ON THE ORIGINAL ONE-WAY FUNCTION AND THE NEW DIGITAL SIGNATURE SCHEME

R. Megrelishvili

Ivane Javakhishvili Tbilisi State University
0186 University Street 2, Tbilisi, Georgia

(Received: 20.12.10; accepted: 25.05.11)

Abstract

In this paper the author presents the original approach to the synthesis of matrix-based one-way function and key exchange algorithm via an open channel together with the new digital signature algorithm, which is highly valuable because of its simultaneous simplicity and steadiness.

Key words and phrases: Open channel, key exchange, commutative set of matrices, one-way function, digital signature.

AMS subject classification: 1594.

1 Introduction

One-way function, obtained by the author of this paper, was first published in [1]. Studies, presented in [2-4], have shown that it's possible to build large sets of $n \times n$ sized commutative matrices over Galois fields $\mathbf{GF}(p)$ for fixed natural values of n. The set is built as a cyclic multiplicative group of cardinality $2^n - 1$ with the specific generator-matrix. These sets of matrices are used in construction of the one-way function and form cryptographic algorithm of key exchange via an open channel as well as digital signature scheme. The first section of the present paper represents the one-way function and key exchange cryptographic algorithm, the second section - principally new digital signature algorithm.

2 Original one-way function and key exchange algorithm

Reader may refer to [5,6] in order to see one-way functions, which are already obtained by other authors. For example, the one-way function $a^x \equiv y \pmod{p}$ (see [7]) is based on the impossibility of computing x-es for given values of y (actually, the impossibility is real time-based for enough high values of parameters p, x and a); at the same time, obviously, it's

easy to calculate y for given values of x. This fact is known as Discrete Logarithm Problem.

Unlike Diffie-Hellman protocol, for the purpose of alternative approach to one-way functions construction, the author uses $n \times n$ matrices over Galois fields $GF(p)$.

The idea of this alternative approach is as follows:

Let $\mathfrak{A}$ be the set of commutative matrices (suppose that cardinality of the set $\mathfrak{A}$ is approximately 10^{30}, i.e. $card(\mathfrak{A}) = 2^{100}$, which equals to the lowest steadiness level of contemporary cryptosystems).

Also let the matrices

$$A^{2^0}, A^{2^1}, \ldots, A^{2^n-1} \tag{2.1}$$

create the basis of the set $\mathfrak{A}$, where $A^{2^i} \neq A^{2^j}$, if $i \neq j$ and A is the generator-matrix.

Therefore, any matrix A_i of the set $\mathfrak{A}$ is obtained as the linear combination of matrices (2.1):

$$A_i = c_0 A^{2^0} + c_1 A^{2^1} + \cdots + c_{n-1} A^{2^n-1}, \tag{2.2}$$

where $c_i \in GF(2)$.

In order to reach the minimal level of contemporary cryptographic steadiness, we should consider only those sets of matrices, which cardinality is above 2^{100}. For example, if matrices are given over $GF(2)$ field, their size must be greater than 100×100.

Now we'll formulate our key exchange scheme:

Let the basis (2.1) or the generator-matrix A be public as well as the vector $v = (v_1, \ldots, v_n)$ $(v_i \in GF(2))$.

Imagine two people, Alice and Bob, forming a secret exchange key via the public channel:

- Alice chooses the $n \times n$ sized secret matrix $A_1 \in \mathfrak{A}$ and sends the following vector to Bob:
$$u_1 = vA_1; \tag{2.3}$$

- Bob chooses the $n \times n$ sized secret matrix $A_2 \in \mathfrak{A}$ and sends Alice the vector
$$u_2 = vA_2; \tag{2.4}$$

- Alice computes $K_1 = u_2 A_1$;

- Bob computes $K_2 = u_1 A_2$.

The crucial point is that K_1 and K_2 are the same secret exchange keys, as $K_1 = v(A_2A_1) = v(A_1A_2) = K_2$ (the author uses here commutativity of the set $\mathfrak{A}$).

3 New digital signature scheme

It is well-known that one-way functions are used to implement digital signature schemes. For example, ElGamal (see [8]) created the original efficient digital signature algorithm using the one-way function of Diffie-Hellman protocol (see [7]). All the same concerns the RSA algorithm (see [9]), which is widely used for digital signature and authentication purposes.

