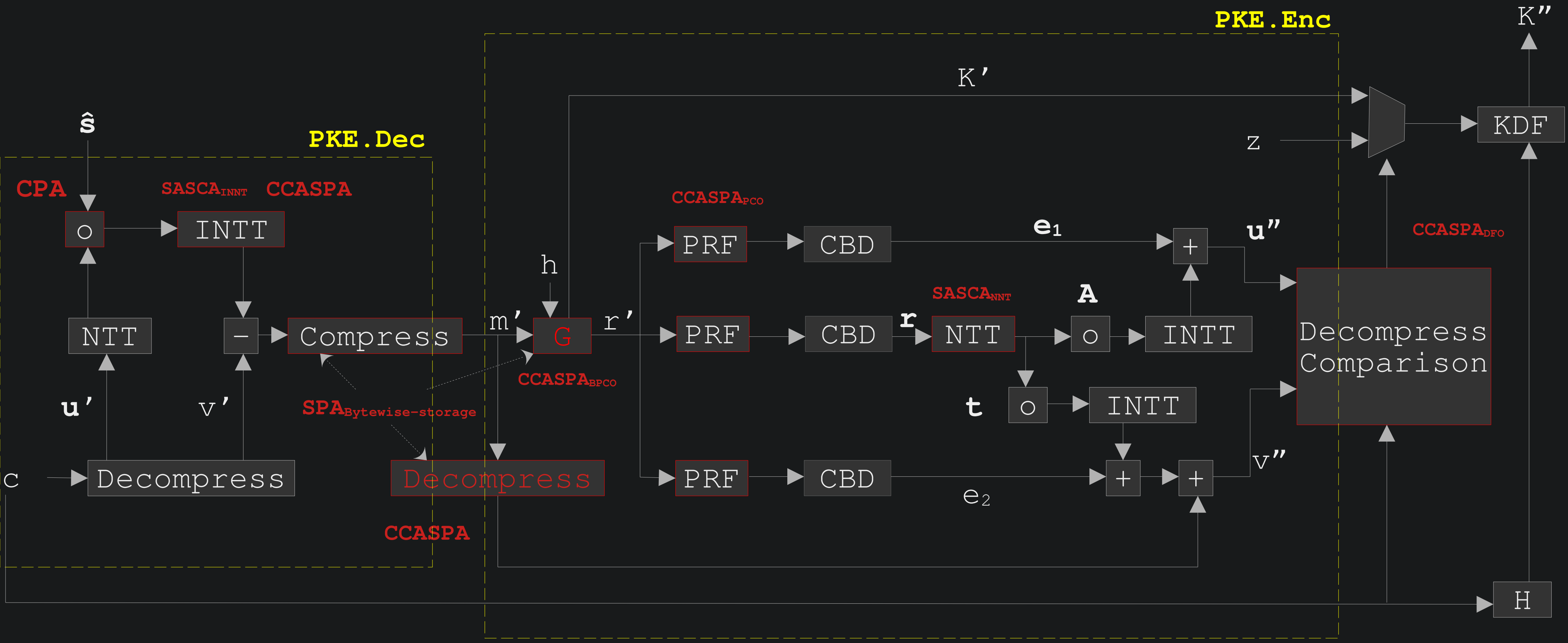


PKE.Enc
K”
K’
z
KDF
ŝ
PKE.Dec
CPA
$\text{SASCA}_{\text{INNT}}$
CCASPA
o
INTT
$\text{CCASPA}_{\text{PCO}}$
PRF
CBD
e_1
+
u”
$\text{CCASPA}_{\text{DFO}}$
h
NTT
−
Compress
m’
G
r’
PRF
CBD
r
$\text{SASCA}_{\text{NNT}}$
NTT
A
o
INTT
Decompress Comparison
$\text{CCASPA}_{\text{BPCO}}$
u’
v’
$\text{SPA}_{\text{Bytewise-storage}}$
t
o
INTT
c
Decompress
Decompress
PRF
CBD
+
+
v”
e_2
CCASPA
H

Algorithms

Doc

Algorithm 4 Kyber.Encaps$(pk = (\mathbf{t}, \rho))$

$m \leftarrow \{0,1\}^{256}$
$(\hat{K}, r) := \mathrm{G}(\mathrm{H}(pk), m)$
$(\mathbf{u}, v) := \mathsf{Kyber.CPA.Enc}\left((\mathbf{t}, \rho), m; r\right)$
$c := (\mathbf{u}, v)$
$K := \mathrm{H}(\hat{K}, \mathrm{H}(c))$
return (c, K)

Algorithm 5 Kyber.Decaps$(sk = (\mathbf{s}, z, \mathbf{t}, \rho), c = (\mathbf{u}, v))$

$m' := \mathsf{Kyber.CPA.Dec}(\mathbf{s}, (\mathbf{u}, v))$
$(\hat{K}', r') := \mathrm{G}(\mathrm{H}(pk), m')$
$(\mathbf{u}', v') := \mathsf{Kyber.CPA.Enc}\left((\mathbf{t}, \rho), m'; r'\right)$
if $(\mathbf{u}', v') = (\mathbf{u}, v)$ **then**
  return $K := \mathrm{H}(\hat{K}', \mathrm{H}(c))$
else
  return $K := \mathrm{H}(z, \mathrm{H}(c))$
end if

Algorithm 3 Kyber.CPA.Dec$(sk = \mathbf{s}, c = (\mathbf{u}, v))$: decryption

$\mathbf{u} := \mathsf{Decompress}_q(\mathbf{u}, d_u)$
$v := \mathsf{Decompress}_q(v, d_v)$
return $\mathsf{Compress}_q(v - \mathbf{s}^T\mathbf{u}, 1)$

Algorithm 2 Kyber.CPA.Enc$(pk = (\mathbf{t}, \rho), m \in \mathcal{M})$: encryption

$r \leftarrow \{0,1\}^{256}$
$\mathbf{t} := \mathsf{Decompress}_q(\mathbf{t}, d_t)$
$\mathbf{A} \sim R_q^{k \times k} := \mathsf{Sam}(\rho)$
$(\mathbf{r}, \mathbf{e}_1, e_2) \sim \beta_\eta^k \times \beta_\eta^k \times \beta_\eta := \mathsf{Sam}(r)$
$\mathbf{u} := \mathsf{Compress}_q(\mathbf{A}^T\mathbf{r} + \mathbf{e}_1, d_u)$
$v := \mathsf{Compress}_q\left(\mathbf{t}^T\mathbf{r} + e_2 + \left\lceil \frac{q}{2} \right\rfloor \cdot m, d_v\right)$
return $c := (\mathbf{u}, v)$

n	k	q	η_1	η_2	(d_u, d_v)	δ
256	2	3329	3	2	$(10, 4)$	2^{-139}
256	3	3329	2	2	$(10, 4)$	2^{-164}
256	4	3329	2	2	$(11, 5)$	2^{-174}

BACK

SC-assisted CCA

SC-info ~ POC => CCA on underlying CPA-PKE

POC or DFO: whether or not PKE.Dec result is equiv to a *reference* plaintext.

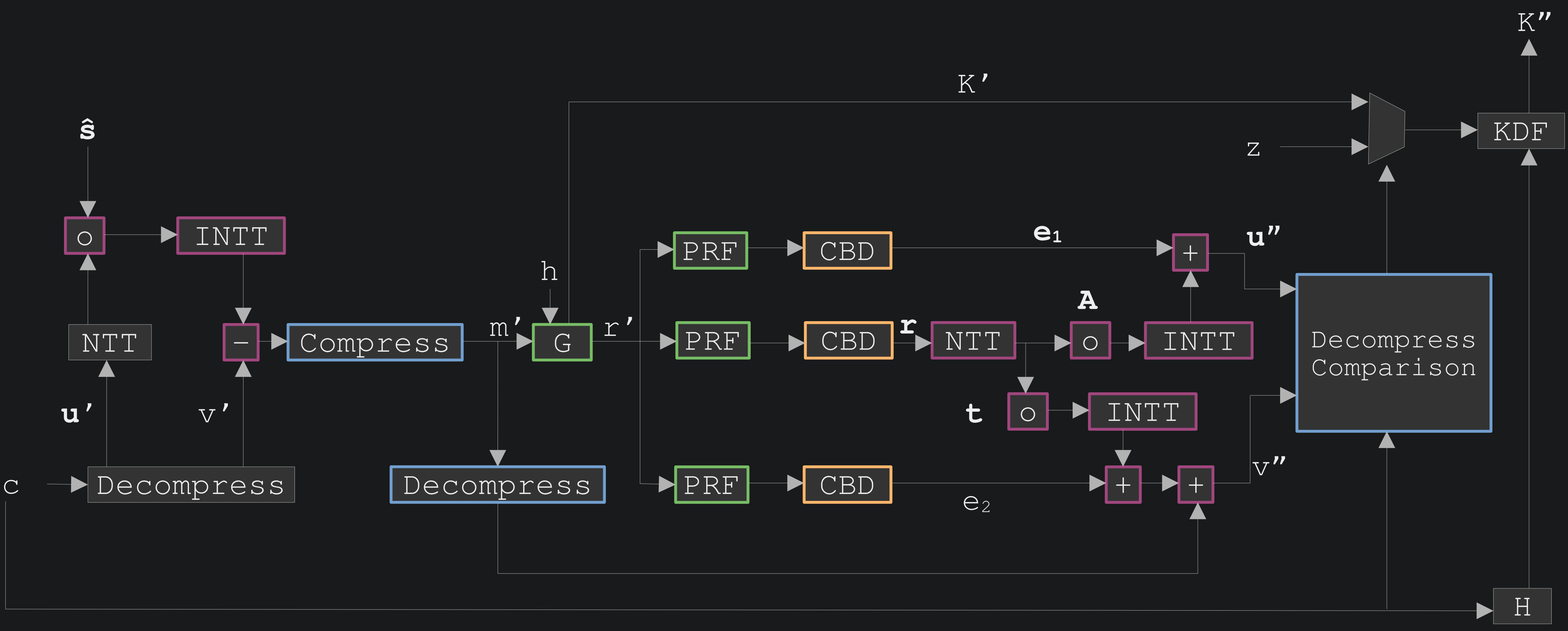
SC-info: SC-leakage of PRF or PRG in re-encryption (SHAKE/AES)

$m = 0$
$\mathbf{c}' = (\mathbf{u}, v)$
$m' = v - \mathbf{s}^T\mathbf{u}$, $\mathbf{u} = (\mathbf{u}_0 = \rho/2, \mathbf{u}_1 = 0)$, $v = (0, \ldots, t, 0, \ldots, 0)$

$\mathrm{POC}(m, \mathbf{c}') = 1 \Leftrightarrow |\mathbf{s}_{0i} \times \rho/2 + t \times \rho/2| \leq \rho$

POC: fixed vs random binary classifier.

