HAWK: Having Automorphisms Weakens Key

Sixth PQC Standardisation Conference

NIST, Gaithersburg, MD

Daniël M. H. van Gent[1], **Ludo N. Pulles**[1]

26th September 2025

[1]Cryptology Group, CWI, Amsterdam

Eurasian sparrowhawk by Bart Hoekstra, CC-BY-NC-SA 4.0

HAWK

HAWK is a signature scheme,
inspired by the Lattice Isomorphism Problem (LIP).

Fast

KeyGen: < 3.5 ms

Sign/Verify: < 0.1 ms

(on a laptop)

Compact

$|\text{pub}|$: 1024 B

$|\text{sig}|$: 555 B

Hardware friendly

≤ 12 KiB RAM

no *floats*

Security of HAWK

- **Practical** security of HAWK is determined by experiments & cryptanalysis.
  $\Longrightarrow$ Parameters of HAWK are robust.

- Private key recovery requires solving a module structured variant of LIP: smLIP.
  smLIP seems as hard as its unstructured counterpart LIP.

- **Theoretical** security of HAWK formally reduces, in qROM, to:
  one more Shortest Vector Problem (omSVP).
  $\Longrightarrow$ Design of HAWK is robust.

Security of HAWK

- **Practical** security of HAWK is determined by experiments & cryptanalysis.
  $\Longrightarrow$ Parameters of HAWK are robust.

- Private key recovery requires solving a module structured variant of LIP: smLIP.
  smLIP seems as hard as its unstructured counterpart LIP.

- **Theoretical** security of HAWK formally reduces, in qROM, to:
  one more Shortest Vector Problem (omSVP).
  $\Longrightarrow$ Design of HAWK is robust.

Security of HAWK

- **Practical** security of HAWK is determined by experiments & cryptanalysis.
  $\Longrightarrow$ Parameters of HAWK are robust.

- Private key recovery requires solving a module structured variant of LIP: smLIP.
  smLIP seems as hard as its unstructured counterpart LIP.

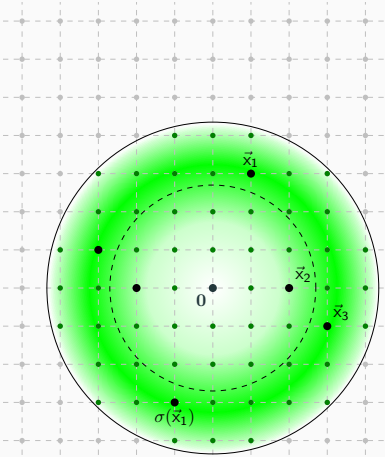
- **Theoretical** security of HAWK formally reduces, in qROM, to:
  one more Shortest Vector Problem (omSVP).
  $\Longrightarrow$ Design of HAWK is robust.

omSVP

one more Shortest Vector Problem (omSVP)

Given: many 'shortish' lattice points $\mathbf{x}_i$,
Challenge: find a "fresh" 'shortish' lattice point $\widetilde{\mathbf{x}}$.

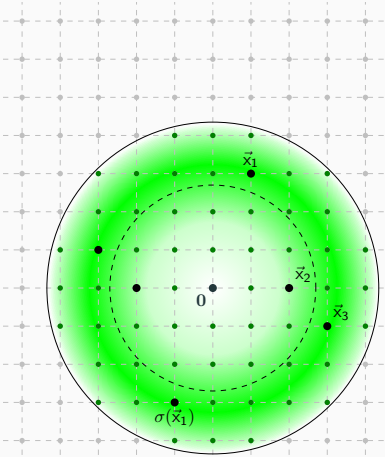
1. **HAWK v1.0 "fresh":**
$$\widetilde{\mathbf{x}} \neq \sigma(\mathbf{x}_i)$$
for a group G of *automorphisms*, i.e., a bijection $\sigma : \Lambda \to \Lambda$ preserving distance. For example: negation.
2. **[LJPW24]:** omSVP is trivial using a 'symplectic automorphism' ω: query $\mathbf{x}_1$ and output $\widetilde{\mathbf{x}} = \omega(\mathbf{x}_1)$.
3. **HAWK v1.1:** add ω to G.
omSVP & formal reduction are updated.



one more Shortest Vector Problem (omSVP)

Given: many 'shortish' lattice points $\mathbf{x}_i$,
Challenge: find a "fresh" 'shortish' lattice point $\widetilde{\mathbf{x}}$.

1. **HAWK v1.0 "fresh":**
$$\widetilde{\mathbf{x}} \neq \sigma(\mathbf{x}_i)$$
for a group G of *automorphisms*, i.e., a bijection $\sigma : \Lambda \to \Lambda$ preserving distance. For example: negation.
2. **[LJPW24]:** omSVP is trivial using a 'symplectic automorphism' ω: query $\mathbf{x}_1$ and output $\widetilde{\mathbf{x}} = \omega(\mathbf{x}_1)$.
3. **HAWK v1.1:** add ω to G.
omSVP & formal reduction are updated.

