HAETAE: Shorter Fiat-Shamir with Aborts Signature

HEAAN CRYPTO LAB

Jung Hee Cheon[1,2], Hyeongmin Choe[1], **Julien Devevey**[3], Tim Güneysu[4,5], Dongyeon Hong[2], Markus Krausz[4], Georg Land[4], Marc Möller[4], Damien Stehlé[2], MinJune Yi[1]

1. Seoul National University
2. CryptoLab Inc.
3. ENS de Lyon
4. Ruhr University Bochum
5. German Research Centre for Artifical Intelligence

SEOUL NATIONAL UNIVERSITY HEAAN CRYPTO LAB ENS DE LYON RUB DFKI

High-level Overview

Same framework as DILITHIUM: Fiat-Shamir with Aborts over lattices

Our goals:

Minimize signature size	Fixed-point arithmetic everywhere
Replace with	Careful analysis of the sampler
Bimodal version of the scheme	

Performances

	FALCON	DILITHIUM	HAETAE	H/D
Level	1	2	2	
$\lvert vk \rvert$	897	1312	992	75.6%
$\lvert \sigma \rvert$	666	2420	1463	60.5%
KG cycles (average)	60M	339K	1.832M	540%
Sign cycles (average)	17M	1.446M	8.903M	616%

Performances

	FALCON	DILITHIUM	HAETAE	H/D
Level	1	2	2	
$\|vk\|$	897	1312	992	75.6%
$\|\sigma\|$	666	2420	1463	60.5%
KG cycles (average)	60M	339K	1.832M	540%
Sign cycles (average)	17M	1.446M	8.903M	616%

- Haetae works over $\mathcal{R} = \mathbb{Z}[x]/(x^{256}+1)$ and uses a modulus $q = 64513$ (16 bits)
- Between 1.5 and $2\times$ less arithmetic operations than Dilithium

Fiat-Shamir with Aborts

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$
with $\mathbf{As} = q\mathbf{j} \bmod 2q$

Sign($\mathbf{A}, \mathbf{s}, \mu$):

do
$\mathbf{y} \hookleftarrow U($$)$
$w = \mathbf{Ay} \bmod 2q$
$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$
$z = \mathbf{y} + (-1)^b \mathbf{s}c$
w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$
while $\mathbf{z} = \perp$
$x = \mathrm{compress}(\mathbf{z})$
return (x, c)

Fiat-Shamir with Aborts

do

$\mathbf{y} \hookleftarrow Q$

$c = H(\mathbf{Ay} \bmod 2q, \mu)$

$\mathbf{z} = \mathbf{y} + (-1)^{U(\{0,1\})}\mathbf{s}c$

w.p. $\frac{2P(\mathbf{z})}{(M(Q(\mathbf{z}-\mathbf{s}c)+Q(\mathbf{z}+\mathbf{s}c)))}$

$\mathbf{z} = \perp$

while $\mathbf{z} = \perp$

return $(\mathbf{z}, c)$

- $\mathbf{z} \hookleftarrow P$
- Verification relies on $\mathbf{Az} - qc\mathbf{j} = \mathbf{Ay} \bmod 2q$ as $\mathbf{As} = -\mathbf{As} = q\mathbf{j} \bmod 2q$
- Bimodal [DDLL13] is more compact [**D**FPS22]
- Compactness depends on $\|\mathbf{s}c\|$

Optimal Choice of Distribution

Our choice: continuous U(●)

- Most compact [**D**FPS22]
- Easier rejection probability than Gaussians
- Rejection probability well-understood
- Rounding step before hashing $\mathbf{A}\lfloor\mathbf{y}\rceil$ and compressing compress($\lfloor\mathbf{z}\rceil$)

Rejection Step

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$
with $\mathbf{As} = q\mathbf{j} \bmod 2q$

Sign$(\mathbf{A}, \mathbf{s}, \mu)$:

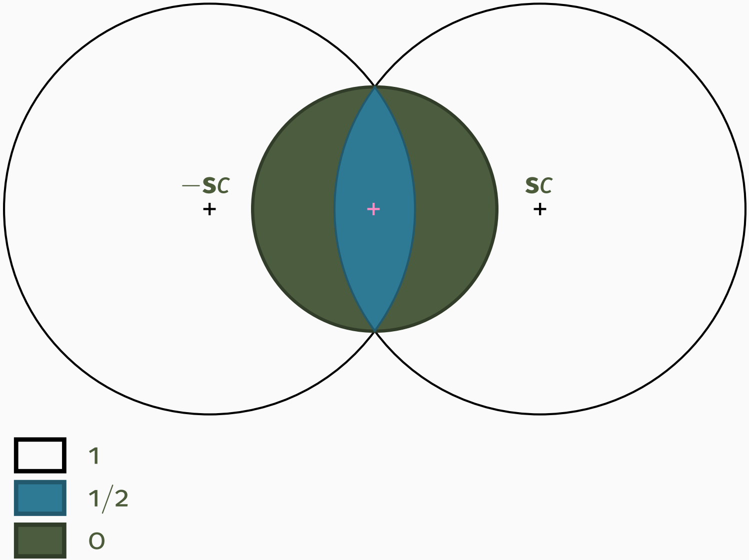
do

$\mathbf{y} \hookleftarrow U($$)$
$w = \mathbf{Ay} \bmod 2q$
$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$
$z = \mathbf{y} + (-1)^b \mathbf{s}c$
w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$

