Kettle

Short Post-Quantum Threshold Signatures from the HAWK Signature Scheme

Calvin Abou Haidar Daniel Escudero Thomas Espitau Clément Hoffmann Haruhisa Kosuge Kaoru Takemure Mehdi Tibouchi Hernán Darío Vanegas Madrigal

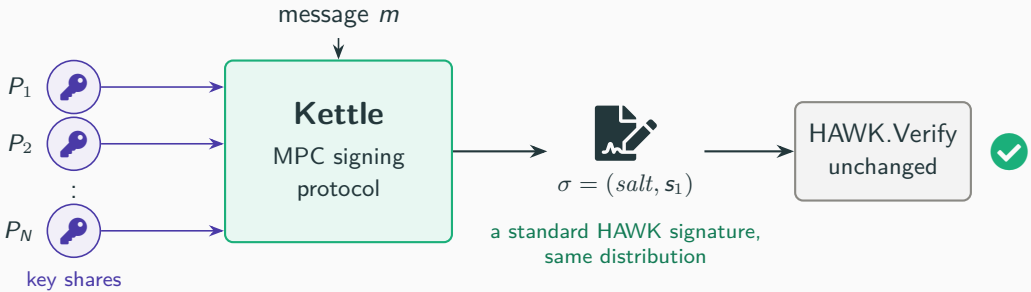
NTT Social Informatics Laboratories ▪ TACEO ▪ PQShield ▪ AIST ▪ HashCloak

Preview Talk @ TCPT3 ▪ NIST Threshold Call (IR 8214C) ▪ Category S1

October 2026

Presented on 2026-10-06 @ NIST TCPT3

Kettle in one slide

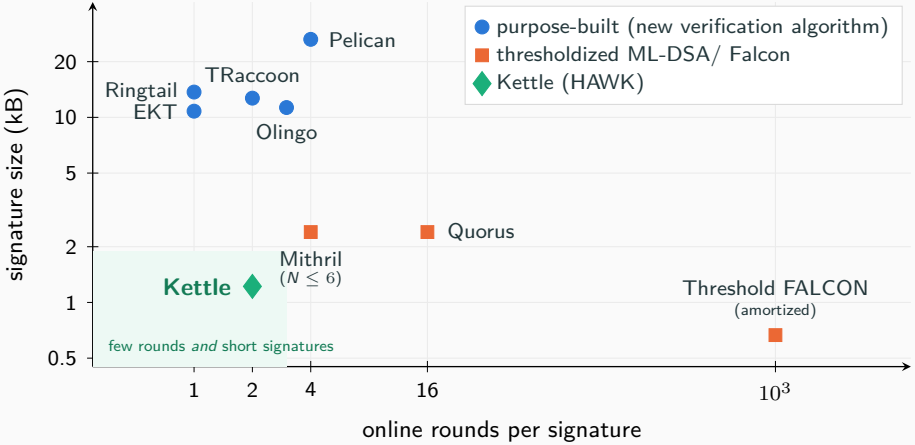


2	**1221 B**	**any N, t**	**UC**
online rounds (+1 if malicious)	signature HAWK-1024, 128 bits	honest or dishonest majority	security set by the MPC backend

Category S1. Keys generated by a trusted dealer (no distributed key generation).

Why HAWK

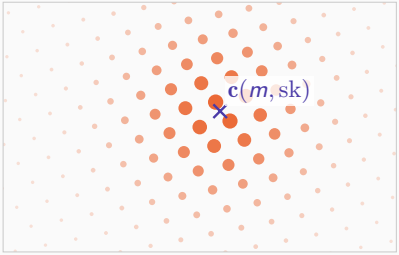
Threshold lattice signatures: two routes



Rounds per signing *attempt* for the ML-DSA-based schemes (rejections restart them). Kettle: 2 rounds, +1 for active security. Values from Table 1 of the Kettle paper and the references therein, for $N = 1024$ or the largest N reported.

