BAT: Small and Fast KEM over NTRU Lattices

Pierre-Alain Fouque, Paul Kirchner, Thomas Pornin, Yang Yu

CHES 2022

September 2022

Our initial Goal

BAT: a New KEM companion for Falcon signature, based on NTRU

- Latency and efficiency of TLS are impacted by lattice schemes
- Maximal IP packet size: 1536 bytes
- Falcon: smallest $|ct| + |pk|$ among all known lattice signature schemes
- Falcon has some drawbacks: FPA and hard to protect against SCA

Our initial Goal

BAT: a New KEM companion for Falcon signature, based on NTRU

- Latency and efficiency of TLS are impacted by lattice schemes
- Maximal IP packet size: 1536 bytes
- Falcon: smallest $|ct| + |pk|$ among all known lattice signature schemes
- Falcon has some drawbacks: FPA and hard to protect against SCA

- BAT avoids these issues: no need to FPA and easier to protect
- BAT ciphertext size:
  1. 473 bytes for BAT-512(NIST-I)
  2. 1006 bytes for BAT-1024 (NIST-V)
  3. LW-BAT: 203 bytes for 80-bit security
- BAT enjoys fast encap/decap $\simeq$ Kyber, but relatively slow keygen

Performance Comparison

	Security	Ciphertext (bytes)	PK (bytes)	Keygen (kcycles)	Encaps (kcycles)	Decaps (kcycles)
BAT	128 bits	473	521	$30.6 \cdot 10^3$	8.4	54.3
BAT	256 bits	1006	1230	$185.7 \cdot 10^3$	18.5	118.6
LW-BAT	80 bits$^{(c)}$	203	225	$23.6 \cdot 10^3$	55.7	248.0
NTRU-HRSS	128 bits	1140	1140	220.3	34.6	65.0
NTTRU	128 bits	1248	1248	6.4	6.1	7.8
Kyber	128 bits	768	800	33.9	45.2	34.6
Kyber	256 bits	1568	1568	73.5	97.3	79.1
Saber	128 bits	736	672	45.2	62.2	62.6
Saber	256 bits	1472	1312	126.2	153.8	155.7
LAC	128 bits	712	544	59.6	89.1	140.2
LAC	256 bits	1424	1056	135.8	208.0	359.2
Round5	128 bits	620	461	46	68	95
Round5	256 bits	1285	978	105	166	247
Round5-iot	96 bits$^{(c)}$	394	342	41	52	28
RSA 4096	128 bits$^{(c)}$	512	512	2.19×10^6	212.1	13690
ECC	128 bits$^{(c)}$	32	32	46	176	130
SIKE p434	128 bits	346	330	5.9×10^3	9.7×10^3	10.3×10^3
Compressed SIKE p434	128 bits	236	197	10.2×10^3	15.1×10^3	11.1×10^3

BAT: a New decryption process for NTRU (more natural ?)

NTRU

Public key: $h = g/f \bmod q$

g, f small coeff. $\in \mathbb{Z}_q[X]$

Secret key: $(f, f^{-1} \bmod p)$

Encrypt: $c = phr + m \bmod q$

p masking mod., r random

Decrypt:

$c' = fc = pgr + fm \bmod q$

$c' = pgr + fm$ over $\mathbb{Z}$

$(f^{-1} \bmod p)c' = m \bmod p$

BAT

Public key: $h = g/f \bmod q$

Secret key: $\mathbf{B}_{f,g} = \begin{pmatrix} g & G \\ f & F \end{pmatrix}$

s.t. $gF - fG = q$

Encrypt: $c = hm + r \bmod q$

Decrypt: Dec. $\begin{pmatrix} c \\ 0 \end{pmatrix}$ w/ $\mathbf{B}_{f,g}$

$L_{h,q} = \{(u, v)^t | u = hv \bmod q\}$

has $\mathbf{B}_{f,g}$ (short basis)

Solving 2 linear equations with 2 unknowns:

$$\begin{pmatrix} Fc \\ -fc \end{pmatrix} = \begin{pmatrix} F & -G \\ -f & g \end{pmatrix} \begin{pmatrix} r \\ -m \end{pmatrix}$$

New efficient NTRU Decoding

Decoding algorithms

- Given c, find short polynomials (e, s) s.t. $c = hs + e$
- All operations simple and efficient: no high-precision arithmetic
- Large decoding distance means high security level: large errors (s, e)
- Babai Rounding: efficient and simple but cannot decode large errors
- Nearest Plane: decode larger errors but require high-prec. arithmetic

New efficient NTRU Decoding

Decoding algorithms

- Given c, find short polynomials (e, s) s.t. $c = hs + e$
- All operations simple and efficient: no high-precision arithmetic
- Large decoding distance means high security level: large errors (s, e)
- Babai Rounding: efficient and simple but cannot decode large errors
- Nearest Plane: decode larger errors but require high-prec. arithmetic

