Analyzing the Shuffling Side-Channel Countermeasure for Lattice-Based Signatures

Peter Pessl
IAIK, Graz University of Technology, Austria

Indocrypt 2016, December 12

Accurate depiction of quantum computing

Credit: *The Binding of Isaac: Rebirth* by Edmund McMillen

Introduction

- Lattice-based cryptography is a promising candidate for PQ
- Efficient schemes and implementations
- Implementation security neglected this far
  - very first attack on lattice-based signatures at CHES 2016
- Shuffling proposed as a possible countermeasure
  - protect Gaussian samplers
  - ...but no analysis given

Our contribution

- In-depth analysis of shuffling in context of lattice-based signatures
- Side-channel analysis of a Gaussian sampler implementation
- New attack on shuffling - *unshuffling* and key recovery
  - exploit properties of intermediates
- Show that shuffling *can* be effective
  - but only if done right

BLISS - Bimodal Lattice Signatures [DDLL13]

- **BLISS** - Bimodal Lattice Signature Scheme
  - Ducas, Durmus, Lepoint, Lyubashevsky (CRYPTO 2013)
- Works over ring $\mathcal{R}_q = \mathbb{Z}_q[x]/\langle x^n + 1\rangle$
  - $n = 512$
  - polynomials $\mathbf{a}$,$\mathbf{b}$, $\mathbf{ab} = \mathbf{aB}$, nega-cyclic rotations
- Discrete Gaussians $D_\sigma(x)$

BLISS - Bimodal Lattice Signatures [DDLL13]

Input: Message μ, public key $\mathbf{A} = (\mathbf{a}_1, q-2)$, private key $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2)$

Output: A signature $(\mathbf{z}_1, \mathbf{z}_2^\dagger, \mathbf{c})$

1: $\mathbf{y}_1 \leftarrow D_\sigma^n$, $\mathbf{y}_2 \leftarrow D_\sigma^n$

2: $\mathbf{u} = \zeta \cdot \mathbf{a}_1 \mathbf{y}_1 + \mathbf{y}_2 \bmod 2q$

3: $\mathbf{c} = \mathrm{H}(\lfloor \mathbf{u} \rceil_d \bmod p \| \mu)$

4: Sample a uniformly random bit b

5: $\mathbf{z}_1 = \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$

6: $\mathbf{z}_2 = \mathbf{y}_2 + (-1)^b \mathbf{s}_2 \mathbf{c}$

7: Continue with some probability $f(\mathbf{Sc}, \mathbf{z})$, restart otherwise

8: **return** $(\mathbf{z}_1, \mathbf{z}_2^\dagger = (\lfloor \mathbf{u} \rceil_d - \lfloor \mathbf{u} - \mathbf{z}_2 \rceil_d), \mathbf{c})$

Efficient Gaussian Sampling [PDG14]

- Gaussian convolution: sample twice from a smaller distribution
  (1) $\sigma' = \sigma/\sqrt{1+k^2}$ (2) $y', y'' \leftarrow D_{\sigma'}$ (3) $y = ky' + y''$
- CDT sampling: precompute $T[y] = \mathrm{P}(x < y | x \leftarrow D_{\sigma}^{+})$
  (1) $r \leftarrow [0, 1)$ (2) return $T[y] \leq r < T[y+1]$ (binary search)
- Guide tables: Speed up binary search
  (1) sample first byte of r (2) lookup range in table

A Cache Attack on BLISS [GBHLY16]

- Partial recovery of the noise vector $\mathbf{y}_1$
  - Equation: $z_{ji} = y_{ji} + (-1)^{b_j} \langle \mathbf{s}_1, \mathbf{c}_{ji} \rangle$
- Filter equations with $z_{ji} = y_{ji} \implies \langle \mathbf{s}_1, \mathbf{c}_{ji} \rangle = 0$
  - gather $n = 512$ equations over multiple signatures into $\mathbf{L}$
- Solve $\mathbf{s}_1 \mathbf{L} = 0$
  - error correction using a lattice reduction

Shuffling as a Countermeasure

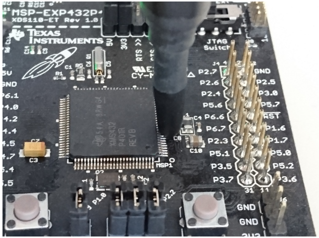
- Protecting samplers appears to be difficult
  - no inherently constant runtime samplers, data-dependent branches
- Idea: sample **y**, then shuffle it
  - breaks connection between sampling time and index
  - simple implementation, low overhead
- Previously proposed [RRVV14, Saa16]
  - ...but no security analysis thus far

Shuffling Variants

- **Single-Stage Shuffling**
  - $\mathbf{y}' \leftarrow D_\sigma^n$, $\mathbf{y} = \text{Shuffle}(\mathbf{y}')$
- **Two-Stage Shuffling** [Saa16]
  - shuffling twice, combine with [PDG14]
  - $\mathbf{y}', \mathbf{y}'' \leftarrow D_{\sigma'}^n$, $\mathbf{y} = k \cdot \text{Shuffle}(\mathbf{y}') + \text{Shuffle}(\mathbf{y}")$

