind of keys pre-distribution. This feature is desirable for wireless ad hoc networks, where there is no predefined infrastructure, as required on classical secure channel encryption. The prototype shows that the proposal is feasible and can be advantageous when compared to One-Time Padding.**

Keywords: Padding-based cryptography, Wireless Networks, No key protocol, Orthogonal Matrices

I. INTRODUCTION

In early times, communication among nodes in a computer or telecommunication network was usually done through protocols that transmitted plain text over a dedicated physical medium. With the evolution and popularization of computer systems and communications, especially the Internet, many services were developed, almost all using the TCP/IP architecture. However, in the Internet it is difficult to keep privacy and confidentiality of the information being transported, due to its public nature. Therefore, cryptographic techniques assume an important role to protect private data being transported over a public network.

The classical cryptographic techniques are classed in two kinds of cryptosystems: symmetric and asymmetric. In symmetric systems, a single secret key should be shared among the communicating nodes. In asymmetric systems, each node has two keys, one of them is kept secret and the other one is publicly available. In such case, there is no need to share the same key to communicate each other. The advantage of symmetric algorithms over asymmetric ones is their low computational complexity [1], [2].

The implementation of cryptography protocols without the need of key exchange is still a less investigated area. The robustness of such methods is based on the introduction of padding and frequent key changing. Shamir [1] presented the first idea of an algorithm in this area, named as “No Key Protocol”, in an unpublished paper.

Today, devices with wireless connectivity require even more for cryptographic protocols, due to the facility to capture packets transmitted between nodes over the air medium. A classical symmetric cryptography system can provide the properties of confidentiality and integrity, but it is not a simple task to ensure such properties in an ad-hoc wireless network (a set of nodes compounding a temporary network without a centralized administration or any kind of previously defined infrastructure). In such environment, a node can also become a router to forward packages from one node to another.

The implementation of public key cryptographic algorithms in ad-hoc wireless networks is not a simple task due the necessity of a centralized entity to manage the public keys required in such kind of context [3].

Another method to implement security in wireless networks is through symmetric cryptographic algorithms, in which a secret key should be shared among the participating nodes. The problem in this approach is the key distribution, because one node must send the key to another one without encryption. This constitutes a weakness, because the key can be easily intercepted in a wireless environment. One way to circumvent this problem is to perform a pre-distribution of the key. In this case, the manufacturers should previously install predefined secret keys into the hardware devices [4]. Such key pre-distribution restricts the possibility to choose the kind of cryptographic algorithm to be used, as well as the key size. Additionally, if an intruder has physical access to the mobile device, the pre-defined keys can be compromised.

The work presented in this paper proposes a three-pass protocol for cryptographic based on padding, aiming to allow any two nodes to establish a secure communication channel without a previous key distribution. The proposed protocol is based on a set of *N*-dimensional orthogonal matrices, which are employed in the encryption and decryption procedures, and a protocol that requires three-pass to securely transport data from one node to another. Using this technique, nodes can start a communication without any previous knowledge, neither a secret key nor a public key of a server.

This paper is organized as follows: Section II presents a brief overview about cryptographic protocols that do not use keys, as well as the message exchanges between the nodes participating of such kind of protocols. The proposed algorithm is presented in Section III. Section IV describes the algorithm implementation prototype. In Section V some relevant related works are presented. In section VI some discussions about the proposed protocol are carried out. Finally, conclusions are drawn in Section VII.

II. NO KEY PROTOCOLS

The cryptography protocol proposed by Shamir [1] – denominated either *no-key protocol* or *three-pass protocol*, which derived from an unpublished previous work – requires keys that can be applied on the message in any order, producing the same encryption result. This property is also denoted *transitive keys*, and is defined in equation (1). Consider a node *i* with encryption and decryption keys denoted by $C_i\{\cdot\}$ and $C_i^{-1}\{\cdot\}$, respectively. Considering the keys from a commutative set *E*, the following property must be verified:

$$C_i^{-1}\{C_j\{C_i\{M\}\}\} = C_j\{M\}, \quad \forall C_j\{\cdot\} \in \mathrm{E} \qquad (1)$$

When a node *A* wants to send a message to node *B*, node *A* encrypts the message using the key $C_A\{\cdot\}$ (Figure 1) and send the result $C_A\{M\}$ to node *B*. Node *B* encrypts the received data using key $C_B\{\cdot\}$ and returns $C_B\{C_A\{M\}\}$ back to *A*. Node A applies $C_A^{-1}\{\cdot\}$ on the received message, $C_B\{C_A\{M\}\}$, getting $C_B\{M\}$, and sends it back to node *B*. Node *B* can then recover the original data, *M*, by applying the key $C_B^{-1}\{\cdot\}$ on $C_B\{M\}$. Using this scheme, there are no keys as those used in symmetric or asymmetric cryptosystems, so this approach is considered to be distinct from classical cryptography ones.

