Linear Algebraic Public Key Encryption Scheme

Gábor Harangozó, Hungary
harangozo.gabor.dr@gmail.com

Abstract A new linear algebraic public key encryption scheme is introduced for post-quantum cryptography. The mathematical problem behind the encryption algorithm is based on matrix factorization and the solution of a linear system of matrix equations including singular matrices as coefficients.

Keywords: post-quantum cryptography, linear algebraic equations, singular matrices, matrix factorization

1. Introduction

Nowadays the most common public key encryption algorithms, like the RSA (Rivest–Shamir–Adleman) algorithm, or the EEC (elliptic-curve cryptography) algorithm, belong to those cryptographic schemes which can be broken using a sufficiently powerful future quantum computer within reasonable time. The security of these algorithms relies on hard mathematical problems, like the integer factorization problem, the discrete logarithm problem or the elliptic-curve discrete logarithm problem, which can be solved on a powerful quantum computer running Shor's algorithm within polynomial time duration. Therefore new algorithms are needed that are substantially secure against the quantum computers. The candidate quantum-resistant encryption algorithms should be based on a mathematical problem, the solution of which has a computational complexity of at least NP-complete. Such a mathematical problem is, among others, the exact nonnegative matrix factorization (NMF), the computational complexity of which is proved to be NP-hard [1]. Various attempts have been previously made to use NMF in public key cryptography, however, most of them applied approximate NMF for decryption [2]. Furthermore, a quantum algorithm has been recently published by Du at al. [3] for solving separable NMF (SNMF) under a logarithmic runtime. However, no sub-exponential algorithm is currently known for general exact NMF.

2.1. Concept of the algorithm

In the proposed encryption algorithm, the plaintext message is represented by a square matrix and the ciphertext message is represented by multiple square matrices. The matrix entries are defined over a finite field F_q, where $q = 2^m$. The ciphertext message is produced using multiple linearly independent linear algebraic equations, in which the variables include the encoded plaintext message and random error components, which are also nxn square matrices. Due to the random error components, the encryption algorithm is probabilistic.

In the linear equations, the random error components are multiplied, at least on their one side, with a respective singular matrix, where the singular matrices themselves are defined as the product of an nxr matrix and an rxn matrix, where r<n. The equation system formed of these linearly independent matrix equations can be solved only through multiplicative decomposition of the singular coefficient matrices into the specific matrix factors. The matrix coefficients of the equation system together form the public key, whereas a specific set of matrices defined using the multiplicative matrix factors of the public key matrices will form the private key.

2.2. Encryption

The ciphered message is computed using the following linear algebraic equations:

$$K_1E_1K_2 + K_3E_2 = Y_1 \qquad \text{(Eq. 1)}$$

$$K_4E_1K_5 + K_6E_2 = Y_2 \qquad \text{(Eq. 2)}$$

$$K_7E_1K_8 + K_9E_2 + M = Y_3 \qquad \text{(Eq. 3)}$$

where M is an nxn matrix representing the plaintext message, E_1 and E_2 are arbitrary nxn random error matrices, K_i are nxn singular matrices, and Y_1, Y_2 and Y_3 are nxn code matrices which together form the ciphered message. The public key matrices K_i are defined as follows:

$$K_1=FT;\ K_2=QA;\ K_3=CR;\ K_4=HT;\ K_5=K_2=QA;\ K_6=DR;\ K_7=JT;\ K_8=QB;\ K_9=GR,$$

where A, B, R and T are arbitrary full-ranked rxn matrices, C, D, F, G, H, J and Q are arbitrary full-ranked nxr matrices, where r<n, A≠ B, and C, D and G are mutually different matrices. Each of the matrices A, B, C, D, F, G, H, J, Q, R and T, which define the public key matrices K_i, should be kept in secret. Additionally, the random error matrices E_1, E_2 are also to be kept in secret by the ciphering party.

From the above definitions of the public key matrices K_i, it is clear that since each of K_i is a singular matrix, the linear equation system cannot be solved using the matrices K_i themselves for determining either the plaintext message matrix M or the random error matrices E_1, E_2.

