Remasking Dilithium

Attacking a Dilithium masking scheme and fixing it

Lukas Prokop

2021-09-09

`https://lukas-prokop.at/talks/seminar-remasking-dilithium`

IAIK – Graz University of Technology

A post-quantum scheme

History and prior work on physical security

2009 V. Lyubashevsky introduces "Fiat-Shamir with Aborts" framework

2012 V. Lyubashevsky: "Lattice Signatures without Trapdoors"

2017 CRYSTALS-Dilithium submission in NIST Standardization Process as lattice-based signature scheme

2020 P. Ravi, R. Poussier, S. Bhasin, A. Chattopadhyay. "On Configurable SCA Countermeasures Against Single Trace Attacks for the NTT - A Performance Evaluation Study over Kyber and Dilithium on the ARM Cortex-M4"

2020 A. P. Fournaris, C. Dimopoulos, O. G. Koufopavlou. "Profiling Dilithium Digital Signature Traces for Correlation Differential Side Channel Attacks"

History

"To the best of our knowledge, Dilithium has the smallest public key + signature size of any lattice-based signature scheme that only uses uniform sampling."

round 1 → 2 deterministic scheme becomes deterministic or randomized

round 2 → 3 parameter set adjustments for NIST, sampling among $2^{\text{something}}$ values

3 variants Dilithium-2, Dilithium-3, Dilithium-5 (corresponding to NIST security categories)

Fiat-Shamir with Aborts

Secret key: $s \xleftarrow{\$} D_s$
Public key: N, g, and $S \leftarrow g^s \bmod N$

Prover		Verifier
$y \xleftarrow{\$} D_y$, $Y \leftarrow g^y \bmod N$	$\xrightarrow{\quad Y \quad}$	
	$\xleftarrow{\quad \mathbf{c} \quad}$	$c \xleftarrow{\$} D_c$
$z \leftarrow sc + y$		
[if $z \notin G$ then $z \leftarrow \perp$]	$\xrightarrow{\quad z \quad}$	
		Accept iff $g^z \equiv YS^c (\bmod N)$

Fig. 1. Factoring-Based Identification Schemes. The parameters for this scheme are in Figure 6. The line in [ ] is only performed in the aborting version of the scheme.

[L09] Vadim Lyubashevsky (2009).
"Fiat-Shamir with Aborts: Applications to Lattice and Factoring-Based Signatures"

Fiat-Shamir with Aborts

Signing Key: $\widehat{\mathbf{s}} \xleftarrow{\$} D_s^m$
Verification Key: $h \xleftarrow{\$} \mathcal{H}(R, D, m), \mathbf{S} \leftarrow h(\widehat{\mathbf{s}})$
Random Oracle: $\mathrm{H} : \{0,1\}^* \rightarrow D_c$

Sign$(\mu, h, \widehat{\mathbf{s}})$
1: $\widehat{\mathbf{y}} \xleftarrow{\$} D_y^m$
2: $\mathbf{e} \leftarrow \mathrm{H}(h(\widehat{\mathbf{y}}), \mu)$
3: $\widehat{\mathbf{z}} \leftarrow \widehat{\mathbf{s}}\mathbf{e} + \widehat{\mathbf{y}}$
4: if $\widehat{\mathbf{z}} \notin \mathrm{G}^m$, then goto step 1
5: output $(\widehat{\mathbf{z}}, \mathbf{e})$

Verify$(\mu, \widehat{\mathbf{z}}, \mathbf{e}, h, \mathbf{S})$
1: Accept iff
$\widehat{\mathbf{z}} \in \mathrm{G}^m$ and $\mathbf{e} = \mathrm{H}(h(\widehat{\mathbf{z}}) - \mathbf{S}\mathbf{e}, \mu)$

Fig. 4. Lattice-Based Signature Scheme

Parameters for Dilithium-3

q	modulus	$2^{23} - 2^{13} + 1$
d	dropped bits from t	13
τ	number of ± 1s in c	49
γ_1	y coefficient range	2^{19}
γ_2	low-order rounding range	$\frac{q-1}{32}$
(k, l)	dimensions of A	$(6, 5)$
η	secret key range	4
β	$\tau \cdot \eta$	196
ω	max. number of 1s in hint h	55
	avg. repetitions	5.1

