Recovering the Secret on Binary Ring-LWE problem with Random Known bits*

Reynaldo C. Villena[1], **Routo Terada**[1]

[1]Institute of Mathematics and Statistics – University of São Paulo – SP – Brazil

reynaldo@ime.usp.br, rt@ime.usp.br

Abstract. *There are cryptographic systems that are secure against attacks by both quantum and classical computers. Some of these systems are based on the Binary Ring-LWE problem which is presumed to be difficult to solve even on a quantum computer. This problem is considered secure for IoT (Internet of things) devices with limited resources. In Binary Ring-LWE, a polynomial* $\mathbf{a}$ *is selected randomly and a polynomial* $\mathbf{b}$ *is calculated as* $\mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e}$ *where the secret* $\mathbf{s}$ *and the noise* $\mathbf{e}$ *are polynomials with binary coefficients. The polynomials* $\mathbf{b}$ *and* $\mathbf{a}$ *are public and the secret* $\mathbf{s}$ *is hard to find. However, there are Side Channel Attacks that can be applied to retrieve some coefficients (random known bits) of* $\mathbf{s}$ *and* $\mathbf{e}$*. In this work, we analyze that the secret* $\mathbf{s}$ *can be retrieved successfully having at least 50 % of random known bits of* $\mathbf{s}$ *and* $\mathbf{e}$*.*

1. Introduction

The cryptographic community is searching for new quantum-resistant primitives because the main asymmetric cryptosystems such as RSA[1], (EC)DLP[2] are insecure against a quantum computer. Hence, the National Institute of Standards and Technology (NIST) initiated a process to search new public key cryptographic algorithms which are post-quantum secure [Roy et al. 2016]. Some proposals submitted to NIST process are based on Ring-Learning-with-Errors (Ring-LWE) problem because its implementation in software and hardware is efficient.

A variant of the Ring-LWE problem, the Binary Ring-LWE problem, has been recently proposed [Buchmann et al. 2016b]. Its security levels are lower than Ring-LWE. However, it is sufficient against conventional and quantum cryptanalysis [Göpfert et al. 2017, Albrecht 2017, Bogdanov et al. 2007]. In Binary Ring-LWE, a polynomial $\mathbf{a}$ is selected randomly and a polynomial $\mathbf{b}$ is calculated ($\mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e}$ since the polynomials $\mathbf{s}$ and $\mathbf{e}$ have binary coefficients). The polynomials $\mathbf{b}$ and $\mathbf{a}$ are public and the secret $\mathbf{s}$ is hard to find because this difficult corresponds to the hardness of lattice problems, which have theoretical guarantees against powerful quantum computers. However, its implementation in software or hardware, specially in IoT devices, can be vulnerable to Side Channel Attacks [Fan and Verbauwhede 2012, Aysu et al. 2018].

A Side Channel Attack (SCA) is any attack based on Side Channel Information that is obtained when protocols or schemes are executed. Some examples are execution time, power consumption, electromagnetic leaks, sound, and other information that is

*Supported by organization CAPES.

[1]Rivest, Shamir and Adleman cryptosystem based on the NP problem of Prime Factoring.

[2]It is a cryptosystem that uses the Discrete Logarithm Problem over the Elliptic curves.

produced during the running process. These Side Channel Information can be applied to retrieve (hints about) the values of some coefficients (bits) of the secret $\mathbf{s}$ [Buchmann et al. 2016a, Dachman-Soled et al. 2020]. Applying the same concepts, the recovery of bits of noise polynomial $\mathbf{e}$ is feasible.

1.1. Our Contribution

Due to limited resources on IoT devices, the polynomials $\mathbf{s}$ and $\mathbf{e}$ are unprotected and some bits of $\mathbf{s}$ and $\mathbf{e}$ can be retrieved using SCA. Analyzing the mathematical properties between the public parameters (polynomials $\mathbf{b}$ and $\mathbf{a}$), the secret key $\mathbf{s}$ and the noise polynomial $\mathbf{e}$, we show that all secret $\mathbf{s}$ can be successfully retrieved having at least 50 % of bits of $\mathbf{s}$ and $\mathbf{e}$.

2. Preliminaries

For an integer $q \geq 1$, let $\mathbb{Z}_q$ be the residue class ring modulo q and $\mathbb{Z}_q = \{0, ..., q-1\}$. Let $\mathcal{R}_q = \mathbb{Z}_q[x]/(x^n+1)$ denote the polynomial ring modulo x^n+1 where the coefficients are in $\mathbb{Z}_q$. The operations (addition and multiplication) of the elements in $\mathcal{R}_q$ are according to these operations on polynomials.

