EHTv3 and EHTv4

On-ramp Digital Signature Candidates for the NIST PQC Standardization Process

Martin Feussner and Igor Semaev

UNIVERSITY OF BERGEN

Contents

What is it?

- Initiative for developing new KEM and DSA standards
- Why? Quantum threat $\rightarrow$ RSA and ECC are broken

- Open call for proposals (turned into competition)
- Round 1: December 2017 - 69 Candidates
- Round 2: January 2019 - 26 Candidates [17 KEMs, 9 DSAs]
- Round 3: July 2020 - 7 Finalists [4 KEMs, 3 DSAs]
  - 8 Alternates [5 KEMs, 3 DSAs]
- Round 4: July 2022 - 4 Candidates [KEMs]

- Round 1 Additional Signatures: July 2023 - **40 Candidates**

What has come out of it so far?

Selected algorithms becoming FIPS (Federal Information Processing Standards):

- FIPS203: ML-KEM: CRYSTALS-KYBER
- FIPS204: ML-DSA: CRYSTALS-DILITHIUM
- FIPS205: SLH-DSA: SPHINCS+

FALCON (FN-DSA) will also have a FIPS (but later... too complicated)

pqc-forum (Google Group) with a tsunami of information (discussions, attacks, developments, ...)

What is EHTv3 and EHTv4?

(formerly) q-ary lattice based digital signature schemes that were inspired by the public key crypto-system EHT [Budroni and Semaev, 2021].

Prior versions, EHTv1 [Semaev, 2022] and EHTv2 [Semaev, 2022], in ePrint and NISK respectively.

EHTv3 $\rightarrow \mathbb{Z}_q$
EHTv4 $\rightarrow$ finite group G ring over $\mathbb{Z}_q \rightarrow G_q$

The schemes are easy to understand and implement and the parameters can be easily modified to vary security.

*The slides that follow mostly refer to EHTv3

Some Definitions

- L the lattice generated by the columns of matrix $A \in \mathbb{Z}_q^{m \times n}$ mod q
- $y \in L$ satisfies $y \equiv Ax \bmod q$ for some $x \in \mathbb{Z}^n$
- $e = (e_1, ..., e_m) \in \mathbb{Z}^m$ is the error vector. We say that $max_l(e) \leq s$ if at least l entries of e are at most s in absolute value
- $f(e) = \sum_{a=0}^{q-1} 2^{f_a}$, where f_a is the number of entries in e equal to a mod q
- $h = \text{HASH}(M) \in \mathbb{Z}_q^m$

Design Rationale

$A \in \mathbb{Z}_q^{m \times n}$ is the public matrix.
$e \in \mathbb{Z}^m$ and $h \in \mathbb{Z}_q^m$

The signature is $x \in \mathbb{Z}_q^n$
Such that $h \equiv Ax + e \bmod q$
Where e is constrained: $\max_l(e) \leq s$ and $s_1 \leq f(e) \leq s_2$.
Harder than CVP $\rightarrow$ given h, find $y \in L$ such that $e = h - y$.

We provide A through some secure trapdoor construction.
Consider 3 secret matrices:
$C \in \mathbb{Z}^{m \times kn}$
$T \in \mathbb{Z}_q^{kn \times n}$
$B \in \mathbb{Z}_q^{n \times n}$

The public key matrix is computed as:
$A \equiv CTB^{-1} \bmod q$

Trapdoor Matrix - $C \in \mathbb{Z}^{m \times kn}$

- 1-norm of each row of C is λ with non-zero entries ± 1
- Let $d = kn - m$, the matrix is constructed as:
  $C = (C_1 \in \mathbb{Z}^{m \times m} | C_2 \in \mathbb{Z}^{d \times m})$
- 1-norm of each row of C_1 is $\tau < \lambda$ and C_1 is invertible mod q.
- Sparse matrix: $\lambda, \tau << kn$

Trapdoor Matrix - $T \in \mathbb{Z}_q^{kn \times n}$

Lower triangular matrix with k-tuple $[t_1, ...t_k]$ populating its diagonal.
Given any tuple of integers $[b_1, ...b_k]$ there exists an integer u such that:

$$\begin{aligned} |(b_1 - t_1 u) \bmod q| &\leq c, \\ |(b_2 - t_2 u) \bmod q| &\leq c, \\ &\vdots \\ |(b_k - t_k u) \bmod q| &\leq c \end{aligned}$$

Entries below the diagonal are secret and chosen at random. Below is an example of the structure for $k = 2$ with random entries denoted by $*$:

$$T = \begin{pmatrix} t_1 & 0 & \cdots & 0 \\ t_2 & 0 & \cdots & 0 \\ * & t_1 & \cdots & 0 \\ * & t_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \cdots & t_1 \\ * & * & \cdots & t_2 \end{pmatrix}$$

Trapdoor Matrix - $B \in \mathbb{Z}_q^{n \times n}$

- An arbitrary square matrix invertible mod q.