Hash-and-sign in MPC: why Falcon is hard and HAWK is not

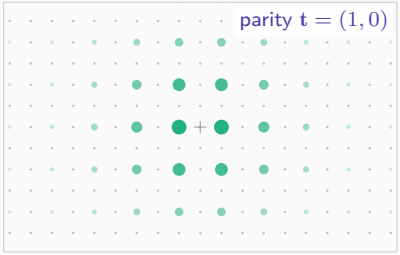
GPV / Falcon (FN-DSA)



Gaussian over a skewed lattice, centred on a secret point. Needs the secret basis: recursive sampler, floating point.

✗ in MPC: $\approx 10^3$ online rounds [Threshold FALCON]

HAWK

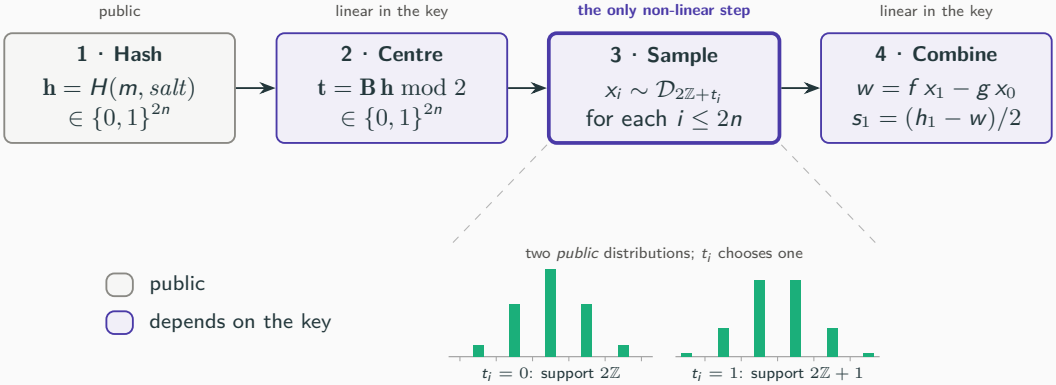


Gaussian over $2\mathbb{Z}^{2n} + \mathbf{t}$, centred at 0: independent coordinates. The secret only picks the bits $\mathbf{t}$.

Integers only: no floating point.

✓ in MPC: table look-ups (comparisons) + bit selection

HAWK signing in four steps

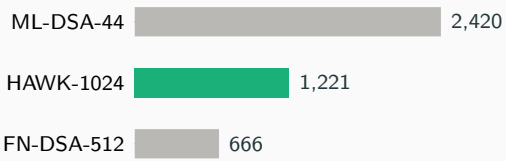


Key: $\mathbf{B} = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$. Simplified: sign convention on w and norm check omitted.

HAWK after the 2026 key-recovery result

- Key recovery for HAWK-n reduces to SVP in dimension $n/2 + 1$ [Straznickas–Weis, Anthropic, Jul. 2026].
- About twice the dimension for the same security: **HAWK-1024 now targets 128 bits**.
- 256-bit level: **HAWK-2048**, to be specified in our package.
- The HAWK team withdrew from the NIST additional signature call.
- ✅ For Kettle, n only scales communication and signature size.