<u>*Gen*</u>

$\zeta \leftarrow \{0,1\}^{256}$

$(\rho, \varsigma, K) \in \{0,1\}^{256\times 3} := H(\zeta)$

$(s_1, s_2) \in S_\eta^l \times S_\eta^k := H(\varsigma)$

$\mathrm{A} \in R_q^{k\times l} := \mathrm{ExpandA}(\rho)$

$\mathrm{t} := \mathrm{As}_1 + \mathrm{s}_2$

$(\mathrm{t}_1, \mathrm{t}_0) := \mathrm{Power2Round}_q(\mathrm{t}, d)$

$tr \in \{0,1\}^{384} := \mathrm{CRH}(\rho \,\|\, \mathrm{t}_1)$

return $(pk = (\rho, \mathrm{t}_1), sk = (\rho, K, tr, \mathrm{s}_1, \mathrm{s}_2, \mathrm{t}_0))$

Core routines

<u>*Sign(sk, M)*</u>

$A \in R_q^{k \times l} := \text{ExpandA}(\rho)$
$\mu \in \{0,1\}^{384} := \text{CRH}(tr \,\|\, M)$
$\kappa := 0$
$(z, h) := \bot$
$\rho' \in \{0,1\}^{384} := \text{CRH}(K \,\|\, \mu)$
while $(z, h) = \bot$ **do**
  ...
return $\sigma = (z, h, \tilde{c})$

$y \in \tilde{S}_{\gamma_1}^{l} := \text{ExpandMask}(\rho', \kappa)$
$w := Ay$
$w_1 := \text{HighBits}_q(w, 2\gamma_2)$
$\tilde{c} \in \{0,1\}^{256} := H(\mu \,\|\, w_1)$
$c \in B_\tau := \text{SampleInBall}(\tilde{c})$
$z := y + cs_1$
$r_0 := \text{LowBits}_q(w - cs_2, 2\gamma_2)$
if $\| z \|_\infty \geq \gamma_1 - \beta$ or $\| r_0 \|_\infty \geq \gamma_2 - \beta$ **then**
  $(z, h) := \bot$
else
  $h := \text{MakeHint}_q(-ct_0, w - cs_2 + ct_0, 2\gamma_2)$
  if $\| ct_0 \|_\infty \geq \gamma_2$ or the number of 1s in $h \geq \omega$
then
    $(z, h) := \bot$
$\kappa := \kappa + l$

<u>*Verify*$(pk, M, \sigma = (\mathrm{z}, \mathrm{h}, \tilde{c}))$</u>

$\mathrm{A} \in R_q^{k \times l} := \mathrm{ExpandA}(\rho)$

$\mu \in \{0,1\}^{384} := \mathrm{CRH}(\mathrm{CRH}(\rho \parallel \mathrm{t}_1) \parallel M)$

$c := \mathrm{SampleInBall}(\tilde{c})$

$\mathrm{w}_1 := \mathrm{UseHint}_q(\mathrm{h}, \mathrm{Az} - c\mathrm{t}_1 \cdot 2^d, 2\gamma_2)$

return $[\![\| z \|_\infty < \gamma_1 - \beta]\!]$ and $[\![\tilde{c} = H(\mu \parallel \mathrm{w}_1')]\!]$ and $[\![$number of 1s in h is $\leq \omega]\!]$

public values $\rho, \mathrm{t}_1, A, z, h, \tilde{c}$

secret values $K, tr, \mathrm{s}_1, \mathrm{s}_2, \mathrm{t}_0, Y, W$

If you know one of the values $\{s_1, s_2\}$, then you can determine the other one [BP18][1]

[1] L. G. Bruinderink, P. Pessl (2018).
"Differential Fault Attacks on Deterministic Lattice Signatures"

Attacking Masked Dilithium

Preliminaries

Preliminary: poly. mult. as matrix mult.