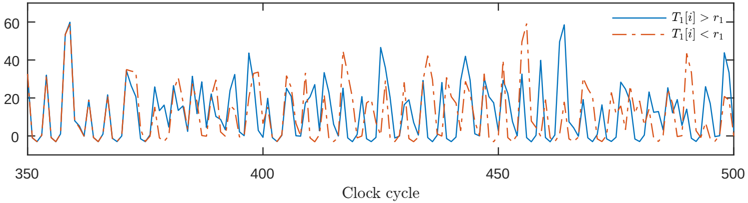
How much do Samplers leak?

- Split-Sampler [PDG14]
  - sampling from *small* distribution $D_{\sigma'}$
  - two classified samples to recover y
- ARM Cortex M4F (TI MSP432)
- EM measurement on core-voltage regulation
- SPA-like attack (single trace)



Recovering the Control Flow

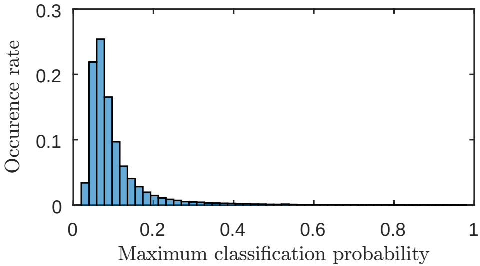
- Recover the steps in the binary search
- Record a reference trace for all possible jumps
  - match using mean of squared error
- Perfect accuracy



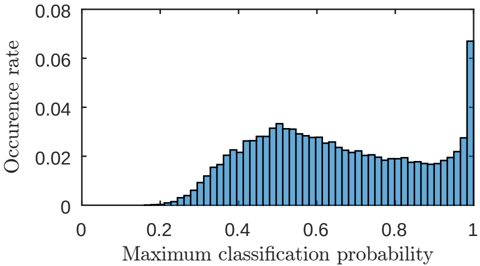
Recover the Sampled Value

- Control flow alone not sufficient
  - guide tables $\rightarrow$ initial range for binary search
- Use template attacks
  - templates for all values and possible flows
- Success highly dependent on nr. of comparisons in binary search

SCA Results



No comparison



1 comparison

Success rate with $>$ 1 comparison: 99.9%

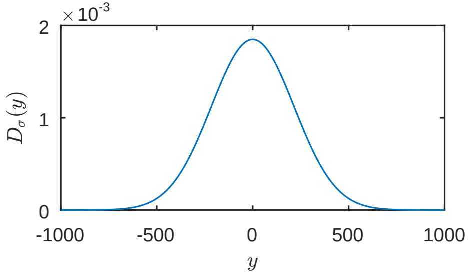
Modeled Adversaries

- **A1 - perfect adversary**
  - knows all sampled values
  - evaluate theoretical limits of shuffling
- **A2 - profiled SCA adversary**
  - recovers all samples requiring 2 or more comparisons
  - $|\text{sample}| > 47$, 1.5%
- **A3 - non-profiled SCA adversary**
  - samples that are uniquely determined by control flow
  - $|\text{sample}| > 54$, 0.5%

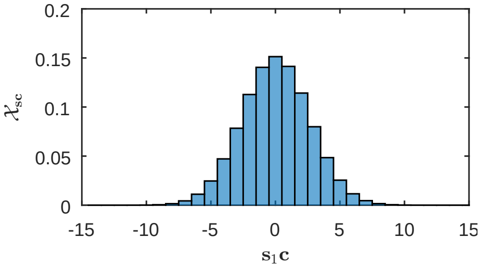
An Attack on Shuffling

- Re-assign samples to index
  - assumption: shuffling is leak-free
- Observation in $\mathbf{z}_1 = \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c}$
  - $\mathbf{y} \leftarrow D_\sigma^n$, $\sigma = 215$
  - $\mathbf{s}_1, \mathbf{c}$ more or less sparse, small coefficients

Coefficient-wise Distributions



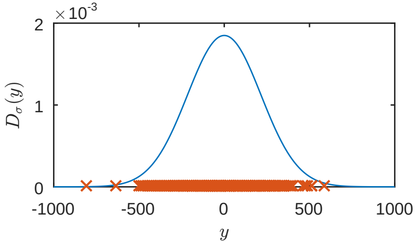
Distribution of **y**: D_σ



Distribution of $\mathbf{s}_1\mathbf{c}$

An Attack on Shuffling

- $\mathbf{z}_1 = \mathbf{y}_1 + (-1)^b \mathbf{s}_1 \mathbf{c} \approx \mathbf{y}_1$
- Given a y, check for *proximity* to all $z_i \in \mathbf{z}$
  - if only one z_i *close*: $z_i - y = (-1)^b \langle \mathbf{s}_1, \mathbf{c}_i \rangle$
- Success for large $|z_i|, |y|$ (tail of D_σ)



