DE GRUYTER
OPEN

TATRA MOUNTAINS
Mathematical Publications

DOI: 10.1515/tmmp-2016-0032
Tatra Mt. Math. Publ. **67** (2016), 85–92

SIMPLE POWER ANALYSIS ATTACK ON THE QC-LDPC McELIECE CRYPTOSYSTEM

Tomáš Fabšič — Ondrej Gallo — Viliam Hromada

ABSTRACT. It is known that a naive implementation of the decryption algorithm in the McEliece cryptosystem allows an attacker to recover the secret matrix P by measuring the power consumption. We demonstrate that a similar threat is present in the QC-LDPC variant of the McEliece cryptosystem. We consider a naive implementation of the decryption algorithm in the QC-LDPC McEliece cryptosystem. We demonstrate that this implementation leaks information about positions of ones in the secret matrix Q. We argue that this leakage allows an attacker to completely recover the matrix Q. In addition, we note that the quasi-cyclic nature of the matrix Q allows to accelerate the attack significantly.

1. Introduction

In 1978, McEliece proposed a public key cryptosystem based on coding theory [6], now called the McEliece cryptosystem. The cryptosystem has never been adopted widely, mainly due to the large size of the public key. The interest in the McEliece cryptosystem has, however, risen recently, since it has become a candidate for post-quantum cryptography.

Since the invention of the McEliece cryptosystem, a number of variants of the cryptosystem have been proposed with the ambition to reduce the size of the public key. An overview of the recent research on McEliece cryptosystem is presented in [9].

In [1], Baldi and Chiaraluce proposed to use quasi-cyclic low-density parity-check codes (QC-LDPC codes) in the McEliece cryptosystem, in order to reduce the size of the public key. Their cryptosystem is now known as the QC-LDPC McEliece cryptosystem. However, in [8], Otmani, Tillich and

2010 Mathematics Subject Classification: 94A60.
Keywords: simple power analysis attack, QC-LDPC McEliece cryptosystem.
This work was supported by NATO's Public Diplomacy Division in the framework of "Science for Peace", Project MD.SFPP 984520.

D a l l o t showed that the proposed system had serious vulnerabilities. In [2], B a l d i et al. proposed an amended version of the cryptosystem which was immunized against the attacks of Otmani, Tillich and Dallot. An important role in the cryptosystem is played by matrices which are formed by blocks of circulant matrices. In [10], it was demonstrated that when the block size is chosen to be an even number, a more efficient information-set decoding attack on the cryptosystem can be executed. However, when the block size is odd, the system remains unbroken. A related cryptosystem was proposed in [7] by M i s o c z k i et al.

In the present paper, we demonstrate that a careless implementation of the QC-LDPC McEliece cryptosystem can allow a simple power analysis attack to be executed on the cryptosystem. The idea of the attack is very similar to the attack on the classical McEliece cryptosystem presented in [4]. In [4], a simple power analysis attack was executed to reveal a permutation matrix P which is a part of the private key in the classical McEliece cryptosystem. The private key in the QC-LDPC McEliece cryptosystem does not contain a permutation matrix P. However, instead of P it contains a matrix Q which has a very small number of ones in every row. In addition, Q has to be formed by blocks of circulant matrices. Our attack recovers this matrix Q. Moreover, we note that the block-circulant structure of Q allows to speed up the attack significantly.

The paper is structured as follows. In Section 2, we review the QC-LDPC McEliece cryptosystem. In Section 3, we describe our implementation of the QC-LDPC McEliece cryptosystem. In Section 4, we describe the attack to recover the secret matrix Q. In Section 5, we discuss a countermeasure against the attack and conclude the paper.