In this paper the author proposes his original approach to digital signature algorithms construction, which is based on the one-way function, presented in section 1 (see (2.1)-(2.4)).

The ElGamal scheme [8] is based on the one-way function of Diffie-Hellman (see [7]):

$$a^x \equiv y \pmod{p}. \tag{3.5}$$

Therefore, this scheme is based on the difficulty of calculating discrete logarithms in finite fields.

RSA algorithm essentially uses the difficulty of prime factorization.

Cryptographic steadiness of our algorithm is based on the difficulty of solving two problems in real time:
(i) Complexity of solution of linear equations systems, when quantity of variables (n^2 in our case) exceeds quantity of equations (n in our case);
(ii) Complexity of exhaustive search in sets of high-sized matrices (even when sets have the structure of a multiplicative group).

The scheme of our digital signature algorithm is as follows:

Let's consider the situation of information exchanging between two persons, Alice and Bob. The main problem is to avoid the third person (for example, Carol) to use this information in her favor.

Suppose that Alice sends information (concatenation $\|M\|\|r\|\|s\|$) to Bob, where M denotes text information, r and s are signatures. Then the digital signature scheme is obtained in the following way:

Alice selects her secret key $x \in V_n$, i.e. x is a vector from n-dimensional vector space (one uses the same secret key x only for the specific period of time - that means that this x will be changed to different one as soon as this period ends). After that Alice calculates $y = xA_0$ open key and sends it to Bob. Secret matrix A_0 is derived from an open matrix $A \in \mathfrak{A}$ by mixing strings in this A matrix (phrase "mixing strings" means permutation of strings; matrix A_0 is defined below). Then Alice selects another secret

key $k \in V_n$ (this key is also used for a certain period of time or even for one session if needed) and calculates open signature $r = kA_0$. Then she computes her auxiliary secret signature $s_0 \in V_n$ using the following formula:

$$s_0 = x + k + m, \tag{3.6}$$

where m is the data, obtained by hashing with some appropriate hash function H:

$$m = H(M). \tag{3.7}$$

After this step everything is ready to compose the concatenation $\|M\|\|r\|\|s\|$, which Alice sends to Bob (for signature s we've got $s = s_0A_0$).

When Bob receives the concatenation, he performs the verification procedure $s_0 = x+k+m$, which is identical to (3.2) (recall that m is obtained from (3.3); $x = yA_0^{-1}$ (y is an open key, A_0^{-1} is defined below); $k = rA_0^{-1}$ and $s_0 = sA_0^{-1}$).

Now it's quite reasonable to define A_0^{-1} matrix (to be more precise, we should clarify, how the inverse matrix of A_0 is obtained by Bob).

First we ought to notice that Bob can directly calculate the inverse matrix of A, as this matrix is public. Then if Bob is aware of the exact permutations of strings, made by Alice, he will undoubtedly obtain A_0^{-1} by performing the same permutations of columns in A^{-1} (note the difference: Alice makes permutation of strings in A and obtains A_0, but for Bob everything is quite vice versa). It's obvious to show, that acting in opposite manner, Bob will definitely obtain the inverse matrix of A_0.

But the following problem is still open: How to make Bob aware of Alice's permutations securely?

Of course, it's possible to use other well-known cryptographic protocols in order to transmit information about permutations made in A, but this way will definitely hurt the elegance of actual work as well as it's significance. Therefore it's quite necessary to invent the way of secure exchange basing on our original key exchange algorithm and, fortunately, such remarkable way exists:

Consider $n \times n$ sized matrix A. Alice and Bob exchange the secret key $K \in V_{nm}$ (m is the least natural number, such that $n < 2^m$; V_{nm} is nm-dimensional vector space over $GF(2)$ field) via performing actions, described above in our key exchange algorithm. Let this K be the vector $(k_1, k_2, k_3, \ldots, k_{nm})$. It's obvious that we had to take nm-dimensional vector K, because any permutation of $n \times n$ sized matrices looks like

$$\begin{pmatrix} 1 & 2 & \ldots & n \\ \alpha_1 & \alpha_2 & \ldots & \alpha_n \end{pmatrix}, \tag{3.8}$$

where each α_i belongs to the set $\{1, 2, 3, \ldots, n\}$. Therefore we need nm bits (m bits for each) to convert all these n numbers (i.e. $1, 2, 3, \ldots, n$) to binary form.