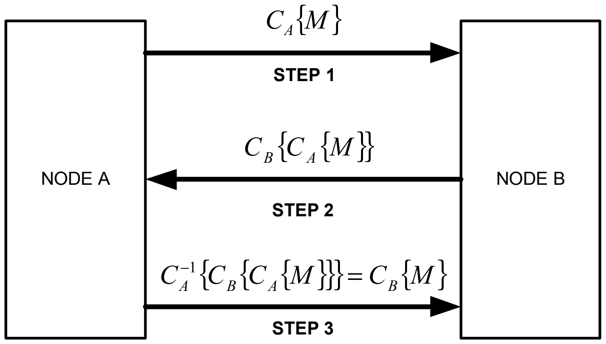

Figure 1. Shamir's three-pass protocol.

According to Shamir [1] – which uses a *commutative cipher* based on the modulus (*mod*) of a large prime number *p* – the communicating parties, *A* and *B*, make an agreement on a prime number *p* to be used. Afterwards, nodes *A* and *B* generate a pair of relatively prime numbers (a, a^{-1}) and (b, b^{-1}), respectively. These pairs should satisfy the properties $a \cdot a^{-1} = 1 \bmod p$ and $b \cdot b^{-1} = 1 \bmod p$, where $a, b \in Z_p^*$, (Z_p^* is the set of strictly positive integers).

To encrypt a message *m*, node *A* computes $c = m^a \bmod p$ and sends c to node *B*. Node *B* will compute $x = c^b \bmod p$ and send x back to node *A*. Node *A* will then compute $y = x^{a^{-1}} \bmod p = m^b$. Afterwards, node *A* will send the message y back to *B,* so node *B* can perform the operation $m = y^{b^{-1}} \bmod p = m^{b.b^{-1}} = m^1$, extracting the original message *m* (plain text).

III. PROPOSED ALGORITHM

The proposed algorithm is based on a three-pass block encryption procedure, using an orthogonal set of rotation matrices [5][6]. The orthogonal rotation matrices are composed by bi-dimensional sub-matrices ordered along the main diagonal. All other matrices elements are nulls. Hence an orthogonal matrix $\mathbf{O}_i$ with dimension $N = 2n$ can be represented in a pseudo-diagonal form [7], defined by (2).

$$\mathbf{O}_i = diagonal\left[\begin{pmatrix} \cos\theta_{i1} & sen\theta_{i1} \\ -sen\theta_{i1} & \cos\theta_{i1} \end{pmatrix}, \begin{pmatrix} \cos\theta_{i2} & sen\theta_{i2} \\ -sen\theta_{i2} & \cos\theta_{i2} \end{pmatrix}, \ldots, \begin{pmatrix} \cos\theta_{in} & sen\theta_{in} \\ -sen\theta_{in} & \cos\theta_{in} \end{pmatrix}\right] \quad (2)$$

The orthogonality property implies on equality between the transposed matrix and its inverse: $\mathbf{O}_i^T = \mathbf{O}_i^{-1}$ [8]. Using a simplified notation, an orthogonal matrix can be represented by a vector defining the rotation angles of each sub-matrix of the main diagonal, that is, $\bar{\theta}_i = (\theta_{i1}, \theta_{i2} \ldots, \theta_{in})$.

For a description of the proposed algorithm, the data block (message) will be denoted by a vector $\mathbf{x}_M$, the encrypted block by a vector $\mathbf{y}_i$ and the orthogonal matrices used by each node as $\mathbf{O}_i$ and $\mathbf{O}_i^C$, where $\mathbf{O}_i \cdot \mathbf{O}_i^C = I$ (*I* denotes the identity matrix). Therefore, the three-pass encryption procedure of Figure 1 can now be defined using the *N*-dimensional rotation matrices. On the proposed algorithm, the data vector is initially encrypted by node *A*: $C_A\{M\} = \mathbf{y}_A = \mathbf{x}_M \cdot \mathbf{O}_A$. This represents one *N*-dimensional rotation of data vector using the rotation matrix $\mathbf{O}_A$.