2. The QC-LDPC McEliece cryptosystem

In [1], B a l d i et al. proposed a variant of the McEliece cryptosystem based on LDPC codes—QC-LDPC McEliece cryptosystem. A part of the private key in this cryptosystem is formed by an $(n-k)\times n$ parity-check matrix H of an LDPC code able to correct t errors. The matrix H is formed by a row $\{H_0,\ldots,H_{n_0-1}\}$ of $n_0 = n/(n-k)$ binary circulant blocks with size $p\times p$, where $p = n-k$. Each block has a row weight (i.e., the number of ones in a row) equal to a number w which is small compared to p. If H_{n_0-1} is invertible, a generator matrix G for the code can be obtained as:

$$G = \left[\begin{array}{c|c} \mathbf{I} & \begin{array}{c} \left(H_{n_0-1}^{-1}\cdot H_0\right)^T \\ \vdots \\ \left(H_{n_0-1}^{-1}\cdot H_{n_0-2}\right)^T \end{array} \end{array}\right].$$

The remaining part of the private key is formed by two other matrices: a $k \times k$ invertible matrix S and a sparse $n \times n$ invertible matrix Q. S and Q are formed by blocks of $p \times p$ circulant matrices. In addition, Q has a fixed row weight m. The public key is then computed as follows:

$$G' = S^{-1} \cdot G \cdot Q^{-1}.$$

Encryption is done as follows. Let the original message be u. Then Alice encrypts u as follows:

$$x = u \cdot G' + e,$$

where e is a randomly generated error vector of length n and weight $wt(e) = t' \leq \frac{t}{m}$.

When Bob receives the encrypted message x, he first computes:

$$x' = x \cdot Q = u \cdot S^{-1} \cdot G + e \cdot Q.$$

Vector x' is a codeword of the LDPC code chosen by Bob (corresponding to the information vector $u' = u \cdot S^{-1}$), affected by the error vector $e \cdot Q$, whose maximum weight is t. Bob is able to correct all the errors with very high probability, by means of LDPC decoding, thus recovering u', and then u through a post-multiplication by S.

In [8], O t m a n i et al. demonstrated that this cryptosystem is vulnerable to attacks which exploit the fact that Q is block-diagonal and S is sparse. In order to immunize their cryptosystem against these attacks, B a l d i et al. proposed a version of the QC-LDPC McEliece cryptosystem with the matrix S dense and the matrix Q no longer block-diagonal in [2]. In [2], they proposed two variants of their cryptosystem: the first with parameters $n_0 = 4$, $w = 13$, $p = 4096$, $m = 7$ and $t' = 27$, and the second with parameters $n_0 = 3$, $w = 13$, $p = 8192$, $m = 11$ and $t' = 40$. They further suggested to choose S, so that every block in S has rows with weight approximately equal to $p/2$, with blocks along the diagonal having rows with an odd weight and blocks away from the diagonal having rows with an even weight. As for the matrix Q, B a l d i et al. suggest to obtain Q in the first variant by constructing a matrix of 4×4 circulant blocks with the blocks on the diagonal having rows of weight 1 and the blocks away from the diagonal having rows of weight 2, and by randomly permuting its block rows and columns. Similarly, in the second variant, they suggest to obtain Q by constructing a matrix of 3×3 circulant blocks with the blocks on the diagonal having rows of weight 3 and the blocks away from the diagonal having rows of weight 4, and by randomly permuting its block rows and columns.

In [10], it was demonstrated that when the value of the block size is chosen to be an even number a more efficient information-set decoding attack on the cryptosystem can be executed. However, this attack is not applicable when the block size is odd.

3. Implementation details

We implemented the QC-LDPC McEliece cryptosystem with parameters $n_0 = 3$, $w = 13$, $p = 8192$, $m = 11$ and $t' = 30$. The matrices S and Q in our implementation follow the above-mentioned recommendations from [2]. Thus every block in S has rows with weight approximately equal to $p/2$, with blocks along the diagonal having rows with an odd weight and blocks away from the diagonal having rows with an even weight. The matrix Q is obtained by constructing a matrix of 3×3 circulant blocks with the blocks on the diagonal having rows of weight 3 and the blocks away from the diagonal having rows of weight 4, and by randomly permuting its block rows and columns.