Falcon and the BAT New NTRU decoding

- Falcon: Signature size $\propto$ maximal Gram-Schmidt norm of $\mathbf{B}_{f,g}$ (NP)
- BAT: Decode larger errors than Babai Rounding
- BAT: All algorithms can be implemented using fixed-point arithmetic
- BAT: expensive computations are pre-computed in the Key Gen
- BAT: Optimization between the distribution of e and s

Babai Rounding vs. Nearest Plane Decoding

- Babai Rounding decoder: $\begin{pmatrix} e' \\ -s' \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix} - \mathbf{B}_{f,g} \left\lfloor \mathbf{B}_{f,g}^{-1} \begin{pmatrix} c \\ 0 \end{pmatrix} \right\rceil$
- As $\mathbf{B}_{f,g}^{-1} = \frac{1}{q} \begin{pmatrix} F & -G \\ -f & g \end{pmatrix}$, $(e', s') = (e, s)$ if

$$\begin{pmatrix} F & -G \\ -f & g \end{pmatrix} \begin{pmatrix} e \\ -s \end{pmatrix} = \begin{pmatrix} Fc \bmod q \\ -fc \bmod q \end{pmatrix}$$

- It is correct iff $\max\{\|fe + gs\|_\infty, \|Fe + Gs\|_\infty\} \leq q/2$

Babai Rounding vs. Nearest Plane Decoding

- Babai Rounding decoder: $\begin{pmatrix} e' \\ -s' \end{pmatrix} = \begin{pmatrix} c \\ 0 \end{pmatrix} - \mathbf{B}_{f,g} \left\lfloor \mathbf{B}_{f,g}^{-1} \begin{pmatrix} c \\ 0 \end{pmatrix} \right\rceil$
- As $\mathbf{B}_{f,g}^{-1} = \frac{1}{q} \begin{pmatrix} F & -G \\ -f & g \end{pmatrix}$, $(e', s') = (e, s)$ if

$$\begin{pmatrix} F & -G \\ -f & g \end{pmatrix} \begin{pmatrix} e \\ -s \end{pmatrix} = \begin{pmatrix} Fc \bmod q \\ -fc \bmod q \end{pmatrix}$$

- It is correct iff $\max\{\|fe + gs\|_\infty, \|Fe + Gs\|_\infty\} \leq q/2$
- NP decoder correct iff $\max\{\|fe + gs\|_\infty, \|F^* e + G^* s\|_\infty\} \leq q/2$, where F^*, G^* are the Gram-Schmidt orthogonalized basis $\mathbf{B}_{f,g}^*$
- $\|(g, f)\| \approx \|(G^*, F^*)\|$, but $\|(G, F)\| \approx \sqrt{\frac{n}{12}} \cdot \|(g, f)\|$
- $\|(e, s)\|$ is dominated by $\|(G, F)\|$ in Babai Rounding, $\|(g, f)\|$ in NP

A New Decoding Algorithm for NTRU

Goal: Replace the large (G, F) by some small (G', F') of size $\approx \|(g, f)\|$

- $\mathbf{B}^*_{f,g} = \begin{pmatrix} g & G^* = G - vg \\ f & F^* = F - vf \end{pmatrix}$ where $v = \frac{F\bar{f} + G\bar{g}}{f\bar{f} + g\bar{g}}$
- $(G', F') = (G - g\lfloor v \rceil_{q'}, F - f\lfloor v \rceil_{q'})$
- If q' is large, (G', F') converges to (G^*, F^*) whose norm is $\approx \|(g, f)\|$

A New Decoding Algorithm for NTRU

Goal: Replace the large (G, F) by some small (G', F') of size $\approx \|(g, f)\|$

- $\mathbf{B}^*_{f,g} = \begin{pmatrix} g & G^* = G - vg \\ f & F^* = F - vf \end{pmatrix}$ where $v = \frac{F\bar{f} + G\bar{g}}{f\bar{f} + g\bar{g}}$
- $(G', F') = (G - g\lfloor v \rceil_{q'}, F - f\lfloor v \rceil_{q'})$
- If q' is large, (G', F') converges to (G^*, F^*) whose norm is $\approx \|(g, f)\|$

- Refinement: Distributions of e and s are not the same $\|s\| \ll \|e\|$
- Better decoding if we tweak the $\begin{pmatrix} g' & G' \\ f' & F' \end{pmatrix}$ with $\|g'\| > \|g\| \approx \|f\| > \|f'\|$ and $\|G'\| > \|G\| \approx \|F\| > \|F'\|$
- $\gamma = \sigma_e / \sigma_s$ tweaking parameter

BAT scheme

BAT Encryption

$c_1 = \left\lfloor \frac{hs \bmod q}{k} \right\rceil$
$e = (hs \bmod q) - kc_1$
if $\|(\gamma s, e)\| >$ thres., return $\perp$
else return $(c_1, c_2 = m \oplus H(s))$

