A New Birthday-Type Algorithm for Attacking the Fresh Re-Keying Countermeasure

Qian Guo[a,1,*], Thomas Johansson[a,1]

[a] *Electrical and Information Technology, Lund University, Lund, Sweden*

Abstract

The fresh re-keying scheme is a countermeasure designed to protect low-cost devices against side-channel attacks. In this paper, we present a new birthday-type attack based on a refined reduction to Ring-LPN with a reducible polynomial. Compared with the previous research, our algorithm significantly reduces the time complexity in the 128-bit leakage model—with an SNR equal to 8 and at most 2^{20} traces, for instance, the key can be recovered using $2^{41.99}$ bit-operations.

Keywords: LPN, Ring-LPN, fresh re-keying, birthday attacks.

1. Introduction

The design of efficient countermeasures to prevent side-channel attacks is one of the most attractive research problems in Lightweight cryptography, a realm of developing efficient and secure low-cost cryptographic primitives for highly constrained environments (e.g., RFID tags, sensors, and other power-constrained devices). This task is very challenging, since after devoting more than 15 years, researchers can hardly find a satisfactory solution—many proposals (e.g., [1, 2]) are considered to be either inefficient or ineffective.

Among all the solutions, the fresh re-keying scheme, first proposed by Medwed et al. in [3], seems to be a promising one, due to a security guarantee provided by never reusing the same key straight-forwardly but generating a fresh session key upon each invocation of the encryption. A natural problem, therefore, is to clarify this intuitive security guarantee, i.e., to determine the concrete complexity to mount an attack on a protected implementation.

Following the Hamming weight leakage model [4], Belaïd et al. reduce this problem to a celebrated hard learning problem, the Learning Parity with Noise (LPN) problem, and resolve it using a BKW [5] variant. This reduction is

[*] Corresponding author

Email addresses: qian.guo@eit.lth.se (Qian Guo), thomas.johansson@eit.lth.se (Thomas Johansson)

[1] Supported by the Swedish Research Council (Grants No. 2015-04528).

further adopted by Pessl and Mangard [6] to form an ISD-style attack employing the soft information of bit reliability. Both algorithms are efficient if 8-bit leakage is measurable; in the worst-case model (e.g., a hardware implementation using a 128-bit operator), however, much effort still needs to be paid to enhance their performance.

In this correspondence, we propose a new algorithm for attacking the fresh re-keying scheme along this line of research. We observe a new reduction to one of the variants—Ring-LPN[2], rather than LPN itself, which provides additional algebraic structures to be further exploited. Noticing that the underlying polynomial is reducible[3], we obtain a much better dimension-bias trade-off, i.e., further diminishing the dimension at the same cost of decreasing the bias, thereby resulting in the enhanced performance.

This attack compares favorably with the previous best algorithm. For example, given an instance noticed in the abstract of [9], we improve the time complexity with a factor of almost one thousand, rather significant in the security level of around 2^{40} bit-operations.

The remaining parts of the paper are organized as follows. We introduce some basic theory in Section 2, and then the main algorithm in Section 3. This is followed by a section stating the numerical results of the new algorithm. We finally conclude the paper in Section 5.

2. Preliminary

We give some preliminaries in this section. Given a positive integer n, we denote the finite field of 2^n elements by $\mathbb{F}_{2^n}$ and the n-dimensional vector space over $\mathbb{F}_2$ by $\mathbb{F}_2^n$. For an irreducible polynomial $P(X)$ with degree n over $\mathbb{F}_2$, we can represent one element in $\mathbb{F}_{2^n}$ as a unique polynomial in $\mathbb{F}_2[X]/(P(X))$, whose degree is less than n. This representation implies a bijection from $\mathbb{F}_{2^n}$ to $\mathbb{F}_2^n$; we, therefore, use the notation $\mathbf{a}$ to denote both $\sum_{i=0}^{n-1} a_i X^i$ and $(a_0, \dots, a_{n-1})$ if there is no ambiguity.

2.1. The Matrix/Vector Representation

Consider two field elements $\mathbf{a} = \sum_{i=0}^{n-1} a_i X^i$ and $\mathbf{k} = \sum_{i=0}^{n-1} k_i X^i$. For the multiplication $\mathbf{a} \cdot \mathbf{k}$ in the field $\mathbb{F}_{2^n}$, we can write it as a matrix/vector product $\mathbf{A}(\mathbf{a})\mathbf{k}^{\mathrm{T}}$, where the i-th column of the matrix $A(\mathbf{a})$ is the transposed coefficient vector of $\mathbf{a} \cdot X^i \bmod P(X)$ and $\mathbf{k}^{\mathrm{T}}$ is the transpose of $\mathbf{k}$. Note that we also have this representation if the polynomial $P(X)$ is reducible.