We chose the parameter $p = 8192$ because it allows simple constructions of the matrices S and Q. The attack described in this paper targets the implementation of the decryption algorithm in the QC-LDPC McEliece cryptosystem. Our implementation of the decryption algorithm does not contain any special features allowed by the choice of $p = 8192$ (like using Winograd convolution for multiplication by circulant matrices, for example). Thus, in instances of the QC-LDPC McEliece cryptosystem with other values of p, the decryption algorithm can be implemented in the same manner. Therefore, the attack presented in this paper is equally feasible for instances of the QC-LDPC McEliece cryptosystem with other values of p, including p odd.

Our implementation is based on the project BitPunch [3], which is a free standalone cryptographic library containing implementations of various variants of the McEliece cryptosystem. For decoding the LDPC code we use a bit-flipping algorithm proposed in [7] (the algorithm is denoted as Approach III in [7]). We implemented the cryptosystem on the STM32F407 microcontroller.

4. The attack

4.1. Previous SPA attack on the classical McEliece

Our attack is inspired by the attack on the classical McEliece cryptosystem presented in [4]. In the classical McEliece, the private key is formed by a parity check matrix H_{Goppa} for a Goppa code, by a "scrambling" matrix S_{Goppa} and by a permutation matrix P. The public key is computed as $G'_{\text{Goppa}} = S^{-1}_{\text{Goppa}} \times G_{\text{Goppa}} \times P^{-1}$, where G_{Goppa} is a generator matrix for the Goppa code. Encryption is done in the same manner as in the case of the QC-LDPC cryptosystem: we add a vector of errors to the message u multiplied by G'_{Goppa}. To decrypt, we firstly multiply the ciphertext x by the permutation matrix P. Afterwards, we apply the Patterson algorithm for decoding Goppa codes to decode

the vector $x' = x \times P$. After decoding by the Patterson algorithm, we multiply the result by S and obtain the plaintext u.

The first step in the Patterson algorithm is to compute the syndrome of x' by multiplying x' by the transpose of H_{Goppa}. Such multiplication is often implemented as follows.

(1) We initialise the syndrome s as the vector of zeros.

(2) We go through the entries in x'. If the ith entry is 1 we add the ith row of H^T_{Goppa} to s. If the entry is zero, we do nothing.

In [4], it is observed that if the multiplication is implemented in the above manner, then the power trace of the cryptographic device may reveal positions of ones in x'. By letting the cryptographic device decrypt all ciphertexts with the hamming weight 1 and by recording and analysing the corresponding power traces, authors of [4] were able to recover the secret permutation matrix P.

4.2. Observing leakage in the QC-LDPC McEliece

Inspired by the attack on the classical McEliece [4], we studied the possibility of performing a simple power analysis attack on the QC-LDPC McEliece cryptosystem.

Let c_i be the ciphertext with 1 in the ith position and with the remaining positions filled with zeros. For various values of i, we let the cryptosystem decrypt the ciphertext c_i and we recorded the power trace. The first step in the decryption is to multiply c_i with Q. The result x' ($x' = c_i \times Q$) is equal to the ith row of Q. The result x' is then decoded using the bit-flipping algorithm. The first step in the bit-flipping algorithm is to compute the syndrome $s = x' \times H^T$. The multiplication of x' by H^T was implemented as follows:

(1) We initialise the syndrome s as the vector of zeros.

(2) We go through the entries in x'. If the ith entry is 1 we add the ith row of H^T to s. If the entry is zero, we do nothing.

Since each row of Q contained exactly $m = 11$ ones, the multiplication of x' by H^T involved 11 additions of rows of H^T to s. We observed that each of these additions is recognizable in the power trace as the pattern shown in Figure 1. Thus the positions of the pattern indicate positions of ones in x' and thus they indicate positions of ones in the ith row of Q.