2.3. Decryption

To determine the plaintext message matrix M from the above equation system, the following steps are to be taken:

a) Let us express the matrix product RE_2 from Eq. 1 as a function of the code matrix Y_1 and the matrix product TE_1Q, i.e. $RE_2 = f_1(Y_1, TE_1Q)$, where C^{-1} is the Moore-Penrose pseudoinverse of C:

$$RE_2 = C^{-1}(Y_1 - FTE_1QA)$$

b) Let us express the matrix product TE_1Q from Eq. 2 as a function of code matrices Y_1 and Y_2, i.e. $TE_1Q = f_2(Y_1, Y_2)$, where A^{-1} is the Moore-Penrose pseudoinverse of A:

$$TE_1Q = (H - DC^{-1}F)^{-1}Y_2A^{-1} - (H - DC^{-1}F)^{-1}DC^{-1}Y_1A^{-1}$$

c) By using the expression of step b) for the matrix product TE_1Q and combining Eq. 1 with Eq. 3, we can express matrix M as a function of the three code matrices Y_1, Y_2 and Y_3, i.e. $M = f_3(Y_1, Y_2, Y_3)$, where $P = (H - DC^{-1}F)$, P is an nxr matrix and is assumed to be full-ranked, P^{-1} is the Moore-Penrose pseudoinverse of P, and I is an nxn identity matrix:

$$M = JP^{-1}DC^{-1}Y_1A^{-1}B - GC^{-1}(I + FP^{-1}DC^{-1})Y_1 - JP^{-1}Y_2A^{-1}B + GC^{-1}FP^{-1}Y_2 + Y_3$$

If P turns out to be rank-deficient, any one or more of C, D, F and H should be changed so that P be full-ranked.

By introducing the following definitions:

$$S_1 = JP^{-1}DC^{-1};\ S_2 = A^{-1}B;\ \ S_3 = GC^{-1}(I + FP^{-1}DC^{-1});\ \ S_4 = JP^{-1};\ S_5 = A^{-1}B;\ \ S_6 = GC^{-1}FP^{-1}$$

the plaintext message matrix M can be expressed as

$$M = S_1Y_1S_2 - S_3Y_1 - S_4Y_2S_5 + S_6Y_2 + Y_3$$

The above defined matrices S_i are nxn square matrices, and they will together form the private key.

2.4. Security considerations

One possible attack against the present encryption scheme is where the attacker attempts to determine the private key matrices on the basis of the public key matrices. As it can be seen from the definition of the public key matrices K_i, the matrices A, B, C, D, F, G, H and J should be determined by the attacker to compute the private key matrices S_i. However, factorization of the public key matrices K_i into the product of two specific matrices is a hard mathematical problem. According to Moitra [4], the best NMF algorithm known runs in time $O(2^r mn)^{O(r^2)}$. The security of the present algorithm against chosen plaintext attacks (CPA) and chosen ciphertext attacks (CCA) has not been deeply analysed yet, but preliminary researches show that with appropriate parameter settings, the algorithm will likely be IND-CPA and IND-CCA2 secure.

3. Conclusion

The proposed encryption scheme is very robust and easy to implement, and it involves a high degree of randomness and great freedom for the selection of the public key matrix factors. Once its security has been justified by the crypto society, it may become a candidate algorithm for post-quantum cryptography.

References

[1] Stephen A. Vavasis, *On the complexity of nonnegative matrix factorization,* SIAM Journal on Optimization, Volume 20, Issue 3, August 2009, pp. 1364-1377.

[2] Shengli Xie et al., *Nonnegative Matrix Factorization Applied to Nonlinear Speech and Image Cryptosystems*, IEEE Transactions on Circuits and Systems, I: Regular Papers, Vol. 55., No. 8, September 2008, pp. 2356-2367.

[3] Yuxuan Du et al., *Quantum Divide-and-Conquer Anchoring for Separable Non-negative Matrix Factorization,* Proc. of the 27th International Joint Conference on Artificial Intelligence (IJCAI-18), 2018, pp. 2093-2099.

[4] A. Moitra, *An almost optimal algorithm for computing nonnegative rank*, Proceedings of SODA, 2013, pp. 1454–1464.