Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 6, pp. 2333–2347
DOI : 10.47974/JDMSC-2598

Solving noisy systems of equations to enhance side-channel attacks on Dilithium

Konstanty Junosza-Szaniawski *
Piotr Lewandowski †
Faculty of Mathematics and Information Science
Warsaw University of Technology
Pl. Politechniki 1
00-661 Warsaw
Poland

Abstract

CRYSTALS-Dilithium, known also as FIPS-204 is one of the post-quantum signature schemes standardized by NIST. In this paper we investigate and enhance a side-channel attack on Dilithium proposed by Marzougui et al. [18], which leverages leaked data to recover parts of the private key. Recovery involves employing a MILP model to solve a noisy system of linear equations. We propose two new integer optimisation models, one linear and one nonlinear, which aimed at improving the efficiency of the attack. In particular, the nonlinear model improved success rates by up to 84 percentage points, significantly improving the practicality of attacks in scenarios with limited data and higher levels of noise. Our results emphasize the need for defensive measures in Dilithium implementations, protecting against side-channel attacks.

Subject Classification: *94A60, 90C05, 90C26.*

Keywords: *Dilithium, Side-channel attacks.*

1. Introduction

The National Institute of Standards and Technology (NIST), which is part of the U.S. Department of Commerce, released a standard for a Module-Lattice-Based Digital Signature on August 13, 2024 [1]. It is one of the results of a 7 years long effort to design and standardize new digital signature and encryption algorithms [2], that would remain secure even in the presence of a

Research was funded by Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme

* *ORCID-ID:* `0000-0003-0352-8583`

* *E-mail:* `konstanty.szaniawski@pw.edu.pl (Corresponding Author)`
† *E-mail:* `piotr.lewandowski11.stud@pw.edu.pl`

results of a 7 years long effort to design and standardize new digital signature and encryption algorithms [2], that would remain secure even in the presence of a general purpose quantum computer. We have known for almost 30 years that the currently used cryptographic methods as well as many designed nowadays [3, 4], based on the number factorisation or discrete logarithm , can theoretically be broken by employing Shor's algorithm [5]. There are also cryptosystems and digital signatures [6, 7] based on quadratic residue problem, but there is also quantum algorithms for that problem [8].

The need to replace those schemes has become more urgent due to rising interest in creating a quantum computer, with research in that field being funded by organisations such as Google, Microsoft and NASA [9, 10, 11].

With Shor's algorithm and a quantum computer at hand, we would have a way of efficiently solving both the factorization and discrete logarithm problems, which are the basis of the widely used RSA and ECDSA schemes. Proposed alternatives to these include lattice, isogeny, code, multivariable and hash based solutions [12, 13, 14, 15, 16]. Out of the three post quantum standards released by NIST two are lattice-based and one is hash-based. The aforementioned Module-Lattice-Based Digital Signature is derived from CRYSTALS-Dilithium (or simply Dilithium) [17], which is the main focus of this paper.

In particular, we will analyse a known side-channel attack on Dilithium, described in Marzougui et al. [18]. Side-channel attacks are one of the more potent and hard to prevent ways of compromising security of a system, as they depend on exploiting an unexpected vector of attack. Some known examples include measuring power consumption of a processor [19], response times of an application, execution time of some particular operations [20] or even the sounds produced by a keyboard or a monitor in some critical moment [21, 22]. In our case it is a power based attack, aimed at one of the helper functions in the reference implementation of Dilithium. It is not necesarilly an issue with the method itself, but still some precautions should be taken to defend against it, whenever one tries to implement either Dilithium or it's successor in FIPS 204.

Security of Dilithium relies on the hardness of the Learning With Errors problem (LWE) [12]. In short, we consider solving a system of equations of the form $As + e = b,$ for known A, b and small but unknown e to be computationally difficult. In particular, at one point in the signature generation algorithm we perform an assignment of the form $z := y + \langle\langle cs_1 \rangle\rangle,$ where z and c are publicly available as a part of the signature and s_1 is a part of the private key. In general, obtaining s_1 is computationally hard, but using a side-channel leak in a different part of the procedure allows an attacker to gain information about which elements of y are equal to zero, with some probability of a false positive [18]. This leads to a new system of equations of the form $z = Cs,$ with some being incorrect. Directly solving this system would therefore yield a solution $\hat{s}$ that is different from s_1 that we would like to obtain. In the original attack a simple MIP model is employed to find an integer solution that maximises the number of solved equations, which, if the majority of equations is correct, should be equal s_1. Ravi *et al.* [23] it is shown that knowing only s_1 is enough to allow an attacker to forge signatures.

