Construction Efficiency of the Public Key Cryptosystem based on Boolean Product of Matrices

Yeghisabet Alaverdyan
European Regional Educational Academy
Yerevan, Armenia
e-mail: ealaverdyan@armline.am

ABSTRACT

Estimation of efficiency of the public key cryptosystem based on Boolean product of matrices is given. It is shown that the high performance of the cryptosystem is conditioned by the use of fast logic operations. The stability of the cryptosystem is based on the computational complexity of decomposing Boolean product and Boolean addition of large matrices. Ease of implementation of the cryptosystem based on Boolean product of matrices is also investigated.

Keywords

Boolean matrix, product of matrices, public key, private key.

1. INTRODUCTION

Most of the existing public key cryptosystems are based on the number theory, providing high stability against attacks by using a large key space [1]. Implementation of such algorithms leads to selection of primes from a sufficiently large set, meanwhile at present, there are no useful techniques to yield arbitrary large primes, and also almost invariably, the tests for primality are still probabilistic. Both key generation and encryption/decryption involve raising an integer to an integer power then reducing modulo n dealing with potentially large exponents and complex calculations, which significantly decreases the level of the cryptosystem performance, especially with procession of large amount of information. The larger size of the key, the slower the system will run. This keeps the number theory based public key cryptosystems currently confined to key distribution and signature applications. To improve the performance of public key cryptosystems and to eliminate the restrictions of their applications, cryptosystems based on mathematical logic are being developed. In particular, the public key cryptosystem based on the Boolean product of matrices can serve as a premise to construct fast and stable public key cryptosystems providing also easiness of implementation. Boolean product of matrices involves usage of so called *zero-one* matrices, with entries either zero or one. Zero - one matrices are often used to represent discrete structures such as functions and binary relations. Algorithms using these structures are based on Boolean arithmetic with zero-one matrices, implementing logical multiplication called as conjunction, and addition of two types, such as disjunction and exclusive *OR* operation. Any algorithm applying logical operations possesses high level of performance, therefore, usage of Boolean logic can result in a general purpose cryptosystem due to ease of its realization. In this paper detailed exploration of essential features of the cryptosystem based on Boolean product of matrices is represented.

2. KEY PAIR GENERATION IN THE PUBLIC KEY CRYPTOSYSTEM BASED ON BOOLEAN PRODUCT OF MATRICES

In this cryptosystem the plaintext and ciphertext symbols are integers between 0 and 2^{n-1}. Any integer between 0 and 2^{n-1} can be expressed as a binary number consisting of n bits with padded starting 0-s if needed. It is known that a Boolean function of n variables is a mapping from an n-dimensional vector space over the binary field $F_2 = \{0, 1\}$ to itself. Such a function can be implemented as a combinational logic unit with one bit output and n -bit input [2]. A mapping from F_2^n to F_2^m is called an (n, m) Boolean function. An (n, m) Boolean function can always be expressed as a collection of m functions in F_n, where F_n is the set of all Boolean functions over n variables. A particular class of these type of multiple output Boolean functions occur when $m = n$ and that different inputs yield different outputs. By treating each input/output as the binary expression of an integer within the range $S = \{0, 1, ..., 2^{n-1}\}$, the above functions perform permutations on S and are called Boolean permutations [3]. Key pair generation starts with the creation of such a collection of Boolean functions over n variables called as initial Boolean permutation, BP , of order n

$$BP = [f_1(x), f_2(x), , f_n(x)]. \tag{1}$$

Here f_1, f_2, ..., f_n are component functions of the initial Boolean permutation BP, and x is the shorthand of all the variables. Like any permutation, the initial permutation BP has own inverse. The initial permutation and its inverse are defined by tables, as shown in Table 1 and Table 2, respectively.