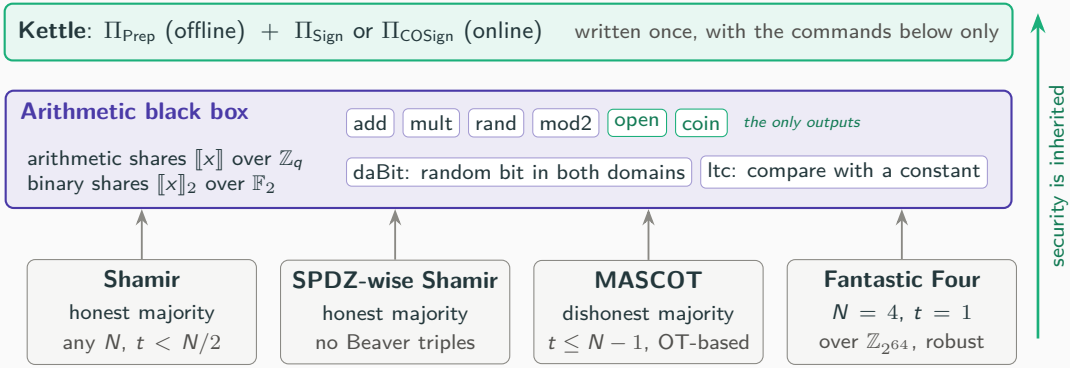
Signature size at the 128-bit level (bytes)



Purpose-built threshold schemes: 10.8–26.4 kB.

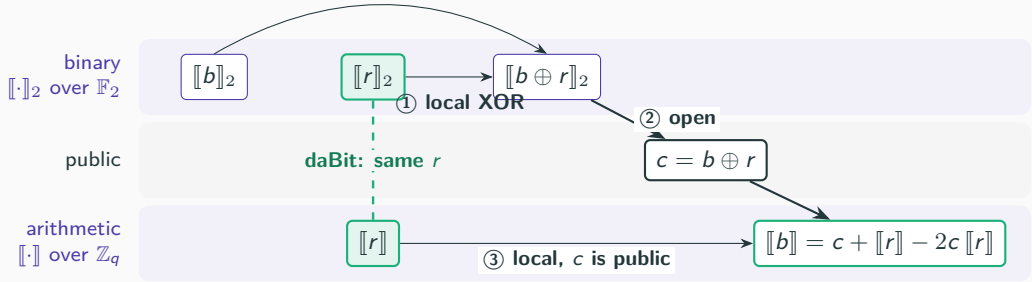
The Kettle protocol

One protocol, many security settings



four maliciously secure backends, all run in our MP-SPDZ proof of concept

Reminder: daBits move a bit between domains



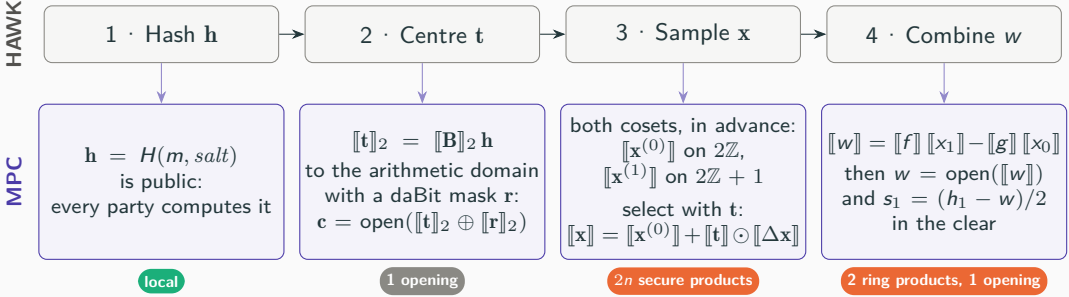
Why it is correct. If $c = 0$ then $b = r$; if $c = 1$ then $b = 1 - r$. With c public, both cases are linear in $[\![r]\!]$.

Why it is private. r is uniform, so c reveals nothing about b.

Cost. One opening, no secure multiplication. daBits are prepared in advance, independently of key and message.

In Kettle. The centre bits $\mathbf{t}$, and the sampler's comparison bits.

A first attempt: each HAWK step in MPC



⚠ Correct, but two layers of secure products *after* the message is known.