In this work, we propose two alternative models for the integer optimisation problem employed in the attack from [18]. It is used to solve a noisy set of equations, by looking for values that maximise the number of fulfilled equations. The main idea behind the new models is expanding the objective function to value not only fulfilled equations, but also equations that are off by a small value. We were looking for a way to increase the value of partially correct solutions, while possibly limiting the impact of errors. The intuition for that approach is that getting most of the values right should result in the differences being small for the correct equations, and large for the erroneous ones. From our tests, the proposed model with an exponential objective function performs significantly better than both the original model, and our linear alternative, which are similar to each other in performance.

2. Background

Dilithium is a variation of the Fiat-Shamir with aborts approach to constructing signature schemes. Particular implementations are dependent on 11 parameters. Following in the footsteps of Marzougui et al. [18] we focus on values corresponding to NIST Security Level 2, which are presented in table 1:

Table 1
Parameters of Dilithium

NIST Security Level	2
Parameters	
n [polynomial degree]	256
q [modulus]	8380417
d [dropped bits from t]	13
τ [# of ±1's in c]	39
Challenge entropy [$\log(256^{\tau}) + \tau$]	192
γ_1 [y coefficient range]	2^{17}
γ_2 [low-order rounding range]	$(q-1)/88$
(k, ℓ) [dimensions of A]	(4, 4)
η [secret key range]	2
β [$\tau \cdot \eta$]	78
ω [max. # of 1's in the hint h]	80

To explain the general notion of the attack we present below a simplified description of the signature generation algorithm in Dilithium. It is important to note that y, z and s_1 are 4--element vectors of polynomials from $\mathbb{Z}_q/_{(x^{256}+1)}$ - the ring of polynomials with integer coefficients modulo q, reduced by the polynomial $x^{256} + 1$, while $c \in \mathbb{Z}_q/_{(x^{256}+1)}$ is just a single polynomial. Polynomials in y and z have potentially large coefficients, up to $\gamma_1 - 1 = 2^{17} - 1$, the set of all such polynomials is denoted $S_{\gamma-1}$. The possible coefficients of s_1 are $\{-2, -1, 0, 1, 2\}$ and in the case of c just $\{-1, 0, 1\}$. B_{τ} denotes a set of polynomials with coefficients $\{-1, 0, 1\}$ that contain exactly τ

non-zero elements. Function SampleInBall generates such a polynomial for a given seed. H is an extended output function (XOF), in particular SHAKE-256. HighBits and LowBits are supporting algorithms specified in [17], that extract either the high or low bits from an element of Z_q. The signature generation algorithm proceeds as follows. First $y \in S^{\ell}_{\gamma-1}$ is generated pseudo-randomly, with the seed depending on the secret key and message. Then using y and the message we generate the polynomial $c \in B_\tau$. With y and c we can calculate $z = y + cs_1$. If z fulfils the condition $\|z\|_\infty \geq \gamma_1 - \beta$ or if $\|\text{LowBits}(Ay - cs_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$ then z is discarded and the procedure is restarted. Otherwise (z, c) is returned as the signature.

Algorithm 1 Dilithium Signature Generation

1: **function** SIGN sk = (A, t, s_1, s_2), m
2: $z := \perp$
3: **while** $z = \perp$ **do**
4: $y \leftarrow S^{\ell}_{\gamma_1-1}$
5: $c \leftarrow B_\tau$
6: $z := y + cs_1$
7: **if** $\|z\|_\infty \geq \gamma_1 - \beta$ **or** $\|\text{LowBits}(Ay - cs_2, 2\gamma_2)\|_\infty \geq \gamma_2 - \beta$ **then**
8: $z := \perp$
9: **end if**
10: **end while**
11: **return** $\sigma = (z, c)$
12: **end function**