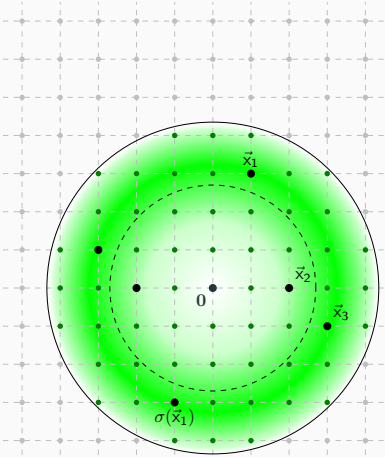


omSVP

one more Shortest Vector Problem (omSVP)

Given: many 'shortish' lattice points $\mathbf{x}_i$,
Challenge: find a "fresh" 'shortish' lattice point $\widetilde{\mathbf{x}}$.

1. **HAWK v1.0 "fresh":**
$$\widetilde{\mathbf{x}} \neq \sigma(\mathbf{x}_i)$$
for a group G of *automorphisms*, i.e., a bijection $\sigma : \Lambda \to \Lambda$ preserving distance. For example: negation.
2. **[LJPW24]:** omSVP is trivial using a 'symplectic automorphism' ω: query $\mathbf{x}_1$ and output $\widetilde{\mathbf{x}} = \omega(\mathbf{x}_1)$.
3. **HAWK v1.1:** add ω to G.
omSVP & formal reduction are updated.



Central Question

Question:
Can we find another automorphism σ with limited effort?

(which would ruin hardness of omSVP once again)

Central Question

Question:
Can we find another automorphism σ with limited effort?

(which would ruin hardness of omSVP once again)

Answer: No.

Any nontrivial σ allows solving smLIP with little effort.

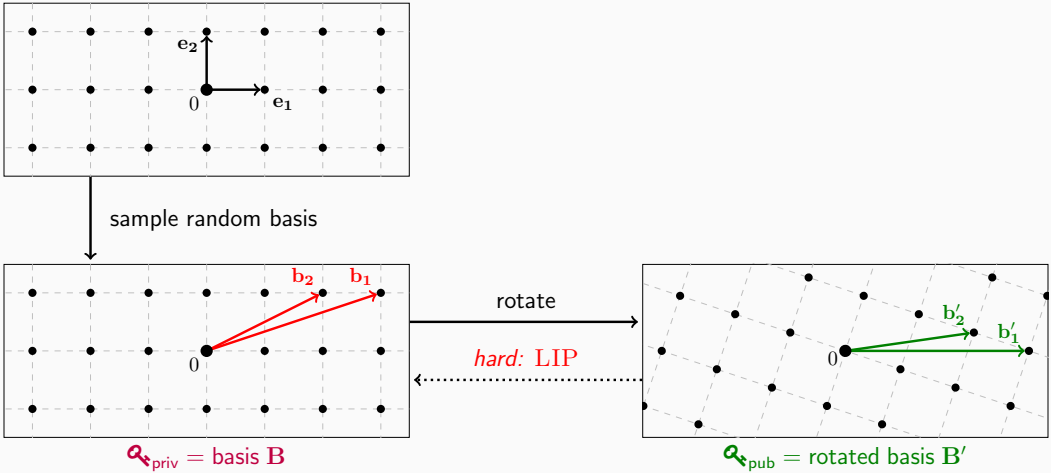
Hence: if smLIP is hard, finding any σ is hard, too.

HAWK in a nutshell

Eurasian sparrowhawk by Bart Hoekstra, CC-BY-NC-SA 4.0

Hiding $\mathbb{Z}^n$ with rotation

Use the 'simplest' lattice: $\mathbb{Z}^n$.



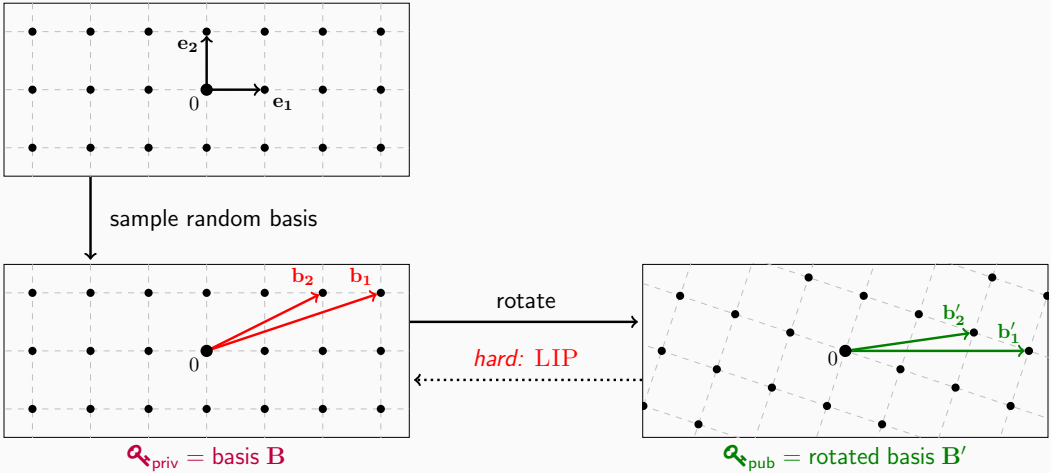
$\mathbf{B}$ = basis $\mathbf{B}$

$\mathbf{B}'$ = rotated basis $\mathbf{B}'$

Give rotation implicitly by giving all inner products $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$, avoiding *floats* 👍.