Let's split the vector K into m-bit blocks (obviously, we'll get n blocks), each representing binary form of a certain number from $\{1, 2, 3, \ldots, n\}$. As a result, we've got:

$$K = (k_1 k_2 \ldots k_m, \ldots, k_{(n-1)m+1} k_{(n-1)m+2} k_{nm}) \equiv (\alpha_1, \alpha_2, \ldots, \alpha_n). \quad (3.9)$$

We should notice, that it's quite impossible to predict exact components of K - thus they could be arbitrary ones and this fact yields two problems:
(i) How to make these m-bit blocks be different from each other?
(ii) How to make them be in range $\{1, 2, 3, \ldots, n\}$ (as we know, m-bit blocks may represent all numbers from 0 up to $2^m - 1$, which could be greater than n in general)?

Well, solutions of these two problems are quite easy: (ii) issue could be fixed if after converting m-bit blocks to numbers, we'll calculate them modulo $n + 1$; for (i) issue we can perform the following actions: if $\alpha_i = \alpha_j$ for $i < j$ (including the case when $\alpha_i = 0$), we'll set α_i to be equal to $\alpha_i + 1$ (if $\alpha_i = 0$ occurs) and α_j - to be equal to $\alpha_j + 1$ (if the value $\alpha_j + 1$ already exists in the set $\{\alpha_1, \ldots, \alpha_{j-1}\}$, we'll repeat the last procedure until $\alpha_k \neq \alpha_l$ for every $k \neq l$, such that $k, l \in \{1, \ldots, j\}$).

Therefore, our original digital signature algorithm is fully presented and our goal is achieved!

The author objectively hopes that the above-described digital signature algorithm will be put beside well-known algorithms of modern cryptography in the nearest future because of its relatively huge steadiness and simplicity at the same time!

In conclusion, we'd like to emphasize the following important, but obvious fact: it's not possible to figure out the exact permutation of $\{1, 2, \ldots, n\}$ for sufficiently big values of n in real time, because exhaustive (i.e. brute-force) search requires $\mathcal{O}(n!)$ operations to be performed. Thus steadiness is even above exponential in this case!

References

1. Megrelishvili, R.; Chelidze, M.; Chelidze, K. On the construction of secret and public-key cryptosystems. Appl. Math. Inform. Mech. 11 (2006), no. 2, pp. 29-36.

2. Megrelishvili R., Sikharulidze A. *New matrix sets generation and the cryptosystems*, Proceedings of the European Computing Conference and the Third International Conference on Computational Intelligence, Tbilisi, Georgia, June, 26-28, 2009, pp. 253-255.

3. Megrelishvili R., Chelidze M., Besiashvili G. *Investigation of new matrix-key function for the public cryptosystems.* The Third International Conference "Problems of Cybernetics and Information", v.1, September, 6-8, Baku, Azerbaijan, 2010, pp. 75-78.

4. Megrelisvili R., Chelidze M., Besiashvili G. *One-way matrix function - analogy of Diffie-Hellman protocol*, Proceedings of the Seventh International Conference, IES-2010, 28 September-3 October, Vinnytsia, Ukraine, 2010, pp. 341-344.

5. Schneier B. *Applied Cryptography*, John Wiley and Sons, Inc, New York, 1996.

6. Menezes A., Oorshot P. van, Vanstone S. *Handbook of Applied Cryptography*, CRC Press, 1996.

7. Diffie W., Hellman M.E. *New Direction in Cryptography*, IEEE Transaction on Information Theory, IT-22, n.6, Now. 1996, pp. 644-654.

8. ElGamal T. *A Public-Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms*, IEEE Transaction on Information Theory, v. IT-31, n. 4, 1985, pp. 469-472.

9. Rivest R.L., Shamir A., Adleman L.M. *A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*, Communications of the ACM, v. 21, n. 2, Feb 1978, pp. 120-126.