In the later sections, we consider a more general case, i.e., operations over polynomial ring R, where $R \coloneqq \mathbb{F}_{2^m}[X]/P(X)$ and $P(X)$ is a polynomial with degree n over $\mathbb{F}_{2^m}$. We can represent an element in R as a vector in $\mathbb{F}_2^{mn}$ by using a two-level transformation, i.e., first rewriting it as a length-n vector in $\mathbb{F}_{2^m}^n$ and then representing each of its entries as a length-m vector in $\mathbb{F}_2^m$. Using

[2] We use a slightly generalized definition compared with the original one in [7].

[3] In this sense the new attack is related to the one [8] for reducible Lapin.

this concatenation, therefore, the ring multiplication can be represented in the matrix/vector product manner similarly.

2.2. Fresh Re-keying

This scheme is first proposed in [3] to protect low-cost devices against side-channel attacks, by using a re-keying function g that generates a fresh session key $\mathbf{k}^*$ for every encryption from a fixed master $\mathbf{k}$. Here g is suggested to be the following modular polynomial multiplication over $\mathbb{F}_{2^8}$:

$$\begin{aligned} g : (\mathbb{F}_{2^8}[X]/P(X))^2 &\rightarrow \mathbb{F}_{2^8}[X]/P(X) \\ (\mathbf{a}, \mathbf{k}) &\rightarrow \mathbf{k}^* = \mathbf{a} \cdot \mathbf{k}, \end{aligned} \tag{1}$$

where $\mathbf{a}$ is generated uniformly at random. They suggest that $P(X)$ should have the form of $X^d + 1$, where $d \in \{4, 8, 16\}$. Throughout the rest of the paper, we assume that $d = 16$, which is the most secure parameter as suggested, and we focus on the worst case, i.e. the 128-bit leakage model.

2.3. Leakage Model

Following previous research (see e.g. [4]), we assume that a noisy observation of the Hamming weight of the processed values is leaked, and extract a problem which is a slightly more general version of the problem stated in [9].

Let $w_H(\mathbf{x})$ denote the Hamming weight of a fixed length vector $\mathbf{x}$. If $\mathbf{x} \in \mathbb{F}_2^n$, then $w_H(\mathbf{x})$ is just a textbook definition. Otherwise, the value $w_H(\mathbf{x})$ equates $\sum_{i=1}^{n} w_H(x_i)$, where $w_H(x_i)$ represents the Hamming weight of the binary representation of x_i in $\mathbb{F}_2^m$, if $\mathbf{x} \in \mathbb{F}_{2^m}^n$, i.e., $\mathbf{x}$ is an n-dimensional vector $(x_1, \ldots, x_n)$ over an extension field $\mathbb{F}_{2^m}$. We define the problem based on these definitions.

Definition 1 (Hidden Multiplier (HM) problem). *Let $\mathbf{k} \in \mathbb{F}_{2^m}^n$. A sequence of samples $(\mathbf{a_i}, \mathcal{L}_i), 1 \leq i \leq l$, where $\mathbf{a_i} \in \mathbb{F}_{2^m}^n$ and $\mathcal{L}_i = w_H(\mathbf{a_i} \cdot \mathbf{k}) + \epsilon_i$, where the multiplication is in the ring $\mathbb{F}_{2^m}[X]/P(X)$, is given. Here $P(X)$ is a polynomial in $\mathbb{F}_{2^m}[X]$ of degree n and ϵ_i is Gaussian distributed with standard deviation σ. Recover the hidden value $\mathbf{k}$.*

We see from this definition that the side-channel attack on the fresh re-keying scheme corresponds to solving a HM problem with $m = 8$ and $P(X) = X^{16} + 1$. Later we denote the dimension mn by N, which is set to be 128 for the fresh re-keying scheme by default.

2.4. Reduction to Ring-LPN

We show that HM can be reduced to Ring-LPN.

2.4.1. Ring-LPN

The Ring-LPN problem is first proposed in [7]. Let U^R denote the uniform distribution over R and Ber_η^R denote a distribution over R that each bit of the output element represented in $\mathbb{F}_2^N$ is distributed according to a Bernoulli distribution with parameter η. Here we describe a generalized version.