For $\mathbf{x} \in \mathcal{R}_q$, let $\mathbf{x}[i]$ be the (i)-th coefficient of $\mathbf{x}$ for $0 \leq i < n$. $\mathbb{Z}_q^l$ denotes a set of vectors of length l and their components belong to $\mathbb{Z}_q$. For $\mathbf{x} \in \mathbb{Z}_q^l$, $\mathbf{x}[i]$ denotes the (i)-th component of $\mathbf{x}$ for $0 \leq i < l$. $\{0,1\}^l$ is a set of strings of length l. For $\mathbf{x} \in \{0,1\}^l$, $\mathbf{x}[i]$ denotes the (i)-th bit of $\mathbf{x}$ for $0 \leq i < l$. For a set $\mathcal{S}$, $x \xleftarrow{\$} \mathcal{S}$ denotes that an element x is chosen from $\mathcal{S}$ uniformly at random. For a distribution χ, $x \leftarrow \chi$ denotes that an element x is sampled according to the distribution χ. A polynomial $\mathbf{x} \xleftarrow{\$} \mathcal{R}_q$ means that each coefficient of $\mathbf{x}$ is chosen randomly from $\mathbb{Z}_q$. A polynomial $\mathbf{x} \xleftarrow{\$} \chi^l$ means that each coefficient of $\mathbf{x}$ is chosen randomly according to χ.

The integer $\lfloor x \rceil$ is defined as $\lfloor x + \frac{1}{2} \rfloor \in \mathbb{Z}$.

2.1. Binary Ring-LWE

The Binary Ring-LWE is a new, promising variant of Ring-LWE that achieves smaller key sizes and more efficient computations [Buchmann et al. 2016b]. The security analysis of Binary Ring-LWE is corroborated by several authors [Wunderer 2016, Göpfert et al. 2017, Albrecht 2017]. In the Ring-LWE problem, the secret is selected uniformly at random over $\mathbb{Z}_q^n$, and the noise polynomial is generated by Gaussian or Binomial distribution. However, both polynomials mentioned above are selected uniformly at random over $\mathbb{Z}_2^n$ in Binary Ring-LWE.

The Binary Ring-LWE problem fixes a size parameter n that is a power of 2, a modulus $q \geq 2$. Define $\mathcal{R}_q$ as the ring $\mathbb{Z}_q[x]/(x^n + 1)$ containing all polynomials over the field $\mathbb{Z}_q$ in which x^n is identified with -1. For secret $\mathbf{s} \in \mathbb{Z}_2^n$, we define the Binary Ring-LWE distribution $\mathrm{A}_{n,q,\mathbf{s}}$ over $\mathbb{Z}_q^n \times \mathbb{Z}_q^n$, obtained by choosing independently a vector $\mathbf{a} \in \mathbb{Z}_q^n$ and $\mathbf{e} \in \mathbb{Z}_2^n$, that are selected uniformly at random, respectively. One sample of distribution $\mathrm{A}_{n,q,\mathbf{s}}$ is $(\mathbf{a}, \mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e})$, where additions and multiplications are performed in $\mathcal{R}_q$:

$$\{(\mathbf{a}, \mathbf{b}) \mid \mathbf{a} \leftarrow \mathbb{Z}_q^n, \mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e}, \mathbf{e} \leftarrow \mathbb{Z}_2^n)\} \in \mathbb{Z}_q^n \times \mathbb{Z}_q^n$$

Definition 1 (Binary Ring-LWE oracle) *A Binary Ring-LWE oracle $\mathcal{A}_{n,q,\mathbf{s}}$ is an oracle which outputs independent random samples according to the* $\mathrm{A}_{n,q,\mathbf{s}}$ *distribution.*

There are two versions of LWE problems:

- **Search Binary Ring-LWE$_{q,\chi,m}$:** Given access to a Ring-LWE oracle $A_{n,q,\mathbf{s}}$, find the vector $\mathbf{s}$.
- **Decision Binary Ring-LWE$_{q,\chi,m}$:** The Decision Ring-LWE problem is to distinguish between the uniform distribution over $\mathbb{Z}_q^{2n}$ and the samples given by the oracle $A_{n,q,\mathbf{s}}$.