BACK



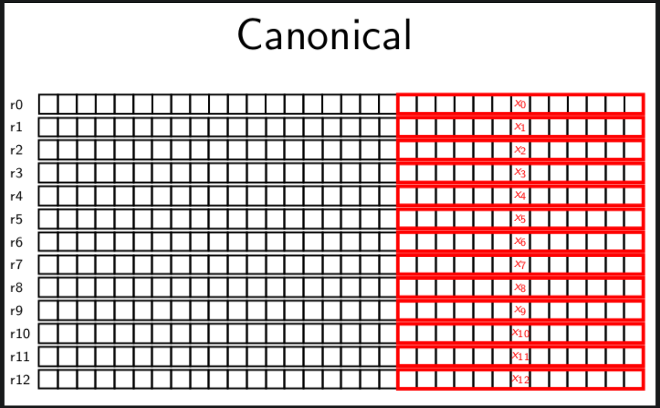
K"
K'
KDF
z
ŝ
∘
INTT
e1
+
u"
PRF
CBD
h
A
NTT
−
Compress
m'
G
r'
PRF
CBD
r
NTT
∘
INTT
Decompress Comparison
u'
v'
t
∘
INTT
c
Decompress
Decompress
PRF
CBD
+
+
v"
e2
H

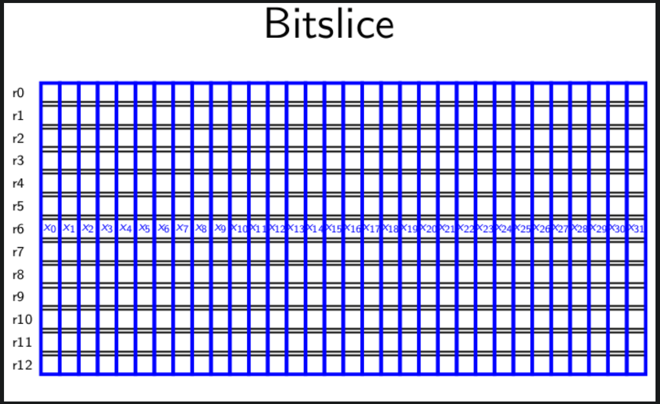
Masked Operations

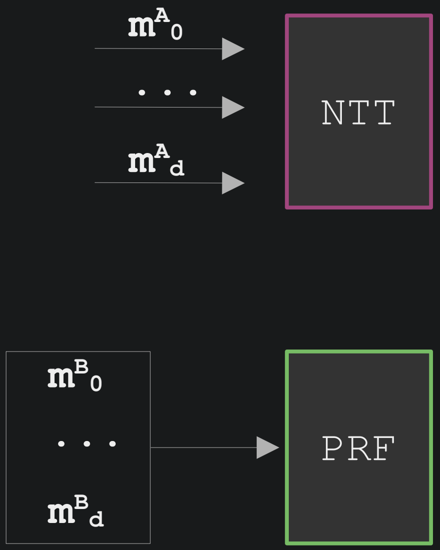
Doc

$$s = \sum^{d-1} m_d^A \mod q$$

$$s = \oplus^{d-1} m_d^B$$







BACK

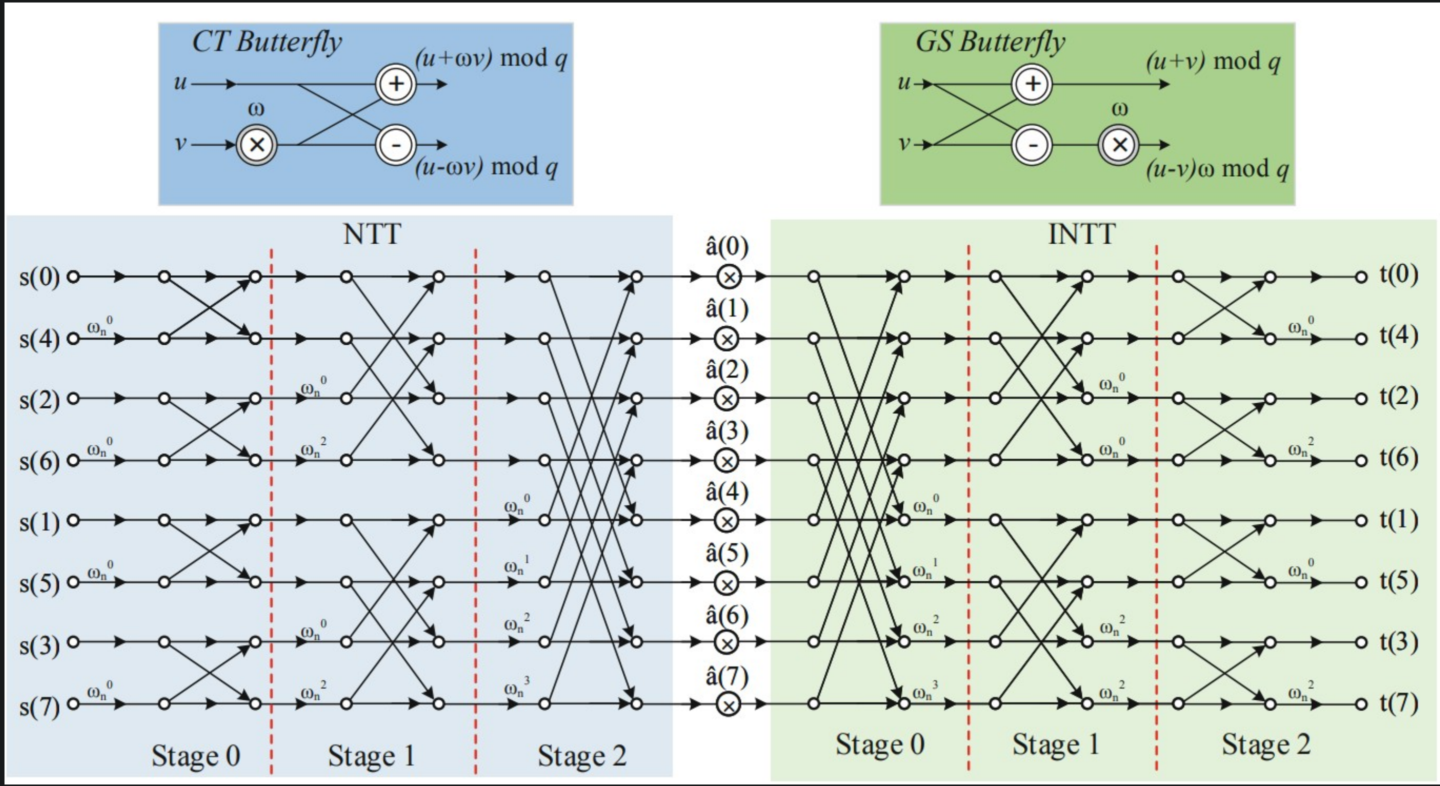
NTT & INTT

Doc

$R_q = \mathbb{Z}_q[X]/(X^{256}+1)$

ω is the n-th root of unity, $\psi^2 = \omega$

Wrapped Convolution to avoid zero padding

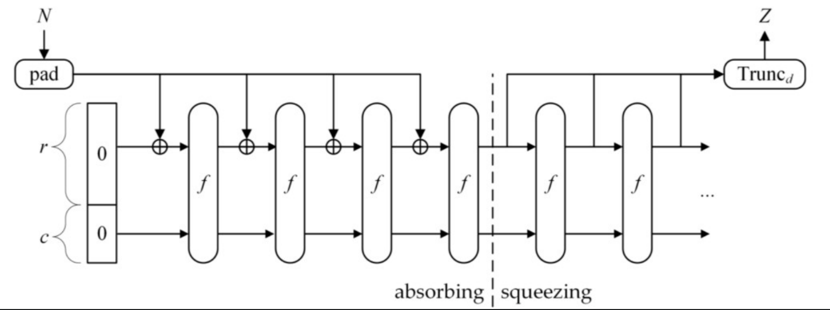


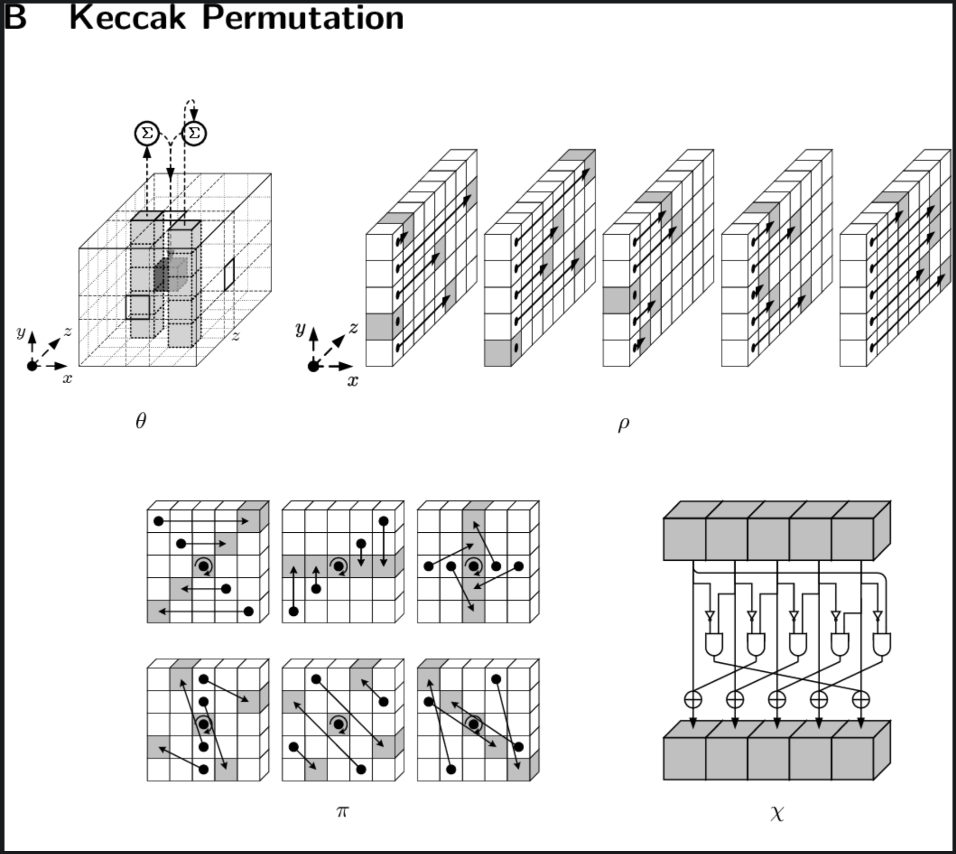
BACK

KECCAK (XOF, H, G, PRF, KDF)