The leak enabling our attack happens in the process of generating y in line 4. Reference implementation of Dilithium uses a function that can be power profiled to reveal whether certain positions of y are equal to 0. In line 6 we can notice that the value of z depends on y, c and s_1. Values of z, c are publicly know, as they form the signature itself. Which means that, with the additional knowledge of y from the leaking function, the only unknown that remains is s_1, which is a part of the secret key. To retrieve s_1 we need to:

1. Sign some messages while monitoring power consumption.
2. On the generated power profile run a machine learning model that recognises whether some coefficients in y are 0, with some error probability.
3. For each found 0 at some position j we get a single linear equation of the form $z_j := c(s_1)_i$, in which the variables are the coefficients of the i--th polynomial in s_1. Incorrect classifications introduce noise into the system.
4. Solve the noisy system of equations using optimisation methods to retrieve s_1.

In this work we focus on improving step 4, by altering the model introduced in [18]. The attack described in [18] followed the framework described above. In

more technical terms, a ChipWhisperer device is used for power profiling the *polyz_unpack* function from the reference implementation of Dilithium. Messages to sign were chosen uniformly at random. For $y_{ij} = 0$ the corresponding coefficient in z is small with high probability - $|z_{ij}| \leq 2\sigma \approx 13.96$. To decrease the chance of a false-positive, the machine learning classifier was only run if z contained such a small coefficient. Each positive classification on y_{ij} resulted in an additional equation $z_j := c(s_1)_i$ being recorded. Afterwards, the generated set of equations was solved using a least squares method to provide a solution candidate: a non-integer vector, with values that might be close to coefficients of s_1. Two MILP models were then run, one base and the other extending it by considering the candidate solution as a hint. We aimed to improve the base model, as all of our modifications could be just as well performed on the model with hints.

3. Models

The noisy system of equations obtained from the leak is solved with the help of a mixed integer programming model. The base model, introduced in [18] aims to maximise the number of fulfilled equations. For an alternative form of the equations, obtained by moving every term to the left hand side to get $z_{ij} - \langle\langle c_{ij}\hat{s}\rangle\rangle = 0$, we consider the value of the left hand side. Every 0 on the left side contributes 1 to the objective, while every other value does not contribute at all. The idea behind the new models is to also reward small, but non-zero left side values, scaled with the distance from 0. Let r be the value of the left hand side, $r = |z_{ij} - \langle\langle c_{ij}\hat{s}\rangle\rangle|$. Then, for the linear model the reward is $\max(0, 1 - \frac{1}{5}r)$, while in the nonlinear case, the reward is $e^{-\frac{1}{2}r}$. Plot in Figure 1 displays the three functions side by side.

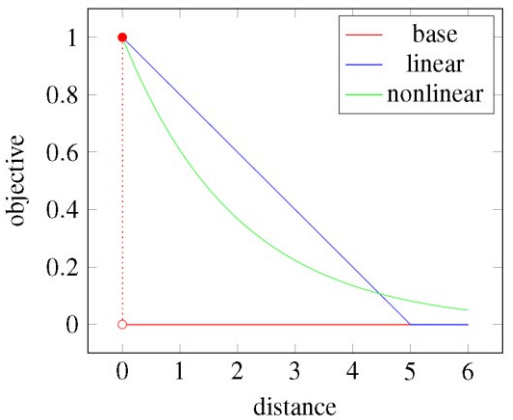

Figure 1

Plot of objective function contribution for an equation that diverges from equality by a certain amount

We limited our experiments to just the first vector in s_1, as every vector can, and even should be retrieved independently, to limit the dimension of the optimization problem. From this point, whenever s_1 is referenced it signifies just the first vector. All of the models are written with just that single vector of 256 variables in mind. Each model takes as input a set of equations indexed by two numbers (i,j), where i is the position in y that was classified as 0, and j is the number of the message that was signed to generate the equation. We will denote the set of indices by M. For every $(i,j) \in M$ we have an equation of the form $z_{ij} - \langle\langle c_{ij}\hat{s}\rangle\rangle = 0$, where $\hat{s}$ is the unknown vector. The parameter $K = 700$ is a rough estimate of a small integer that is still larger then the left hand side values.