Core Theorem

Theorem: For every $a \in \mathbb{Z}_q^{kn}$ there exists $y \in \mathbb{Z}_q^n$ and $z \in \mathbb{Z}^{kn}$ such that $\max(z) \leq c$ and $a \equiv Ty + z \bmod q$

The proof to the above theorem provides an efficient algorithm to compute $y \in \mathbb{Z}_q^n$ and $z \in \mathbb{Z}^{kn}$ such that $\max(z) \leq c$ and $a \equiv Ty + z \bmod q$.

With this we provide that:
$h \equiv Ca$
$e \equiv Cz$

Signature:
$x \equiv By$

Verification:
$e \equiv h - Ax \bmod q$
$\max_l(e) \leq s$ and $s_1 \leq f(e) \leq s_2$

Verification Proof

$A \equiv CTB^{-1} \rightarrow AB \equiv CT$
$a \equiv Ty + z \rightarrow z \equiv a - Ty$
$h \equiv Ca$
$x \equiv By$

Proof:
$e \equiv h - Ax \equiv Cz$

$e \equiv (Ca) - A(By) \equiv Ca - ABy \equiv Ca - CTy \equiv C(a - Ty) \equiv Cz$

Simplified Protocol - Key Generation

Protocol: EHTv3 Keypair Generation

Input: secure entropy source
Output: pk - public key, sk - secret key

1: Initialise RNG with the seed and store in sk.
2: Generate C as described.
3: Generate T as described.
4: Generate B^{-1} as described.
5: Compute $A \equiv CTB^{-1} \bmod q$ and efficiently store it in pk.

Simplfied Protocol - Signature Generation

Protocol: EHTv3 Signature Generation

Input: M, sk, secure entropy source
Output: sm - signed message

1: Initialise RNG with seed in sk.
2: Generate C, T and B.
3: Compute $h = \text{HASH}(M)$ and store M in sm.
4: Initialise RNG with secure entropy source.
5: Fix part of z from centered multinomial distribution.
6: Compute y and z by the Core Theorem.
7: Compute $e = Cz$.
8: If $\max_l(e) \leq s$ and $s_1 \leq f(e) \leq s_2$, then the signature is $x \equiv By$ which is stored efficiently in sm; otherwise, repeat from step 5.

Simplified Protocol - Signature Verification

Protocol: EHTv3 Signature Verification

Input: pk, sm
Output: true if valid, false otherwise

1: Recover A from pk and M and x from sm.
2: Compute $h = \text{HASH}(M)$
3: Compute $e \equiv h - Ax \bmod q$ and center entries about 0.
4: If $\max_l(e) \leq s$ and $s_1 \leq f(e) \leq s_2$, then accept the signature; otherwise, reject.

Protocols - EHTv4

In EHTv4 the protocols are almost identical to that of EHTv3.

The main change is that EHTv4 operates in some finite group G ring over $\mathbb{Z}_q$. Basically changing from $\mathbb{Z}_q$ to G_q or $\mathbb{Z}_q[G]$.

The group has order r, $G = \{\alpha_0 = 1, \alpha_1, \ldots, \alpha_{r-1}\}$, and the set G_q contains all formal sums $\alpha = \sum_{i=0}^{r-1} a_i \alpha_i$ where $a_i \in \mathbb{Z}_q$.

EHTv4 can be easily transformed to EHTv3 with more structure.

EHTv3 Optimizations/Considerations

- Compiler flag `-O3`
- RNG buffer
- Pre-computed inverses of elements in $\mathbb{Z}_q$
- Characteristic polynomial of C_1, denoted as p_{C_1}. Cayley–Hamilton theorem.
- Construction of B and B^{-1}.
- Pre-computed operation tables - situational use and small $q = 47$
- Choice for storage of A in pk
- Distinguisher function f

EHTv4 Optimizations/Considerations

- Compiler flag `-O3`
- RNG buffer
- v4-I1: $G = GL(3,2)$ and $r = 168$. Composition lookup with binary value of length $2^{3\times 3} = 512 \rightarrow$ at index stores index from the ordered list G.
- v4-I5: $G = A_6$ and $r = 360$. Composition lookup using lexicographic ordering of length $7! = 720$ *.
- Inverses of intermediate diagonal entries of C_1 (G_q elements)
- Block construction for B and B^{-1}.

