Matrix Analogues of the Diffie-Hellman Protocol

Alexsander Beletsky[1], Anatoly Beletsky[1] and Roman Kandyba[1]

[1]Department of Electronics National Aviation University of Kiev,
1, av. Cosmonaut Komarov, 03680, Kiev, Ukraine

alexander.beletsky@gmail.com, abelnau@ukr.net,
romankandyba@mail.ru

Abstract. This paper presents a comparative analysis of several matrix analogs of the Diffie-Hellman algorithm, namely, Yerosh-Skuratov and Megrelishvili protocols, as well as alternative protocols based on irreducible polynomials and primitive Galois or Fibonacci matrices. Binary matrix is primitive, if the sequence of its powers in the ring of residues mod 2 forms a sequence of maximum length (m – sequence). Offer alternative protocols and discuss ways to improve the reliability of their.

Keywords. Encryption key exchange protocol, the irreducible polynomials, a primitive element of Galois field, primitive binary matrix

Key terms. Research, CryptographyTheory, MathematicalModelling

1 Introduction

The Diffie-Hellman algorithm (DH-algorithm) [1] assumes that two subscribers – Alice and Bob both know the public keys p and q , where p is a large prime number, and q is a primitive root. Subscriber Alice generates a random big number a , computes $A = q^a \bmod p$ and sends it to Bob. In turn, Bob generates a random big number b , computes $B = q^b \bmod p$ and sends it to Alice. Then subscriber Alice raises number B received from Bob to her random power a and calculates $K_a = B^a \bmod p = q^{ba} \bmod p$. Subscriber Bob acts similarly, calculating $K_b = A^b \bmod p = q^{ab} \bmod p$. It is obvious that both parties receive the same number K because $K_a \equiv K_b$. Then Alice and Bob can use this number K as a secret key, e.g. for symmetric encryption because a foe who intercepts numbers A and B faces with virtually unsolvable (in a reasonable time) the problem of calculation K , under the condition, that numbers p , a and b were chosen big enough.

2 Yerosh-Skuratov Protocol

In order to form a secret encryption key in the public network by subscribers Alice and Bob, the authors [2] propose to use DH protocol in the cyclic group of matrices $\langle M \rangle$, and the matrix M is considered as public information. It is assumed that Alice generates a random index x, calculates the matrix M^x and sends it to Bob. In turn, Bob generates a random index y, calculates the matrix M^y and sends it to Alice. Then both subscribers raise the matrices obtained from a partner in their secret powers and calculate the sheared matrix (encryption key) $K = M^{xy} \equiv M^{yx}$. The matrix M must be a high-order matrix (at least 100); so, the authors assert (by the way, without a proof), cracking key has invincible complexity. However, in [3] it has been proved, that Yerosh-Skuratov protocol can easily be cracked based on the generalized Chinese remainder theorem.

3 Megrelishvili Protocol

The essence of the protocol [4] is following. Binary initialization vector V and primitive matrix M of order n are accepted as a public key. Subscriber Alice generates a random index x, calculates the vector $V_a = V \cdot M^x$ and sends it to Bob. In turn, Bob generates a random index y, calculates the vector $V_b = V \cdot M^y$ and sends it to Alice. Then Alice computes the key $K_a = V_b \cdot M^x = V \cdot M^{y+x}$, and Bob computes the key $K_b = V_a \cdot M^y = V \cdot M^{x+y}$. It is quite obvious that using such data exchange protocol, both parties receive the same private key K, because $K_a \equiv K_b = K$.

The algorithm of generating the matrices in Megrelishvili protocol is fairly simple and can be explained by the following calculation scheme

$$M_1 = 1, \quad M_3 = \begin{bmatrix} 1 & 0 & 1 \\ 1 & M_1 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \quad M_5 = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & & & & 0 \\ 0 & & M_3 & & 1 \\ 1 & & & & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}, \ldots \tag{1}$$

As it follows from (1), the matrices $M_i, \ i = 1, 2, \ldots,$ are matrices of odd order only that can cause some difficulties when they are used in cryptography. This shortcoming was remediated by replacing matrices of type (1) by primitive matrices of an arbitrary order that is synthesized based on the so-called generalized Gray transforms [5]. The essence of these transforms is explained below.

The matrix form of direct (for simplicity denoted by number 2) and inverse (denoted by number 3) classical Gray transforms (codes) [6] can be presented in the form

$$2 := \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad 3 := \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}; \qquad (2)$$

where as an example, the order of the matrix n is set $n = 4$.