Definition 2 (Ring-LPN **oracle).** *A* Ring-LPN *oracle* $\Pi_{\text{Ring-LPN}}$ *for an unknown polynomial* $\mathbf{k} \in \mathbf{R}$ *with* $\eta \in (0, \frac{1}{2})$ *returns pairs of the form*

$$(\mathbf{a},\ \mathbf{a} \cdot \mathbf{k} + \mathbf{e})\,,$$

where $\mathbf{a} \overset{\$}{\leftarrow} \mathsf{U}^{\mathbf{R}}$ *and* $\mathbf{e} \overset{\$}{\leftarrow} \mathsf{Ber}_\eta^R$*.*

There are two versions of the Ring-LPN problem, i.e., the decision one and the search one. Aiming to give a key-recovery attack, we focus on the search Ring-LPN problem to recover the unknown value $\mathbf{k}$ after a number q of queries to the oracle.

2.4.2. Transformation By Using Filtering

Only samples with leakage values outside the range $[N/2 - \lambda s, N/2 + \lambda s]$ are kept, where $s = \sqrt{N}/2$ the standard deviation of the Hamming weight. We then rewrite the problem in a form close to the Ring-LPN problem: if $\mathcal{L}_i < N/2$ set $\mathbf{z_i} = \mathbf{0}$; otherwise set $\mathbf{z_i} = \mathbf{1}$. For sample i we can now write

$$\mathbf{a_i} \cdot \mathbf{k} = \mathbf{z_i} + \mathbf{e_i},$$

where $\mathbf{e_i}$ is a length N binary noise vector. It has a distribution determined from the known $\mathcal{L}_i$ value.

We see that this is a Ring-LPN problem, with the difference that in the usual description of Ring-LPN, the noise consists of N independent noise variables with Bernoulli distribution, whereas here the distribution is different. However, we may consider the marginal distribution of single variables and this will give us exactly the Ring-LPN case. The noise variable p of this Ring-LPN sample, given the leakage $\mathcal{L}_i < N/2$, is formulated by $\mathcal{L}_i/N$, if $\mathcal{L}_i < N/2$, and by $1 - \mathcal{L}_i/N$ otherwise.

2.5. Computing the Error Probability in a Folded Setting

Similar to the formulation in [9], we assume that the multiplication $\mathbf{a} \cdot \mathbf{k}$ is uniformly distributed in $\mathbb{F}_2^N$. Thus, the pdf h of $\mathcal{L}(\mathbf{z})$ can be computed by

$$h(x) = 2^{-N} \sum_{y=0}^{N} \binom{N}{y} \phi_{y,\sigma}(x). \tag{2}$$

The proportion of filtered acquisition $F(\lambda)$ is then

$$F(\lambda) = 1 - 2^{-N} \sum_{y=0}^{N} \binom{N}{y} \int_{\frac{N}{2}-\lambda s}^{\frac{N}{2}+\lambda s} \phi_{y,\sigma}(t)dt, \tag{3}$$

for any $\lambda \in \mathbb{R}$, and the error probability averaged over all filtered samples is

$$p(\lambda) = \frac{1}{F(\lambda)} \sum_{y=0}^{N} \frac{\binom{N}{y}}{2^N} \left(\frac{y}{N} \int_{-\infty}^{\frac{N}{2}-\lambda s} \phi_{y,\sigma}(t)dt + \left(1 - \frac{y}{N}\right) \int_{\frac{N}{2}+\lambda s}^{+\infty} \phi_{y,\sigma}(t)dt \right). \tag{4}$$

Notice that the noise parameter σ is a function of the signal-to-noise ratio (SNR for short), i.e., $\sigma = \sqrt{\frac{n}{4 \cdot \mathsf{SNR}}}$. Given the number Q of the obtained traces and the value of SNR, therefore, we can theoretically compute the number q of the filtered samples and the error probability p according to Eq. (3) and Eq. (4) by choosing a filter parameter λ.

3. The Main Result

We now proceed with a formulation of an attack on the fresh re-keying scheme that uses $\mathbb{F}_{2^8}[X]/P(X)$, where $P(X) \in \mathbb{F}_{2^8}[X]$ and $P(X) = X^{16} + 1$.