Doc

θ: XOR bits w/ parities of 2 col
ρ: Rotate bit by an offset
π: Rearrange positions of the lanes
χ: XOR bits w/ non-linear function of 2 other bits in its row
ι: Modify some bits of lane(0,0) (depend on round index)





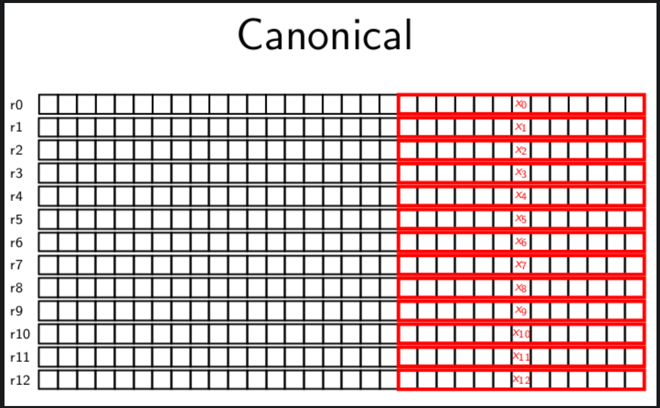
BACK

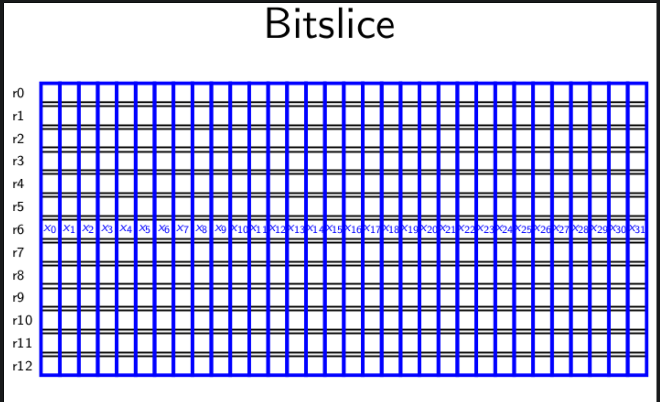
Masked Operations

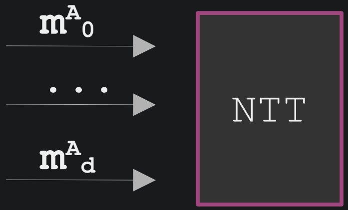
Doc

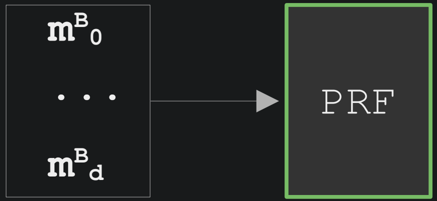
$$s = \sum^{d-1} m_d^A \mod q$$

$$s = \bigoplus^{d-1} m_d^B$$





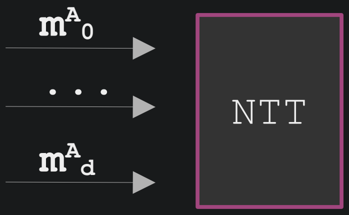




BACK

First experiment

Doc



Distinguish decrypted message m' from *reference* message m
Even if m differs m' 1 bit,
$\mathbf{r} = G(m)$ can differ $\mathbf{r}' = G(m')$ in several coefficients

First order masking:
$\mathbf{r} = [r_0, \ldots, r_{255}]$, $r_i \in [-2, 2]$
$\mathbf{x} = [x_0, \ldots, x_{255}]$, $x_i \in [0, q]$, $\mathbf{l}_0 = HW(\mathbf{x}) + \beta$
$\mathbf{y} = (\mathbf{r} - \mathbf{x})\%q$, $\mathbf{l}_1 = HW(\mathbf{y}) + \beta$

Goal: Distinguish $\mathbf{r}^0$, $\mathbf{r}^1$ given $\mathbf{L}$

Leakage:
- Jointly (SASCA)
- Prod (normalized prod)
- Sum
- Abs diff

Noise from low to high & increasing #shares

Independent coefficients =>
- Equiv with whole key distinguishing
- Large #coeff <=> higher distinguishability
- Higher complexity (nonD&C)
- IT value can be deduced from IT of each coeff

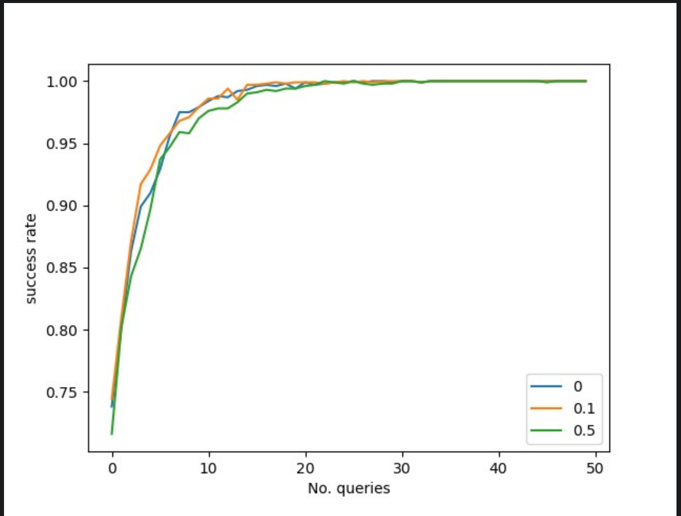
Unprotected:

$L = HW(\mathbf{R}) + \beta$
SR = 1 as soon as $HW(\mathbf{r}^0{}_i) \neq HW(\mathbf{r}^1{}_i)$