In the second step, node *B* computes $C_B\{\mathbf{y}_A\} = \mathbf{y}_B$, $\mathbf{y}_B = \mathbf{y}_A \cdot \mathbf{O}_B$ by applying the rotation matrix $\mathbf{O}_B$. This implies that original information is rotated by two distinct and independent matrices, which represent the "encryption keys" for nodes *A* and *B*. In the third step, node *A* could remove its rotation applying one complementary rotation matrix, denoted by $\mathbf{O}_A^C$, thus undoing its encryption: $\mathbf{y}_C = \mathbf{y}_B \cdot \mathbf{O}_A^C = \mathbf{x}_M \cdot \mathbf{O}_B$. Node *B* recovers the information data by applying its complementary rotation matrix $\mathbf{O}_B^C$. Each node should choose rotation angles for the matrices $\mathbf{O}_i$ in a pseudo-random manner. Additionally, such angles can be modified for each transmission.

It is important to point out that this encryption protocol using orthogonal rotation matrices can be easily compromised if a third party monitors the communication channel and captures the three exchanges of data information. It is possible to determine the rotation angles used in each matrix $\mathbf{O}_A$ from vectors $\mathbf{y}_A$ and $\mathbf{y}_C$ and thus to discover the original data vector. This kind of problem can be eliminated by using two additional matrices $\mathbf{A}_1$ and $\mathbf{A}_2$, named *nonces matrices*; on (3) one such matrix is given as example. In that case, the tasks executed on the three-pass should be redefined by (4), (5), and (6).

$$\mathbf{A} = \begin{bmatrix} a_i & b_i \\ -b_i & a_i \end{bmatrix} \tag{3}$$

$$\mathbf{y}_A = [\mathbf{x}_M \cdot \mathbf{O}_A \cdot \mathbf{A}_1] \tag{4}$$

$$\mathbf{y}_B = [\mathbf{y}_A \cdot \mathbf{O}_B \cdot \mathbf{A}_2] \tag{5}$$

$$\mathbf{y}_C = \lfloor \mathbf{y}_B \cdot \mathbf{O}_A^C \cdot \mathbf{A}_1^T \div \lambda_{i1} \rfloor \tag{6}$$

Node *B* recovers the original data information by applying the operation $\mathbf{x}_M = \lfloor \mathbf{y}_C \cdot \mathbf{O}_B^C \cdot \mathbf{A}_2^T \div \lambda_{i2} \rfloor$, where $\lambda_i = (a_i^2 + b_i^2)$ and subscript *T* represents transpose matrix. The usage of *nonces matrices* prevents the possibility of transmitted message replay and prevents an *attacker* from recovering the transmitted information through operations on the encrypted vectors exchanged during the transmission procedure.

A. Example

This section presents the pseudo-code of the implemented algorithm, as shown in Figure 2, and a simplified numerical example of the proposed cryptographic algorithm.

Consider the communication between two nodes *A* and *B* over an ad-hoc network, where node *A* wants to send a message with text "GH" to node *B*. In such proposed scenario, every plain text character will be represented by its respective ASCII code. The data vector (message) is represented by (7) – variable INFO in Figure 2.

$$\mathbf{x}_M = ['G','H'] = [71, 72] \tag{7}$$

$$\mathbf{O}_i = \begin{bmatrix} \cos\theta_{i1} & sen\theta_{i1} \\ -sen\theta_{i1} & \cos\theta_{i1} \end{bmatrix} \tag{8}$$

For practical reasons, bi-dimensional rotation matrices will be used, as defined by (8). In this case, nodes *A* and *B* must randomly choose angles θ_1 and θ_2 (function ANGLEGEN in Figure 2) with their respective complementary angles $\theta_1^c = 2\pi - \theta_1$ e $\theta_2^c = 2\pi - \theta_2$. To numerically exemplify the algorithm procedures, it will be arbitrated $\theta_1 = \pi/6$ and $\theta_2 = \pi/3$, although these angles should be generated randomly to minimize their predictability.

$$\mathbf{y}_A = [-119.244 \quad 344.537] \tag{9}$$

Consider the chosen parameters values for *nonces matrices* as being $a_1 = 3$ and $b_1 = 2$ (for node *A*), $a_2 = 5$ and $b_2 = 6$ (for node *B*). In real implementations the values of a_1, a_2, b_1, b_2 should be random and thus unpredictable (function FRAND, Figure 2).