3. Recovering the secret using random known bits

We have a Binary Ring-LWE instance $\mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e}$, and some random bits of $\mathbf{s}$ and $\mathbf{e}$ are known (the recovery of these bits can be seen in Appendix A). The Binary Ring-LWE instance $\mathbf{b} = \mathbf{a}.\mathbf{s} + \mathbf{e}$ can be written as matrix operations.

$$\begin{bmatrix} \mathbf{b}[0] \\ \mathbf{b}[1] \\ \vdots \\ \mathbf{b}[n-1] \end{bmatrix} = \begin{bmatrix} \mathbf{a}[0] & -\mathbf{a}[n-1] & \dots & -\mathbf{a}[1] \\ \mathbf{a}[1] & \mathbf{a}[0] & \dots & -\mathbf{a}[2] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}[n-1] & \mathbf{a}[n-2] & \dots & \mathbf{a}[0] \end{bmatrix} \begin{bmatrix} \mathbf{s}[0] \\ \mathbf{s}[1] \\ \vdots \\ \mathbf{s}[n-1] \end{bmatrix} + \begin{bmatrix} \mathbf{e}[0] \\ \mathbf{e}[1] \\ \vdots \\ \mathbf{e}[n-1] \end{bmatrix}$$

Each $\mathbf{b}[i]$ can be expressed as a system of equations

$$\mathbf{b}[i] = \sum_{j=0}^{i} \mathbf{a}[i-j].\mathbf{s}[j] - \sum_{j=i+1}^{n-1} \mathbf{a}[j].\mathbf{s}[n+i-j] + \mathbf{e}[i] \qquad \text{for } 0 \leq i \leq n-1 \qquad (1)$$

It results in n equations with $2n$ variables (bits of $\mathbf{s}$ and $\mathbf{e}$) that results hard to solve. However, some bits of $\mathbf{s}$ and $\mathbf{e}$ are known.

Let $\mathbf{e}_k$ and $\mathbf{e}_u$ be the sets of known bits and unknown bits of noise polynomial $\mathbf{e}$ ($|\mathbf{e}_k| + |\mathbf{e}_u| = n$). Let $\mathbf{s}_k$ and $\mathbf{s}_u$ be the sets of known bits and unknown bits of the secret polynomial $\mathbf{s}$ ($|\mathbf{s}_k| + |\mathbf{s}_u| = n$). Let α be the percentage of known bits of $\mathbf{s}$ and $\mathbf{e}$ ($|\mathbf{s}_k| + |\mathbf{e}_k| = \alpha.2n$). Therefore, considering the known bits of $\mathbf{e}_k$ and $\mathbf{s}_k$ we have n equations with $|\mathbf{e}_u| + |\mathbf{s}_u|$ variables. One condition to have the solution of a system of equations is that the number of variables must be lower than or equal to the number of equations:

$$\begin{aligned} |\mathbf{e}_u| + |\mathbf{s}_u| &\leq n \\ |\mathbf{e}_u| &\leq |\mathbf{s}_k| \qquad \text{because } |\mathbf{s}_k| + |\mathbf{s}_u| = n \end{aligned}$$

The above condition is always accomplished since we can set $|\mathbf{e}_u| = 0$. One way to get $|\mathbf{e}_u| = 0$ is discarding all equations in Equations (1) where the value of $\mathbf{e}[i]$ is unknown, resulting in $|\mathbf{e}_k|$ equations and $|\mathbf{s}_u|$ variables. This new system of equations needs $|\mathbf{e}_k| \geq |\mathbf{s}_u|$ to be solved.

$$\begin{aligned} |\mathbf{e}_k| + |\mathbf{s}_k| + &\geq |\mathbf{s}_u| + |\mathbf{s}_k| && \text{because } |\mathbf{e}_k| \geq |\mathbf{s}_u| \\ |\mathbf{e}_k| + |\mathbf{s}_k| &\geq n && \text{because } |\mathbf{s}_k| + |\mathbf{s}_u| = n \\ \alpha.2n &\geq n && \text{because } |\mathbf{s}_k| + |\mathbf{e}_k| = \alpha.2n \\ \alpha &\geq \frac{1}{2} \end{aligned}$$

In other words, we need at least 50 % of bits of $\mathbf{s}$ and $\mathbf{e}$ to retrieve all unknown bits of $\mathbf{s}$, allowing us to know the actual value of the secret $\mathbf{s}$.

