Side-Channel Analysis of Kyber's Arithmetic Encodings: a Cautionary Note

No Author Given

No Institute Given

Abstract.

1 Introduction

PKE schemes are threatening under quantum computing. This thread leads to the need for new cryptographical schemes that are secure even if the adversary has sufficient quantum power. In 07/2023, Kyber was officially selected as the standard post-quantum cryptographical scheme by NIST. The security of Kyber is based on the hardness of solving the learning-with-errors problem in module lattices (MLWE problem [66]). Despite the concrete CPA-secure for public key encryption (PKE) and CCA-secure for key encapsulation mechanism (KEM), Kyber has been proven to be vulnerable under various Side-channel Attacks (SCA), from SPA in FiXme: citation to DPA in FiXme: citation. Therefore, many publications were dedicated to improving side-channel resilience for Kyber's implementation. The most prominent countermeasure used is masking with several optimizations. Masking for Kyber differs from masking for former symmetric or asymmetric key schemes since the computations happen in binary and prime fields and the special distribution of the secret. Even though it can use ready tools available for both cases, Boolean masking for the first field and arithmetic masking for the latter. Both masking schemes have been carefully studied and yielded very concrete, provable security levels FiXme: cite Provable proof for Boolean masking and arithmetic masking. However, constraints over these concrete results do not allow a direct induction to arithmetic masking for Kyber since the coefficients of the secret polynomials have centered binomial distribution on relatively small support. Therefore, in this note, we, for the first time, take a closer look at arithmetic masking, specifically for Kyber, and provide a distinct angle of how different the scheme is compared to former well-studied encodings, thus motivating a deeper look at the subject.

Add brief SCA into

add relevant citations

2 Background

We now recall some necessary notations and notions for the rest of the paper.

maybe split it to subsection? Masking proofs, arithmetic noise amplification, ...

Crystal-Kyber

Polynomial Arithmetic Kyber computations work on polynomial ring $R_q = \mathbb{Z}_q[X]/(X^n + 1)$ where q is a prime that allow efficient Number Theory Transformation and n is the size of message space. The polynomials are denoted as lower case $f \in R_q$ and have the form:

$$f = f_0 + f_1 \cdot X^1 + \cdots + f_{n-1} \cdot X^{n-1} \tag{1}$$

where each coefficient f_i lies in one residue class in $\mathbb{Z}_q$. The secret $\mathbf{s}$ in Kyber is a vector of k polynomials in R_q^k such that each polynomial $\mathbf{s}[i] \in R_q$.

Mention PKE, KEM Encaps Decaps?

Centered binomial distribution For the sake of simplicity we use upper cases to denote random variables and calligraphic letters to denote sets. A random variable X we denote $X \xleftarrow{\$} \mathcal{X}$ where X is uniform distributed over $\mathcal{X}$ and $X \leftarrow D$ when X is chosen according to the distribution D.
Noise in Kyber is sampled from centered binomial distribution β_{eta} for $\eta = 2$ or $\eta = 3$ and is defined as:

$$s \leftarrow \beta_\eta(a_1, a_2, \ldots, a_\eta, b_1, b_2, \ldots, b_\eta) \xleftarrow{\$} \{0,1\}^{2\eta} s = \sum_{i=1}^{\eta} (a_i - b_i)$$

Add spec table

The parameters set of Kyber for Round 3 of the NIST competition is summarized in below table: Since this note focus on encoding of polynomials we take n, q, η value as in the table but keep $k = 1$ for simplicity of the analysis.