Polynomial multiplication in $\mathbb{Z}[X]$.

$$(2x^2 + 1) \cdot (3x^2 + 4x + 5) = 6x^4 + 8x^3 + (10 + 3)x^2 + 4x + 5$$

$$[0\ 0\ 2\ 0\ 1] \cdot [0\ 0\ 3\ 4\ 5] = [6\ 8\ 13\ 4\ 5]$$

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 2 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 1 & 0 \\ 0 & 0 & 2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 4 \\ 3 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 5 \\ 4 \\ 10 + 3 \\ 8 \\ 6 \end{pmatrix}$$

Polynomial multiplication in $\mathbb{Z}_{47}[X]/(X^3+1)$.

$$
\begin{aligned}
(2x^2+1)\cdot(3x^2+4x+5) &\equiv 6x^4+8x^3+13x^2+4x+5 \\
&\equiv (6x+8)x^3+13x^2+4x+5 \qquad \text{with } x^3=-1 \\
&\equiv 13x^2-2x-3 \\
[2\ 0\ 1]\cdot[3\ 4\ 5] &= [13\ \ -2\ \ -3]
\end{aligned}
$$

$$
\begin{pmatrix} 1 & -2 & 0 \\ 0 & 1 & -2 \\ 2 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 5 \\ 4 \\ 3 \end{pmatrix} = \begin{pmatrix} -3 \\ -2 \\ 13 \end{pmatrix}
$$

Prior work

[MGTF19] V. Migliore, B. Gérard, M. Tibouchi, P.A. Fouque
"Masking Dilithium - Efficient Implementation and Side-Channel Evaluation" (2019)[2]

- Evaluation of unprotected implementation on ARM Cortex-M3
- Propose masking scheme for power-of-two moduli
- T-test evaluation shows no leakage[3]

Assuming the t-probing model, this scheme is not secure.

---

[2]based on [BBEFGRT18] G. Barthe, S. Belaïd, T. Espitau, P.A. Fouque, B. Grégoire, M. Rossi, M. Tibouchi (2018). "Masking the GLP Lattice-Based Signature Scheme at Any Order."

[3]Implementation not available

The attack

The issue

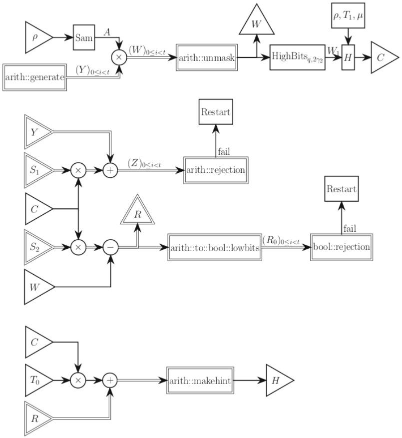


Fig. 4. Masked implementation of DILITHIUM.Sign. Masked functions are represented with a double-lined box.

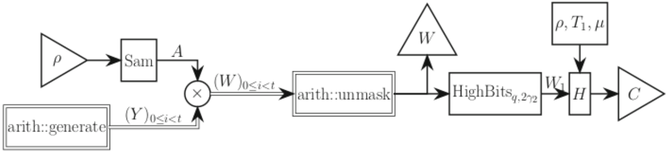


Fig. 4. Masked implementation of DILITHIUM.Sign. Masked functions are represented with a double-lined box.

- The value W is not masked.
- W contains k values of 23 bits. A probe could read 23 bits of the value W.
- Getting 23 accurate bits would allow an attack, I am about to describe.