Table 1. Permutation table for BP

n-bit binary representation of integers	Initial BP			
	f_1	f_2	f_3	...f_n
000....0	0	0	0	...1
000....1	1	0	1	...0
000...10	0	1	1	...1
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
111...10	1	0	1	...0
111...11	1	1	0	...1

The input to a table consists of n bits numbered from 0 to

2^{n-1}. Notice that the initial Boolean permutation BP *has no cryptographic value* and actually represents a bijective encoding scheme available to any user applying this cryptosystem.

Table 2. Permutation table for BP^{-1}

BP^{-1}	x_1	x_2	x_3	$...x_n$
000....1	0	0	0	...0
101....0	0	0	1	...1
011....1	0	1	0	...0
...	...	...	...	...
...	...	...	...	...
...	...	...	...	...
101....0	1	0	1	...0
110....1	1	1	1	...1

Key pair generation of the proposed public key cryptosystem involves the following steps:

1. An initial Boolean permutation of order n over k variables is constructed, then its transpose is taken. This results in a matrix of size $k \times n$. Denote it by BM_1.

2. For the matrix BM_1 two more matrices, BM_2 and BM_3 are created. They have property, that

$$BM_1 \times BM_2 = I_k, \qquad (2)$$

$$BM_3 \times BM_2 = 0. \qquad (3)$$

Here I_k is the identity matrix of size $k \times k$. A convenient way to create such matrices is to choose an arbitrary nonsingular matrix M of size $n \times n$ such that its first k rows are occupied by BM_1, the remaining $n-k$ rows represent BM_3, and BM_2 will be composed by the first left k columns of M^{-1}, which is the inverse of M [2].

3. Another arbitrary collection of Boolean functions, the second Boolean permutation, is created, as follows:

$$R = [r_1,\ r_2,\ ...\ r_{n-k}] \qquad (4)$$

4. A secret Boolean matrix, BMS , of size $k \times k$ is created.

5. $R \times BM_3 \oplus BMS$ is calculated. Denote it by BM_4.

The **public key** of the proposed public key cryptosystem is the pair of only two matrices, (BM_1, BM_4).

The **private key** of this cryptosystem is the triple of Boolean matrices (BM_2, BMS, BP^{-1}).

3. MESSAGE ENCRYPTION/DECRYPTION

The plaintext, that is a collection of integers between 0 and 2^{n-1} converted into binary, is processed according to transformations, indicated above. That is, the plaintext symbols binary values are replaced with the initial Boolean permutation component functions values, resulting in BM_1 and then XOR-ed with BM_4, which is a puzzled composition of R, BM_3 and BMS. The encryption is performed according to the following algorithm:

$$C = (BP \times BM_1) \oplus BM_4, \qquad (5)$$

where C is the ciphertext.

To decrypt the ciphertext, BMS is removed from the encryption matrix through XOR-ing operation, and the initial permutation BP is released through the following operation:

$$PB = C_{PK} \times BM_2, \qquad (6)$$

where C_{PK} is the ciphertext obtained by applying the public key, PK, over the plaintext.

Detailed exploration of stability of the presented algorithm is given in [4]. The cryptosystem security is analyzed against the following main types of attacks:

- probable-message attack
- algebraic attack
- the private key exhaustive search attack
- attack by computing the private key from the given public key.

The analysis given in [4] shows that it is accomplished so that the degree of security is great enough to delay solution by the opponent for such a length of time that when the solution is finally reached, the information thus obtained has lost its value.

4. COMPUTATIONAL ASPECTS

We now turn to the issue of the complexity of the computation required to use Boolean product of matrices, (BPM). There are actually two issues to consider: key generation and encryption/decryption. Before the application of BPM, a pair of keys must be generated. For public key generation this involves the following tasks:

1. Clarifying the range of positive integers used to cover the plaintext characters and find the number of bits, k, required to represent them in binary.
2. Constructing the appropriate truth table mapping the binary patterns into functions values. The number of rows of that table is n.
3. Creating the transpose of the resulting matrix with the functions values of size $n \times k$. This is the initial permutation, BP.
4. Constructing an $n \times n$ matrix, M, such a way to satisfy the conditions (3)and (4).
5. Creating another arbitrary permutation, R.
6. Constructing BMS.
7. Calculating BM_4.