Matrices (2), which we call left-sided Gray transform matrices, are in correspondence with the right-sided transform matrix defined by the following relations:

$$4 := 121 = 2^T; \quad 5 := 131 = 3^T, \qquad (3)$$

where

$$1 := \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \qquad (4)$$

is the matrix (operator) of the inverse permutation.

The set of operators (2) – (4), supplemented by the operator 0, or *e* (identity matrix), forms a complete set of simple Gray operators. From the elements of simple Gray operators, one can form so-called composed Gray codes (CGC) generated by the product of simple (elementary) Gray codes. The simplest examples of CGC 121 and 131 can be seen in (3). Both simple and composed Gray codes have a number of remarkable properties. Firstly, the corresponding transformation matrices are nondegenerate and, therefore, are reversible. Secondly, there are simple inversing algorithms for CGC. And, finally, there are “crypto-order” CGC which have the property of primitiveness. Examples of such codes are given in Tab. 1.

Table 1. Gray Composite codes delivering binary matrices property of primitiveness

The order of the matrix (n)			
32	64	128	256
2244424	22533435	2425535	22533435
2442224	22534335	2433534	22534335
12242253	24334225	2435334	24334225
12242443	25224334	22524224	25224334
12252242	222524424	22533334	2222535224

Suppose M is a primitive binary matrix generated by the CGC G. With respect to such matrices, the following assertion can be easily proved by the test method.

Assertion. *The primitiveness of matrices M is invariant to the group of linear transformations Ω of the CGC G generating matrix M and transformations of similarity Π over these matrices.*

The $\Omega-$group includes the following operators: cyclical shift, assess statement, inversion and conjugation as well as arbitrary combinations of these operators. Transformation Π forms matrix M_p, which is similar to M and determined by the relation

$$M_p = P \cdot M \cdot P^{-1},$$

where P is a permutation matrix.

4 Alternative Protocols

This section proposes two options for alternative matrix protocols of secret key exchange on the open channel of communications. The procedure for the formation of the encryption key K in the first version of the protocol is based on the use of two public and one private key for both subscribers. As a public key a binary initialization vector V of n order and any irreducible polynomial (IP) φ_n of n order are chosen. Private keys are primitive (forming) elements ω of the Galois field $GF(2^n)$ over the IP φ_n, from which the subscribers (Alisa and Bob) form the primitive secret transformation matrices $G_{\varphi_n}^{(\omega_a)}$ and $G_{\varphi_n}^{(\omega_b)}$ respectively. The element ω of the field $GF(2^n)$ is primitive over IP φ_n, if the minimum rate e, at which $(\omega^e \equiv 1) \bmod \varphi$ assumes the value $e = 2^n - 1$.

Matrix $G_{\varphi_n}^{(\omega)}$ we call Galois matrices. The synthesis of algorithm for such matrices is explained on a concrete example. Let's IP $\varphi_8 = 100101101$, and the generating element (GE) of subscriber Alisa $\omega_a = 111$. We obtain

$$A = G_a = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix}. \tag{5}$$

According to (5), the procedure of filling in the matrix G_a is carried out under the following scheme. First, the GE ω_a is arranged in the bottom row of the matrix. The elements of this row in the left from the GE elements are filled with zeros. Subsequent rows of the matrix (in the direction from bottom to top) are produced by a shift of previous lines. If left element of shifted line is 0, then the cyclical shift by one bit to the left (circular scrolling clockwise). In the case where the left element of shifted line is 1, the conventional shift of the line on one bit to the left and 0 is written to the vacant right element in line. Digit capacity of these lines is one bit more than the order of the matrix. The vectors corresponding to these lines are given to the residue modulo IP φ_n that returns them the capacity, which coincides with the order of the matrix n. Subscriber Bob forms similarly the Galois matrix $B = G_b$ using his primitive element ω_b.

The introduced Galois matrices have some interesting properties. First, the matrix product is commutative, i.e. $A \cdot B = B \cdot A$. At the same time, secondly, if at least one of the GE is not a primitive of the IP, the commutative property of matrices is lost. Based on the above properties of Galois matrices a key exchange protocol was proposed.

We consider that initialization vector V and the IP φ are known. Alice chooses a secret primitive over φ GE ω_a, forms a Galois matrix A, calculates the vector $V_a = V \cdot A$ and sends it to Bob. In turn, the subscriber Bob selects a primitive GE ω_b, forms a matrix B that calculates the vector $V_b = V \cdot B$ and sends it to Alice. After that, both parties multiply vectors obtained from the partner, in own secret Galois matrix. Thus, a shared secret key K will be formed by the fact that the product of primitive Galois matrices over the same IP φ is commutative, and this implies the identity

$$K_a = V_b \cdot A = V \cdot B \cdot A \quad \equiv \quad K_b = V_a \cdot B = V \cdot A \cdot B .$$

Instead of Galois matrices G, Fibonacci matrices F can be used in the protocol with the same success. Fibonacci matrices are associated with Galois matrices by equation

$$F \xleftrightarrow{\perp} G, \quad \text{or} \quad F = G^{\perp}; \quad G = F^{\perp},$$

where $\perp$ – means the operator of right transposition, i.e. transposition with respect to the auxiliary diagonal matrix.