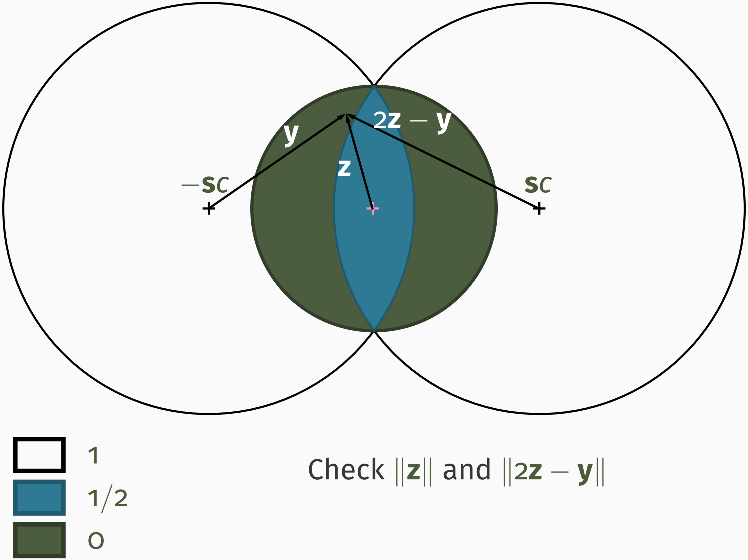
while $\mathbf{z} = \perp$

$x = \mathrm{compress}(\mathbf{z})$
return (x, c)

Rejection Probability



Rejection Probability



Check $\|\mathbf{z}\|$ and $\|2\mathbf{z} - \mathbf{y}\|$

Hyperball Sampler

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$
with $\mathbf{As} = q\mathbf{j} \bmod 2q$

Sign$(\mathbf{A}, \mathbf{s}, \mu)$:

do

$\mathbf{y} \hookleftarrow U($$)$

$w = \mathbf{Ay} \bmod 2q$

$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$

$z = \mathbf{y} + (-1)^b \mathbf{s}c$

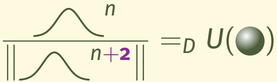
w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$

while $\mathbf{z} = \perp$

$x = \mathrm{compress}(\mathbf{z})$

return (x, c)

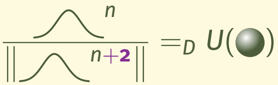
Back to Gaussian sampling [VG17]



- Works for continuous distributions

Main Theorem

Back to Gaussian sampling [VG17]



- Works for continuous distributions
- Adapted to work from discrete gaussian over $\frac{1}{N}\mathbb{Z}^{n+2}$ to $U(\bullet \cap \frac{1}{N}\mathbb{Z}^{n})$
- Requires large enough standard deviation and N

Implementation with Fixed-point Arithmetic

- Reject from discrete  to discrete 
- New average rejection probability?
- Close enough to the previous one for large N
- Balanced out with the previous constraint

Performances

Up to 80% of signing runtime!

Hashing to a Ball

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$
with $\mathbf{As} = q\mathbf{j} \bmod 2q$

Sign$(\mathbf{A}, \mathbf{s}, \mu)$:

do
$\mathbf{y} \hookleftarrow U($$)$
$w = \mathbf{Ay} \bmod 2q$
$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$
$z = \mathbf{y} + (-1)^b \mathbf{s}c$
w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$
while $\mathbf{z} = \perp$
$x = \mathrm{compress}(\mathbf{z})$
return (x, c)

Hash-related Choices

	DILITHIUM	HAETAE
Challenge	Ternary	Binary
Entropy	$\binom{256}{\tau}+\tau$	$\binom{256}{\tau}$
Level II τ	39	58

NB: for Level V, to get 255 bits of entropy, we take τ with Hamming weight $<$ 128 and half of those with Hamming weight 128

SampleInBall(τ):

$c_0 \dots c_{255} = 0^{256}$
For $i = 256 - \tau$ to 255
    $j \hookleftarrow U(\{0 \dots i\})$
    $c_i = c_j$
    $c_j = 1$
return c

