BIKE

4th NIST PQC Standardization Workshop

November 30th, 2022



Nicolas Aragon, University of Limoges, France Paulo S. L. M. Barreto, University of Washington Tacoma, USA Slim Bettaieb, Worldline, France Loïc Bidoux, Worldline, France Olivier Blazy, University of Limoges, France Jean-Christophe Deneuville, ENAC, Federal University of Toulouse, France Philippe Gaborit, University of Limoges, France Santosh Ghosh, Intel, USA Shay Gueron, University of Haifa, and Amazon Web Services, Israel & USA

Tim Güneysu, Ruhr-Universität Bochum & DFKI, Germany Carlos Aguilar Melchor, University of Toulouse, France <u>Rafael Misoczki</u>, Google, USA Edoardo Persichetti, Florida Atlantic University, USA Jan Richter-Brockmann, Ruhr-Universität Bochum, Germany Nicolas Sendrier, INRIA, France Jean-Pierre Tillich, INRIA, France Valentin Vasseur, INRIA, France Gilles Zémor, IMB, University of Bordeaux, France

https://bikesuite.org

Agenda

- BIKE 4th-Round
- A data-oblivious rejection sampling algorithm
- Updated performance numbers



BIKE Recap

- Niederreiter-based KEM instantiated with QC-MDPC codes
- Leverage Fujisaki-Okamoto Transform¹
- State-of-the-art QC-MDPC Decoding Failure Rate analysis²
- Black-Gray-Flip Decoder implemented in constant time³

1: For a detailed analysis of the FO transform applied to BIKE, see: [DGK+'21].

2: For a comprehensive discussion on Decoding Failure Rate of BIKE decoders, see [Vas'21].

3: For BGF decoder implementation strategies, see [DGK'20].



What have we changed?

- A minor refinement
 - We have changed how we implement the hash function H to be data-oblivious
- Otherwise, BIKE remains very stable
 - No changes to the proposed parameters
 - No changes to the overall algorithmic specification
 - No changes to the decoding strategy⁴



BIKE - 4th Round Spec

$\mathbf{KeyGen}: () \mapsto (h_0, h_1, \sigma), h$	Encaps : $h \mapsto K, c$		
Output: $(h_0, h_1, \sigma) \in \mathcal{H}_w \times \mathcal{M}, h \in \mathcal{R}$	Input: $h \in \mathcal{R}$		
1: $(h_0, h_1) \xleftarrow{\mathcal{D}} \mathcal{H}_w \qquad \triangleright^{(1)}$	Output: $K \in \mathcal{K}, c \in \mathcal{R} \times \mathcal{M}$		
2: $h \leftarrow h_1 h_0^{-1}$	1: $m \stackrel{s}{\leftarrow} \mathcal{M}$		
3: $\sigma \stackrel{\hspace{0.1em}\scriptscriptstyle\$}{\leftarrow} \mathcal{M}$	2: $(e_0, e_1) \leftarrow \mathbf{H}(m)$		
	3: $c \leftarrow (e_0 + e_1h, m \oplus \mathbf{L}(e_0, e_1))$		
	4: $K \leftarrow \mathbf{K}(m, c)$		
Decaps : $(h_0, h_1, \sigma), c \mapsto K$			
Input: $((h_0, h_1), \sigma) \in \mathcal{H}_w \times \mathcal{M}, c = (c_0, c_1) \in \mathcal{R} \times \mathcal{M}$			
Output: $K \in \mathcal{K}$			
1: $e' \leftarrow \texttt{decoder}(c_0h_0,h_0,h_1)$	$ hinspace e' \in \mathcal{R}^2 \cup \{ot\}$		
2: $m' \leftarrow c_1 \oplus \mathbf{L}(e')$	\triangleright with the convention $\perp = (0,0)$		
3: if $e' = \mathbf{H}(m')$ then $K \leftarrow \mathbf{K}(m', c)$ else $K \leftarrow \mathbf{K}(\sigma, c)$			

⁽¹⁾: \mathcal{D} a distribution stemming from Algorithm 3, the constant weight sampler of §2.4