4. Experiments and Results

An algorithm was implemented in 20 lines of code using sageMath. This algorithm contains the Gaussian Elimination method to solve equations. It was executed on a processor Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz with 3 Mb of cache and 8 GB of DDR4 Memory. The code is available online at `https://github.com/reynaldocv/sbseg2023`.

For our experiments, we work with parameters $n = 256, q = 256$ (parameters defined in [Aysu et al. 2018]) and $\alpha \in [49, 60]$. For each value of α, 1000 public keys $\langle \mathbf{b}, \mathbf{a} \rangle$ were generated and for each public key, 100 samples were generated with α percentage of random known bits of $\mathbf{s}$ and $\mathbf{e}$. In total, 1200000 experiments were executed. For all experiments, the method Gaussian Elimination was applied and the unknown bits of the secret $\mathbf{s}$ were successfully retrieved since $\alpha \geq 50\%$. Each experiment takes at most 6 seconds.

For experiments with $\alpha = 49\%$, the number of variables are greater than number of equations, therefore there are many candidates to $\mathbf{s}$. With a smaller value α, the number of candidates generated is increased exponentially.

5. Conclusion

We described a scenario where some random bits of the polynomials $\mathbf{s}$ and $\mathbf{e}$ can be retrieved. Using the mathematical definition of the Binary Ring-LWE problem and these retrieved known bits, the unknown bits of the secret $\mathbf{s}$ can be retrieved using the Gaussian Elimination method. Our result was proved mathematically and experimentally where we need at least 50 % of random known bits of $\mathbf{s}$ and $\mathbf{e}$ to retrieve the actual value of the secret $\mathbf{s}$. In other words, with a sufficient number of known bits of $\mathbf{s}$ and $\mathbf{e}$, the (Binary) Ring-LWE is a solvable system of equations.

This work can be extended to Ring-LWE problem giving us the same result. However, we need to retrieve some random known coefficients of $\mathbf{s}$ and $\mathbf{e}$, but this task can be more difficult, yet not impossible because the coefficients of $\mathbf{s}$ and $\mathbf{e}$ are integers, not bits (0 or 1).

As we know, the hardness of the (Binary) Ring-LWE problem is to find $\mathbf{s}$. There are some works focused on the protection of the secret $\mathbf{s}$ [Aysu et al. 2018] and the polynomial $\mathbf{e}$ is left out since $\mathbf{e}$ is only used one time (in the KEYGEN process, see Appendix A). We must be more careful with the noise polynomial $\mathbf{e}$ because the recovery of its coefficient makes the (Binary) Ring-LWE problem weaker.

Acknowledgement This paper was partially funded by the project INCT of the Future Internet for Smart Cities: FAPESP proc. 2014/50937-1 / CNPq proc. 465446/2014-0

References

Albrecht, M. R. (2017). On dual lattice attacks against small-secret lwe and parameter choices in helib and seal. In *Advances in Cryptology–EUROCRYPT 2017: 36th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Paris, France, April 30–May 4, 2017, Proceedings, Part II*, pages 103–129. Springer.

Aysu, A., Orshansky, M., and Tiwari, M. (2018). Binary ring-lwe hardware with power side-channel countermeasures. In *2018 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, pages 1253–1258. IEEE.

Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J., Seurin, Y., and Vikkelsoe, C. (2007). Present: An ultra-lightweight block cipher. In *Cryptographic Hardware and Embedded Systems-CHES 2007: 9th International Workshop, Vienna, Austria, September 10-13, 2007. Proceedings 9*, pages 450–466. Springer.

Buchmann, J., Göpfert, F., Güneysu, T., Oder, T., and Pöppelmann, T. (2016a). High-performance and lightweight lattice-based public-key encryption. In *Proceedings of the 2nd ACM international workshop on IoT privacy, trust, and security*, pages 2–9.

Buchmann, J., Göpfert, F., Player, R., and Wunderer, T. (2016b). On the hardness of lwe with binary error: Revisiting the hybrid lattice-reduction and meet-in-the-middle attack. In *Progress in Cryptology–AFRICACRYPT 2016: 8th International Conference on Cryptology in Africa, Fes, Morocco, April 13-15, 2016, Proceedings*, pages 24–43. Springer.