Moving every secure product offline

$$\mathbf{x} = \mathbf{x}^{(0)} + \mathbf{t} \odot \Delta\mathbf{x}, \qquad \mathbf{t} = \mathbf{c} + (\mathbf{1} - 2\mathbf{c}) \odot \mathbf{r} \quad \text{(selection; daBit)}$$

violet: secret, prepared offline black: public

Moving every secure product offline

$$\mathbf{x} = \mathbf{x}^{(0)} + \mathbf{t} \odot \Delta\mathbf{x}, \qquad \mathbf{t} = \mathbf{c} + (\mathbf{1} - 2\mathbf{c}) \odot \mathbf{r} \quad \text{(selection; daBit)}$$

$$\Rightarrow \quad \mathbf{x} = \underbrace{\mathbf{x}^{(0)} + \mathbf{r} \odot \Delta\mathbf{x}}_{\mathbf{s}_0} + \mathbf{c} \odot \underbrace{(\mathbf{1} - 2\mathbf{r}) \odot \Delta\mathbf{x}}_{\mathbf{s}_1}$$

violet: secret, prepared offline black: public

Moving every secure product offline

$$\mathbf{x} = \mathbf{x}^{(0)} + \mathbf{t} \odot \Delta\mathbf{x}, \qquad \mathbf{t} = \mathbf{c} + (\mathbf{1} - 2\mathbf{c}) \odot \mathbf{r} \quad \text{(selection; daBit)}$$

$$\Rightarrow \quad \mathbf{x} = \underbrace{\mathbf{x}^{(0)} + \mathbf{r} \odot \Delta\mathbf{x}}_{\mathbf{s}_0} + \mathbf{c} \odot \underbrace{(\mathbf{1} - 2\mathbf{r}) \odot \Delta\mathbf{x}}_{\mathbf{s}_1}$$

$$\Rightarrow \quad w = \mathbf{M}\,\mathbf{x} = \underbrace{\mathbf{M}\mathbf{s}_0}_{\mathbf{w}_0} + \mathbf{V}\,\mathbf{c}, \qquad V_{j,i} = M_{j,i}\,(s_1)_i$$

violet: secret, prepared offline black: public

Moving every secure product offline

$$\mathbf{x} = \mathbf{x}^{(0)} + \mathbf{t} \odot \Delta\mathbf{x}, \qquad \mathbf{t} = \mathbf{c} + (\mathbf{1} - 2\mathbf{c}) \odot \mathbf{r} \quad \text{(selection; daBit)}$$

$$\Rightarrow \quad \mathbf{x} = \underbrace{\mathbf{x}^{(0)} + \mathbf{r} \odot \Delta\mathbf{x}}_{\mathbf{s}_0} + \mathbf{c} \odot \underbrace{(\mathbf{1} - 2\mathbf{r}) \odot \Delta\mathbf{x}}_{\mathbf{s}_1}$$

$$\Rightarrow \quad w = \mathbf{M}\,\mathbf{x} = \underbrace{\mathbf{M}\mathbf{s}_0}_{\mathbf{w}_0} + \mathbf{V}\,\mathbf{c}, \qquad V_{j,i} = M_{j,i}\,(s_1)_i$$

violet: secret, prepared offline black: public

Offline, before m
- Gaussians $[\![\mathbf{x}^{(0)}]\!]$, $[\![\mathbf{x}^{(1)}]\!]$
- daBit mask $\mathbf{r}$
- products $[\![\mathbf{w}_0]\!]$, $[\![\mathbf{V}]\!]$