3.1 *Base model*

This model uses two sets of variables: integer $\{\hat{s}_i \in Z: i = 0,1,2, \dots 255\}$ and binary $\{x_{ij} \in \{0,1\}: (i,j) \in M\}$.

$$\max \sum_{(i,j)\in M} x_{ij}$$

$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \le (1 - x_{ij})K \qquad (i,j) \in M$$

$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \ge -(1 - x_{ij})K \quad (i,j) \in M$$

$$-2 \le \hat{s}_i \le 2 \qquad i = 0,1,2, \dots 255$$

For a given $(i,j) \in M$ an assignment $x_{ij} = 1$ indicates that

$$z_{ij} = \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k}$$

so the equation $z_{ij} = \langle\langle c_{ij}\hat{s}\rangle\rangle$ holds. Our objective is to maximise the number of $(i,j) \in M$ for which $x_{ij} = 1$, so the number of fulfilled equations.

3.2 *Linear relaxation*

This model builds on the base model by modifying existing constraints and adding new ones for the purpose of altering the objective function. The goal was to assign a positive value to equations that are slightly off. Position of the cut-off point, which corresponds also to the slope of the objective function, was determined experimentally and set to 5. Hence in this model the input to the objective function for each $(i,j) \in M$ is $\max(0, 1 - \frac{1}{5}|z_{ij} - \langle\langle c_{ij}\hat{s}\rangle\rangle|)$. To achieve that, we use four sets of variables:

- $\{\hat{s}_i \in Z: i = 0,1,2, \dots 255\}$ - represent the value of the current solution,
- $\{0 \le r_{ij} \le 91: (i,j) \in M\}$ - represent the distance from 0 in the equation indexed (i,j). We can estimate the possible distance for the incorrect equations using the bounds introduced in [18]. We know that $|\langle\langle c_{ij}\hat{s}\rangle\rangle| \le \beta = 78$ and that $|z_{ij}| \le 2\sigma \approx 13.96$ with high probability, where β, σ, η are parameters of Dilithium (as specified in table 1) and

$$\sigma^2 = \frac{(2\eta)^2 - 1}{12\tau}$$

Therefore $\left|z_{ij} - \langle\langle c_{ij}\hat{s}\rangle\rangle\right| \le 91$.

- $\{x_{ij} \in \{0,1\}: (i,j) \in M\}$ - equal to 1 when the distance is smaller than 5 and equal to 0 when it is larger,
- $\{v_{ij} \in [0,1]: (i,j) \in M\}$ - value of the objective function contribution for the equation, as described above.

$$\max \sum_{(i,j)\in M} v_{ij}$$
$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \le r_{ij} \qquad (i,j) \in M$$
$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \ge -r_{ij} \qquad (i,j) \in M$$
$$r_{ij} - 5 \le K(1 - x_{ij}) \qquad (i,j) \in M$$
$$5 - r_{ij} \le K x_{ij} \qquad (i,j) \in M$$
$$v_{ij} \le 1 - \frac{r_{ij}}{5} + K(1 - x_{ij}) \qquad (i,j) \in M$$
$$v_{ij} \le x_{ij} \qquad (i,j) \in M$$
$$-2 \le \hat{s}_i \le 2 \qquad i = 0,1,2,\dots 255$$

Substituting the right hand side in the first two constraints of the base model gives us

$$r_{ij} = \left| z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \right|$$

We then relate the other variables to r_{ij} as follows. For a given $(i,j) \in M$ an assignment $x_{ij} = 1$ indicates that $r_{ij} - 5 \le 0 r_{ij} \le 5$, conversely an assignment $x_{ij} = 0$ indicates that $5 - r_{ij} \le 05 \le r_{ij}$. So $x_{ij} = 1$ if and only if the distance r_{ij} is not larger than 5. Furthermore for $x_{ij} = 1$ we get $v_{ij} \le 1 - \frac{r_{ij}}{5}$ and $x_{ij} = 0$ we get $v_{ij} \le 0$ which means that we get the desired objective function contribution:

$$v_{ij} = \begin{cases} 1 - \frac{r_{ij}}{5} & \text{if } \ r_{ij} < 5 \\ 0 & \text{otherwise} \end{cases}$$

3.3 *Nonlinear relaxation*

This model achieves a similar objective function to the linear model (see figure 1) with three times fewer auxiliary variables, by using a nonlinear objective function. The only variables used are:

- $\{\hat{s}_i \in Z: i = 0,1,2,\dots 255\}$ - represent the value of the current solution,
- $\{0 \le r_{ij} \le 91: (i,j) \in M\}$ - represent the distance from 0 in the equation indexed (i,j), with the constraints derived the same way as in the linear model above.