Dachman-Soled, D., Ducas, L., Gong, H., and Rossi, M. (2020). Lwe with side information: Attacks and concrete security estimation. Cryptology ePrint Archive, Paper 2020/292. `https://eprint.iacr.org/2020/292`.

Fan, J. and Verbauwhede, I. (2012). An updated survey on secure ecc implementations: Attacks, countermeasures and cost. *Cryptography and Security: From Theory to Applications: Essays Dedicated to Jean-Jacques Quisquater on the Occasion of His 65th Birthday*, pages 265–282.

Göpfert, F., van Vredendaal, C., and Wunderer, T. (2017). A hybrid lattice basis reduction and quantum search attack on lwe. In *Post-Quantum Cryptography: 8th International Workshop, PQCrypto 2017, Utrecht, The Netherlands, June 26-28, 2017, Proceedings 8*, pages 184–202. Springer.

Lyubashevsky, V., Peikert, C., and Regev, O. (2013). On ideal lattices and learning with errors over rings. *Journal of the ACM (JACM)*, 60(6):1–35.

Roy, S. S., Karmakar, A., and Verbauwhede, I. (2016). Ring-lwe: applications to cryptography and their efficient realization. In *Security, Privacy, and Applied Cryptography Engineering: 6th International Conference, SPACE 2016, Hyderabad, India, December 14-18, 2016, Proceedings 6*, pages 323–331. Springer.

Wunderer, T. (2016). Revisiting the hybrid attack: Improved analysis and refined security estimates. *Cryptology ePrint Archive*.

A. Recovering random known bits

In this section, we describe one Public Key Encryption Scheme based on Binary Ring-LWE problem, and the recovery of some bits of the secret and noise polynomials is explained.

A.1. Binary Ring-LWE Public Key Encryption Scheme

Binary Ring-LWE Public Key Encryption was proposed in [Lyubashevsky et al. 2013]. The algorithms are shown in Figure 1 and described below.

- **KEYGEN**: Key generation sets a polynomial $\mathbf{a}' \in \mathcal{R}_q$ and samples two polynomials $\mathbf{r}_1, \mathbf{r}_2 \in \{0,1\}^n$ and compute $\mathbf{p} = \mathbf{r}_1 - \mathbf{r}_2.\mathbf{a}' \in \mathcal{R}_q$.
  The public key is $\mathbf{pk} = \langle \mathbf{p}, \mathbf{a}' \rangle$ and the private key (secret key) is $\mathbf{sk} = \langle \mathbf{r}_2 \rangle$.
- **ENCRYPTION**: Firstly, three polynomials $\mathbf{e}_1, \mathbf{e}_2$ and $\mathbf{c}_3$ are selected uniformly random over $\mathbb{Z}_2^n$. The ciphertext is the pair of polynomials $\mathbf{c}_1 = \mathbf{a}'.\mathbf{e}_1 + \mathbf{e}_2$ and $\mathbf{c}_2 = \mathbf{p}.\mathbf{e}_1 + \mathbf{e}_3 + \bar{\mathbf{m}} \in \mathcal{R}_q$. The value of $\bar{\mathbf{m}}$ is obtained by multiplying each coefficient of message $\mathbf{m}$ with $\lfloor \frac{q}{2} \rfloor$.
- **DECRYPTION**: This algorithm reconstructs the message $\mathbf{m}$ by using the secret key $\mathbf{sk} = \langle \mathbf{r}_2 \rangle$. It Computes $\mathbf{m}' = \mathbf{c}_1.\mathbf{r}_2 + \mathbf{c}_2$ and decodes the coefficients of $\mathbf{m}'$ using the threshold decoder $th(.)$. Each coefficient of $\mathbf{m}'$ is processed separately, returning a binary value, if $\mathbf{m}'[i]$ lies in the range $(q/4, 3q/4)$, then the value of $\mathbf{m}[i]$ is 1 else the value of $\mathbf{m}[i]$ is 0. The threshold decoder $th(.)$ can be defined as $th(x) = \lfloor 2.x/q \rceil \pmod 2$.