Key Recovery

- Keep only highly probable equations (P $>$ 0.99)
- Key recovery: similar to Groot Bruinderink et al. [GBHLY16]
  - gather equations $z_{ji} = y_{ji} + (-1)^{b_j} \langle \mathbf{s}_1, \mathbf{c}_{ji} \rangle$
  - b recoverable with SCA: $n = 512$ equations
  - b not recoverable: filter $z_{ji} = y_{ji}$ (factor 6.6)

Results - Single Stage

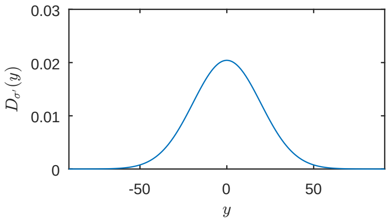
- Number of required signatures increases only slightly
- A2, A3: classifiable samples in the tail of D_σ
  - ... which is where the matching works

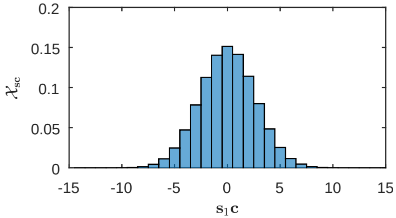
	A1	A2	A3
no shuffling	1	4 400 (29 000)	36 000 (239 000)
single-stage	40 (264)	7 000 (46 000)	46 000 (301 000)

Adaptation to Two-Stage Shuffling

$\mathbf{y} = k \cdot \text{Shuffle}(\mathbf{y}') + \text{Shuffle}(\mathbf{y}")$

1. $\mathbf{z}_1 = k\mathbf{y}' + \mathbf{y}'' + (-1)^b \mathbf{s}_1 \mathbf{c} \approx k\mathbf{y}'$
   - match $\mathbf{z}_1$ and $k\mathbf{y}'$
2. $z_i - ky' = y'' + (-1)^b \langle \mathbf{s}_1, \mathbf{c}_i \rangle \approx y''$
   - match $z_1 - ky'$ and $\mathbf{y}''$





Results on Two-Stage Shuffling

- Number of required signatures increases drastically
  - need to match twice, lower difference of std. dev.
- Small difference between A1 and A2
  - ”matcheable” samples are in the tail, where A2 can detect them

	A1	A2	A3
no shuffling	1	4 400 (29 000)	36 000 (239 000)
single-stage	40 (264)	7 000 (46 000)	46 000 (301 000)
two-stage	260 000 (1 550 000)	285 000 (1 880 000)	575 000 (3 800 000)

Conclusion

- Shuffling once is pointless
- Shuffling twice increases signature requirements drastically
  - effective countermeasure, but still circumventable
  - different splittings and more stages might be more effective
- Generic analysis with simplifications
  - no leakage from shuffling as such, from PRNG, from additions etc.
  - further reduces signature count

Analyzing the Shuffling Side-Channel Countermeasure for Lattice-Based Signatures

Peter Pessl
IAIK, Graz University of Technology, Austria

Indocrypt 2016, December 12

Bibliography I

[DDLL13] Léo Ducas, Alain Durmus, Tancrède Lepoint, and Vadim Lyubashevsky. Lattice Signatures and Bimodal Gaussians. In Ran Canetti and Juan A. Garay, editors, *CRYPTO 2013*, volume 8042 of *LNCS*, pages 40–56. Springer, 2013.

[GBHLY16] Leon Groot Bruinderink, Andreas Hülsing, Tanja Lange, and Yuval Yarom. Flush, Gauss, and Reload - A Cache Attack on the BLISS Lattice-Based Signature Scheme. In Benedikt Gierlichs and Axel Y. Poschmann, editors, *CHES 2016*, volume 9813 of *LNCS*, pages 323–345. Springer, 2016. full version available at `http://eprint.iacr.org/2016/300`.

[PDG14] Thomas Pöppelmann, Léo Ducas, and Tim Güneysu. Enhanced Lattice-Based Signatures on Reconfigurable Hardware. In Lejla Batina and Matthew Robshaw, editors, *CHES 2014*, volume 8731 of *LNCS*, pages 353–370. Springer, 2014. VHDL source code available at `http://sha.rub.de/research/projects/lattice`.

[RRVV14] Sujoy Sinha Roy, Oscar Reparaz, Frederik Vercauteren, and Ingrid Verbauwhede. Compact and Side Channel Secure Discrete Gaussian Sampling. Cryptology ePrint Archive, Report 2014/591, 2014. `http://eprint.iacr.org/2014/591`.

[Saa16] Markku-Juhani O. Saarinen. Arithmetic Coding and Blinding Countermeasures for Lattice Signatures: Engineering a Side-Channel Resistant Post-Quantum Signature Scheme with Compact Signatures. Cryptology ePrint Archive, Report 2016/276, 2016. `http://eprint.iacr.org/2016/276` Note: to appear in Journal of Cryptographic Engineering.