Saber: Status update

A. Basso[1] J. P. D'Anvers[2] A. Karmakar[2] J. M. Bermudo Mera[2]
S. S. Roy[3] M. Van Beirendonck[2] F. Vercauteren[2]

[1] University of Birmingham, [2] ESAT-COSIC - KU Leuven, [3] IAIK - Graz University of Technology

Saber - Mod-LWR

Mod-LWE	Mod-LWR
$\left(\boldsymbol{a}, b = \boldsymbol{a}^T \boldsymbol{s} + e\right) \in R_p^{l\times 1} \times R_p$	$\left(\boldsymbol{a}, b = \left\lfloor \frac{p}{q}(\boldsymbol{a}^T \boldsymbol{s}) \right\rceil\right) \in R_q^{l\times 1} \times R_p$
$e \leftarrow \chi(R_p)$ small error	q/p determines inherent noise

- No error generation required
- Public key and ciphertext compression
- Saber parameters: same as in Round 1
  - Fixed ring $R_q = \mathbb{Z}_q[x]/(x^{256}+1)$, power-of-two moduli $q = 2^{13}$, $p = 2^{10}$
  - Modules of rank l, with $l = 2, 3, 4$
  - Secrets sampled from binomial β_μ with $\mu = 5, 4, 3$ (values in $[-\mu, \mu]$)

Saber - Specification

Alice		Bob
$\boldsymbol{A} \leftarrow \mathcal{U}(R_q^{l\times l})$		
$\boldsymbol{s} \leftarrow \text{small}(R_q^{l\times 1})$		
$\boldsymbol{b} = (\boldsymbol{A}\cdot\boldsymbol{s} + \boldsymbol{h}) \gg \log_2(\frac{q}{p})$	$\xrightarrow{\boldsymbol{b},\boldsymbol{A}}$	$\boldsymbol{s}' \leftarrow \text{small}(R_q^{1\times l})$
		$\boldsymbol{b}'^T = (\boldsymbol{A}^T\cdot\boldsymbol{s}' + \boldsymbol{h}) \gg \log_2(\frac{q}{p})$
$v = \boldsymbol{b}'\cdot\boldsymbol{s}$	$\xleftarrow{\boldsymbol{b}', v'}$	$v'^T = (\boldsymbol{b}^T\cdot\boldsymbol{s}' + h_1 + \frac{p}{2}m) \gg \log_2(\frac{p}{T})$
$m' = \lfloor \frac{2}{p}(v' - \frac{p}{T}v) \rceil$		

- Mod-LWR used twice
- Equivalent of standard Regev-type LWE encryption

Saber - Parameters

Common parameters: $q = 2^{13}$, $p = 2^{10}$, $f(x) = x^{256} + 1$

Security Category	Failure Probability	Classical Core-SVP	Quantum Core-SVP	pk (B)	sk (B)	ct (B)
LightSaber-KEM: $l = 2$, $T = 2^3$, $\mu = 5$						
1	2^{-120}	2^{118}	2^{107}	672	1568 (992)	736
Saber-KEM: $l = 3$, $T = 2^4$, $\mu = 4$						
3	2^{-136}	2^{189}	2^{172}	992	2304 (1440)	1088
FireSaber-KEM: $l = 4$, $T = 2^6$, $\mu = 3$						
5	2^{-165}	2^{260}	2^{236}	1312	3040 (1760)	1472

Saber - Parameter Choices

- **Simplicity**: moduli $T|p|q$ are powers of 2
  - ⊕ all security levels $p = 2^{10}$ and $q = 2^{13}$
  - ⊕ easy uniform sampling
  - ⊕ no modular arithmetic, no real rounding
  - no native NTT for fast multiplication
    - working modulo larger prime allows NTT
- **Modular**: Only one polynomial ring $R_q = \mathbb{Z}_q[x]/(x^{256} + 1)$ with $q = 2^{13}$
- **Flexibility**: Rank of module $2, 3, 4$ depending on security level

Saber - Security

- Parameters same since original submission
- Security estimates corrected and verified by 3 independent teams:
  - Original LWE estimator (Albrecht et. al)
  - Leaky LWE estimator (Léo Ducas)
  - Script by Dan Bernstein
- Saber ciphertexts uniformly random bytes due to power of 2
- Damien Stehlé: Security of Saber can be based on **Search** Mod-LWR (not just decision)
  - Core idea: prove OW-CPA instead of IND-CPA
  - Proof technique: Section 5 of J. Devevey, A. Sakzad, D. Stehlé, R. Steinfeld: **On the Integer Polynomial Learning with Errors Problem**. Public Key Cryptography (1) 2021: 184-214

Saber - Side Channel Security

- No error sampling required vs. LWE based schemes
- Implies less pseudo-random bits and thus less hash calls
  - Saber: 4/5/5 Keccak-f calls vs. Kyber: 7/7/9 Keccak-f calls
- No rejection sampling: less randomness, easier to mask
  - Masked randomness sampling: Kyber overhead of 17.5 vs. Saber
- Masked implementations: B2A and A2B conversions more efficient for power of two moduli $q = 2^k$ than for prime q
  - B2A: Kyber overhead of 7 vs. Saber

Saber - Multiplications

- All multiplications in Saber are uniform random $\times$ small element from β_μ
- Bounds on coefficients of product is $256 \cdot q \cdot \mu$ instead of $256 \cdot q^2$
- Flexibility: schoolbook / Karatsuba / Toom-Cook / NTT-based
- NTT-based multiplication: can choose smaller NTT-friendly prime
  - C.-M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C.-J. Shih, B.-Y. Yang. **NTT Multiplication for NTT-unfriendly Rings.**
- Good for use on large-integer arithmetic co-processor
  - B. Wang, X. Gu, Y. Yang: **Saber on ESP32**. ACNS (1) 2020: 421-440
  - J. W. Bos, J, Renes, C. van Vredendaal: **Polynomial Multiplication with Contemporary Co-Processors: Beyond Kronecker, Schönhage-Strassen & Nussbaumer.** IACR Cryptol. ePrint Arch. 2020: 1303 (2020)
- Open problem: can this be exploited in masked implementations?