4.3. Recovering the matrix Q in the QC-LDPC McEliece—first method

The leakage observed in Section 4.2 can be used to recover the secret matrix Q by a similar method to the one used in the attack described in Section 4.1.

An attacker can make the cryptosystem decrypt all possible ciphertexts of the hamming weight one and for each decryption record the power trace. Afterwards, the attacker can determine positions of the pattern from Figure 1 in each

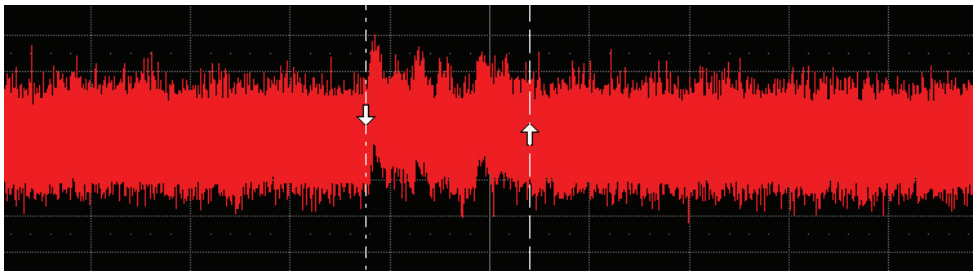

FIGURE 1. The pattern corresponding to the addition of a row of H^T to s. The pattern is bordered by the two vertical white lines in the figure.

power trace. As noted in [4], these positions can be determined automatically by crosscorrelating the power traces with a reference pattern corresponding to the addition of a row of H^T to s.

When trying to recover the matrix Q from the positions of the pattern in the power traces, the attacker will be aided by the knowledge that Q is composed of circulant blocks. Since Q is composed of circulant blocks, it has to have m ones in every column. Thus the attacker knows that the m power traces where the pattern appears earliest correspond to the m rows of Q containing 1 at the first position. Thus the attacker can determine the first column of Q. The first column of Q determines all columns of Q up to the $(p+1)$th column. Analogously, the attacker can recover the remaining columns of Q.

4.4. Recovering the matrix Q in the QC-LDPC McEliece—second method

In Section 4.3, we explained that an attacker can recover the matrix Q after recording power traces of decryptions of every ciphertext with the hamming weight 1. This would require the attacker to run n decryptions. However, an attacker can utilize the quasi-cyclic nature of Q to recover Q with significantly fewer decryptions.

Suppose an attacker records a power trace of the decryption of the ciphertext c_1 (i.e., the ciphertext with one in the first position and zeroes everywhere else). Based on the positions of the pattern from Figure 1 in the power trace, the attacker will try to guess positions of ones in the first row of Q. To improve his guess, the attacker can also record the power trace from the decryption of c_2. Since Q is composed of circulant blocks, by comparing the positions of the pattern in the power trace for c_1 to the positions in the power trace for c_2, the attacker can obtain an estimate of the duration of processing one 0 in x' during the multiplication of x' by H^T.

Suppose that the attacker's guess for the position of the first one was j. To verify the guess, the attacker can utilize the quasi-cyclic nature of Q. He can

record power traces for the decryption of ciphertexts c_{p-j+1} and c_{p-j+2}. If the guess was correct, then in the power trace for c_{p-j+1} the pattern should appear in approximately one third of the part of the power trace corresponding to the multiplication of x' by H^T and in the power trace for c_{p-j+2} it should appear at the very beginning of the part corresponding to the multiplication of x' by H^T. If the guess turns out incorrect, the guess can be altered until it is verified.

Similarly, the attacker can obtain the correct positions of the remaining ones in the first row of Q. The first row of Q determines all rows of Q up to the $(p+1)$th row. To learn the next p rows of Q, the attacker can repeat the process with c_{p+1} instead of c_1. Analogously, the attacker can learn all rows in Q.

Again, the process can be automated using crosscorrelation.

