EHTv3 and EHTv4
Lattice-Based Digital Signature Schemes

Martin Feussner

Igor Semaev

University of Bergen

Tuesday September 5, 2023

Overview

- Some similarity with the public key crypto-system EHT [Budroni, Semaev].
- Prior versions: EHTv1 [Semaev] and EHTv2 [Semaev]
- EHTv4 is very similar to EHTv3 but the arithmetic is in a finite group ring G_q over $\mathbb{Z}_q$ instead of $\mathbb{Z}_q$ itself.
- Schemes are easy to understand and implement.
- Parameters can be easily modified to increase security levels if needed.
- Hardness is based on solving some linear algebra problems (CVP, etc.)

EHTv3 Definitions

$A \rightarrow$ public key matrix

$C, T, B \rightarrow$ secret key matrices

$$A \equiv CTB^{-1} \in \mathbb{Z}_q^{m\times n}$$

$$C = \left(C_1 \in \mathbb{Z}_q^{m\times m} \middle| C_2 \in \mathbb{Z}_q^{m\times d}\right) \in \mathbb{Z}_q^{m\times kn}$$

- Is a sparse matrix where the 1-norm of each row of C and C_1 is λ and τ respectively.

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

- Is a special rectangular matrix that contains tuples $[t_{1j}, t_{2j}, \dots, t_{kj}]$ on its main diagonal $\longrightarrow$

$$T = \begin{pmatrix} t_{11} & 0 & \dots & 0 \\ t_{21} & 0 & \dots & 0 \\ t_{k1} & 0 & \dots & 0 \\ * & t_{12} & \dots & 0 \\ * & t_{22} & \dots & 0 \\ * & t_{k2} & \dots & 0 \\ * & * & \dots & t_{1n} \\ * & * & \dots & t_{2n} \\ * & * & \dots & t_{kn} \end{pmatrix}$$

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

- Is an arbitrary matrix invertible modulo q

Core Theorem

Theorem 1:

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

Because T is triangular (the trapdoor), it allows us solve systems of equations to recursively construct y and z that satisfy the above – efficiently!

$h \rightarrow$ hash of message

$a, y, z \rightarrow$ part of core theorem

$e \rightarrow$ error vector

$x \rightarrow$ signature

$h = \mathrm{HASH}(M) \in \mathbb{Z}_q^m$

$a = \left(a_1 \in \mathbb{Z}_q^m \middle| a_2 \in \mathbb{Z}_q^d\right) \in \mathbb{Z}_q^{kn}$

$y \in \mathbb{Z}_q^n$

$z \in \mathbb{Z}^{kn}$ and $\max(z) \leq c$

$e \in \mathbb{Z}^m$ and we want $\max_l(e) \leq s$

$x \in \mathbb{Z}_q^n$

v3-1:
$(m, n, l) = (460, 242, 451)$
$(q, k, \lambda, \tau, c, s) = (47,2, 9, 4, 3, 13)$

Scheme Formulations

$AB \equiv CT$

$Ca \equiv h$

$a \equiv Ty + z$

$e = Cz$

$x \equiv By$

$A \equiv CTB^{-1}$

$a_1 \equiv C_1^{-1}h - C_1^{-1}C_2a_2$

$y, z \rightarrow$ Solved by core theorem

$e \equiv Cz$

$e \equiv C(a - Ty)$

$e \equiv Ca - CTy$

$e \equiv h - ABy$

$e \equiv h - Ax$

EHTv3 Key Generation

1. Initialize RNG with some seed **[sk]**.
2. Generate C_1. Go back to 1. if C_1 is not invertible.
3. Generate B. Try and compute B^{-1}, if not invertible go back to 1.
4. Generate C_2
5. Generate T
6. Compute $\mathrm{A} \equiv CTB^{-1}$ **[pk]**.

EHTv3 Signature Generation

1. Initialize RNG with the seed **[sk].**
2. Generate C, T, B
3. Compute h using message.
4. Randomly generate a_2.
5. Compute a_1.
6. Compute y and z by Theorem 1.
7. Compute $e = Cz$. If $\max_l(e) \leq s$ is not satisfied, go back to 4.
8. Compute signature: $x \equiv By$

EHTv3 Signature Verification

1. Compute h using message.
2. Compute $e \equiv h - Ax$.
3. If $\max_l(e) \leq s$, then accept the signature, otherwise reject.