By applying (4) on $\mathbf{O}_A$ and $\mathbf{A}_1$ the vector $\mathbf{y}_A$ is obtained, as shown in (9) (functions FRAND, ANGLEGEN and MULTMAT in Figure 2). This vector is transmitted to node *B* through the communication channel, by function SEND in Figure 2.

```
PROPOSED ALGORITHM
(Here it is presented just the operations from node A side; their
counterparts on node B side follow a very similar scheme.)

VARIABLES Ai, Bi, ANG1, COMP, RES1, RES2, INFO

FRAND = { Ai = CHOOSE N from 1 ≤ N ≤ 2^64;
          Bi = CHOOSE N from 1 ≤ N ≤ 2^64 }

ANGLEGEN = { ANG1 = CHOOSE R from 0 ≤ R ≤ 2π;
          COMP = 2π - ANG1}

MULTMAT()= {
          VARIABLES INFO, ORTMAT, ANG1
          RES1 = MULTMAT(INFO, ORTMAT, ANG1, Ai, Bi) }

SEND ( NODEX = RES1)
(Where x could be either Node A or B)

RECEIVE (RES2 = NODE X * RES1)
(Where x could be either Node A or B)

MULTMAT() = {
          RES1 = MULTMAT(RES2, ORTMAT, ANG1, Ai, Bi); }
```

Figure 2. Pseudo code of the implemented algorithm.

The second step, given by (5), generates the vector shown in (10), which is transmitted back to node *A*, where node B uses function FRAND, ANGLEGEN, MULTMAT and SEND to perform operations similarly done by node *A* (Figure 2).

$$\mathbf{y}_B = [-2204.000 \quad -1803.000] \tag{10}$$

In the third step, node *A* will perform the operation $\mathbf{y}_C = \lfloor \mathbf{y}_B \cdot \mathbf{O}_A^C \cdot \mathbf{A}_1^T \div \lambda_{i1} \rfloor$, where $\mathbf{O}_A^C$ is the orthogonal matrix with complementary angle θ_1^c, and $\mathbf{A}_1^T$ is the transpose matrix of $\mathbf{A}_1$, obtaining vector (11) that is transmitted to node *B*. Following the same idea, this is done by function MULTMAT with variable COMP (complementary angles) and function SEND from Figure 2.

$$\mathbf{y}_C = [-719.196 \quad 326.316] \tag{11}$$

Finally, node *B* receives the encrypted vector $\mathbf{y}_C$ (this is done by function RECEIVE from Figure 2) and perform the operation $\mathbf{x}_M = \lfloor \mathbf{y}_C \cdot \mathbf{O}_B^C \cdot a_2^T \div \lambda_{i2} \rfloor$, where $\mathbf{O}_B^C$ is the orthogonal matrix with complementary angle θ_2^c, and $\mathbf{A}_2^T$ is the transpose matrix of $\mathbf{A}_2$, which is done by function MULTMAT with variable COMP (complementary angles). The result of this operation is shown in (12) and represents the original message sent by node *A*. For node *B* it is only necessary to transform the recovered information sequence from the numerical format to its corresponding plain text "GH".

$$\mathbf{x}_M = [71, 72] = ['G','H'] \tag{12}$$

IV. Implementation Issues

This section discusses some implementation aspects of the prototype for the proposed scheme. The first version of the proposed scheme did not consider the multiplication of encrypted data by *nonces*. However, a simple analysis of the three-pass procedure showed that the algorithm security can be easily compromised if the three messages exchanged are captured – as discussed in Section III, an eavesdropper can easily recover the original data information without knowing the rotation angles. Then it was added a multiplication by *nonces matrices* (pseudo-random numbers), containing values ranging from 1 to 2^{64}. In this case, the only identified way one can compromise the protocol security is by applying brute force attacks. This attack technique is effective only if the transmitted values are readable ASCII characters; if any kind of scrambler technique is applied before encryption, even brute force attacks will become infeasible.

The prototype was implemented in C Language [9], using the MPFR library [10] to support high precision floating-point numbers [11]. Due to the required high precision floating-point numbers on this method, only using such library it was possible to measure the limit of bytes per block that can be encrypted at once. Extensive tests shown that the precision limits of MPFR library allowed data blocks with up to 24 bytes (192 bits).