$$\max \sum_{(i,j)\in M} e^{-\frac{1}{2} r_{ij}}$$
$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \le r_{ij} \qquad (i,j) \in M$$
$$z_{ij} - \sum_{k=0}^{j} (c_{ij})_k \hat{s}_{j-k} + \sum_{k=j+1}^{255} (c_{ij})_k \hat{s}_{256+j-k} \ge -r_{ij} \qquad (i,j) \in M$$
$$-2 \le \hat{s}_i \le 2 \qquad i = 0,1,2,\dots 255$$

This model changes existing constraints from the base model in the same way as the linear model, without introducing new ones. The $\frac{1}{2}$ coefficient in the objective function was chosen experimentally.

4. Experimental Setup and Results

4.1 *Experimental Setup*

We simulated the leak by collecting the relevant data from the signing process for a known private key. We collected 900 equations and about 40000 potential false positives from 300000 signatures. Then, for certain values of parameters e, o we have randomly chosen $o + 256$ equations from the first set and e of the equations from the second set. We wanted to minimize the number of equations that would still allow us to retrieve s_1 with a certain number of errors. Using more than 256 correct equations for 256 variables in s_1, means that they form an overdetermined system, with a solution equal to s_1. For each pair of $e \in \{20,30,40,50,60\}$ and $o \in \{10,20,30\}$ we generated 50 data sets.

Each set was run with each model on an Intel i5-1340P processor, using up to 16 threads. Execution was stopped after 5 minutes or after reaching a 6% primal/dual gap, at which point, the returned solution was recorded. We marked the run as a success if the obtained solution was equal to s_1, and as a failure otherwise.

4.2 *Experimental Results*

The main conclusion states that the exponential model emerged as a clear winner, with noticeably better performance, especially in the data sets with larger numbers of errors. For the harder inputs it outperforms the other models by 50, 52 or even 84 percentage points, which significantly improves the potential for a successful attack. This advantage is especially crucial in scenarios with limited amounts of leaked information, further enhancing the practical applicability of the nonlinear model.

Table 2

Number of timeouts out of 50 runs

data set		model		
(lr)1-5 errors	**correct**	**base**	**linear**	**nonlinear**
20	256	32	27	15
20	261	23	22	4
20	266	16	15	4

A second conclusion is that the base and linear models exhibit similar performance; although, the linear model has an objective function closer to that of the nonlinear model. On the downside, it involves the highest number of variables and constraints out of our three models, which appears to outweigh the advantages granted by the improved objective function. However there are still cases in which the linear model performs better than the base one. Table 2 shows the results for cases in which the linear model reached the time limit in fewer runs

than the base model. This gives hope that increasing the timeout may allow the linear model to achieve better results than the base model.

[H] Exact results of the experiments are presented in figures 2, 3 and 4.

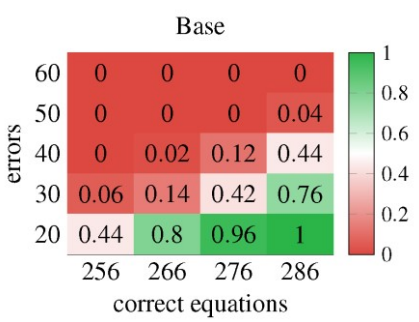

Figure 2

Performance of the original model, x axis signifies the number of correctly classified equations, the y axis indicates the number of errors

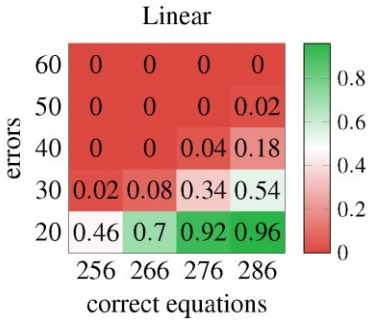

Figure 3

Performance of the new model, x axis signifies the number of correctly classified equations, the y axis indicates the number of errors

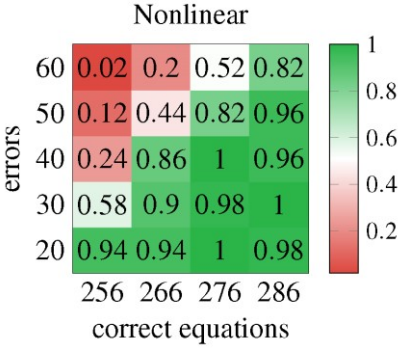

Figure 4

Performance of the new model, x axis signifies the number of correctly classified equations, the y axis indicates the number of errors

5. Future work

We believe that there are still some aspects of our work that could benefit from further investigation. Future research could focus on tweaking parameters in our test setup, changing the used model and solver, or using different data sets, including real life data.