The key observation is that if the element in $\mathbb{F}_{2^8}$ is represented as a polynomial (or a vector), then the addition is component-wise. We formulate the problem as follows. A trail kept after the filtering gives an instance which can be represented by a ring equation

$$(\sum_{i=0}^{15} a_i(Y)X^i) \cdot (\sum_{i=0}^{15} k_i(Y)X^i) \equiv \sum_{i=0}^{15} (\delta \mathbb{1}(Y) + e_i(Y))X^i \bmod (X^{16} + 1),$$

where $e_i(Y) = \sum_{j=0}^{7} e_{ij} Y^j$, $e_{ij} \in \mathbb{F}_2$, e_{ij} is biased and δ is 0 or 1. Since the polynomial $P(X)$ in this case is reducible, the problem instance can be reduced using the Chinese Remainder Theorem (CRT). The above equation then yields,

$$(\sum_{i=0}^{7} (a_i(Y) + a_{i+8}(Y))X^i) \cdot (\sum_{i=0}^{7} (k_i(Y) + k_{i+8}(Y))X^i) \equiv \sum_{i=0}^{7} e'_i(Y)X^i \bmod (X^8 + 1), \tag{5}$$

where $e'_i(Y) = e_i(Y) + e_{i+8}(Y) = \sum_{j=0}^{7} (e_{ij} + e_{i+8,j})Y^j$. Denoting $e_{ij} + e_{i+8,j}$ by e'_{ij}, we see the bias of the noise variable decreased.

We may perform this process iteratively to reduce the dimension of the problem further. This process, however, also increases the noise level, and thus should be performed only once for an optimized attack on this fresh rekeying countermeasure. Then we can solve it much faster than the previous best algorithm by using birthday-type procedures.

After recovering the partial secret $k_i(Y) + k_{i+8}(Y)$, we determine each of them by solving an LPN problem with dimension halved and noise unchanged. This process requires negligible cost compared with that of solving the original problem.

3.1. Detailed Algorithm Description

Here we describe this new algorithm in steps.

Algorithm 1 A New Attack on Fresh Re-keying

```
1 Reduce the dimension of the transformed Ring-LPN problems
  and represent each of them as 64 LPN samples.
2 Perform two birthday steps.
3 Hypothesis testing using FWHT.
```

3.1.1. Dimension Reduction

The following elaborates the method described in Eq. (5). Given a filtered sample $\mathbf{a_i}$ with bias $\epsilon = 1 - 2p$, where p is the expected error probability, we denote it by a concatenation $(\mathbf{a_i^1}, \mathbf{a_i^2})$, where $\mathbf{a_i^1}$ $(\mathbf{a_i^2})$ is the first (last) length-64 sub-vector of the sample $\mathbf{a_i}$. Bit-wise adding $\mathbf{a_i^1}$ and $\mathbf{a_i^2}$, we obtain a new Ring-LPN sample $(\hat{\mathbf{a}}_\mathbf{i}, \mathbf{0})$ with dimension 64 and bias ϵ^2. We then rewrite the multiplication inside the Ring-LPN sample in the matrix/vector manner as described in Sec. 2.1, thereby obtaining 64 LPN samples. For the fresh rekeying scheme, in particular, the cost for transforming one Ring-LPN sample is at most $4 \cdot 7 \cdot 8 = 224$ bit-operations and several shifts, since the underlying AES polynomial is $x^8 + x^4 + x^3 + x + 1$.

Suppose that q samples are filtered. Then the complexity of this step is about

$$C_1 = 64 \cdot q + 224 \cdot q = 288q \tag{6}$$

bit-operations, which is negligible compared with the complexity of the remaining steps. After this step, we obtain $64q$ LPN samples $(\bar{\mathbf{a}}_\mathbf{i},\ 0)$ with length 64 and bias ϵ^2.

Note: Compared with the algorithm in [9], the improvement of Alg. 1 mainly comes from this step as we reduce the dimension by 64, much larger than the dimension reduced by one birthday-type step in [9], while they both decrease the bias from ϵ to ϵ^2.

3.1.2. Birthday Steps

After the previous dimension reduction step, the traces are transformed to a series of LPN samples. We can then use some standard techniques (like BKW or Information Set Decoding (ISD)) for solving this problem. Since the bias is small and the number of traces is limited, we prefer to use a birthday-type variant, which can also be viewed as one heuristic version of the BKW algorithm similar to those in [10][9].