Parameters

The parameters are provided below for the various security levels: NIST Level 1 (L1), NIST Level 3 (L3) and NIST Level 5 (L5).

v3	q	k	m	n	l	λ	τ	c	s	$\log_2 s_1$	$\log_2 s_2$
L1	47	2	460	242	451	9	4	3	13	35.807	40
L3	47	2	696	367	684	9	4	3	13	52.807	58
L5	47	2	940	495	921	9	4	3	13	69.807	76

v4	q	k	m	n	G	r	l	λ	c	s	s_1	s_2
L1	439	2	3	2	GL(3,2)	168	492	54	10	100	5203	7913
L5	839	2	3	2	A_6	360	1076	100	14	244	19503	30043

EHTv3 uses the 2-tuple $[1, 7]$ to construct T. Same for all security levels in v3 due to same q, k, c.

EHTv4-L1 uses $[1, 21]$ and EHTv4-L5 uses $[1, 29]$.

Sizes and Performance

Operating System: Windows10 64-bit
Processor, RAM: 12thGen Intel(R) Core(TM) i7-12800H@2.40 GHz, 16.0 GB
Iterations: 10^3 EHTv3, 10^4 EHTv4

	v3-L1	v3-L3	v3-L5	v4-L1	v4-L5
Public Key (bytes)	83490	191574	348975	1107	2623
Secret Key (bytes)	368	532	701	604	1363
Signature (bytes)	169	255	344	369	875
Keypair Generation (ms)	149	503	1303	16.5	158
Signature Generation (ms)	21.6	90.8	187	5.03	25.6
Signature Verification (ms)	0.76	1.68	3.02	3.72	21.9
Trials for signature	3.46	4.43	3.17	1.89	2.04

Sizes vs Others Schemes (L1)

"... we concluded that early adoption of post-quantum signatures on the Internet would likely be more successful if those six signatures and two public keys would fit in 9KB." - [Bas Westerbaan, Cloudflare]

Scheme	Category	pk size (B)	sig size (B)	2*pk + 6*sig (KB)
SQIsign	Isogenies	64	177	1.19
RSA	Factoring	272	256	2.08
$SNOVA_1$	Multivariate	1016	248	3.52
MAYO	Multivariate	1168	321	4.26
EHTv4	Harder than lattices	1110	369	4.43
HAWK	Lattices	1024	555	5.38
$SNOVA_2$	Multivariate	2320	165	5.63
Falcon	Lattices	897	666	5.79
MAYO	Multivariate	5488	180	12.06
SLH-DSA $(SPHINCS+)_1$	Symmetric	32	7856	47.20
SLH-DSA $(SPHINCS+)_2$	Symmetric	32	17088	102.59
EHTv3	Harder than lattices	83500	169	168.01

Cycles vs Others Schemes (L1)

Scheme	Verify (kilocycles)	Scheme	Sign (kilocycles)
RSA	45	HAWK	85
Falcon	81	MAYO	461
HAWK	148	Falcon	1010
MAYO	175	$SNOVA_2$	12408
EHTv3	1900	EHTv4	12575
$SNOVA_2$	3960	$SNOVA_1$	19681
SLH-DSA (SPHINCS+)$_1$	4764	RSA	27000
$SNOVA_1$	8087	EHTv3	54000
EHTv4	9300	SLH-DSA (SPHINCS+)$_2$	239794
SLH-DSA (SPHINCS+)$_2$	12910	SLH-DSA (SPHINCS+)$_1$	4682571
SQIsign	108000	SQIsign	5669000

Attack 1

By:
Eamonn Postlethwaite and Wessel van Woerden (team HAWK)

Attack:
Given A and h, finds x_1 and e_1 such that $e_1 \equiv h - Ax_1$ and $max_l(e_1) \leq s$. Forged by solving the underlying CVP with BKZ reduction and Babai nearest plane algorithm for a lattice generated by the columns of A. They were able to break the posted 80-bit security challenge.