NOTATION		Functions
\mathbb{F}_2 :	Binary finite field.	
\mathcal{R} :	Cyclic polynomial ring $\mathbb{F}_2[X]/(X^r-1)$.	• $\mathbf{H}: \mathcal{M} \to \mathcal{E}$
\mathcal{H}_w :	Private key space $\{(h_0, h_1) \in \mathcal{R}^2 \mid h_0 = h_1 = w/2\}$	
\mathcal{E}_t :	Error space $\{(e_0, e_1) \in \mathcal{R}^2 \mid e_0 + e_1 = t\}$	• $\mathbf{K} : \mathcal{M} \times \mathcal{R} \times \mathcal{M} \to \mathcal{K}.$
g :	Hamming weight of a binary polynomial $g \in \mathcal{R}$.	• $\mathbf{L}: \mathcal{R}^2 \to \mathcal{M}$
u U:	Variable u is sampled uniformly at random from the set U .	
⊕:	exclusive or of two bits, componentwise with vectors	

Parameters

r : block lengthw : row weightt : error weight $\ell : shared secret size$ $<math>\mathcal{M}$: message space in $\{0, 1\}^{\ell}$ \mathcal{K} : key space in $\{0, 1\}^{\ell}$



BIKE - 4th Round Spec



⁽¹⁾: \mathcal{D} a distribution stemming from Algorithm 3, the constant weight sampler of §2.4

NOTATION	r	Functions
\mathbb{F}_2 :	Binary finite field.	
\mathcal{R} :	Cyclic polynomial ring $\mathbb{F}_2[X]/(X^r-1)$.	• $\mathbf{H} \cdot \mathbf{M} \to \mathcal{E}$
\mathcal{H}_w :	Private key space $\{(h_0, h_1) \in \mathcal{R}^2 \mid h_0 = h_1 = w/2\}$	
\mathcal{E}_t :	Error space $\{(e_0, e_1) \in \mathcal{R}^2 \mid e_0 + e_1 = t\}$	• $\mathbf{K} : \mathcal{M} \times \mathcal{R} \times$
g :	Hamming weight of a binary polynomial $g \in \mathcal{R}$.	• $\mathbf{L}: \mathcal{R}^2 \to \mathcal{M}$
$u \stackrel{\hspace{0.1em}{\leftarrow}}{\leftarrow} U$:	Variable u is sampled uniformly at random from the set U .	
\oplus :	exclusive or of two bits, componentwise with vectors	

 $\mathcal{R} \times \mathcal{M} \to \mathcal{K}.$



Constant-Time Rejection Sampling

- [GHJ+'21]
 - A variable-time rejection sampling algorithm used to produce fixed-weight vectors can lead to side-channel attacks in both HQC and BIKE⁵
 - In BIKE, the time variability can be used to distinguish decoding failures
 - Select m such as H(m) has a very distinct number of rejections (vs. the average case)
 - In Decaps, if time greatly differs from the one expected for the right H(m) -> decoding failure
 - Once enough decoding failures are identified, [GJS'18] attack can be be applied



⁵: Note that this attack would be restricted to the static-key setting (IND-CCA security)

Data-Oblivious Rejection Sampling

Old Variable-Time Rejection-Sampling Algorithm (BIKE Round 3)

Algorithm 3 WSHAKE256-PRF(seed, wt, len) Require: seed, wt (32 bits), len **Ensure:** A list (wlist) of wt bit-positions in $[0, \ldots, len - 1]$. 1: wlist = ϕ ; ctr = 0; i = 02: s = SHAKE256-Stream(seed, ∞) $\triangleright \infty$ denotes "sufficiently large" 3: $mask = (2^{ceil(\log_2 len)} - 1)$ 4: while ctr < wt dopos = s[32(i+1) - 1:32i] & mask▷ & denotes bitwise AND 5: if $((pos < len) AND (pos \notin wlist))$ then 6: wlist = wlist \cup {pos}; ctr = ctr + 1; 7: i = i + 18: return wlist

New Data-Oblivious Rejection-Sampling Algorithm [Sen'21]