Key Generation

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$
with $\mathbf{As} = q\mathbf{j} \bmod 2q$

Sign($\mathbf{A}, \mathbf{s}, \mu$):

do

$\mathbf{y} \hookleftarrow U($$)$

$w = \mathbf{Ay} \bmod 2q$

$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$

$z = \mathbf{y} + (-1)^b \mathbf{s}c$

w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$

while $\mathbf{z} = \perp$

$x = \mathrm{compress}(\mathbf{z})$

return (x, c)

Key Generation

$\mathbf{A}_0 \hookleftarrow U(\mathcal{R}_q^{k\times \ell-1})$
$\mathbf{s}_0, \mathbf{e}_0 \hookleftarrow U([-\eta \ldots \eta])^{\ell-1+k}$
$\mathbf{b} \leftarrow \mathbf{A}_0\mathbf{s}_0 + \mathbf{e}_0 \bmod q$
$\mathbf{A} \leftarrow (-2\mathbf{b} + q\mathbf{j}|2\mathbf{A}_0|2\mathbf{I}_k) \bmod 2q$
$\mathbf{s} \leftarrow (1|{\mathbf{s}_0}^\top|{\mathbf{e}_0}^\top)^\top$
restart if $f_\tau(\mathbf{s}) > n\beta^2/\tau$
return $\mathsf{vk} = \mathbf{A}, \mathsf{sk} = \mathbf{s}$

- $\mathbf{j} = (1, 0 \ldots 0)^\top$
- Add a trapdoor in the public matrix
- f_τ ensures that $\|\mathbf{s}c\| \leq \beta$ for any c with Hamming weight τ
- Acceptance rate from 10 to 25%

Signature Compression (Two Ways)

KeyGen(1^λ):

return $\mathbf{A}, \mathbf{s}$

with $\mathbf{A}\mathbf{s} = q\mathbf{j} \bmod 2q$

Sign($\mathbf{A}, \mathbf{s}, \mu$):

do

$\mathbf{y} \hookleftarrow U(\bullet)$

$w = \mathbf{A}\mathbf{y} \bmod 2q$

$c = H(\mathrm{HB}(w), \mathrm{LSB}(w), \mu)$

$z = \mathbf{y} + (-1)^b \mathbf{s}c$

w.p. $p(\mathbf{z})$, set $\mathbf{z} = \perp$

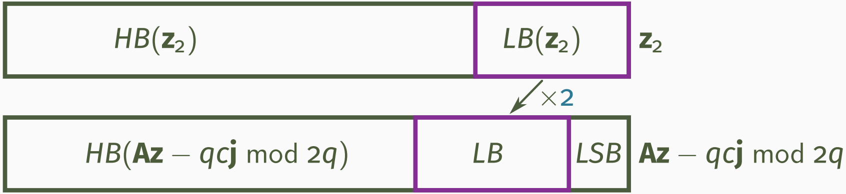
while $\mathbf{z} = \perp$

$x = \mathrm{compress}(\mathbf{z})$

return (x, c)

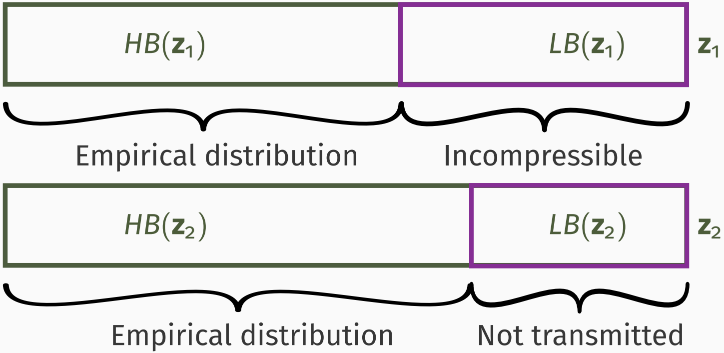
Low Bits Truncation

- Truncation technique from Bai and Galbraith
- $\mathbf{Ay} = \mathbf{A}_1\mathbf{z}_1 + 2\mathbf{z}_2 - q c \mathbf{j} \bmod 2q$ for some $\mathbf{A}_1$



- Exclude $LB(\mathbf{z}_2)$ from the signature
- Hash $HB(\mathbf{w})$ and $LSB(\mathbf{w})$

rANS



- Similar to [ETWY22]
- range Asymmetric Numeral System used to encode/decode
- Swapped with tANS to reduce RAM usage
- Negligible cost in sign runtime ($< 1\%$)

Security Estimation

Security

Theoretical	Practical
Similar to Dilithium	Similar to Dilithium
• Reduction in the QROM depends on SelfTargetMSIS	• Key Recovery attacks solve an LWE instance
• Lossy-soundness in specific parameters regime	• Forgery attacks solve a SIS instance

Thank you!