5. Conclusion

We demonstrated that the naive implementation of the multiplication $x' \times H^T$ leaks information about positions of ones in the secret matrix Q in the QC-LDPC McEliece cryptosystem. We argued that this leakage allows an attacker to completely recover the matrix Q. This is a very similar situation to the situation observed in [4], where the leakage of the classical McEliece cryptosystem was analyzed. In both cases the leakage was caused by a multiplication of a vector v and a matrix A being implemented as follows:

(1) The result u is initialized as the vector of zeros.

(2) We go through the entries in v. If the ith entry is 1 we add the ith row of A to u. If the entry is zero, we do nothing.

In [4], a countermeasure was proposed to prevent the leakage in the classical McEliece cryptosystem. It was suggested to implement the multiplication of v and A so that the order of processing the bits of v changes randomly for every ciphertext. This countermeasure can be equally applied to the QC-LDPC McEliece cryptosystem.

REFERENCES

[1] BALDI, M.—CHIARALUCE, F.: *Cryptanalysis of a new instance of McEliece cryptosystem based on QC-LDPC codes,* in: Proceedings IEEE ISIT '07, Nice, France, 2007, pp. 2591–2595.

[2] BALDI, M.—BODRATO, M.—CHIARALUCE, F.: *A new analysis of the McEliece cryptosystem based on QCLDPC codes,* in: 6th Internat. Conf. on Security and Cryptography for Networks—SCN '08 (R. Ostrovsky et al., eds.), Lecture Notes in Math., Vol. 5229, Springer-Verlag, Berlin, 2008, pp. 246–262.

[3] *BitPunch,* `https://github.com/FrUh/BitPunch`

[4] HEYSE, S.—MORADI, A.—PAAR, C.: *Practical power analysis attacks on software implementations of McEliece*, in: Post-Quantum Cryptography (N. Sendrier, ed.), Lecture Notes in Math., Vol. 6061, Springer-Verlag, Berlin, 2010, pp. 108–125.

[5] LÖNDAHL, C.–JOHANSSON, T.–SHOOSHTARI, M. K.–AHMADIAN-ATTARI, M.––AREF, M. R.: *Squaring attacks on McEliece public-key cryptosystems using quasi-cyclic codes of even dimension*, Des. Codes Cryptogr. **80** (2016), pp. 359–377.

[6] MCELIECE, R. J.: *A public-key cryptosystem based on algebraic coding theory*, Deep Space Network Progress Report **44** (1978), 114–116.

[7] MISOCZKI R.—TILLICH J-P.—SENDRIER N.—BARRETO P. S. L. M.: *MDPC--McEliece: new McEliece variants from moderate density parity-check codes*, in: IEEE Internat. Symp. on Information Theory—ISIT '13), Istanbul, 2013, pp. 2069–2073.

[8] OTMANI, A.—TILLICH, J.P.—DALLOT, L.: *Cryptanalysis of two McEliece cryptosystems based on quasi-cyclic codes*, in: The 1st Internat. Conf. on Symbolic Computation and Cryptography—SCC '08, Beijing, China, 2008, Math. Comput. Sci., **3** (2010), no. 2, 129–140.

[9] REPKA, M.—ZAJAC, P.: *Overview of the McEliece cryptosystem and its security,* Tatra Mt. Math. Publ. **60** (2014), pp. 57–83.

[10] KOOCHAK SHOOSHTARI, M.—AHMADIAN-ATTARI, M.—JOHANSSON, T.––REZA AREF, M.: *Cryptanalysis of McEliece cryptosystem variants based on quasi-cyclic low-density parity check codes,* IET Information Security **10** (2016), 194–202.

Received December 1, 2016

Slovak University of Technology
in Bratislava
Faculty of Electrical Engineering and
Information Technology
Ilkovičova 3
SK–812-19 Bratislava
SLOVAKIA

E-mail: tomas.fabsic@stuba.sk
ondrej.gallo@stuba.sk
viliam.hromada@stuba.sk