BAT Decryption

$(e, s) \leftarrow$ Decode (c_1, k, sk)
if $\|(\gamma s, e)\| >$ thres., return $\perp$
else if $c_1 = \left\lfloor \frac{hs \bmod q}{k} \right\rceil$, return $H(s) \oplus c_2$

Key Encapsulation Method derived from the Encryption Scheme using Duman et al. [eprint 2021/1352]. Security proof in the QROM

Security Assumptions

- (Decision) NTRU assumption: $\mathcal{R} = \mathbb{Z}[x]/(x^n+1)$, χ distrib. of f, g

$$\begin{aligned}\mathrm{Adv}^{\mathrm{NTRU}}_{\mathcal{R},q,\chi}(\mathrm{A}) &= \Pr[b=1 \mid f,g \hookleftarrow \chi; b \leftarrow \mathrm{A}(f^{-1}g \bmod q)] - \\ &\qquad \Pr[b=1 \mid u \hookleftarrow U(\mathcal{R}_q^\times); b \leftarrow \mathrm{A}(u)]\end{aligned}$$

- (Search) Ring-LWR assumption: $\chi = U(\mathcal{R} \bmod 2)$

$$\mathrm{Adv}^{\mathrm{RLWR}}_{\mathcal{R},q,k,\chi}(\mathrm{A}) = \Pr_{a \hookleftarrow U(\mathcal{R}_q^\times), s \hookleftarrow \chi}\left[\mathrm{A}\left(a, \left\lfloor \frac{(as \bmod q)}{k} \right\rceil\right) = s\right]$$

Security parameters and attack cost

- $n = 2^\ell$ and q' is used to control the decryption failure rate.
- $q = bk + 1$: b size of each ciphertext coefficient, k decoding distance.

Security	n	(b, k, q)	σ_f	q'	Decrypt. Fail.
80 bits	256	$(64, 2, 128)$	0.595	64513	$2^{-71.9}$
128 bits	512	$(128, 2, 257)$	0.596	64513	$2^{-146.7}$
256 bits	1024	$(192, 4, 769)$	0.659	64513	$2^{-166.7}$

Security parameters and attack cost

- $n = 2^\ell$ and q' is used to control the decryption failure rate.
- $q = bk + 1$: b size of each ciphertext coefficient, k decoding distance.

Security	n	(b, k, q)	σ_f	q'	Decrypt. Fail.
80 bits	256	$(64, 2, 128)$	0.595	64513	$2^{-71.9}$
128 bits	512	$(128, 2, 257)$	0.596	64513	$2^{-146.7}$
256 bits	1024	$(192, 4, 769)$	0.659	64513	$2^{-166.7}$

Security	Key Recovery		Message Recovery	
	primal	hybrid	primal	hybrid
80 bits	87.8 / 236	90.3	83.3 / 219	79.5
128 bits	152.1 / 475	164.1	144.6 / 447	140.1
256 bits	274.4 / 933	314.4	278.6 / 949	275.7

Table: Concrete security estimate. "A/B": attack cost A and BKZ blocksize B.

BAT: Storage and Speed Performances

Table: The required storage (full format, including the header byte)

Security	Public Key (bytes)	Ciphertext (with FO, bytes)	Private key (short, bytes)	Secret Key (long, bytes)
80 bits	225	203	225	1473
128 bits	521	473	417	2953
256 bits	1230	1006	801	6030

BAT: Storage and Speed Performances

Table: The required storage (full format, including the header byte)

Security	Public Key (bytes)	Ciphertext (with FO, bytes)	Private key (short, bytes)	Secret Key (long, bytes)
80 bits	225	203	225	1473
128 bits	521	473	417	2953
256 bits	1230	1006	801	6030

Table: The performance of the plain C implementation

Security	Key Generation (cycles)	Encapsulation (cycles)	Decapsulation (cycles)
80 bits	$\approx 23.8 \times 10^6$	82131	392036
128 bits	$\approx 37.2 \times 10^6$	35785	279260
256 bits	$\approx 264.7 \times 10^6$	71007	537580

Measured on Intel i5-8259U CPU clocked at 2.3 GHz; TurboBoost is disabled

Conclusion

We present a new NTRU-based KEM, called BAT

- more compact than all known lattice-based KEMs
- encap/decap are fast comparable to Kyber

Conclusion

We present a new NTRU-based KEM, called BAT

- more compact than all known lattice-based KEMs
- encap/decap are fast comparable to Kyber

The complexity of the code as well as its running time is asymmetric

- cheap daily operations are favourable to small devices
- expensive keygen can be compensated by frequent key usage or regular key creation for forward secrecy

Conclusion

We present a new NTRU-based KEM, called BAT

- more compact than all known lattice-based KEMs
- encap/decap are fast comparable to Kyber

The complexity of the code as well as its running time is asymmetric

- cheap daily operations are favourable to small devices
- expensive keygen can be compensated by frequent key usage or regular key creation for forward secrecy

BAT and Falcon use similar key structure

- BAT has simpler and faster daily operations
- BAT is implemented fully over integers and smaller, thus more compatible with small devices