- I use Dilithium-3 to describe the attack because $k \neq l$ $(k = 6, l = 5)$
- [MGTF19] assumes the t-probing model[4]

[4] [ISW03] Y. Ishai, A. Sahai, D. Wagner. "Private Circuits: Securing Hardware against Probing Attacks" (2003)

Attack

$$\mathrm{w} := \mathrm{Ay}$$

$$\mathrm{z} := \mathrm{y} + c\mathrm{s}_1 \qquad \text{from the spec.}$$

$$Z = Y + C \cdot S_1 \qquad \text{switching notation}$$

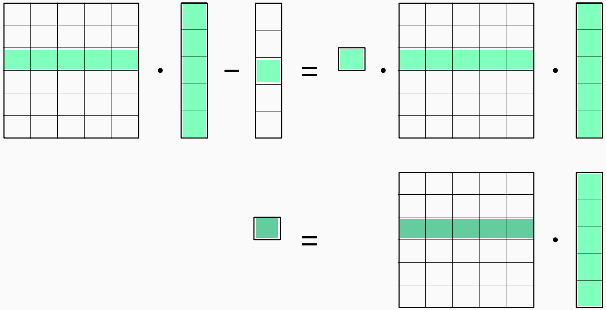
$$A \cdot Z = A \cdot Y + A \cdot C \cdot S_1$$

$$A \cdot Z = W + A \cdot C \cdot S_1$$

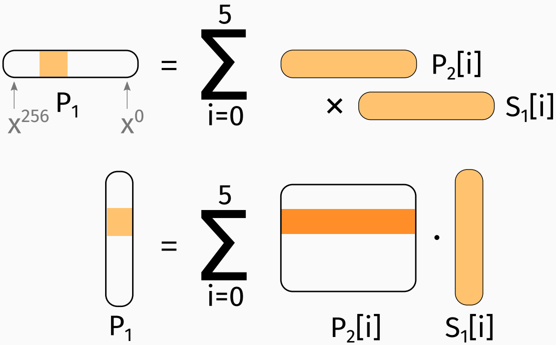
$$A \cdot Z - W = A \cdot C \cdot S_1$$

$$\underbrace{\underbrace{A}_{k \times l} \cdot \underbrace{Z}_{l \times 1}}_{\text{known}} - \underbrace{\underbrace{W}_{k \times 1}}_{\text{known/observed}} = \underbrace{C}_{1 \times 1} \cdot \underbrace{A}_{k \times l} \cdot \underbrace{\underbrace{S_1}_{l \times 1}}_{\text{unknown}}$$

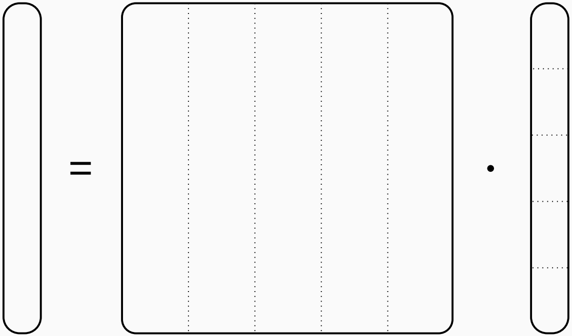
$$A \cdot Z - W = C \cdot A \cdot S_1$$



$$P_1 = P_2 \cdot S_1$$



Take $256 \cdot 5$ such measurements



(256×1) $(256 \cdot 5) \times (256 \cdot 5)$ $(256 \cdot 5) \times 1$

Attack summary

1. The attack shows that the masking scheme of [MGTF19] is insecure
2. We observe 23 bits of variable W (or $256 \cdot 23$ bits)
3. We build a linear equation system with S_1 as unknown variable
4. We reconstruct the secret key (S_1, S_2) from S_1 [BP18]

A masking scheme

(non)-linear operations

KeyGen
- uniform sampling with SHAKE-256
- ExpandA
- NTT/NTT^{-1}
- matrix multiplication
- vector addition
- Power2Round
- CRH

Sign
- ExpandMask
- HighBits/LowBits/Decompose
- SampleInBall
- MakeHint
- counting number of 1s ($>$)

Verify
- UseHint and counting number of 1s ($\leq$)

Goal

- Publish attack for ref/optimized implementation
- Show leakage on a microcontroller
- Propose masking scheme for original Dilithium
- Evaluate leakage on a microcontroller empirically

Thank you!

Remasking Dilithium

Attacking a Dilithium masking scheme and fixing it

Lukas Prokop

2021-09-09

https://lukas-prokop.at/talks/seminar-remasking-dilithium

IAIK – Graz University of Technology