EHTv3 Improvements

1. We can check if C_1 is invertible from its characteristic polynomial (p_{C_1}) determined from its Hessenberg form which can be stored as part of the secret key. Allows us to speed up signature generation process (Cayley-Hamilton Theorem) when computing a_1 at the cost of a larger **[sk]**.
2. We can generate invertible B by construction:
   - Generate B_u (UTM) and B_l (LTM), with the fact that $B \equiv B_u B_l$.
   - Compute ${B_u}^{-1}$ and ${B_l}^{-1}$.
   - Compute $B^{-1} \equiv {B_l}^{-1} {B_u}^{-1}$

EHTv3 Key Generation

1. Initialize RNG with some seed and store in **[sk]**.
2. Generate C_1. Go back to 1. if C_1 is not invertible, otherwise store p_{C_1} in **[sk]**.
3. Generate C_2.
4. Generate T.
5. Generate B^{-1} by construction.
6. Compute $\mathrm{A} \equiv CTB^{-1}$ and store in **[pk]**.

EHTv3 Cryptanalysis

1. ***Private Key Recovery and Algebraic Attacks:***

   Analysis Focus: $CT_n \equiv C_{2n-1} + tC_{2n} \equiv AB_n$, where indices denote columns. This leads to m linear equations with $n + 2m$ unknowns, resulting in q^{n+m} potential solutions.

   Alternatively, guess n zero entries of $V = C_{2n-1} + tC_{2n}$ and solve n equations with n variables.

   Success probability: $P_{Al} = \frac{\binom{\theta m}{n}}{\binom{m}{n}} \approx 2^{-246}$ for v3-1, $\theta = (1 - (\lambda - \tau)/d)^2$

2. *Existential Forgery by Guessing:*

Given $h = \mathrm{HASH}(M)$, one may guess small values ($\leq s$) of some n entries of $e \equiv h - Ax$ and then compute x by solving a system of n linear equations modulo q. One then checks if among other $m - n$ entries of e there are at least $l - n$ entries that are $\leq s$. Let $p = \frac{2s+1}{q}$ be the probability that a random entry is at most s in absolute value. The attack success probability is:

$$P_G = \sum_{i=l-n}^{m-n} \binom{m-n}{i} p^i (1-p)^{m-n-i}$$

The attack may be optimized and result in these many operations modulo q:

$$Q = P_G^{-1}(\log_2 P_G^{-1})(m-n)/2$$

For v3-1, Q $\approx 2^{140.69}$

3. *Adaptive Forgery under Known Message Attack:*

A message M with $h = \mathrm{HASH}(M)$ may have multiple valid signatures like $x_1, x_2, \ldots$

Suppose for them we have: $h \equiv Ax_1 + e_1, h \equiv Ax_2 + e_2, \ldots$

Modify M_0's signature x_0 to get $h_0 \equiv A(x_0 + x_1 - x_2) + e_0 + e_1 - e_2$

Possible when $\max_l(e_0 + e_1 - e_2) \leq s$

Assuming entries of e are independently distributed, the probability is:

$$P_A = \sum_{i=l}^{m} \binom{m}{i} p^i (1-p)^{m-i}$$

For v3-1 this is $2^{-101.14}$. This would require a little over $2^{101.14}$ independently generated triplets e_0, e_1, e_2 for attack probability to be close to 1 which is greater than the cap for this analysis indicated by NIST: 2^{64}.

EHTv4 Definitions

Similar to EHTv3, but while EHTv3 operates in the ring $\mathbb{Z}_q$, EHTv4 operates in some finite group G over $\mathbb{Z}_q \to G_q$ or $\mathbb{Z}_q[G]$

$G = \{\alpha_0 = 1, \alpha_1, \ldots, \alpha_{r-1}\}$ contains r elements (order)

Set G_q consists of all formal sums: $\alpha = \int_{i=0}^{r-1} a_i \alpha_i$ where $a_i \in \mathbb{Z}_q$

v4-1, $G = PSL(2,7) = GL(3,2)$ and $r = 168$:

$$A = \begin{pmatrix} c_{11} & c'_{12} & c_{13} & c'_{14} \\ c'_{21} & c_{22} & c'_{23} & c_{24} \\ c_{31} & c'_{32} & c_{33} & c'_{34} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 21 & 0 \\ * & 1 \\ * & 21 \end{pmatrix} \begin{pmatrix} * & * \\ * & * \end{pmatrix} \quad |c_{ij}| = 1 \text{ and } |c'_{ij}| = 26$$

$(q, r, \lambda, c, s, l) = (439, 168, 54, 10, 100, 492)$
$(m, n, k) = (3, 2, 2)$

v4-5, $G = A_6$ and $r = 360$:

$$A = \begin{pmatrix} c_{11} & c'_{12} & c_{13} & c'_{14} \\ c'_{21} & c_{22} & c'_{23} & c_{24} \\ c_{31} & c'_{32} & c_{33} & c'_{34} \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 29 & 0 \\ * & 1 \\ * & 29 \end{pmatrix} \begin{pmatrix} * & * \\ * & * \end{pmatrix} \quad |c_{ij}| = 1 \text{ and } |c'_{ij}| = 49$$

EHTv4 Improvements

1. When checking if C_1 is invertible, we compute the inversions of resulting diagonal elements. These inversions can be stored as part of the secret key **[sk]** and allows us to skip any inversions in the signature generation process which is the most computationally expensive operation in the scheme.