The most natural follow-up to this work would be a more in-depth analysis of model performance in relation to test parameters. Adjusting the primal-dual gap might improve execution time, by allowing the solver to finish earlier. In particular it might be interesting to experiment with gaps correlated to the number of errors in data, as that is possible to approximate based on the accuracy of

whatever classifier we are using. Setting a different timeout might also provide more accurate data about the differences in speed between the models. Our chosen timeout of 5 minutes turned out to be adequate to compare their performance, but there is a possibility that a longer, but still acceptable in practice timeout, for example 10 or 15 minutes could increase the number of successful runs and further differentiate between the linear and base models.

Another potential area of investigation covers the use of other solvers. For our experiments we have used Gurobi while the original work of Marzogui et al. [18] used SCIP. It might be interesting to see whether the differences between model performance are preserved when using a different solver.

There is also a direction that we have not explored at all - attempting to retrieve the key from a smaller number of equations. Using an underdetermined system would mean foregoing the guarantee that the key is the optimal solution, but it might still be recovered as one of the possible solutions. With some additional method of checking alternative solutions, it might provide a way to facilitate attacks requiring an even smaller number of equations.

Performing the tests on "real life" data, which was generated by actually performing the power analysis of a processor would verify our results in a practical setting. Unfortunately that requires access to specialised hardware, for example a ChipWhisperer device, which was used in the original attack. Access to this kind of data would also provide opportunities to devise more specialised test cases, as well as attempt extending them to a full end-to-end attack.

Appendix

A. Code Listings of AMPL models

A.1 *Base*

```
# global parameters
param POLY_DEGREE = 255;
set INDICES = 0 .. POLY_DEGREE;

# set of message numbers
set L dimen 2;

# system of equations we are trying to solve
# z = c * s
param zL integer;
param CL, INDICES integer;
var sINDICES integer;

# auxiliary variable to model equality as two inequalities
param K = 700;

# whether a given equation is satisfied
var xL binary;

# we are looking for the solution that satisfies the most equations
maximize number_of_equations: sum (m,j) in L x[m,j];
```

```
subject to equation_right (m,j) in L:
    z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
        + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
            <= (1 - x[m,j]) * K;

subject to equation_left (m,j) in L:
z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
    + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
    >= - (1 - x[m,j]) * K;

subject to small_s i in INDICES: -2 <= s[i] <= 2;
```

A.2 *Linear*

```
# global parameters
param POLY_DEGREE = 255;
set INDICES = 0 .. POLY_DEGREE;

# set of message numbers
set L dimen 2;

# system of equations we are trying to solve
# z = c * s
param zL integer;
param CL, INDICES integer;
var sINDICES integer;

# auxiliary variable to model equality as two inequalities
param K = 700;

# relaxation parameters
param max_relaxation = 91;
param rel_coef = 1 / card(L);
var rL >= 0, <= max_relaxation;

# cut-off reward function
param cutoff = 5;
param reward_coef = 1;
var reward_valueL >= 0, <= reward_coef;
var reward_activeL binary;

# reward small errors as well as satisfied equations
maximize distnace_rewards: sum (m,j) in L (reward_value[m,j]);

subject to equation_right (m,j) in L:
    z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
        + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
        <= r[m,j];

subject to equation_left (m,j) in L:
z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
    + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
    >= - r[m,j];
```

```
subject to no_reward_for_big_errors (m,j) in L:
    r[m,j] - cutoff <= K * (1 - reward_active[m,j]);

subject to reward_smaller_errors (m,j) in L:
    cutoff - r[m,j] <= K * reward_active[m,j];

subject to reward_value_bound (m,j) in L:
    reward_value[m,j] <= reward_coef - (reward_coef / cutoff) * r[m,j] + K * (1 -
reward_active[m,j]);

subject to reward_only_when_active (m,j) in L:
    reward_value[m,j] <= reward_active[m,j];

subject to s_size i in INDICES : -2 <= s[i] <= 2;
```