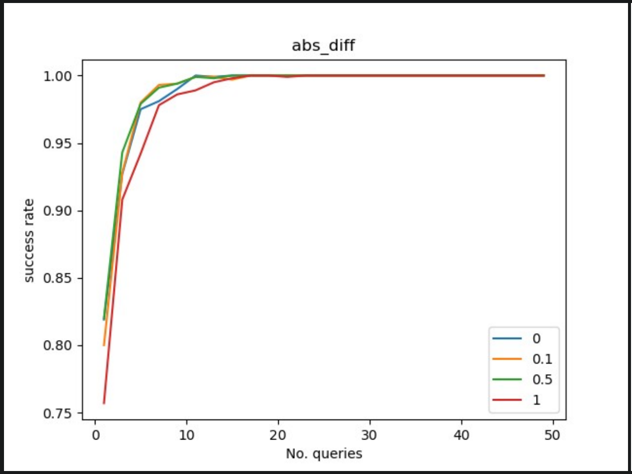
BACK

First results-SR

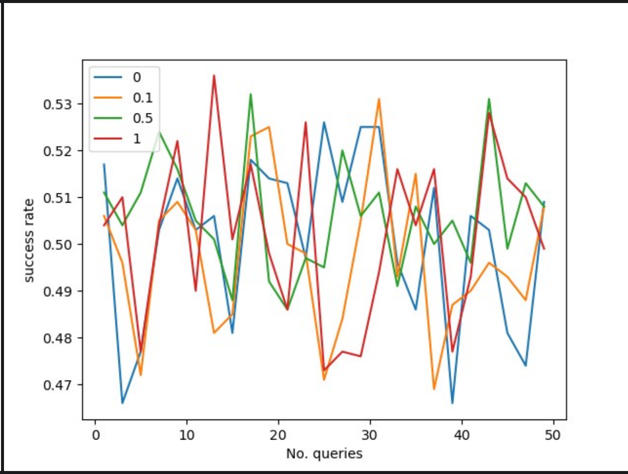
2 shares



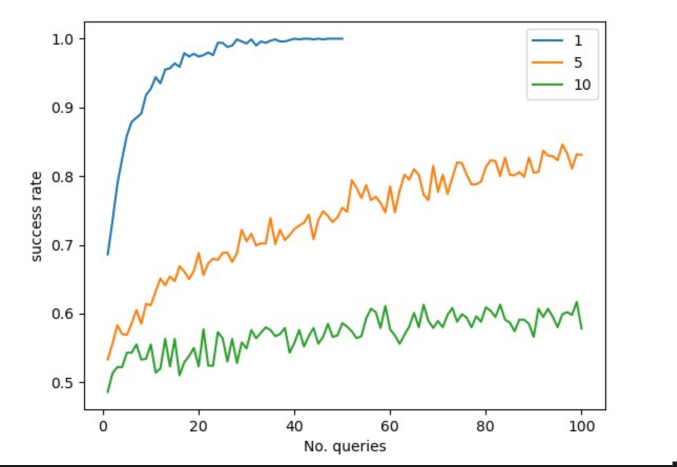
abs_diff



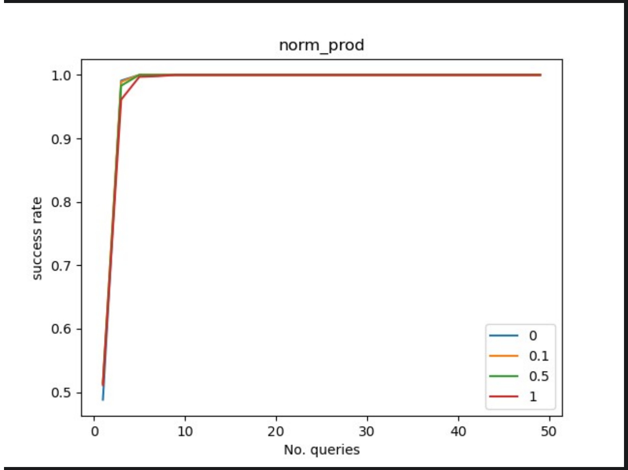
sum



prod



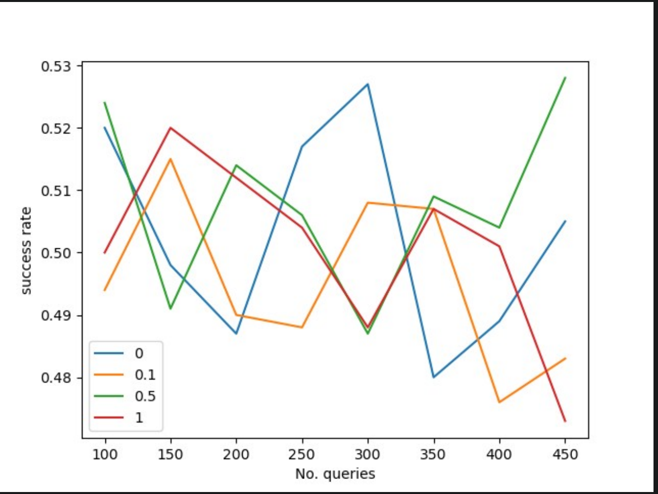
Normalized prod



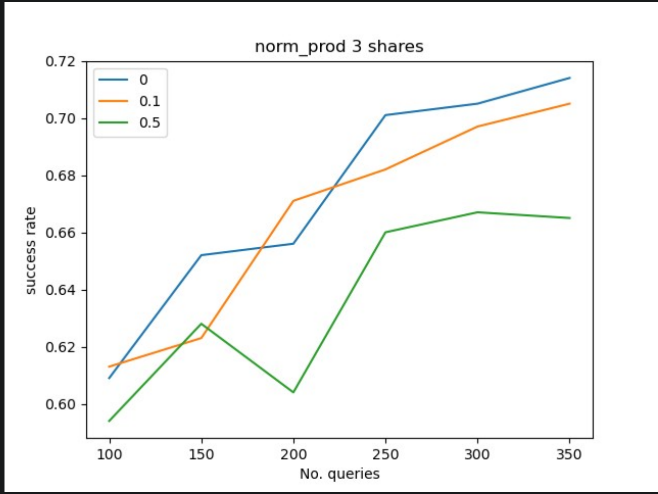
First results-SR

3 shares

prod



Normalized prod

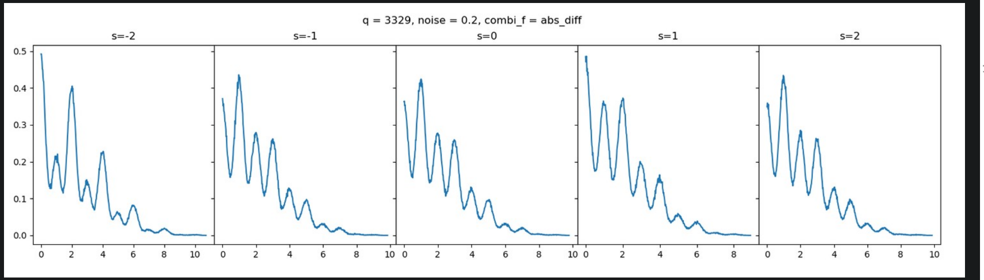


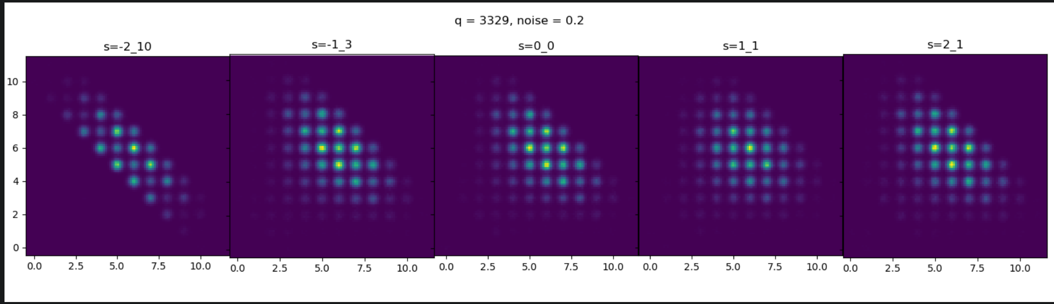
First results

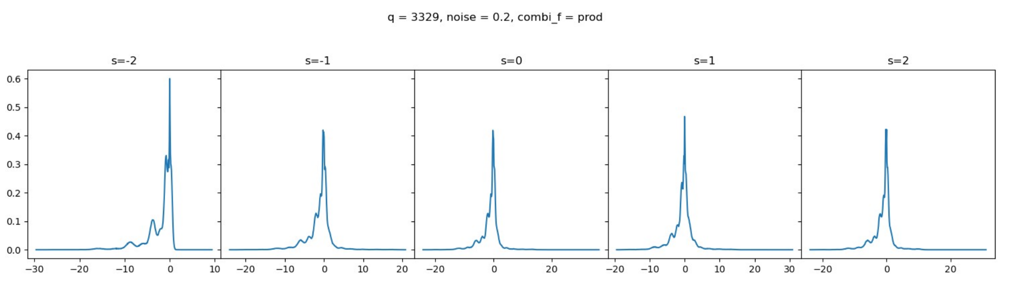
Doc

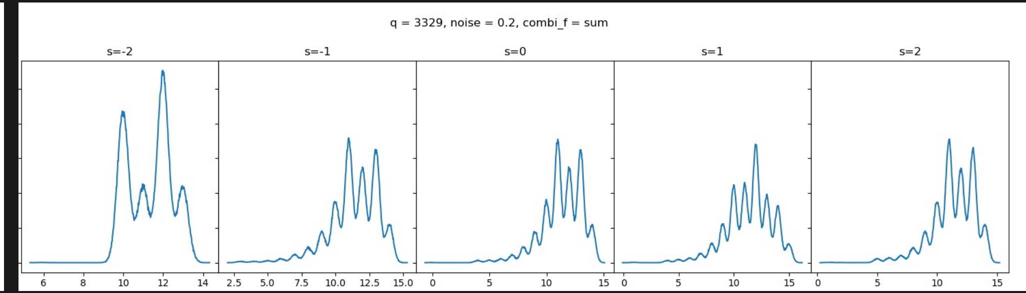
Distribution:

abs_diff









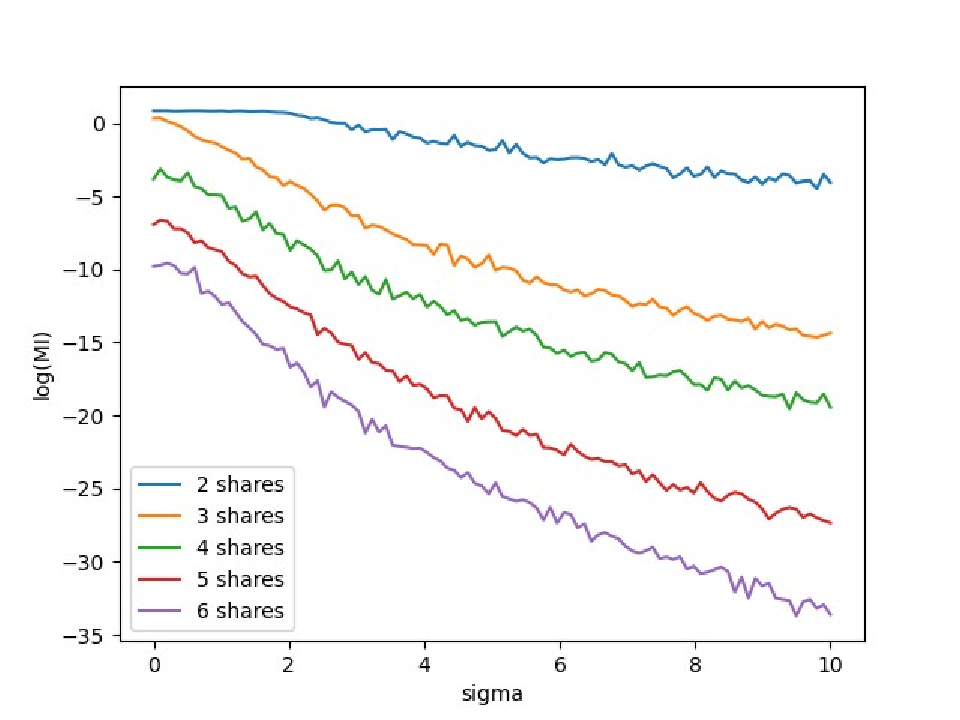
norm_prod

sum

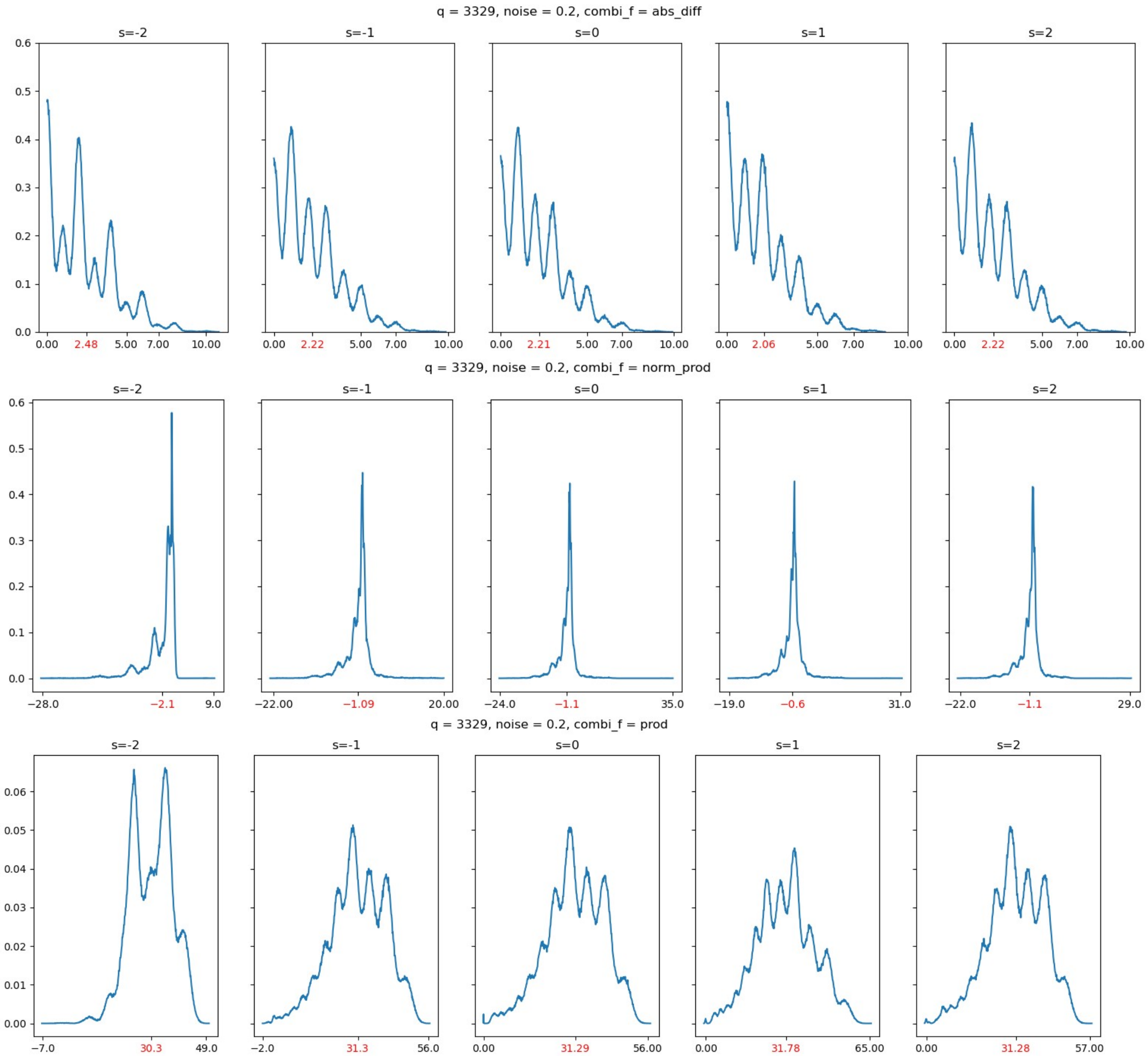
BACK

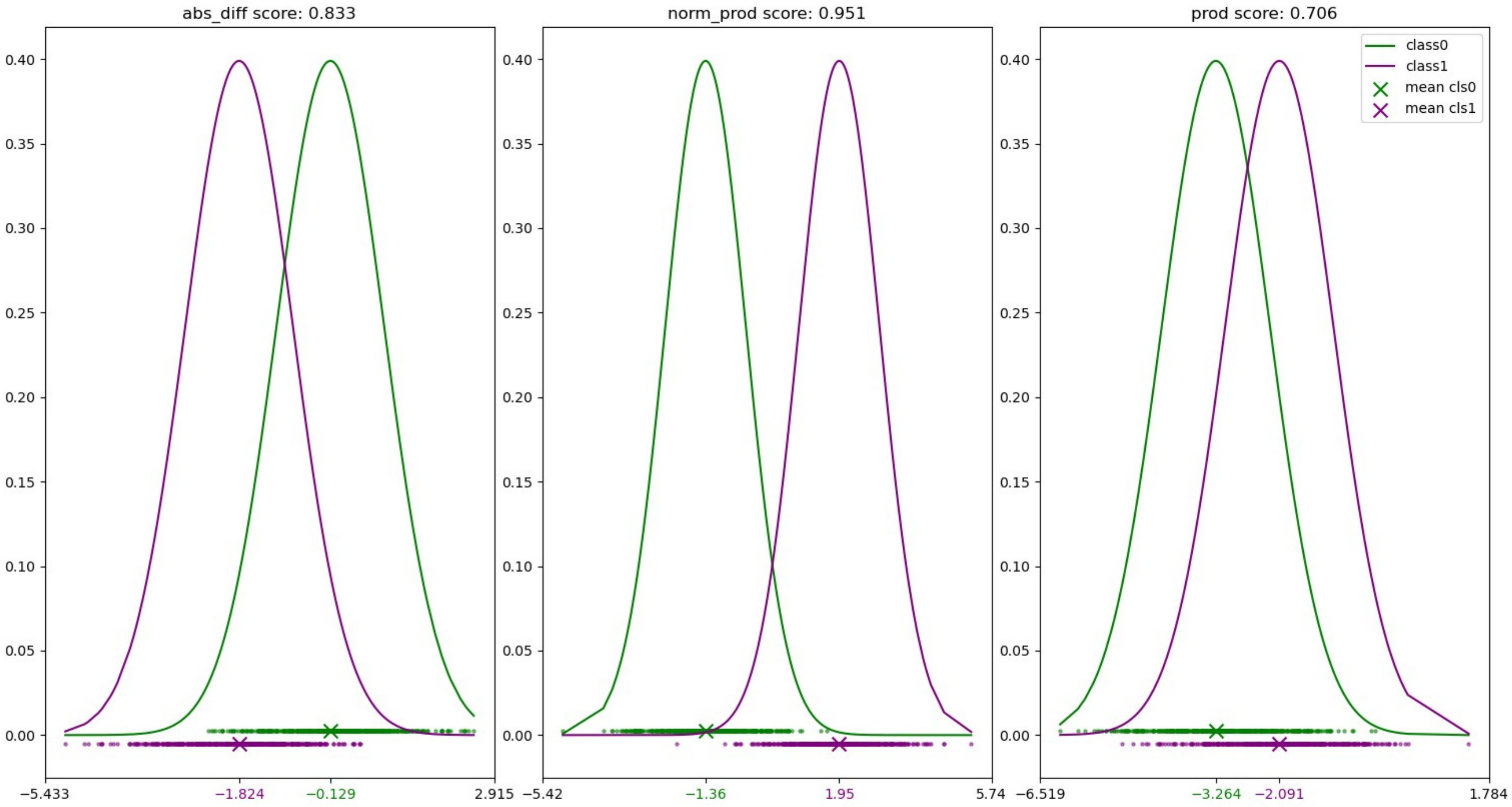
First results–MI

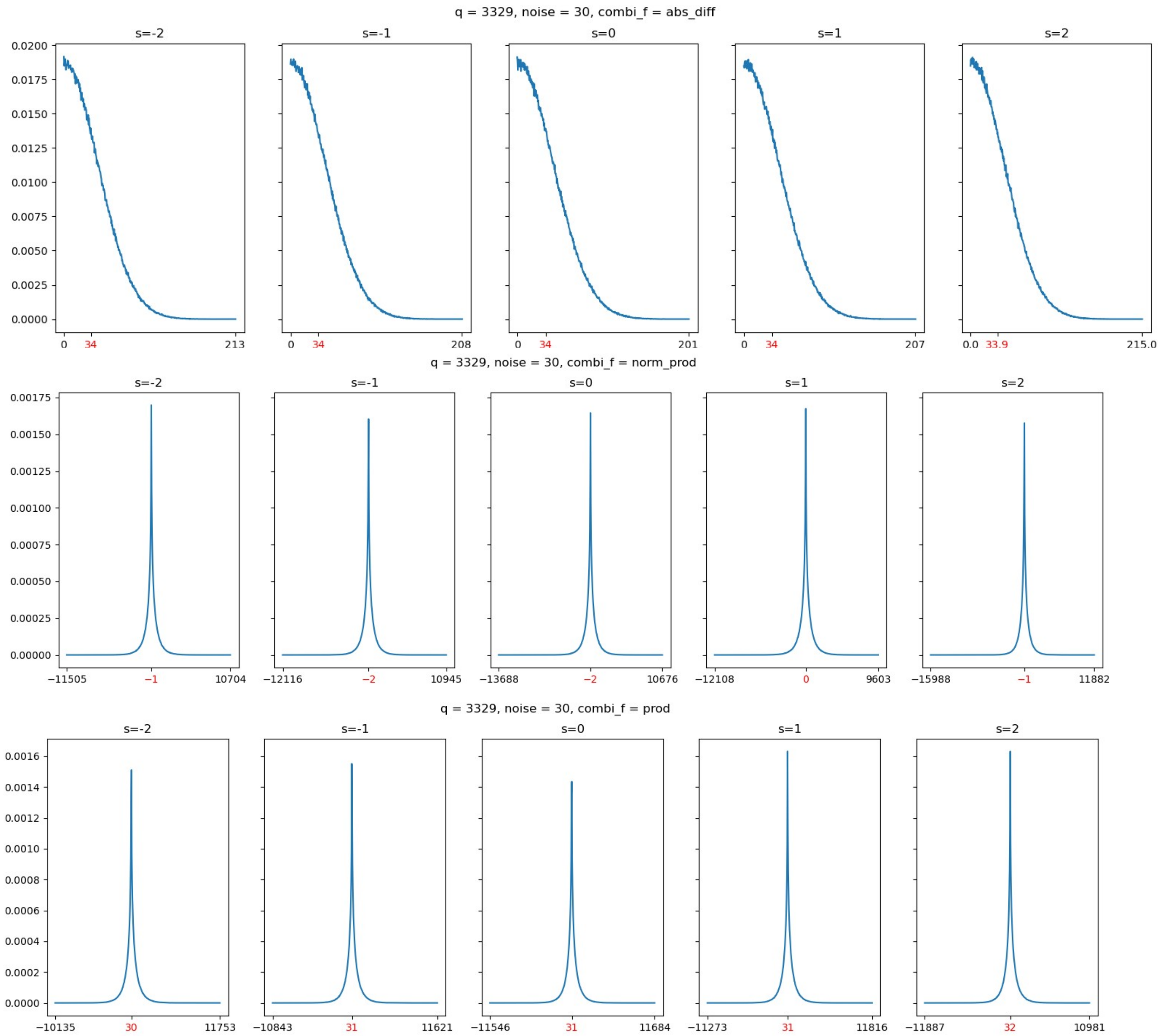
SASCA

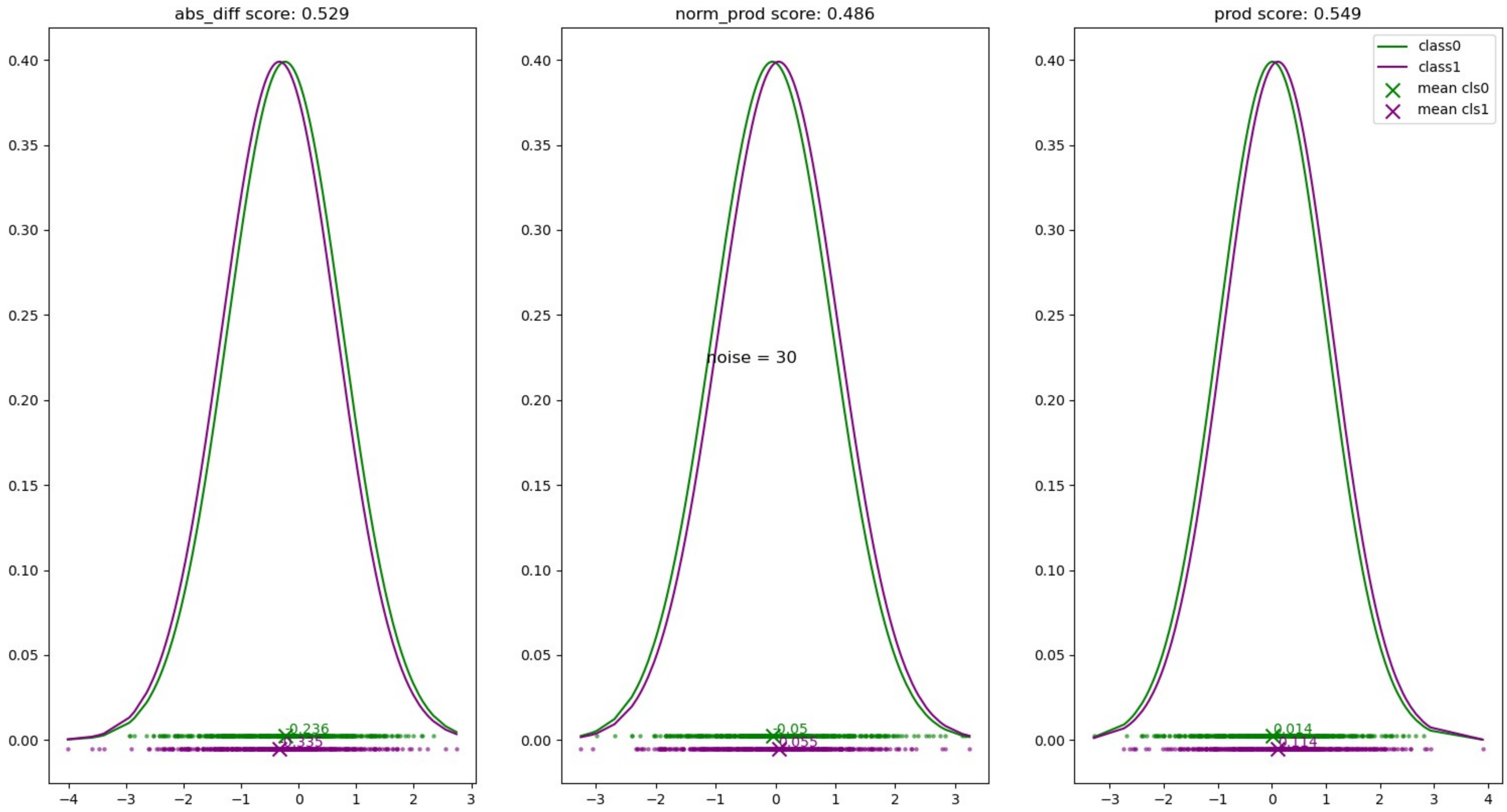


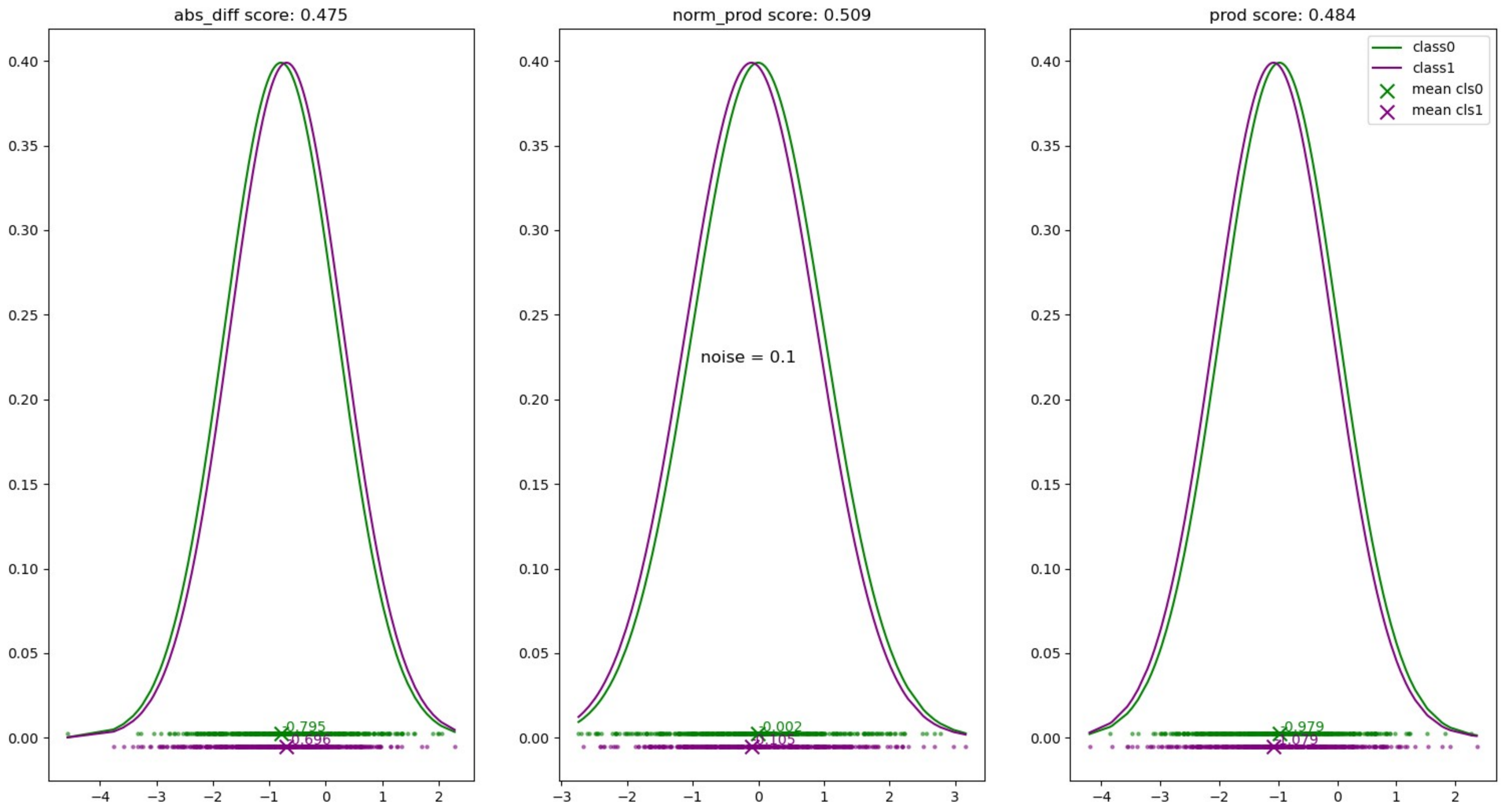