Sharing Encoding Masking is a popular countermeasure against DPA that aims to randomizing the intermediate values that are processed by the device thus making the side-channel leakage independent of the sensitive values. It generally works for most of the cryptographic schemes and has been extensively studied, even got provable security. The core idea of masking is to probabilistically split the sensitive variables into d shares, the cryptographic computations process on shares and only combined at the end to get the correct output. If the information of any $d-1$ tuple of shares reveals nothing about the sensitive value then d-share masking (i.e. $d+1$ masking) was proven to be secure against $d+1$-order SCA. FiXme: Double check and add citation

In a d-share masked implementation, each intermediate variable is concealed by d shares. The process of splitting a target value into d *random* values is called sharing encoding. The encoding representation determines the relations among shares and its target value, thus determines how the computations on shares are combined in the end. We extend the definition in [PR13] in order to simplify the notations.

Definition 1 (d-share encoding). *Let $\mathcal{X}$ is a set in a group $(G, *)$ where $*$ is some group operation, let d be a positive integer. The d^{th}-share encoding of*

$x \in \mathcal{X}$ is a maps

$$Enc_d^* : \mathcal{X} \to G^d :$$
$$x \mapsto (x_1, x_2, \ldots, x_d)$$

such that $(x_i)_{i=1}^{d-1} \xleftarrow{\$} G$ *and* $x = x_1 * x_2 * \cdots * x_d$

Difference masking schemes have different encoding representation, for example, in symmetric schemes where the working group is $\mathbb{Z}_{256}$, for 2-share Boolean masking $\text{Enc}_2^{\oplus}(X) = (X_1, X_2)$ where $X_1 \xleftarrow{\$} \mathbb{Z}_{256}$, $X_2 = X \oplus X_1$ and for multiplicative masking, the encoding representation is in the form $\text{Enc}_2^{\otimes}(X) = (X_1, X_2)$ where $X_1 \xleftarrow{\$} \mathbb{Z}_{256}$, $X_2 = X \otimes X_1$ or $X_2 = X \otimes X_1^{-1}$.

As mentioned earlier, the secret polynomial in Kyber can be express as a vector in $\mathbb{Z}_q$: $s = [s_0, s_1, \ldots, s_n]$, $s_i \leftarrow \beta_\eta$, all available masking schemes for polynomial computation of Kyber FiXme: citations use additive arithmetic encoding, e.g.

$$Enc_2^+(s) = (x_1, x_2)$$
$$x_1 \xleftarrow{\$} \mathbb{Z}_q^n$$
$$x_2 = s - x_1 \mod q$$

For the rest of the note, we only study additive arithmetic encoding, thus, to simplify the notation we omit the operation specification of the encoding and keep d subscript to specify the number of shares. Furthermore, the coefficients are independent FiXme: Kyber algo citation and the encoding acts on each coefficient independently, hence, we only consider encoding for one coefficient (i.e. 1-D variable) instead of the full polynomial (i.e. 256-D variable).

Similar to multiplicative encoding in $\mathbb{Z}_{256}$, additive arithmetic encoding in $\mathbb{Z}_q$ can have different representation, e.g. for $\text{Enc}_2(X) = (X_1, X_2)$ then there are two different ways to combine the two shares into protect variable X as:

$$X_2 = (X - X_1) \mod q \qquad \text{i.e.} X = (X1 + X2) \mod q \text{ or}$$
$$X_2 = (X + X_1) \mod q \qquad \text{i.e.} X = (X2 - X1) \mod q$$

To distinguish different representations, as an abuse of notation, we denote:

$$\text{Enc}_d^{\text{sum}}(X) = (X_1, X_2, \ldots, X_d), \qquad \text{Enc}_d^{\text{diff}}(X) = (X_1, X_2, \ldots, X_d)$$
$$X_i \xleftarrow{\$} \mathbb{Z}_q^n, \text{for } i \in \{1, \ldots, d-1\}$$
$$X_d = X + \sum_{i=1}^{d-1} X_i \mod q, \qquad X_d = X - \sum_{i=1}^{d-1} X_i \mod q$$

Metrics

We next recall some notions relevant in SCA.