A.3 Nonlinear

```
# global parameters
param POLY_DEGREE = 255;
set INDICES = 0 .. POLY_DEGREE;

# set of message numbers
set L dimen 2;

# system of equations we are trying to solve
# z = c * s
param zL integer;
param CL, INDICES integer;
var sINDICES integer;

# auxiliary variable to model equality as two inequalities
param K = 700;

# relaxation parameters
param max_relaxation = 91;
param rel_coef = 1 / card(L);
var rL >= 0, <= max_relaxation;
param exp_coef = 0.5;

# reward small errors as well as satisfied equations
maximize distnace_rewards: sum (m,j) in L (exp(-r[m,j] * exp_coef));

subject to equation_right (m,j) in L:
    z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
        + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
            <= r[m,j];

subject to equation_left (m,j) in L:
z[m,j] - sum i in INDICES: i <= j C[m,j,i]*s[j - i]
    + sum i in INDICES: i > j C[m,j,i]*s[POLY_DEGREE + 1 + j - i]
        >= - r[m,j];

subject to s_size i in INDICES : -2 <= s[i] <= 2;
```

Acknowledgement

Research was funded by Warsaw University of Technology within the Excellence Initiative: Research University (IDUB) programme

References

[1] N. I. of Standards and Technology, "Module-lattice-based digital signature standard. (department of commerce, washington, d.c.), federal information processing standards publication (fips) nist fips 204." (2024). [Online]. Available: https://doi.org/10.6028/NIST.FIPS.20

[2] N. I. of Standards and Technology, Kevin Kimball, "Announcing request for nominations for publickey post-quantum cryptographic algorithms, docket no. 161116999-6999-02," (2017). [Online]. Available: https://www.federalregister.gov/documents/2016/12/20/2016-30615/announcing-request-for-nominations-for-public-keypost-quantum-cryptographic-algorithms

[3] N. Tahat, A. K. Alomari, O. M. Al-Hazaimeh, and M. F. Al-Jamal, "An efficient self-certified multi-proxy signature scheme based on elliptic curve discrete logarithm problem," *Journal of Discrete Mathematical Sciences and Cryptography,* vol. 23, no. 4, pp. 935–948 (2020). [Online]. Available: https://doi.org/10.1080/09720529.2020.1734293

[4] A. Pandey and I. Gupta, "A new undeniable signature scheme on general linear group over group ring," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 25, no. 5, pp. 1261–1273 (2022). [Online]. Available: https://doi.org/10.1080/09720529.2020.1744814

[5] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Journal on Computing*, vol. 26, no. 5, pp. 1484–1509 (1997). [Online]. Available: https://doi.org/10.1137/S0097539795293172

[6] L. Harn and T. Kiesler, "Improved rabin's scheme with high efficiency," *Electronics Letters*, vol. 25, pp. 726–728 (1989).

[7] M. Alaroud, N. Tahat, A. K. Alomari, and O. M. Al-hazaimeh, "A novel chaotic map partially blind signature scheme based on quadratic residue problems," *Journal of Discrete Mathematical Sciences and Cryptography*, vol. 27, no. 3, pp. 1059–1071 (2024). [Online]. Available: https://doi.org/10.47974/JDMSC-1823.

[8] H.-F. Wang, S. Zhang, Y.-F. Zhao, and K.-H. Yeon, "Quantum mechanical algorithm for solving quadratic residue equation," *International Journal of Theoretical Physics*, vol. 48, no. 11, pp. 3262–3267 (Nov.