m arrives

Round 1
open
$\mathbf{c} = \mathbf{t} \oplus \mathbf{r}$

local
$[\![w]\!] = [\![\mathbf{w}_0]\!] + [\![\mathbf{V}]\!]\,\mathbf{c}$

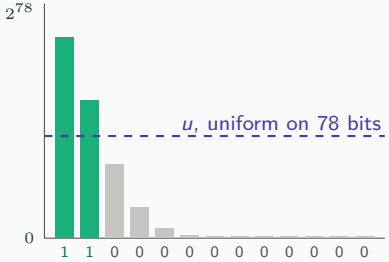
Round 2
open w
$\to \sigma$

time

online: no secure product

Sampling the Gaussians in MPC

HAWK's reverse CDT sampler

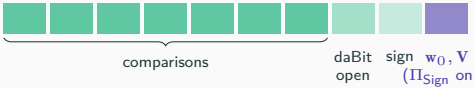


$h = \sum_i b_i = 2, \quad x = \pm(2h + t)$

In MPC: all $4n$ samples in parallel

u: 78 random shared bits	**local**
13 comparisons $u < T[i]$	**7 rounds**
bits to arithmetic (daBits)	**1 round**
random sign	**1 round**

offline rounds

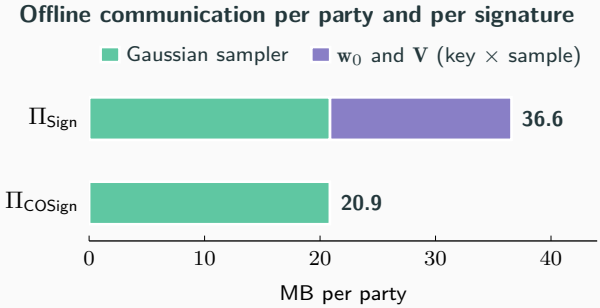


= 10 (Π_{Sign}), 9 (Π_{COSign})

Two variants: online rounds vs. offline communication

Π_{Sign} **round-optimized**
2 online rounds **1.9 kB** online

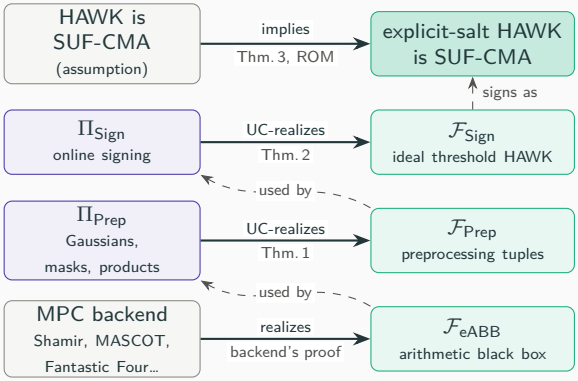
Π_{COSign} **communication-optimized**
3 online rounds **5.8 kB** online



HAWK-1024, up to 1023 parties. We provide a script that computes these costs for other security settings.

Security

Security: a chain of UC proofs



What the adversary sees

- salt
  uniform coin
- $52n$ bits $b \oplus r$ (sampler)
  uniform: daBit mask
- $\mathbf{c} = \mathbf{t} \oplus \mathbf{r}$
  uniform: daBit mask
- w
  determined by the signature

✅ The simulator samples all of them: **identical** views.

Corrupted parties are fixed before the run. How many may be corrupted, and whether they may deviate from the protocol, is set by the backend.