KEYGEN(n)
1 $\mathbf{a}', \mathbf{r}_1, \mathbf{r}_2 \xleftarrow{\$} \mathcal{R}_q, \{0,1\}^n, \{0,1\}^n$
2 $\mathbf{p} = \mathbf{r}_1 - \mathbf{a}'.\mathbf{r}_2$
3 **return** $\mathbf{pk} = \langle \mathbf{p}, \mathbf{a}' \rangle, \mathbf{sk} = \langle \mathbf{r}_2 \rangle$

ENCRYPTION($\mathbf{pk} = \langle \mathbf{p}, \mathbf{a}' \rangle, \mathbf{m}$)
1 $\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_3 \xleftarrow{\$} \{0,1\}^n, \{0,1\}^n, \{0,1\}^n$
2 $\mathbf{c}_1 = \mathbf{a}'.\mathbf{e}_1 + \mathbf{e}_2$
3 $\bar{\mathbf{m}} = \lfloor \frac{q}{2} \rfloor.\mathbf{m}$
4 $\mathbf{c}_2 = \mathbf{p}.\mathbf{e}_1 + \mathbf{e}_3 + \bar{\mathbf{m}}$
5 **return** $\mathbf{c}_1, \mathbf{c}_2$

DECRYPTION($\mathbf{c}_1, \mathbf{c}_2, \mathbf{sk} = \langle \mathbf{r}_2 \rangle$)
1 $\mathbf{m}' = \mathbf{c}_1.\mathbf{r}_2 + \mathbf{c}_2$
2 $\mathbf{m} = th(\mathbf{m}')$
3 **return** m

Figure 1. Binary Ring-LWE Public Key Encryption

The Binary Ring-LWE PKE scheme was proposed for Lightweight applications (e.g. constrained IoT nodes). The security level achieved is 84 bits against conventional computers and 73 bits against quantum computers [Wunderer 2016, Göpfert et al. 2017]. This scheme was implemented in hardware using a configuration that sets $n = 256$ and $q = 256$ [Aysu et al. 2018].

In Binary Ring-LWE PKE scheme, the public key $\mathbf{pk} = \langle \mathbf{p}, \mathbf{a}' \rangle$ and the secret key $\mathbf{sk} = \langle \mathbf{r}_2 \rangle$ are mathematically related $\mathbf{p} = \mathbf{r}_1 - \mathbf{r}_2.\mathbf{a}'$, and it can be expressed as a Binary Ring-LWE instance $\mathbf{b} = \mathbf{s}.\mathbf{a} + \mathbf{e}$ with $\mathbf{b} = \mathbf{p}, \mathbf{s} = \mathbf{r}_2, \mathbf{e} = \mathbf{r}_1$, and $\mathbf{a} = -\mathbf{a}'$.

We know the value of $\mathbf{r}_1$ is used only in the KEYGEN process, therefore we can retrieve some bits of $\mathbf{r}_1$ applying a SCA when the KEYGEN process is executed. As we know, the value $\mathbf{p}$ is defined as $\mathbf{p} = \mathbf{r}_1 - \mathbf{r}_2.\mathbf{a}$. Firstly, the value $\mathbf{r}_2.\mathbf{a}$ is calculated, and the value of $\mathbf{r}_1$ is added. Each i-th bit of $\mathbf{r}_1$ with a value equal to zero, does not modify the value of the bit $(\mathbf{r}_2.\mathbf{a})[i]$. However, when the i-th bit of $\mathbf{r}_1$ is one, the value of $(\mathbf{r}_2.\mathbf{a})[i]$ is altered. This adjustment provokes a power consumption, timing delay, and other information that can be measured, allowing us to differentiate the bits one from zero of $\mathbf{r}_1$ [Aysu et al. 2018]. Using other SCA, the absolute value of one coefficient of $\mathbf{s}$ can be retrieved. Therefore the recovery of some bits of $\mathbf{s}$ is feasible since $\mathbf{s} \in \{0,1\}^n$ [Dachman-Soled et al. 2020]. The value of $\mathbf{r}_2$ is used in a multiplication operation in KEYGEN and DECRYPTION processes. These multiplications are exploited to retrieve the bits of $\mathbf{r}_2$. If the bit $\mathbf{r}_2[i]$ is zero no register is modified; else, when the bit $\mathbf{r}_2[i]$ is equal to one, a sum operation is done. This difference can help retrieve the bits of $\mathbf{r}_2$. Aysu analyzed the correlation between the bits and power consumption (Simple Power Analysis and Differential Power Analysis) to retrieve the bits of $\mathbf{r}_2$ [Aysu et al. 2018].

Summarizing, the recovery of some bits of $\mathbf{s} = \mathbf{r}_2$ and $\mathbf{e} = \mathbf{r}_1$ is feasible using some Side Channel Attacks,