The BKW algorithm, proposed in [5], is the first sub-exponential algorithm for solving the LPN problem with a constant error-rate. The key idea is sorting the received samples by the last l_1 bits and adding two collided samples, i.e., one find two samples $(\bar{\mathbf{a}}_{\mathbf{i_0}},\ 0)$ and $(\bar{\mathbf{a}}_{\mathbf{i_1}},\ 0)$ such that

$$\bar{\mathbf{a}}_{i_0} + \bar{\mathbf{a}}_{i_1} = (* \quad * \quad \cdots \quad * \quad \underbrace{0 \quad 0 \quad \cdots \quad 0}_{l_1 \text{ symbols}}), \tag{7}$$

where $*$ means any value. We call such a sort-and-merge process one BKW step and will iteratively perform more until the bias is too small to distinguish[4].

Due to the sample limit, we will keep every pair with the same sorted bits and use them to generate a new sample for the future steps. When attacking the fresh re-keying scheme, in particular, only two BKW steps will be performed to remove the last $l = l_1 + l_2$ bits for an optimized attack. Finally, we obtain $2^{21-2l_1-l_2} \cdot q^4$ LPN samples $(\bar{\mathbf{a}}_\mathbf{i},\, 0)$ with length $l_3 = 64 - l$ and bias ϵ^8 as the input to the future hypothesis testing step. The time complexity of this step is

$$C_2 = 2^{11-l_1} \cdot q^2 \cdot (64 - l_1) + 2^{21-2l_1-l_2} \cdot q^4 \cdot l_3 \tag{8}$$

bit-operations.

3.1.3. Hypothesis Testing

The remaining is a hypothesis testing problem, which can be accelerated by making use of Fast Walsh-Hadamard Transform (FWHT). The procedure is as follows. We define $f_{\bar{\mathbf{a}}}$ as the number of samples $(\bar{\mathbf{a}}_\mathbf{i},\, 0)$ with $\bar{\mathbf{a}}_\mathbf{i} = \bar{\mathbf{a}}$. Then the Walsh transform of $f_{\bar{\mathbf{a}}}$ is defined as

$$F(\bar{\mathbf{k}}) = \sum_{\bar{\mathbf{a}} \in \mathbb{F}_\mathbf{2}^{\mathbf{l_3}}} (-\mathbf{1})^{\bar{\mathbf{a}} \cdot \bar{\mathbf{k}}} \cdot \mathbf{f}_{\bar{\mathbf{a}}}. \tag{9}$$

If enough samples are tested, then the right key is

$$\arg \max_{\bar{\mathbf{k}} \in \mathbb{F}_\mathbf{2}^{\mathbf{l_3}}} |F(\bar{\mathbf{k}})| \tag{10}$$

with high probability, where $|F(\bar{\mathbf{k}})|$ denotes the absolute value of $F(\bar{\mathbf{k}})$.

The complexity of this step is about

$$C_3 = l_3 \cdot 2^{l_3} + 2^{21-2l_1-l_2} \cdot q^4 \tag{11}$$

bit-operations.

3.2. Complexity Analysis

In this section, we present the complexity formula of the new algorithm. Let Q be the number of obtained traces. Given SNR and a filtering parameter λ, we compute the number q of the filtered samples and the error probability p as stated in Sec. 2.5. Thus, we can view q and p as inputs to the new algorithm (Alg. 1).

Theorem 1 (The complexity of Alg. 1). *Let q be the number of the filtered* Ring-LPN *samples with length* 128 *and expected error probability p. Then the number of required bit-operations for a successful run of Alg. 1 is*

$$C^* = 288 \cdot q + 2^{11-l_1} \cdot q^2 \cdot (64 - l_1) + 2^{21-2l_1-l_2} \cdot q^4 \cdot (l_3 + 1) + l_3 \cdot 2^{l_3}, \tag{12}$$

[4] The bias is squared after each BKW step.

under the assumption that

$$2^{21-2l_1-l_2} \cdot q^4 \geq \frac{4 \ln 2 \cdot l_3}{\epsilon^{16}}, \tag{13}$$

where l_1 and l_2 are algorithmic parameters, $l_3 = 64 - l_1 - l_2$ and $\epsilon = 1 - 2p$.