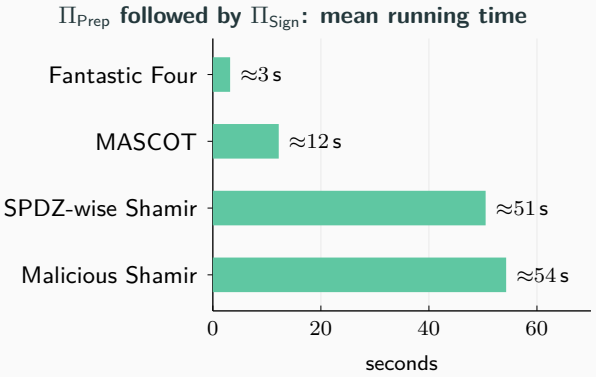
Performance

Comparison with lattice threshold signatures

	online		offline			
Scheme	rounds	comm./party	rounds	comm./party	**signature**	N
Kettle (HAWK-1024)	**2**	**1.9 kB**	10	36.6 MB	**1.2 kB**	1023
thresholdized standard schemes						
Threshold FALCON	$\approx 10^3$	800.3 MB	0	–	0.7 kB	16
Quorus (ML-DSA)	16	1.44 MB	108	330.1 MB	2.4 kB	63
Mithril (ML-DSA)	4	1.05 MB	2	32 B	2.4 kB	6
purpose-built schemes						
Pelican	4	$(53.8 + 116N)$ kB	0	–	26.4 kB	1024
TRaccoon	2	28.2 kB	1	32 B	12.7 kB	1024
Ringtail	1	10.7 kB	1	$(602 + 16N)$ kB	13.7 kB	1024
EKT	1	14.4 kB	1	282.3 kB	10.8 kB	1024
Olingo	3	852.0 kB	0	–	11.3 kB	1024

Values from Table 1 of the Kettle paper and the references therein, for $N = 1024$ or the largest N reported; generation of correlated randomness excluded. ML-DSA-based schemes: rounds per signing attempt; Threshold FALCON: amortized. Kettle: honest majority, $+1$ online round for active security.

Proof of concept in MP-SPDZ



8 parties, 3 corrupted
Fantastic Four: 4 parties, 1 corrupted

all four maliciously secure

emulated LAN
0.5 ms latency, 1 Gbit/s

30 runs each
95% CI within 2.7% of the mean

MPC preprocessing
(Beaver triples, daBits) excluded

One generic circuit, no scheme-specific optimization: these timings compare security settings for the same protocol. The package will add an optimized implementation.

Plan for the package

What the package will contain



Specification

Kettle over the arithmetic black box

HAWK-1024 (128 bits)
HAWK-2048 (256 bits)

Proof for Π_{COSign}; adaptive corruptions

Comparison with ML-DSA, SLH-DSA, FN-DSA



Implementation

Open source

Generic: MP-SPDZ, all settings

Optimized: selected settings

Known-answer tests



Experiments

NIST baseline platform

Rounds, communication, time

Several settings and numbers of parties

Reproduction scripts



Patent notes

No known patents

Not claimed

Distributed key generation (S4): keys from a trusted dealer

Open questions: feedback welcome

Distributed key generation

A HAWK key is an NTRU trapdoor: generating it in MPC means solving the NTRU equation on shared data. Open, as for Threshold FALCON. Ideas welcome.

Better MPC, better Kettle

Kettle only uses generic MPC commands: any progress on daBits, Beaver triples or new backends carries over directly.

A reusable sampler

The shared table sampler (random bits, comparisons, daBits) works for any table-based discrete Gaussian. Other teams may use it.

Which settings matter to you?

Number of parties, threshold, honest or dishonest majority: tell us what to prioritize in the optimized implementation.

Summary



HAWK is MPC-friendly

integers, public tables, one secret bit per coordinate



Standard HAWK signatures

2 online rounds, no secure product online; any N, t and backend; UC



A complete package

specification, generic and optimized implementations, HAWK-1024 and HAWK-2048



Paper: `eprint.iacr.org/2026/2288`

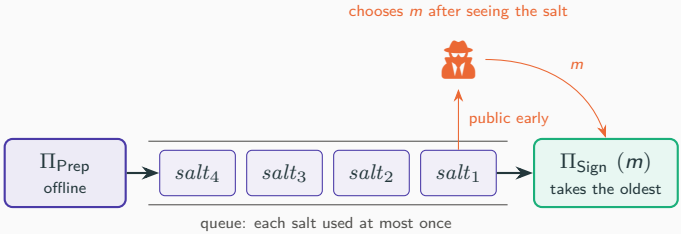


`kettle-team@googlegroups.com`

Thank you!

Questions?