In the second alternative embodiment of the protocol the secret key K is computed in two rounds. In the first round, which repeats the above-considered first version of the protocol, a common to both subscribers secret binary vector of n – th order V_p is formed. On the basis of this vector, Alice and Bob compute the common permutation matrix P. One can propose different ways of constructing matrices P. Let us consider one of them. Let's $n = 8$ and N is the decimal equivalent of the vector V_p. The task is to create permutation matrix P_8 of order eight for value N. Choose one or another way of numbering elements of matrices P_8 from 0 to 63. Calculate the value $n_8 = N \bmod 64$ and write 1 in that element of the matrix, whose number is equal n_8. After that, delete from the matrix P_8 the row and column, which contains 1. We obtain a matrix P_7 of 7-th order, whose elements are numbered from 0 to 48. Find the value $n_7 = N \bmod 49$, which is determined by the location 1 of the matrix P_7 and, consequently, in the matrix P_8. Following the proposed method, one can simply construct a permutation matrix P of any order.

Let proceed to the second variant of the encryption keys protocol. This protocol uses two public keys, which are the initialization vector V, and the irreducible polynomial φ, and also two private keys. These keys are generated by Alice and Bob as a random primitive over IP φ Gees ω and υ. The protocol runs in two rounds. In the first round based on public keys V, φ and secret GE ω network operators calculate the total permutation matrix P. The second round is performed in the following order. Alice chooses a primitive over φ GE υ_a, forms Galois matrix A_υ, then similar matrix $A_p = P \cdot A_\upsilon \cdot P^{-1}$, computes a vector $V_a = V \cdot A_p$, and sends it to Bob. In turn, Bob chooses a primitive over φ GE υ_b, forms Galois matrix B_υ, then similar matrix $B_p = P \cdot B_\upsilon \cdot P^{-1}$, computes a vector $V_b = V \cdot B_p$ and sends it to Alice. After that, both parties multiply vectors obtained from partners on their secret similar Galois matrix. Thus, the shared key K will be generated due to the fact that the matrices A_p and B_p maintain the properties of primitiveness and commutatively of primary matrices A_υ and B_υ, respectively.

5 Protocol of Vagus Keys

One of the major drawbacks of alternative algorithms key generation algorithms for open key cipher infrastructure, in particular the mentioned above the way of synthesis Galois matrix (by the diagonal fill method), is that it could be easily compromised. To prove that, let's see the vector

$$V_a = V \cdot G_{f_n}^{(\omega_a)}, \tag{6}$$

created by Alice.

By the theory of polynomials of one variable x, we know that product of any polynomial $\omega_n(x)$ power of n by x is equivalently either simple shift of polynomial for one bit left or incrementing the power of polynomial,

$$x \cdot \omega_n(x) \to \omega_{n+1}(x). \tag{7}$$

Taking formula (7), let's represent the Galois matrix $G_{f_n}^{(\omega_a)}$ the power of n by,

$$G_{f_n}^{(\omega)}(\mathrm{mod}\, f_n) = \begin{pmatrix} x^{n-1} \cdot \omega \\ x^{n-2} \cdot \omega \\ \dots \\ x \cdot \omega \\ \omega \end{pmatrix} (\mathrm{mod}\, f_n) = \omega \cdot \begin{pmatrix} x^{n-1} \\ x^{n-2} \\ \dots \\ x \\ 1 \end{pmatrix} = \omega \cdot E = \omega, \tag{8}$$

where E – the unit matrix.

From formulas (6) and (8) we can get,

$$V_a = V \cdot \omega_a (\text{mod } f_n), \quad (9)$$

where all parts are known, except ω_a. Solving the equation (9), we found:

$$\omega_a = V_a \cdot V^{-1} (\text{mod } f_n). \quad (10)$$

For example, let's use the matrix $G_{f_n}^{(\omega_a)}$, given by expression (5), where $n = 8$, $\omega_a = 101101$, $f_8 = 101001101$, so f_8 – is public, ω_a – is private keys of protocol. As initialization vector we choose $V = 11010010$, that corresponds to invert by modulus f_8 vector $V^{-1} = 110010$. By formula (9) we get $V_a = 10111111$. Putting the V_a and V^{-1} is the right side of expression (10) and taking modulus f_8 of vectors multiplication results, enemy (Eva) is getting private key ω_a of Alice. The same way, Eva could found secret key ω_b of Bob. After secret keys ω_a and ω_b are found it's trivial to calculate secret key K.