MI: prod sum

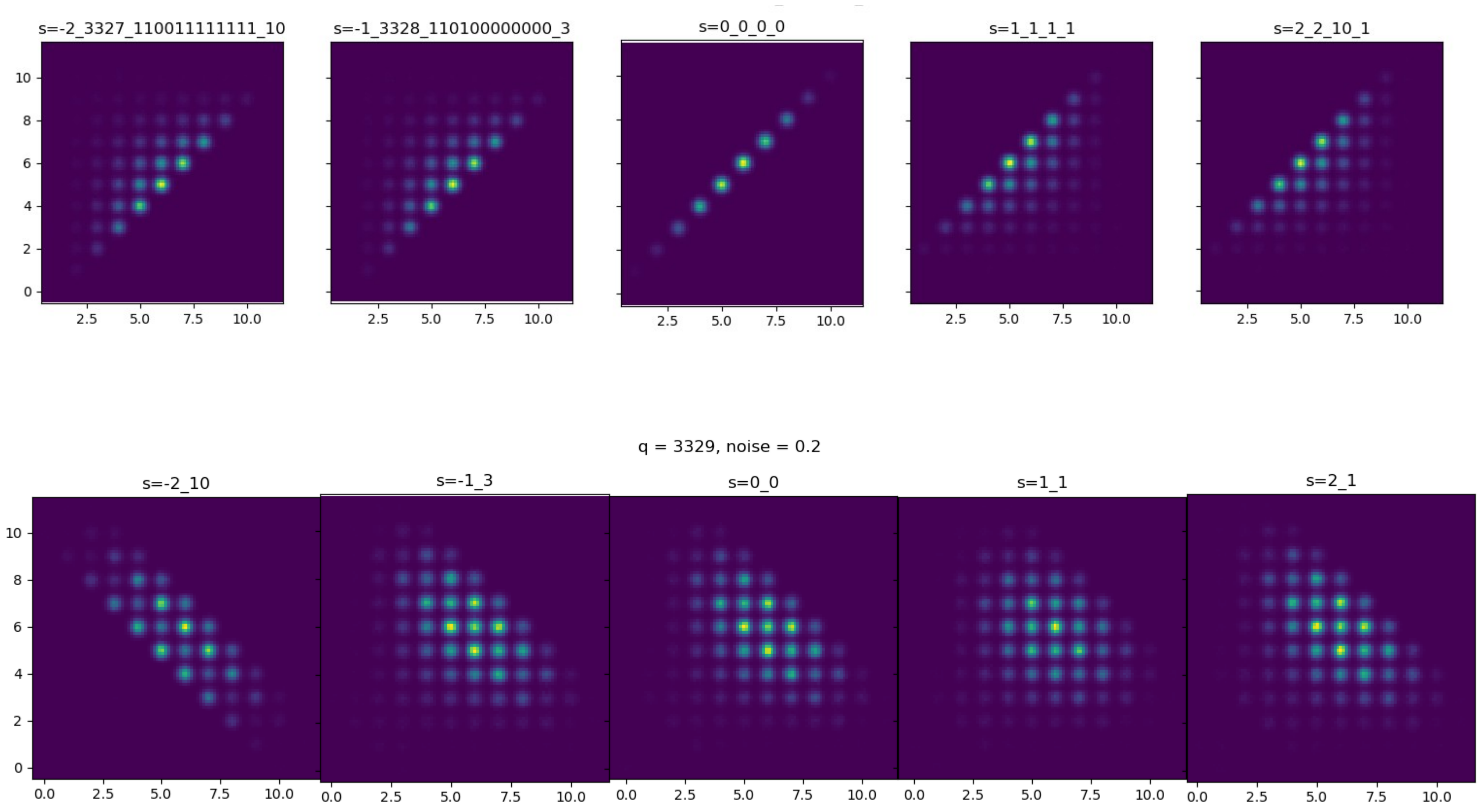

q = 3329, noise = 0.2, combi_f = abs_diff
s=-2
s=-1
s=0
s=1
s=2
2.48
2.22
2.21
2.06
2.22
q = 3329, noise = 0.2, combi_f = norm_prod
-2.1
-1.09
-1.1
-0.6
-1.1
q = 3329, noise = 0.2, combi_f = prod
30.3
31.3
31.29
31.78
31.28