First, let X be the sensitive variable that is masked using the d-share encoding $\text{Enc}_d(X) = (X_1, X_2, \ldots, X_d)$, the leakage of each share can be written as $L(X_i) = \delta(X_i) + N_i$ where δ is a deterministic function of X_i and N_i denotes the random noise. The full leakage vector corresponding the process on X can be express as $\mathbf{L} = (L(X_1), L(X_2), \ldots, L(X_d))$.

Signal-to-Noise Ratio Signal-to-Noise Ration (SNR) indicates the ratio between the signal and the noise component of a measurement and is widely used in electrical engineering and signal processing. In this note, we use SNR as a initial tool to detect relevant points that seems to carry useful information about target value in the measurements (i.e. Point of Interest (PoI)). SNR of variable X in leakage L is computed as

$$\text{SNR} = \frac{Var_X[\mathbf{E}_L[L_x]]}{\mathbf{E}_X[Var_L[L_x]]}$$

Mutual Information and Perceived Information Information theoretic framework FiXme: IT citation is an usual tool to quantitatively analyze the worst-case security provided by a countermeasure. The mutual information (MI) measures the 'amount of information' obtained about one random variable (e.g. the targeting variable X) by observing the other random variable (e.g. the leakage L corresponding to the computation on X) and is computed by:

$$\text{MI}(X; \mathbf{L}) = \text{H}(X) - \text{H}(X|\mathbf{L}), \tag{2}$$

where $\text{H}(X)$ is the entropy of variable X and $\text{H}(X|L)$ is the conditional entropy of X given L and respectively computed by:

$$\text{H}(\boldsymbol{X}) = \sum_{x \in \mathcal{X}} p(x) \cdot \log_2 p(x) \tag{3}$$

$$\text{H}(X; \boldsymbol{L}) = \sum_{x \in \mathcal{X}} p(x) \int_{\boldsymbol{l} \in \mathcal{L}^d} f(\boldsymbol{l}|x) \cdot \log_2 p(x|\boldsymbol{l}), \tag{4}$$

where $p(x)$ is the probability mass function of X at the point x and $p(x|\boldsymbol{l})$ is computed as $\frac{f(\boldsymbol{l}|x)}{\sum_{x' \in \mathcal{X}} f(\boldsymbol{l}|x')}$, where $f(\boldsymbol{l}|x)$ the Probability Density Function (PDF) of the leakage for known value of $X = x$. In d-share encoding implementation, this PDF can be re-written as $f(\boldsymbol{l}|x) = \sum_{r \in \mathcal{X}^d} f(\boldsymbol{l}|x, r) \cdot p(r)$

The MI value between X and its leakage L directly links to the minimum number of measurements $N^\star$ that an adversary must obtained in order to recover specific value of X FiXme: citation

Computing MI directly is hard in practice as it quickly becomes intractable as the dimension of the variables increases and the distributions p, f are usually unknown. The perceived information (PI) is an alternative approach since it is an estimator of MI based on a *model* of the unknown PDF. Generally state, PI

quantifies the amount of information of X can be extracted from Y using an estimated model $\hat{p}$ of p. The PI is theoretically defined as:

$$PI(X;L) = \mathrm{H}(X) + \sum_{x \in \mathcal{X}} p(x) \int_{l \in \mathcal{L}} f(l|x) \cdot \log_2 \hat{p}(x|l)$$

and is practically computed as a sampling process as:

$$\widehat{PI}(X;L) = \mathrm{H}(X) + \sum_{x \in \mathcal{X}} p(x) \sum_{i=1}^{n_x} \frac{1}{n_x} \cdot \log_2 \hat{p}(x|l_{x,i}) \tag{5}$$

Distinguishers

We now detail several tool to estimate the model $\hat{p}(x|\boldsymbol{l})$

3 Simulated analysis

4 Real-world analysis

5 Conclusion

References

PR13. Emmanuel Prouff and Matthieu Rivain. Masking against side-channel attacks: A formal security proof. In Thomas Johansson and Phong Q. Nguyen, editors, *Advances in Cryptology – EUROCRYPT 2013*, pages 142–159, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.