2. We can also generate invertible B by construction:

$$B = \begin{pmatrix} u & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ y & 1 \end{pmatrix} \begin{pmatrix} z & 1 \\ 1 & 0 \end{pmatrix}$$

$$B^{-1} = \begin{pmatrix} 0 & 1 \\ 1 & -z \end{pmatrix} \begin{pmatrix} 1 & 0 \\ -y & 1 \end{pmatrix} \begin{pmatrix} 1 & -x \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 \\ 1 & -u \end{pmatrix}$$

$u, x, y, z \in G_q$

EHTv3/EHTv4 Sizes and Timings

EHT version -NIST category	v3-1	v3-3	v3-5	v4-1	v4-5
Signature (bytes)	169	255	344	369	857
Private Key (bytes)	368	532	701	419	925
Public Key (Kbytes)	83.5	191.6	349.0	1.11	2.63
Key Generation (msec)	194	597	1530	12.1	115
Signature Generation (msec)	75.8	206	305	9.0	59.3
Signature Verification (msec)	0.82	1.78	3.16	3.85	26.2
# trials for a signature	2.6	3.22	2.01	4.97	3.46

Timings from a common computer with Windows 10 64-bit operating system and x64-based processor: 12thGen Intel(R) Core(TM) i7-12800H@2.40 GHz with 16.0 GB Ram.

Size Comparisons

		EHTv3	EHTv4	EagleSign	HAETAE	HAWK	HuFu	Raccoon	SQUIRRELS	Dilithium	Falcon
LEVEL 1	Public Key	83490	1107	-	-	1024	1059	2256	681780	-	897
	Secret Key	368	419	-	-	184	-	14800	-	-	1281
	Signature	169	369	-	-	555	2455	11524	1019	-	666
LEVEL 3	Public Key	191574	-	1824	1472	-	2177	3160	1629640	1952	-
	Secret Key	532	-	-	2080	-	-	18840	-	4000	-
	Signature	255	-	2336	2337	-	3540	14544	1554	3293	-
LEVEL 5	Public Key	348975	2623	3616	2080	2440	3573	4064	2786580	2592	1793
	Secret Key	701	925	-	2720	360	-	26016	-	4864	2305
	Signature	344	875	3488	2908	1221	4520	20330	2025	4595	1280

Published Attacks (pqc-forum)

1. *Wessel van Woerden and Eamonn Postlethwaite **(HAWK)***

Attack:	Given A and h, a BKZ based attack finds x and e_1 such that $e_1 \equiv h - Ax$ and $\max_l(e_1) \leq s$. Recently broke EHTv3 challenge of 80-bit security.
Countermeasure:	e and e_1 are distributed differently. Take a real-valued distinguisher f and bounds b_1 and b_2. Additional signature generation and verification rule: $b_1 \leq f(e) \leq b_2$. So far none of 10^3 e_1 from the attack passed verification. We will further study $\mathbf{Pr}(b_1 \leq f(e_1) \leq b_2)$ to better the choice of f.

2. *Keegan Ryan and Adam Suhl*

Attack:	HZP attack recovers some columns of C from $e = Cz = h - Ax$, where z is distributed uniformly. 5×10^5 signatures were enough to break v3-1 and similar has been verified by us for v4-1.
Countermeasure:	As C is rectangular, we may provide that a significant portion of z has the distribution of our choice. We have tested the attack on this modification with a proper distribution. With 5×10^6 signatures, no information of matrix C was leaked.

**The countermeasures do not affect current parameters and the efficiency. An updated specification will be published soon.*

Closing Remarks

1. EHTv3 and EHTv4 are still not fully optimized (sizes, timing)
2. Shorter signatures when compared to most other schemes.
3. The schemes are transparent and easy to understand and implement.
4. EHTv3 might perform well on 8-bit platforms as its arithmetic is modulo a relatively small positive integer $q = 47$.
5. Main operations in both schemes are easily parallelizable.

Questions?

igor.semaev@uib.no
martin.feussner@uib.no