First consider the performance of the point 4, as the first three points are procedures consuming no time and calculations. The same holds with point 5 and 6. Now it is time to estimate the generation of matrices BM_1, BM_2 and BM_3.

Matrices BM_1 and BM_3 are occupying k and $n-k$ rows of the matrix M and their generation does not require any technique. Concerned with the matrix BM_2 generation, recall that it is composed from k left columns of the matrix M^{-1}. Finding M^{-1} is known to be non NP problem.

Now we proceed with calculation of the number of bit operations used to find the Boolean product of two matrices. Considering the worst case in such calculation, when two matrices both of size n are multiplied. Note that the Boolean product of A and B matrices is obtained in analogous way to the ordinary product of those matrices, but with addition replaced with the operation $\vee$ and with multiplication replaced with the operation $\wedge$. There are n^2 entries in both matrices A and B. Total n ORs and n ANDs are used to find an entry of $A \times B$ matrix. Hence, $2n$ bit operations are used to find each entry. Therefore, $2n^3$ bit operations are required to compute $A \times B$.

As the XOR combination of two matrices of size $n \times n$ is carried out in n^2 steps, the calculation of BM_4 will take $2n^3 + n^2$ bit operations.

The private key generation computational cost is the following:

- Constructing the BMS, that is not a calculation consuming operation.
- Calculating the inverse Boolean permutation, BP^{-1}, with n number of terms, which involves n bit operations
- Determining BM_2 from the matrix M of size $n \times n$ taking total $k \times n$ bit operation, as BM_2 occupies the left k columns of M.

Encryption and Decryption. Encryption involves two Boolean matrices multiplication and one XOR operation to obtain ciphertext. As it was shown above, this will take, at worst, $2n^3 + n^2$ bit operations. Decryption involves a XOR operation to remove BMS from the ciphertext item matrix, another Boolean product and, finally, inversion of the initial Boolean permutation, BP^-1. Therefore, $2n^3 + n^2 + n$ bit operations are required to decrypt a plaintext symbol.

Thus, we have shown, that in the presented public key cryptosystem both key pair generation and encryption/decryption are polynomial time calculations.

5. CONCLUSION

The presented public key cryptosystem possesses high level of performance due to the usage of only logic operations. Through trivial modifications the cryptosystem can be applied for key exchange and digital signatures purposes as well. One can design hash functions based on Boolean product of matrices as well as with existing methods, reducing the size of initial matrices to an offered final key length providing non reversibility of the procedure.

The security of the BPM cryptosystem is based on the computational complexity of decomposing the Boolean product of large matrices.

The decomposition of the Boolean product of matrices is analogous to integer factorization problem with an important distinguishing feature: unlike the existing cryptosystems, where large numbers are used as products of primes and these products are independent of the component multipliers order relation, the order relation of the component matrices in their product is substantial as Boolean product of matrices is not commutative.

The above analysis shows the efficiency of construction of public key cryptosystems based on Boolean product of matrices, which can significantly enlarge the application frame of public key cryptosystems.

6. ACKNOWLEDGEMENT

The author thanks Prof. G.I. Margarov for valuable advice and support of the idea behind this article.

REFERENCES

[1] B. Schneier, Applied Cryptography, Second Edition, John Wiley & Sons (1996).

[2] Chrystos H.Papadimitriou, "Computatonal Complexity". (1994).

[3] Chuan-Kun Wu, Vijay Varadharajan. Public Key Cryptosystems Based on Boolean Permutations and their Applications. School of Computing & Information Technology, University of Western Sydnay (Nepean), PO Box 10, Kingswood, NSW 2747, Australia.

[4] Yeghisabet Alaverdyan, Gevorg Margarov, Fast asymmetric cryptosystem based on Boolean product of matrices, Proceedings of the 7th IEEE/ACS International Conference on Computer Systems and Applications, Rabat, Morocco (2009).