abs_diff score: 0.833
norm_prod score: 0.951
prod score: 0.706
class0
class1
mean cls0
mean cls1
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
−5.433
−1.824
−0.129
2.915
−5.42
−1.36
1.95
5.74
−6.519
−3.264
−2.091
1.784


q = 3329, noise = 30, combi_f = abs_diff
s=-2
s=-1
s=0
s=1
s=2
0 34 213
0 34 208
0 34 201
0 34 207
0.0 33.9 215.0
q = 3329, noise = 30, combi_f = norm_prod
s=-2
s=-1
s=0
s=1
s=2
−11505 −1 10704
−12116 −2 10945
−13688 −2 10676
−12108 0 9603
−15988 −1 11882
q = 3329, noise = 30, combi_f = prod
s=-2
s=-1
s=0
s=1
s=2
−10135 30 11753
−10843 31 11621
−11546 31 11684
−11273 31 11816
−11887 32 10981


abs_diff score: 0.529
norm_prod score: 0.486
prod score: 0.549
class0
class1
mean cls0
mean cls1
noise = 30
-0.236
-0.335
-0.05
0.055
0.014
0.114


abs_diff score: 0.475
norm_prod score: 0.509
prod score: 0.484
noise = 0.1
class0
class1
mean cls0
mean cls1


s=-2_3327_110011111111_10
s=-1_3328_110100000000_3
s=0_0_0_0
s=1_1_1_1
s=2_2_10_1
q = 3329, noise = 0.2
s=-2_10
s=-1_3
s=0_0
s=1_1
s=2_1

$$
\begin{aligned}
f(\mathbf{l}|s) &= \sum_{r\in[0,q]} f(\mathbf{l}|s,r)\cdot p(r) \\
&= \sum_r f([l_r, l_{ms}|s,r)\cdot p(r) \\
&= \sum_r f([l_r, l_{ms}|r,ms)\cdot p(r) \\
&= \sum_r f(l_r|r)\cdot f(l_{ms}|ms)\cdot p(r) \\
&= \frac{1}{q}\sum_r f(l_r|r)\cdot f(l_{ms}|ms)
\end{aligned}
$$

$$L_r|R \sim \mathcal{N}(HW(R),\sigma^2), \quad L_{ms}|MS \sim \mathcal{N}(HW(MS),\sigma^2)$$

$$p(s|\mathbf{l}) = \frac{f(\mathbf{l}|s)}{\sum_{s^*\in\mathcal{S}} f(\mathbf{l}|s^*)}$$

$$\tilde{\mathbf{s}} = [\tilde{s}_0, \tilde{s}_1, \ldots, \tilde{s}_{255}]$$

$$\tilde{\mathbf{l}} = [\tilde{\mathbf{l}}_0, \tilde{\mathbf{l}}_1, \ldots, \tilde{\mathbf{l}}_{255}], \quad \tilde{\mathbf{l}}_i = [l_{r_i}, l_{ms_i}]$$

$$p(\tilde{\mathbf{s}}|\tilde{\mathbf{l}}) = \prod_{i=0}^{255} p(\tilde{s}_i|\tilde{\mathbf{l}}_i)$$

$$\arg\max_{\tilde{\mathbf{s}}} \sum \log(p(\tilde{\mathbf{s}}|\tilde{\mathbf{l}}))$$

CPA

Doc

Target schoolbook multiplications (incomplete NTT)(KC)

Exploit leakage of storing resulting coefficients in memory

$n = 256 = 2^8, \quad q = 3329 = 13 \cdot n + 1 \Rightarrow$ 7 layers incomplete NTT followed by 2-coefficient schoolbook multiplication.
$\mathbf{a} \mapsto \prod^{128}(a_{0i} + a_{1i}X)$

For each result coefficient that is stored in memory:

$c_{0i} = a_{0i}b_{0i} + \zeta a_{1i}b_{1i}, \quad c_{1i} = a_{0i}b_{1i} + a_{1i}b_{0i}$, where a_{0i}, a_{1i} are known,

Two resulting coefficient are stored by the same instruction instead of consecutively
⇒ Need to guess 2 coefficient at once within the range $(-q/2, q/2] \Rightarrow q^2$ combination of guessing values.

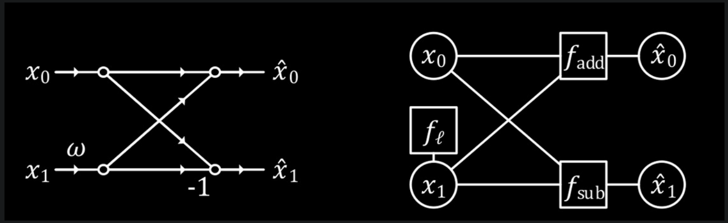
⇒ Select the pair of values resulting in the largest correlation coefficient as key guess.

BACK

SASCA on INNT [Doc]

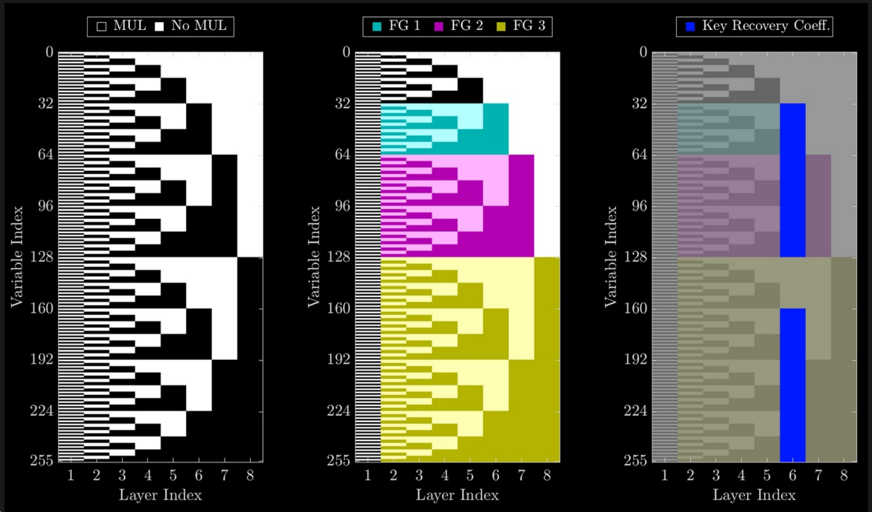
Target the computation of INTT($\hat{\mathbf{s}}^{\mathbf{T}} \circ$ NTT($\mathbf{u}'$))(KC)

1. Build template for MulMod input
(q.n/2 = 983168 templates for all possible combination of operand/twiddle)
2. Template matching on each butterfly.
3. Combine SC-info over entire INTT using BP.



Notes:
- Multiplication operation is the source of SC-info
- Apply BP on disjoint subgraphs => incomplete key recovery

=> Lattice reduction for key recovery
- In masking: BP twice to get the intermediates in all INNT invocations.



(a) Location of multiplications

(b) Three used factor subgraphs

(c) Intermediates used for key recovery

Cost:
- 983168 templates
- 100 traces for each template
- Lattice reduction: 5min (>160 coeff), hours (~150) Unsuccessful (<140)

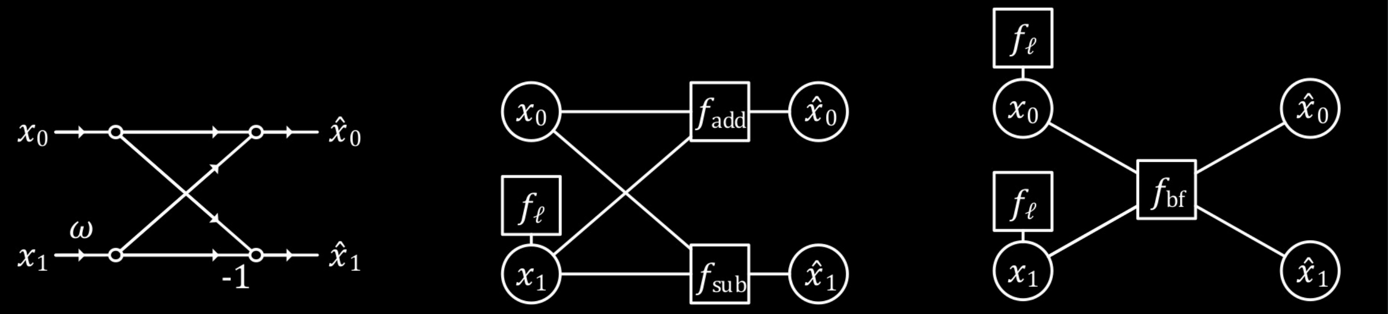
BACK

SASCA on NNT

Doc

Target the computation of NTT(**r**) then compute the decrypted message m’ = Decode(c_2 − $\mathbf{t}^T$ o **r**).

Improvements (wrt SASCA on INNT):
- **r** has narrower support (η = 4)
- HW templates instead of ID templates (70000x less)
- Cluster factors of same butterfly into one (f_{add}, f_{sub} => f_{bf}) => avoid small loopy BP



(a) Single butterfly (b) FG of [23] (c) Our FG

BACK

SPA INNT

Doc

SC-assisted CCA.

Target the output of INTT: fqmul() function

- Secret key coefficients are in small range (η = 2, [-2,2])
- Select appropriate ciphertext s.t HW(INTT output) is partitioned
- Query chosen ciphertexts => classify PoI into different classes => recover key coefficients based on the partitions.

=> full key recovery using 4 traces.