Backup: the salt is public early (explicit-salt HAWK)



Theorem 3 (ROM). Explicit-salt HAWK, whose signer also reveals w, is **SUF-CMA** if HAWK is:

$$\varepsilon_{\mathcal{A}} \leq 4(q_s + 1)\,\varepsilon_{\mathcal{B}} + q_s\, e^{-\Theta(n)}$$

as for threshold MAYO [ABCN26], which proves EUF-CMA; Coron's FDH technique.

Distribution-preserving. Same tables as the HAWK signer: outputs are distributed exactly as HAWK signatures.

Norm check omitted. An output is invalid with probability $< 2^{-121}$ (HAWK-1024). Never a forgery.

Backup: preprocessing Π_{Prep} (one tuple per signature)

Π_{Prep}, in the $\mathcal{F}_{\mathsf{eABB}}$-hybrid model

1. $salt \leftarrow$ coin.
2. $(\llbracket \mathbf{x}^{(0)} \rrbracket, \llbracket \mathbf{x}^{(1)} \rrbracket) \leftarrow$ Gaussian: $2n$ coordinates on $2\mathbb{Z}$ and on $2\mathbb{Z}+1$ (table sampler).
3. $\llbracket \Delta\mathbf{x} \rrbracket \leftarrow \llbracket \mathbf{x}^{(1)} \rrbracket - \llbracket \mathbf{x}^{(0)} \rrbracket$ — local
4. $2n$ daBits $(\llbracket \mathbf{r} \rrbracket, \llbracket \mathbf{r} \rrbracket_2)$.
5. $\llbracket \mathbf{m} \rrbracket \leftarrow \llbracket \mathbf{r} \rrbracket \odot \llbracket \Delta\mathbf{x} \rrbracket$ — $2n$ products
6. $\llbracket \mathbf{s}_0 \rrbracket \leftarrow \llbracket \mathbf{x}^{(0)} \rrbracket + \llbracket \mathbf{m} \rrbracket, \quad \llbracket \mathbf{s}_1 \rrbracket \leftarrow \llbracket \Delta\mathbf{x} \rrbracket - 2\llbracket \mathbf{m} \rrbracket$ — local
7. $\llbracket \mathbf{w}_0 \rrbracket \leftarrow \llbracket f \rrbracket \cdot \llbracket s_{0,1} \rrbracket - \llbracket g \rrbracket \cdot \llbracket s_{0,0} \rrbracket$ — 2 ring products
8. $\llbracket V_{j,i} \rrbracket \leftarrow \llbracket M_{j,i} \rrbracket \cdot \llbracket (s_1)_i \rrbracket$ for all $(j,i) \in [n] \times [2n]$, with $\mathbf{M} = [-\mathrm{rot}(g) \mid \mathrm{rot}(f)]$ — $2n^2$ products
9. Store $(salt, \llbracket \mathbf{w}_0 \rrbracket, \llbracket \mathbf{V} \rrbracket, \llbracket \mathbf{r} \rrbracket_2)$.

Π_{COSign}: stop after step 6 and store $(salt, \llbracket \mathbf{s}_0 \rrbracket, \llbracket \mathbf{s}_1 \rrbracket, \llbracket \mathbf{r} \rrbracket_2)$.

Backup: online signing Π_{Sign} and Π_{COSign}

Π_{Sign}: 2 rounds

Take the oldest unused tuple $(salt, [\![\mathbf{w}_0]\!], [\![\mathbf{V}]\!], [\![\mathbf{r}]\!]_2)$.

1. $\mathbf{h}$ from m and $salt$; $[\![\mathbf{t}]\!]_2 = [\![\mathbf{B}]\!]_2\,\mathbf{h}$ — local
2. **Round 1:** $\mathbf{c} \leftarrow \mathsf{open}([\![\mathbf{t}]\!]_2 \oplus [\![\mathbf{r}]\!]_2)$
3. $[\![w]\!] \leftarrow [\![\mathbf{w}_0]\!] + [\![\mathbf{V}]\!]\,\mathbf{c}$ — local
4. **Round 2:** $w \leftarrow \mathsf{open}([\![w]\!])$
5. Sign convention on w; $s_1 = (h_1 - w)/2$; output $(salt, s_1)$.