Proof. The overall complexity is just the summation of that of all the three steps, i.e.,

$$C^* = C_1 + C_2 + C_3,$$

where C_1, C_2 and C_3 are computed according to Eq. (6), (8) and (11). We add the assumption, Eq. (13), to assure that the samples employed for hypothesis testing are sufficient, thereby yielding a success probability close to 1. □

4. Results

In this section, in the 128-bit leakage model, we present the numerical results for attacking the fresh re-keying countermeasure, assuming for 2^{20}, 2^{22} and 2^{24} traces, respectively. As shown in Tab. 1, with more traces and higher SNR ratio, we provide a more efficient attack. In particular, if obtaining 2^{24} traces and SNR equals 128, we can solve this problem in $2^{37.7}$ bit-operations; even if SNR equals 2 and only 2^{20} traces are measured, the attacking complexity is $2^{45.5}$ bit-operations, still practical for recent computers.

Table 1: The theoretical complexity for attacking the fresh re-keying scheme.

SNR	Parameters						$\log_2 C^*$
	$\log_2(Q)$	l_1	l_2	λ	p	$\log_2(q)$	
128	20	0	30	3.11	0.35	10.93	40.92
	22	2	30	3.39	0.34	11.49	39.29
	24	4	29	3.70	0.33	11.76	37.73
8	20	0	29	3.27	0.36	11.02	41.99
	22	2	29	3.57	0.35	11.57	40.22
	24	2	30	4.00	0.33	11.28	38.83
2	20	0	26	3.73	0.38	11.22	45.54
	22	0	29	4.17	0.37	11.39	43.12
	24	1	29	4.61	0.36	11.38	41.34

Compared with the state-of-the-art work [9], the new algorithm improves substantially. Specifically, if SNR equals 8, they claim[5] a theoretical attack with time complexity $2^{51.68}$ bit-operations and memory complexity 2^{36} bytes

[5] Their analysis is optimistic.

using 2^{20} traces; on the other hand, with the same sample-limit, Alg. 1 recovers the key using solely $2^{41.99}$ bit-operations. Thus, we obtain an improvement with a significant factor of almost 2^{10}, while keeping the memory cost since the vector length of FWHT testing here is only 35 bits. We improve more compared with the recent best attack [6] using an ISD-type algorithm, since according to their estimation, though fewer traces are required, it costs more then 2^{75} bit-operations for the 128-bit leakage case.

5. Conclusion

In this correspondence we have presented a new algorithm for side-channel attacking the fresh re-keying scheme designed to protect constraint devices. Using the reducibility of the transformed Ring-LPN problem, we improve significantly compared with the previous best algorithm in the 128-bit leakage model. One of the future directions is to design a new re-keying scheme to thwart this type of attack.

[1] M.-L. Akkar, C. Giraud, An implementation of DES and AES, secure against some attacks, in: CHES 2001, Springer, 2001, pp. 309–318.

[2] K. Tiri, I. Verbauwhede, Securing encryption algorithms against DPA at the logic level: Next generation smart card technology, in: CHES 2003, Springer, 2003, pp. 125–136.

[3] M. Medwed, F.-X. Standaert, J. Großschädl, F. Regazzoni, Fresh re-keying: Security against side-channel and fault attacks for low-cost devices, in: AFRICACRYPT 2010, Springer Berlin Heidelberg, pp. 279–296.

[4] E. Brier, C. Clavier, F. Olivier, Correlation power analysis with a leakage model, in: CHES 2004, Springer Berlin Heidelberg, pp. 16–29.

[5] A. Blum, A. Kalai, H. Wasserman, Noise-tolerant learning, the parity problem, and the statistical query model, J. ACM 50 (4) (2003) 506–519.

[6] P. Pessl, S. Mangard, Enhancing side-channel analysis of binary-field multiplication with bit reliability, in: CT-RSA, 2016.

[7] S. Heyse, E. Kiltz, V. Lyubashevsky, C. Paar, K. Pietrzak, Lapin: An Efficient Authentication Protocol Based on Ring-LPN, in: Fast Software Encryption, 2012, Springer Berlin Heidelberg, pp. 346–365.

[8] Q. Guo, T. Johansson, C. Löndahl, A new algorithm for solving Ring-LPN with a reducible polynomial, IEEE Transactions on Information Theory 61 (11) (2015) 6204–6212.

[9] S. Belaïd, J.-S. Coron, P.-A. Fouque, B. Gèrard, J.-G. Kammerer, E. Prouff, Improved side-channel analysis of finite-field multiplication, in: CHES 2015, Springer Berlin Heidelberg, pp. 395–415.

[10] É. Levieil, P.-A. Fouque, An improved LPN algorithm, in: R. D. Prisco, M. Yung (Eds.), SCN, Springer-Verlag, 2006, pp. 348–359.