Countermeasure:
New generation and verification rule: $s_1 \leq f(e) \leq s_2$

Countermeasure Visual

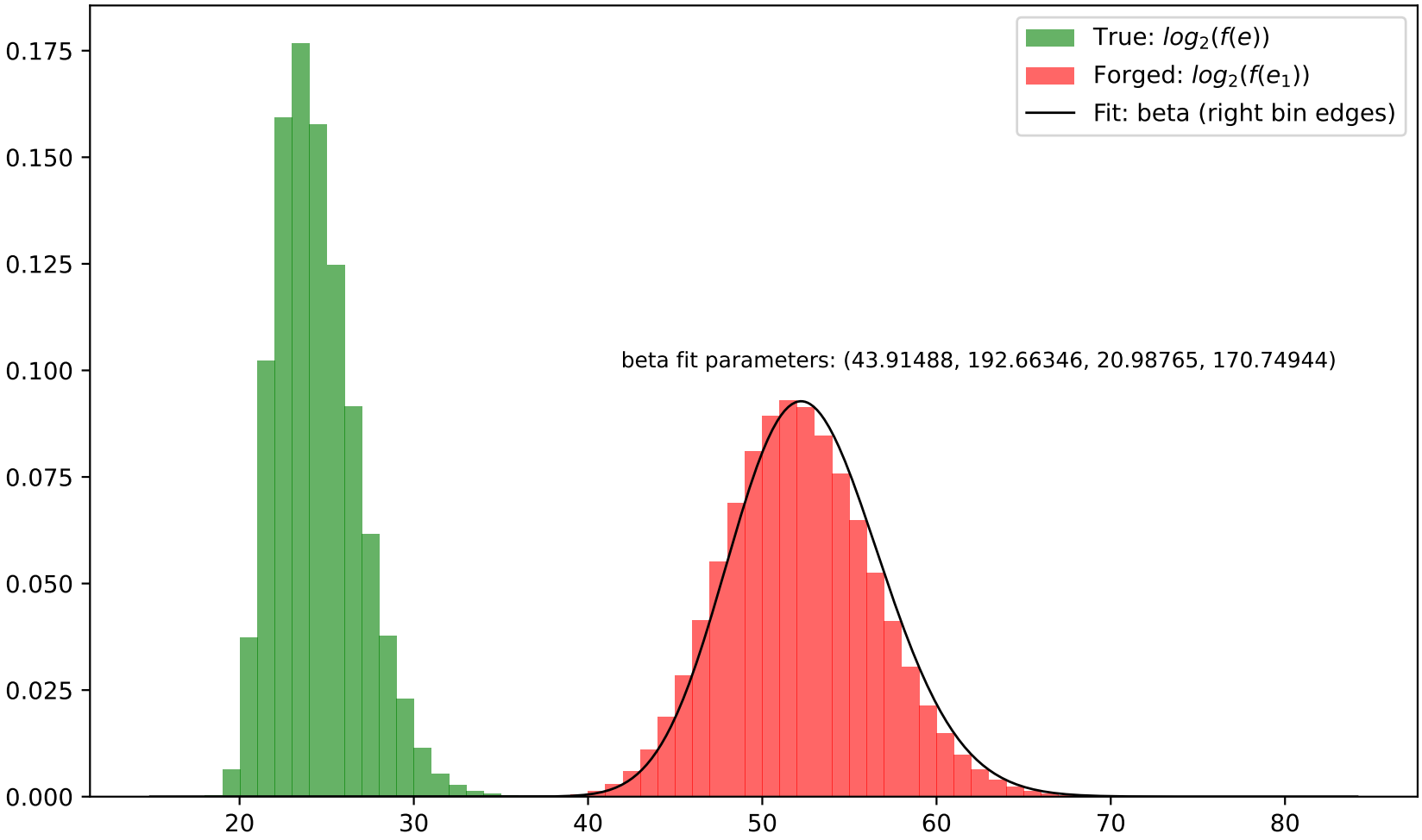

Figure: Histogram comparison of f on e and e_1

Attack 2

By:
Keegan Ryan and Adam Suhl

Attack:
Hidden Zonotope Problem (HZP) attack recovers some columns of C from observing $e = Cz$, where z is distributed uniformly. 5×10^5 signatures were enough to break EHTv3-L1 and similar was verified by us for EHTv4-L1.

Countermeasure:
We provide a large portion of z with the multinomial distribution. We also slighlty redefine the construction of C. No leakage with 5×10^6 signatures.

Next Steps?

- Work towards an optimized implementation: AVX2 and ARM Cortex M4. (**help!?**)
- Constant time implementation.
- Better benchmarking.
- More cryptanalysis to tighten parameters.
- Properly explore use cases. 8-bit implementation?

Thank you for following!

Any Questions?

References

Igor Semaev and Martin Feussner, EHTv3 and EHTv4 Updated Submission Package. Available at:
https://universityofbergen-my.sharepoint.com/:u:/g/personal/martin_feussner_uib_no/ETB7SQoM8XVGpoFtEtRLgaYBNVvKcjEo3mTZ-a0Mz1FsRA?e=Thi5gk

Igor Semaev and Martin Feussner, EHTv3 and EHTv4 Initial Submission Specifications. Available at:
https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/round-1/spec-files/EHTv3v4-spec-web.pdf

pqc-forum. Available at:
https://groups.google.com/a/list.nist.gov/g/pqc-forum

Post-Quantum signatures zoo. Available at:
https://pqshield.github.io/nist-sigs-zoo/index.html