The function that generates the matrix angles computes the value $2\pi/k$, where π has a precision of 200 bits in the MPFR library, and k is a 32-bit pseudo-random number. Therefore, the generation space of encrypted blocks is 224 bits (192 bits + 32 bits). These 224 bits obtained on the matrix are multiplied by the *nonce*s *matrices* (a_i and b_i parameters ranging from 1 to 2^{64}), resulting in a final data block of 352 bits (224 + 64 + 64).

V. Related Works

Conceptually, cryptographic systems do not introduce padding and is defined a set of transformations of one space (the set of possible messages) into a second space (the set of possible cryptograms). Security is based on some secret, like a cryptographic key, which need to be known by both communicating parties. The technique proposed in this paper is based on another encryption principle, which was not been much explored until now, as discussed in the following.

In the technique proposed by Shamir and described by [1], the key is continuously modified and three exchanges are needed to transmit data information. The algorithm introduces padding into information being transmitted and does not require the use of cryptographic keys. That technique is based on exponential functions with modulus p. Thus, before the message exchange starts, it is required that both nodes agree over a large prime number p to be used on message operations, as shown on Section II. The main advantage of such technique is that there is no need to previously exchange any kind of cryptographic keys, although the operations using exponential functions on modulus p generate a lot of padding. Additionally, there is the requirement to previously "share" some information between the communicating parties, the prime number p.

The *One-Time Pad* (OTP) algorithm [1][12] is an example of algorithm that uses pseudo-random numbers to encrypt the data information. For every message (plain text), the algorithm generates a key (a pseudo-random number) with the same size of the message. The encrypted text is obtained by executing byte-wise XOR operations between each byte of message and its corresponding byte on the key. Supposing that an attacker can capture the encrypted text, she should make a brute force attack to recover the original message. Additionally, the attacker would obtain a huge amount of possible decrypted messages and would not know which one is the correct, turning the scheme hardly breakable [13].

The OTP algorithm presents the advantage of being conceptually unbreakable, due to fact that as bigger is the key size, bigger will also be the range of possible correct messages. However, constant key changing is needed in order to avoid dictionary attacks. In addition, the average padding generated by the OTP algorithm is around 100% of the original message size.

VI. Evaluation

In the same way that OTP algorithm [12] continuously changes the key used to transmit data through the communication channel, in our method new values for the rotation angles and the *nonces* parameters are generated for each new message to be transmitted. However, those values do not need to be transmitted over the communication channel and neither is required to be previously known.

Based on our prototype, the impact of nonce range was investigated to determine the average total *padding* (considering the sum of padding for all three messages exchanged) necessary to transmit an information data block. Table 1 summarizes the average percentage of padding (obtained through 1000 complete transmissions with the same number of bits for a_i and b_i parameters) considering parameters precision of 8, 16, 32, and 64 bits.

TABLE I. THE PERCENT OF PADDING FOR NONCE RANGING FROM 8 UP TO 64 BITS.

Parameters Range	**$1 - 2^8$**	**$1 - 2^{16}$**	**$1 - 2^{32}$**	**$1 - 2^{64}$**
Average Padding	13%	31%	63%	124%

The padding of 13%, for example, was obtained summing the padding of 3.5% from first exchange, to a padding of 6 % from the second exchange, and to a padding of 3.5% from the third exchange. The other average padding values were obtained similarly. The OTP algorithm generates padding around to 100% of message size, while our proposed method can combine security levels (by choosing different parameters values, from 8 to 64 bits), therefore obtaining padding as low as 13% for 8 bits, for example.

Considering the proposal, whether an attacker attempts to obtain data information by brute force attack, she would have to test approximately 2^{160} possibilities (2^{32} possibilities for the angle and 2^{64} possibilities for each a_i and b_i parameters).

Considering that the processor utilized to test the 2^{160} possibilities is a 4 GHz Intel Pentium IV (thus having a clock cycle of 0.25 ns). Also, assuming that one possibility can be checked per clock cycle the attacker would need around $(0.25\times10^{-9} \times 2^{160}) / (60\times60\times24\times365) \cong 10^{31}$ years to test all possibilities by using a brute force attack. The consideration "one possibility can be checked per clock cycle" is very rough, because no known algorithm can execute divisions considering a π number with 200 bits precision, calculating angles with 32 bits precision, and executing matrices multiplications with 64 bits values, in just one clock cycle.

Evidently, the padding on proposed algorithm would be bigger than OTP algorithm's padding, by 24% in average (124% against 100% for OTP). Although, according to Table 1, one can utilize a padding of 31%, using parameters precision of 16 bits, for example. Thus, considering the same processor, the time needed to break it by brute force would be around 146 years. In this case, the proposed scheme is 69% better than OTP, for example.