The security of alternative protocols could be increased up to security level of algorithms based on problem of factorization of modular multiplication of big numbers if we assume that there is secret parameter θ, both known to Bob and Alice.

The modification of protocol [6] is the be following. Assume, there are authorized subscribers that have secret parameter θ as binary vector of n – order. Parameter θ could be transported from Alice to Bon (or otherwise), e.g. by RSA protocol. Alice is generating random of n – order number ω_a and computing generating element

$$\theta_a = \omega_a \cdot \theta (\text{mod } f_n), \quad (11)$$

by means of generating element Alice is forming Galois matrix $G_{f_n}^{(\theta_a)}$, calculating vector $V_a = V \cdot G_{f_n}^{(\theta_a)}$ and sends it to Bob. In the same way, Bob send to Alice vector

$$V_b = V \cdot G_{f_n}^{(\theta_b)}, \text{ where } \theta_b = \omega_b \cdot \theta (\text{mod } f_n).$$

As it shown above, generating elements θ_a and θ_b could be easily computed, so authorized subscribers Alice and Bob, but not Eva, could calculate secret parameter ω of partner. As example, by formula (11) Bob calculates $\omega_a = \theta_a \cdot \theta^{-1} (\text{mod } f_n)$, that gives him and Alice ability to calculate secret key $K = \omega_a \cdot \omega_b (\text{mod } f_n)$. Key K as well as any function of it, could be taken as a secret parameter $\theta^{+} = K$ for session key generation for public key cipher channels.

We call that way of key generation – protocol (algorithm) of vagus keys. Vagus keys algorithm could be used in both motioned above protocols. The major benefit of vagus key generation algorithm is protection from "man in a middle" type of attack. It's been archived by including in Galois matrices key generation elements of secret element θ, known only by Bob and Alice. In case of secret element θ is changed by element θ_e of Eva, makes it impossible to Eva to calculate parameters ω_a , ω_b as well as general cipher key K .

6 Conclusions

The article analyzes the known matrix algorithms for exchanging encryption keys between subscribers of a network of open communication channels. The algorithms are based on the modified asymmetric Diffie-Hellman protocol. The essence of the modification is reduced to replacing the large prime numbers of Diffie-Hellman algorithm by assurance nondegenerate primitive binary matrices of high order. Methods of synthesis of these matrices are proposed based on both the generalized Gray codes, and irreducible polynomials. New key exchange matrix protocols have been developed. The protocols developed are superior for cryptographic strength to known cryptographic protocols, particularly Yerosh-Skuratov and Megrelishvili protocols described in this paper.

The proposed variants of vector-matrix protocols for exchanging by cryptographic keys on open communication channels have a good prospect to be applied for symmetric encryption in computer networks protected from the substitution of data, providing the necessary level of protection of private keys from unauthorized access. These protocols can make a strong competition to more resource-intensive RSA protocol.

References

1. Diffie, W., Hellman, M. E.: New Directions in Cryptography.IEEE Transactions on Information Theory, IT-22(6), 644–654 (1976)
2. Eros, I. L., Skuratov, V. V.: Addressing Message Transmitting Using Matrices Over GF (2). Problems of Information Security. Computer Systems, 1, 72–78 (2004) (In Russian)
3. Rostovtsev, A. G.: On the Matrix Encryption (Criticism Yerosh-Skuratov Cryptosystem), http: www.ssl.stu.neva.ru/psw/crypto/rostovtsev/Erosh_Skuratov.pdf (In Russian)
4. Megrelishvili, R. P., Chelidze, M. A., Besiashvili, G. M.: Unidirectional Matrix Function - High-Speed Diffie – Hellman's Analog. In: Proc. 7-th Int. Conf. Internet - Education - Science 2010. VNTU, Vinnitsya, 341–344 (2010) (In Russian)
5. Beletsky, A. Ja., Beletsky, A. A., Beletsky, E. A.: Gray Transformations.V.1. Fundamentals of the theory. V. 2. Applied aspects. NAU Publishing House, Kiev (2007) (In Russian)
6. Beletsky, A. Y., Beletsky, A. A.: Synthesis of Primitive Matrices over a Finite Galois Fields and their Applications. Information Technology in Education: Collected Works, 13. Kherson: KSU, 23–43 (2012) (In Russian)