Algorithm 3 WSHAKE256-PRF(seed, len, v	wt)
Require: seed (32 bytes), len, wt	
Ensure: A list (wlist) of wt distinct elemen	ts in $\{0,, len - 1\}$.
1: wlist \leftarrow ()	\triangleright empty list
2: $s_0, \ldots, s_{wt-1} \leftarrow SHAKE256$ -Stream(see	$1,32 \cdot wt)$
▷ parse as a sequ	nence of wt non negative 32-bits integers
3: for $i = (wt - 1), \dots, 1, 0$ do	$\triangleright i$ decreasing from wt -1 to 0
4: $\operatorname{pos} \leftarrow i + \lfloor (\operatorname{len} - i) s_i / 2^{32} \rfloor$	
5: wlist \leftarrow wlist, (pos \in wlist) ? i : pos	
6: return wlist	

- The new rejection-sampling algorithm is:
 - Data-oblivious (it doesn't depend on the input)
 - It introduces a small bias that does not impact security⁶
 - It can be used in KeyGen step 1 as well (to minimize binary size)

⁶: A comprehensive analysis supporting this claim is presented in Sections 3 and 4 of [Sen'21]

Updated Performance Numbers

- BIKE Additional Implementation
 - Developed by Nir Drucker, Shay Gueron, and Dusan Kostic
 - Available at: <u>https://github.com/awslabs/bike-kem</u>

	Level 1	Level 3
KeyGen	602	1,825
Encaps	130	287
Decaps	1,185	3,956

Intel(R) Xeon(R) Platinum 8175M CPU @ 2.50GHz

No support to vector-PCLMUL

	Level 1	Level 3
KeyGen	371	1,064
Encaps	96	205
Decaps	1,194	3,532

Supports vector-PCLMUL

Intel(R) Xeon(R) Platinum 8375C CPU @ 2.90GHz



Measurements given in kilo cycles

Conclusion

- The BIKE team would like to thank the community for the significant amount of work produced on BIKE
- The [GHJ+'21] attack is a clever idea that would affect a variable-time implementation of H in the static-key setup
 - The fix to this side-channel attack is simple and doesn't incur in any significant performance penalty
 - It also allowed us to converge to a single rejection sampling algorithm for all KeyGen, Encaps and Decaps

Thank you



https://bikesuite.org

References

- [DGK+'21]: Drucker, N., Gueron, S., Kostic, D., & Persichetti, E. (2021). On the applicability of the Fujisaki-Okamoto transformation to the BIKE KEM. Intl. Journal of Computer Mathematics: Computer Systems Theory.
- [GHJ+'21]: Qian Guo, Clemens Hlauschek, Thomas Johansson, Norman Lahr, Alexander Nilsson, and Robin Leander Schröder. Don't reject this: Key-recovery timing attacks due to rejection-sampling in HQC and BIKE. IACR Trans. Cryptogr. Hardw. Embed. Syst., 2022(3):223–263, 2022
- [GJS'16]: Qian Guo, Thomas Johansson, and Paul Stankovski. A key recovery attack on MDPC with CCA security using decoding errors. In Jung Hee Cheon and Tsuyoshi Takagi, editors, ASIACRYPT 2016, Part I, volume 10031 of LNCS, pages 789–815.
 Springer, Heidelberg, December 2016.
- · [PT'22]: Paiva, T.B. and Terada, R., 2022. Faster Constant-Time Decoder for MDPC Codes and Applications to BIKE KEM. IACR Transactions on Cryptographic Hardware and Embedded Systems, pp.110-134.
- [Sen'21]: Nicolas Sendrier. Secure sampling of constant-weight words Application to BIKE. Cryptology ePrint Archive, Report 2021/1631, December 2021.
- [Vas'21]: Valentin Vasseur's PhD thesis "Post-quantum cryptography: study on the decoding of QC-MDPC codes", 2021, available at: <u>https://who.rocq.inria.fr/Valentin.Vasseur/phd-defence/</u>
- . [DGK'20]: N. Drucker, S., Gueron, D. Kostic. "QC-MDPC Decoders with Several Shades of Gray". PQCrypto 2020: 35-50

Questions?

https://bikesuite.org