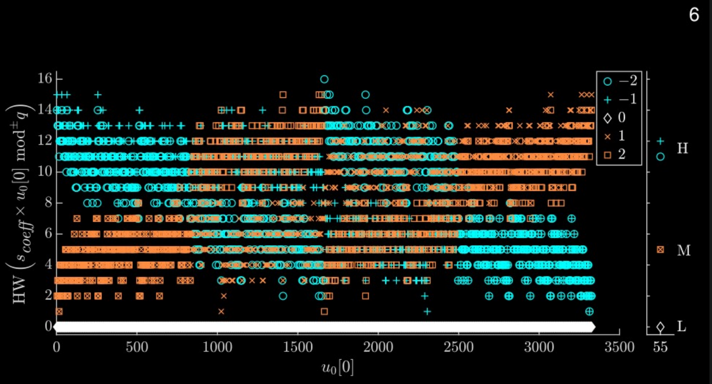


Fig. 5. HWs for all possible choices of $u_0[0] \in U$ and s_{coeff} either $\{-2, -1\}$ (red), 0 (black), or $\{1, 2\}$ (blue). Case for $u_0[0] = 55$.

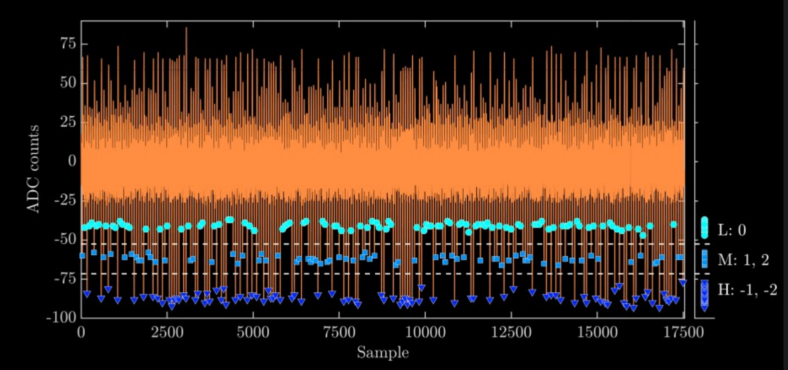


Fig. 8. EM trace with classification into three classes for $u_0[0] = 55$.

BACK

SPA Bytewise-storage

Doc

Message recovery => session key recover
(Chosen) Message recovery => long term key recover

Target any operation involves storage of decrypted message in memory:
- Message Encoding/Decoding
- KeccakAbsorb

1. Determiner-leakage: Distinguish mask value correspond to message bit => recover message bit.

2. Incremental storage: Distinguish intermediate value (HW) after each update (per bit)=> recover message bit.

3. Bytewise storage:
Exploit ciphertext malleability: Bit-Flip or Message-Rotation
Recover message bit by comparing the difference between original and modified decrypted message process => recover message word

SC-assisted CCA
- Choose ciphertext s.t the decrypted message has strong relation with secret key
- Recover message during decaps (message decoding) using SC-info (BPCO)
- Recover secret key from the relation with the decrypted message

BACK

CCASPA–BPCO

Doc

SC-assisted CCA.

SC-info <=> Binary Plaintext Checking Oracle

-Craft ciphertext s.t decrypted message only depends upon one coefficient of the secret key.
- Decrypted message can only be 0 or 1.
- Target hash operation over decrypted message (towards the end of hash computation).
- EM(G): BPCO
=> key recover

$$c''_i = \begin{cases} \mathcal{D}(\mathbf{s}[1]), & \text{if } i = 0 \\ 0, & \text{for } 1 \leq i \leq \mu - 1 \end{cases}$$

Table 1: Unique distinguishability of every key candidate based on the validity of the codeword for the chosen values of $(k_{\mathbf{u}}, k_{\mathbf{v}})$ for our attack on the R5ND_1KEM_5d variant of IND-CCA secure Round5 KEM. **O** and **X** refer to valid ($c'' = 0$) and invalid codewords ($c'' = 1$) respectively.

Secret Coeff.	$c'' = 0$ (**O**) / $c'' = 1$ (**X**)	
	$(k_{\mathbf{u}}, k_{\mathbf{v}})$	
	(21,3)	(12,1)
-1	**X**	**X**
0	**X**	**O**
1	**O**	**O**

Extend to Parallel BPCO

BACK

CCASPA-PCO

Doc

SC-assisted CCA.

SC-info <=> Plaintext Checking Oracle.

Target any operations that show the different behaviors in processing reference/modified decrypted message:

- PRF
- Poly comparison

Ciphertext is correctly decrypted and decoded if noise is below a threshold.
The size of the noise provides info about the original noise (original ciphertext)- linearly depends on secret key.

- Adding small noise into original ciphertext.
- SC-info <=> POC (whether modified ciphertext decrypted, decoded to original decrypted message).
- Query modified ciphertext + PCO + solve linear equations (key dependent) => KC

BACK

SPA Message Decode

Doc

(Chosen) Message recovery => long term key recover
Target poly_frommsg() function in decoding step (Decompress)

SC-assisted CCA

- Choose ciphertext s.t the decrypted message has strong relation with secret key
- Recover message during decaps (message decoding) using SC-info (BPCO)
- Recover secret key from the relation with the decrypted message

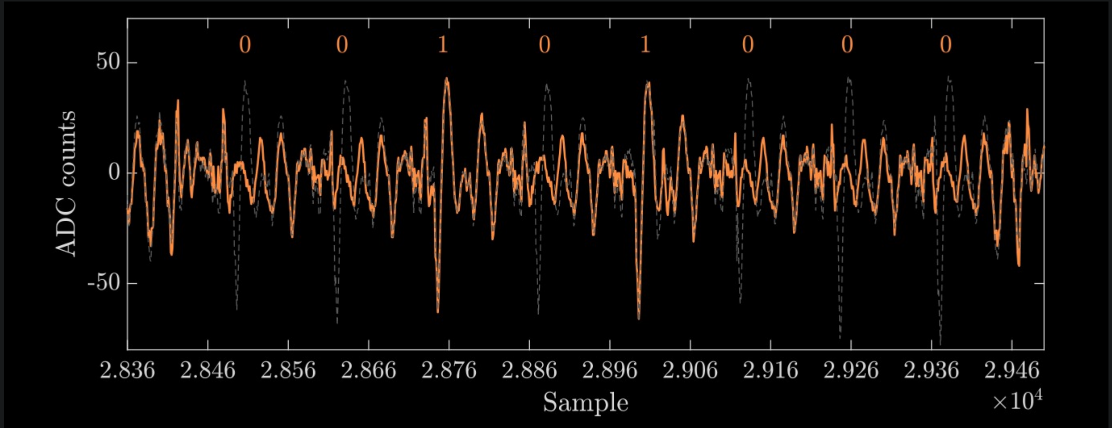


Fig. 11. Example trace at -O0 (blue) showing the differences between message bit = 0 and 1. Reference trace r_1 (with all bits set 1) in gray.

$$\begin{cases} \mathbf{m}_i^{211} = 1, & \text{iff } s_0[i] = -2; \\ \mathbf{m}_i^{419} - \mathbf{m}_i^{211} = 1, & \text{iff } s_0[i] = -1; \\ \mathbf{m}_i^{2705} - \mathbf{m}_i^{2913} = 1, & \text{iff } s_0[i] = \ \ 1; \\ \mathbf{m}_i^{2913} = 1, & \text{iff } s_0[i] = \ \ 2; \\ \text{else}, & \text{iff } s_0[i] = \ \ 0. \end{cases}$$

$$\mathbf{s_0} = (-2) \cdot \mathbf{m}^{211} + (-1) \cdot \left(\mathbf{m}^{419} - \mathbf{m}^{211}\right) + 1 \cdot \left(\mathbf{m}^{2705} - \mathbf{m}^{2913}\right) + 2 \cdot \mathbf{m}^{2913}.$$

BACK

DOC

SASCA on INTT: Single-trace side-channel attacks on masked lattice-based encryption, Peter Pessl and Robert Primas

SASCA on NTT : More practical single-trace attacks on the number theoretic transform, Peter Pessl and Robert Primas

SPA on INTT & KC on Message Decoding:
Magnifying Side-Channel Leakage of Lattice-Based Cryptosystems with Chosen Ciphertexts: The Case Study of Kyber

Bytewise storage:
On Exploiting Message Leakage in (few) NIST PQC Candidates for Practical Message Recovery and Key Recovery Attacks

CCA-SPA_PCO:
Curse of Re-encryption: A Generic Power/EM Analysis on Post-Quantum KEMs

A key-recovery timing attack on post-quantum primitives using the Fujisaki-Okamoto transformation and its application on FrodoKEM

Attacking and Defending Masked Polynomial Comparison for Lattice-Based Cryptography

CCA-SPA_BPCO: Generic Side-channel attacks on CCA-secure lattice-based PKE and KEMs

CCA-SPA_Paralle_BPCO:
Pushing the Limits of Generic Side-Channel Attacks on LWE-based KEMs - Parallel PC Oracle Attacks on Kyber KEM and Beyond

Multiple-Valued Plaintext-Checking Side-Channel Attacks on Post-Quantum KEMs

Efficient SCA on Masked Implementation.

BACK

SCA on Masked Implementation

Problems with masked implementation:

- PDF is a mixture distribution of shares.
- Bitslice: each native word has 1 bit info of the intermediate variable (or *i*-th bit all shares)

Goals

- PDF modeling
- PI evaluation
- Validate with GE

Tools

- GMTA
- MLP
- SASCA/ESASCA/GESASCA/MLPSASCA
- EM
- KDE
- RF

BACK

Distinguishers

Tools	Pros	Cons
GMTA	- Simple estimations	- Exp templates. - Gaussian assumption (bitslice) - Randomness knowledge
SASCA	- Linear templates - Assumption adaptive (ESASCA, G-ESASCA, MLP-ESASCA)	- Complex setup
MLP	- Estimate full PDF - No randomness knowledge - Weak PoIs requirement	- High profiling complexity(vague bounds)
KDE, RF		

BACK

Variable representation

$\mathbb{Z}_q[X]/(X^{256}+1)$

$x = (x_0, x_1, \ldots, x_{255})$
$\mathbf{s} = (s_0, s_1, \ldots, s_k) \leftarrow B_\eta^{k}$, $s_i \in R_q$
$\mathbf{c} = (\mathbf{u}, v)$

BACK