2009). [Online]. Available: https://doi.org/10.1007/s10773-009-0115-x

[9] P. Klimov, A. Bengtsson, C. Quintana, A. Bourassa, S. Hong, A. Dunsworth, K. Satzinger, W. Livingston, V. Sivak, M. Niu, T. Andersen, Y. Zhang, D. Chik, J. Chen, C. Neill, C. Erickson, A. G. Dau, A. Megrant, P. Roushan, A. Korotkov, J. Kelly, V. Smelyanskiy, Y. Chen, and H. Neven, "Optimizing quantum gates towards the scale of logical qubits," *Nature Communications*, vol. 15, p. 2442 (2024). [Online]. Available: https://www.nature.com/articles/s41467-024-46623-y

[10] M. A. Quantum, "Interferometric single-shot parity measurement in an inas-al hybrid device," arXiv (April 2024). [Online]. Available: https://www.microsoft.com/enus/research/publication/interferometric-single-shot-paritymeasurement-in-an-inas-al-hybrid-device/

[11] E. G. Rieffel, A. A. Asanjan, M. S. Alam, N. Anand, D. E. Bernal Neira, S. Block, L. T. Brady, S. Cotton, Z. Gonzalez Izquierdo, S. Grabbe, E. Gustafson, S. Hadfield, P. A. Lott, F. B. Maciejewski, S. Mandr`a, J. Marshall, G. Mossi, H. M. Bauza, J. Saied, N. Suri, D. Venturelli, Z. Wang, and R. Biswas, "Assessing and advancing the potential of quantum computing: A nasa case study," *Future Generation Computer Systems*, vol. 160, pp. 598–618 (2024). [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X24003121

[12] O. Regev, "On lattices, learning with errors, random linear codes, and cryptography," in Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing, ser. STOC '05. New York, NY, USA: Association for Computing Machinery, p. 84–93 (2005). [Online]. Available: https://doi.org/10.1145/1060590.1060603

[13] D. Jao and L. De Feo, "Towards quantum-resistant cryptosystems from supersingular elliptic curve isogenies," in Post-Quantum Cryptography, B.-Y. Yang, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 19–34 (2011).

[14] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in Applied Cryptography and Network Security, J. Ioannidis, A. Keromytis, and M. Yung, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 164–175 (2005).

[15] R. J. McEliece, "A public key cryptosystem based on algebraic coding theory," in DSN Progress Report (1978). [Online]. Available: https://api.semanticscholar.org/CorpusID:56502909

[16] D. J. Bernstein, D. Hopwood, A. H¨ulsing, T. Lange, R. Niederhagen, L. Papachristodoulou, M. Schneider, P. Schwabe, and Z. Wilcox-

O'Hearn, "Sphincs: Practical stateless hash-based signatures," in Advances in Cryptology – EUROCRYPT 2015, E. Oswald and M. Fischlin, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 368–397 (2015).

[17] L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, P. Schwabe, G. Seiler, and D. Stehl´e, "Crystals-Dilithium – algorithm specifications and supporting documentation (version 3.1)," (2021). [Online]. Available: https://pq-crystals.org/dilithium/

[18] S. Marzougui, V. Ulitzsch, M. Tibouchi, and J. Seifert, "Profiling side-channel attacks on dilithium: A small bitfiddling leak breaks it all," IACR Cryptol. ePrint Arch., pp. 106 (2022). [Online]. Available: https://eprint.iacr.org/2022/106

[19] L. Lerman, G. Bontempi, and O. Markowitch, "Power analysis attack: an approach based on machine learning," *International Journal of Applied Cryptography*, vol. 3, no. 2, pp. 97–115 (2014), pMID: 62722. [Online]. Available: https://www.inderscienceonline.com/doi/abs/10.1504/IJACT.2014.062722

[20] B. B. Brumley and N. Tuveri, "Remote timing attacks are still practical," in Computer Security – ESORICS 2011, V. Atluri and C. Diaz, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 355–371 (2011).

[21] M. Guri, "Pixhell attack: Leaking sensitive information from air-gap computers via 'singing pixels'," in 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC), pp. 976–987 (2024).

[22] L. Simon and R. Anderson, "Pin skimmer: inferring pins through the camera and microphone," in Proceedings of the Third ACM Workshop on Security and Privacy in Smartphones & Mobile Devices, ser. SPSM '13. New York, NY, USA: Association for Computing Machinery, pp. 67–78 (2013). [Online]. Available: https://doi.org/10.1145/2516760.2516770.

[23] P. Ravi, M. P. Jhanwar, J. Howe, A. Chattopadhyay, and S. Bhasin, "Side-channel assisted existential forgery attack on dilithium - a NIST PQC candidate," Cryptology ePrint Archive, Paper 2018/821, (2018). [Online]. Available: https://eprint.iacr.org/2018/821

Received April, 2025