Considering the processor characteristics above and a nonce of 8 bits, the number of possibilities to test will be 2^{48} (10^{14}), so in 19 hours these possibilities could be tested. However, as said before, if the message content is not text (binary data, for example), the question is: which one of the 10^{14} possibilities is the right one? Our evaluations suggest that the communicating nodes can start a transmission using a_i and b_i parameters with 64 bits, for example, to exchange some character codes to be transmitted. After this, they can continue to communicate using a precision of just 8 bits, due to the arguments exposed before.

The proposed scheme seems to be adequate for wireless networks, by avoiding the necessity of entities acting like servers and by avoiding the requirement of previous key exchanges. It was successfully tested on a wireless ad-hoc network composed by some notebook computers.

VII. CONCLUSIONS

This work presented a three-pass cryptographic protocol based on padding. The proposed protocol is based on a set of *N*-dimensional orthogonal matrices, which are used for the encryption and decryption process. The protocol needs three steps to deliver a message securely through a communication channel.

The proposed scheme shown achieves the confidentially property using rotation matrices and *nonces matrices*, thus obtaining the privacy of messages being transmitted through a public communication channel by a three-pass protocol.

The main contribution of the proposed method against other cryptographic techniques is its independency from any previous knowledge (shared secrets or keys) in order to achieve secure data information exchange; previous key distribution is not required.

The prototype demonstrated the algorithm feasibility and allowed to evaluate some aspects related to security levels. The prototype also allowed pointing out that the dynamic and security characteristics of a wireless network can be reached more effectively through the proposal.

Regarding other works on padding-based cryptography, the proposed scheme has shown to be more efficient, by generating less padding, whether applied with different security levels (by changing the number of bits on the *nonce* parameters) and without losses that could compromise its security.

Is important to point out that, even if a single message is successfully captured and broken, the angles and the *nonce* parameters can be modified for each new message. Therefore, it is unlikely that the scheme itself can be compromised and, even whether one message was compromised, the number of possibilities to test by brute force attack for another message is the same as if the first one was not compromised.

REFERENCES

[1] A. J. Menezes, P. C. Oorschot, and S. A. Vanstone, "Handbook of Applied Cryptography", CRC Press, [online] available in http://www.cacr.math.uwaterloo.ca/hac/, last access on April 2006.

[2] R. Venugopalan, P. Ganesan, P. Peddabachagari, A. Dean, F. Mueller and M. Sichitiu, "Encryption Overhead in Embedded Systems and Sensor Network Nodes: Modeling and Analysis", ACM CASES'03, San Jose, California, USA, pp. 188-197, 2003.

[3] R. Housley, W. Ford, W. Polk and D. Solo, RFC 3280, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)": [online] available in http://www.ietf.org/rfc/rfc3280.txt?number=3280, last access on April 2006.

[4] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen and D. E. Culler, "SPINS: Security Protocols for Sensor Networks", Wireless Networks 8, pp. 521-534, 2002.

[5] D. Slepian, "Group Codes for the Gaussian Channel", Bell Sys. Tech. J., pp. 575-602, 1968.

[6] I. Ingemarsson, "Commutative Group Codes for the Gaussian Channel", IEEE Transactions on Information Theory, vol. IT-19, no. 5, pp. 215-219, March 1973.

[7] E. Biglieri and M. Elia, "Signal Sets Generated by Groups", The Information Theory Approach to Communications, Ed. G. Longo, Springer, pp. 263-306,1977.

[8] D. S. Dummit and R. M. Foote, "Abstract Algebra", Prentice-Hall International, 1991.

[9] GNU GCC 4.0.2, A Compiler for Language C, C++, Fortran and Java, [online] available in http://gcc.gnu.org/, last access on July 2005.

[10] MPFR Group, Multi Precision Floating Pointing library for Language C, [online] available in http://www.mpfr.org/, last access on March 2006.

[11] D. H. Bailay, "High-Precision Floating-Point Arithmetic in Scientific Computing", IEEE Computing in Science & Engineering, pp. 54 – 61, May/June 2005.

[12] G. S. Vernam, "Cipher printing telegraph systems for secret wire and radio telegraphic communications", *JAIEE* 45, pp. 109-115, 1926.

[13] C. Shannon, "Communications theory of secrecy systems", Bell System Tech, J. 28, pp 657–715, 1949.