Π_{COSign}: 3 rounds

Take the oldest unused tuple $(salt, [\![\mathbf{s}_0]\!], [\![\mathbf{s}_1]\!], [\![\mathbf{r}]\!]_2)$.

1. $\mathbf{h}$ and $[\![\mathbf{t}]\!]_2$ as in Π_{Sign} — local
2. **Round 1:** $\mathbf{c} \leftarrow \mathsf{open}([\![\mathbf{t}]\!]_2 \oplus [\![\mathbf{r}]\!]_2)$
3. $[\![\mathbf{x}]\!] \leftarrow [\![\mathbf{s}_0]\!] + \mathbf{c} \odot [\![\mathbf{s}_1]\!]$ — local
4. **Round 2:** $[\![f]\!] \cdot [\![x_1]\!]$ and $[\![g]\!] \cdot [\![x_0]\!]$ (Beaver triples over $\mathcal{R}$)
5. **Round 3:** $w \leftarrow \mathsf{open}([\![fx_1]\!] - [\![gx_0]\!])$
6. As in Π_{Sign}.

$[\![\mathbf{B}]\!]_2$: the key modulo 2, in the binary domain. Active security adds one round for the final MAC check.

Backup: communication per party and per signature

		Π_{Sign}	Π_{COSign}
offline	13 comparisons $\times$ $4n$ samples (≈ 156 AND each)	20.77 MB	20.77 MB
	13 daBit openings $\times$ $4n$ samples	0.07 MB	0.07 MB
	random signs ($4n$ products)	0.02 MB	0.02 MB
	$\mathbf{w}_0$ and $\mathbf{V}$ ($4n^2$ products)	15.73 MB	–
	total	**36.6 MB**	**20.9 MB**
online	opening of w_1 (n coefficients)	1.92 kB	1.92 kB
	2 ring products	–	3.84 kB
	total	**1.9 kB**	**5.8 kB**
	opening of $\mathbf{c}$ (not counted, as in the paper)	2.56 kB	2.56 kB

HAWK-1024 ($n = 1024$). Shamir sharing, semi-honest: arithmetic shares of 15 bits, binary shares in $\mathbb{F}_{2^{10}}$ (up to 1023 parties); a product costs two openings. Beaver triples and daBits are not counted.

Backup: fewer offline rounds

Trade-off	**rounds**	**price**
Full table instead of half-Gaussian + sign	-1	26 comparisons instead of 13
Degree-3 Beaver triples in the comparisons	-1	about twice the communication of each comparison
Precomputed cross terms (daBits, $\Delta\mathbf{x}$)	-1	$14\times$ more arithmetic products in the sampler
All three	**10 $\rightarrow$ 7**	

Worth it with few parties, when the comparisons are a small part of the offline communication. Offline, we prefer less communication: the phase is batched in advance.

Backup: choice of the modulus q

q prime

+ an arithmetic share is one field element (Shamir: $q > N$)
− binary domain in an unrelated field $\mathbb{F}_{2^\kappa}$, $2^\kappa > N$: daBits need a dedicated protocol
− $\mathrm{mod}2$ is not local

Better for Π_{Sign}: dominated by the $\Theta(n^2)$ arithmetic products.

$q = 2^k$

+ $\mathrm{mod}2$ is local on the shares
+ a daBit is a shared random bit, reduced locally
− Shamir needs a Galois ring $\mathrm{GR}(2^k, \kappa)$: shares κ times larger

Better for Π_{COSign}: almost purely Boolean offline phase.

Few parties: replicated sharing over $\mathbb{Z}_{2^{64}}$ needs no extension (Fantastic Four). HAWK only needs q large enough to avoid reductions: 15 bits.