NTT-based Saber

C.-M. M. Chung, V. Hwang, M. J. Kannwischer, G. Seiler, C.-J. Shih, B.-Y. Yang.
NTT Multiplication for NTT-unfriendly Rings.

- Use larger NTT-friendly prime or a pair of two smaller NTT-friendly primes
- Negacyclic transformation to compute product modulo $x^{256} + 1$
- Matrix-vector and inner-product allow to save on inverse NTT's

	Cortex-M4 (E/D)		AVX2 (E/D)	
	Toom-Cook	NTT	Toom-Cook	NTT
LigthSaber	653k / 678k	513k / 498k	75k / 70k	72k / 64k
Saber	1103k / 1127k	864k / 835k	125k / 118k	118k / 107k
FireSaber	1642k / 1679k	1255k / 1227k	184k / 174k	172k / 160k

Saber in pqm4

- Significantly reduced stack usage in pqm4 starting from NTT-based Saber

	Cortex-M4 (E/D)	
	cycles	bytes
LigthSaber	485k / 460k	5,156/5,172
Saber	828k / 786k	6,180/6,196
FireSaber	1214k / 1167k	7,204/7,220
Kyber-512	556k/516k	2,308/2,324
Kyber-768	907k/848k	2,780/2,804
Kyber-1024	1383k/1304k	3,292/3,324

Masked Saber in HW/SW

Algorithm	Device	Decapsulation unmasked	Decapsulation masked
Saber	ARM M4	$1,123,280$	$2,833,348$ ($\times 2.52$)
Kyber*	ARM M4	$847,584$	$3,596,193$ ($\times 4.24$)
Saber	RISC-V	$347,323$	$914,925$ ($\times 2.63$)
Kyber	RISC-V	$338,746$	$1,402,650$ ($\times 4.14$)

M. Van Beirendonck, J.-P. D'anvers, A. Karmakar, J. Balasch, and I. Verbauwhede. **A Side-Channel-Resistant Implementation of Saber.**
T. Fritzmann, M. Van Beirendonck, D. B. Roy, P. Karl, T. Schamberger, I. Verbauwhede, and G. Sigl. **Masked Accelerators and Instruction Set Extensions for Post-Quantum Cryptography.**
* D. Heinz, P. Schwabe, M. J. Kannwischer, G. Land, D. Sprenkels, T. Pöppelmann. **First-Order Masked Kyber on ARM Cortex-M4.**

Masked Saber on FPGA

Operation	Cycles		Overhead
	Unmasked	Masked	
Polynomial arithmetic (256 DSPs)	4,484	8,968	2.00×
SHA-256	303	1,344	4.44×
SHA-512	62	124	2.00×
Binomial Sampler	176	339	1.92×
A2A			
⌊ Rounding and Scaling	339	682	2.01×
⌊ Ciphertext compression	107	561	5.24×
⌊ Message extraction	167	985	5.90×
Other operations	993	1,986	2.00×
Total (256 DSPs)	8,034	**16,392**	2.04×

A. Basso, L. Prokop, S. S. Roy. **A side-channel resistant hardware implementation of Saber.**

Saber on ASIC

- Tsingua university: Y. Zhu, M. Zhu, B. Yang, W. Zhu, C. Deng, C. Chen, S. Wei and L. Liu;
  LWRpro: An Energy-Efficient Configurable Crypto-Processor for Module-LWR
  - 40nm, 400MHz, 1456/1701 E/D cycles, 275k Enc/sec, 0.15μJ/op, 0.38 mm^2
  - Very energy efficient, only CPA version
- TalTech: Malik Imran, Felipe Almeida, Samuel Pagliarini (EU H2020 952252)
  - 65nm, 1GHz, 6880/8630 E/D cycles, 145k Enc/sec, 4.2μJ/op, 0.49 mm^2
  - Full CCA version, no masking
- Purdue/Intel: A. Ghosh, S. Shreyas, D. Das (Purdue) and S. Ghosh (Intel)
  - 65nm, 200MHz, 18705/23390 E/D cycles, 10.6k Enc/sec, 1.12 μJ/op, 0.74 mm^2
  - Full CCA version, circuit level side-channel protection, no masking

Saber is . . .

- **Secure**
  - Security can be based on Search Mod-LWR
  - Security levels confirmed by 3 teams
  - Stable: parameters same as in Round 1
- **Easy to implement** (less footguns than other schemes)
  - No modular reduction
  - No rejection sampling
  - Modular: only arithmetic in one fixed R_q
- Efficient to **protect against side-channels** (see presentation Michiel)
  - Power-of-two moduli
  - Less hashing (due to rounding)
  - For higher order, difference gets larger