Hiding $\mathbb{Z}^n$ with rotation

Use the 'simplest' lattice: $\mathbb{Z}^n$.



Give rotation implicitly by giving all inner products $\langle \mathbf{b}_i, \mathbf{b}_j \rangle$, avoiding *floats* 👍.

Module structure

- Let $n = 1024$, $\zeta = e^{2\pi i/n}$ and $R = \mathbb{Z}[\zeta]$. HAWK uses the *module lattice* $R \oplus R \cong \mathbb{Z}^n$.
- Complex conjugation $(-)^* \colon R \to R, r(\zeta) \mapsto r(\overline{\zeta})$ extends to module,
- **HAWK v1.0** takes $G = \langle \zeta \rangle$ generated by the lattice automorphism $(r_1, r_2) \mapsto (\zeta r_1, \zeta r_2)$.
- **HAWK v1.1** takes $G = \langle \zeta, \omega \rangle$ where $\omega \colon (r_1, r_2) \mapsto (r_2^*, -r_1^*)$.
  Note: ω is publicly computable.

Private

$$B = [b_1, b_2] = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

Public

$$Q = B^*B = \begin{pmatrix} b_1^* b_1 & b_1^* b_2 \\ b_2^* b_1 & b_2^* b_2 \end{pmatrix}$$

Computing norms & inner products:

$$\langle x, y \rangle_Q = x^* Q y \big|_{\zeta=0} = \langle Bx, By \rangle,$$
$$\|x\|_Q^2 = x^* Q x \big|_{\zeta=0} = \|Bx\|^2.$$

Module structure

- Let $n = 1024$, $\zeta = e^{2\pi i/n}$ and $R = \mathbb{Z}[\zeta]$. HAWK uses the *module lattice* $R \oplus R \cong \mathbb{Z}^n$.
- Complex conjugation $(-)^* : R \to R, r(\zeta) \mapsto r(\overline{\zeta})$ extends to module,
- **HAWK v1.0** takes $G = \langle \zeta \rangle$ generated by the lattice automorphism $(r_1, r_2) \mapsto (\zeta r_1, \zeta r_2)$.
- **HAWK v1.1** takes $G = \langle \zeta, \omega \rangle$ where $\omega : (r_1, r_2) \mapsto (r_2^*, -r_1^*)$.
  Note: ω is publicly computable.

Private

$$\mathsf{B} = [\mathsf{b}_1,\ \mathsf{b}_2] = \begin{pmatrix} f & F \\ g & G \end{pmatrix}$$

Public

$$\mathsf{Q} = \mathsf{B}^*\mathsf{B} = \begin{pmatrix} \mathsf{b}_1^*\mathsf{b}_1 & \mathsf{b}_1^*\mathsf{b}_2 \\ \mathsf{b}_2^*\mathsf{b}_1 & \mathsf{b}_2^*\mathsf{b}_2 \end{pmatrix}$$

Computing norms & inner products:

$$\langle \mathbf{x}, \mathbf{y} \rangle_{\mathsf{Q}} = \mathbf{x}^*\mathsf{Q}\mathbf{y}\big|_{\zeta=0} = \langle \mathsf{B}\mathbf{x}, \mathsf{B}\mathbf{y} \rangle,$$
$$\|\mathbf{x}\|_{\mathsf{Q}}^2 = \mathbf{x}^*\mathsf{Q}\mathbf{x}\big|_{\zeta=0} = \|\mathsf{B}\mathbf{x}\|^2.$$

Three stage rocket

- Forget structure & rotation. Work with $\mathbb{Z}^n$.
- Structure gives $2n$ lattice automorphisms: $G = \langle \zeta, \omega \rangle$!

Suppose we have $\sigma \notin G$. We prove three things:

1. We can construct sublattice $\Lambda' \subset \mathbb{Z}^n$ of rank $\leq \frac{n}{2}$, with a vector $\mathbf{v}$ of norm $\leq \sqrt{2}$.
2. We gain a *quadratic speedup* by solving SVP in Λ' instead of $\mathbb{Z}^n$.
3. One vector $\mathbf{v} \in \mathbb{Z}^n$ with $\|\mathbf{v}\| \in \{1, \sqrt{2}\}$ solves SMLIP.

Having $\sigma \notin G$, reduces bit security of SMLIP by *at least 50%*.

Bonus: one uniformly sampled σ solves SMLIP in poly(n) time.

Red hawk by VIT DUCKEN from Pixabay.

Three stage rocket

- Forget structure & rotation. Work with $\mathbb{Z}^n$.
- Structure gives $2n$ lattice automorphisms: $G = \langle \zeta, \omega \rangle$!

Suppose we have $\sigma \notin G$. We prove three things:

1. We can construct sublattice $\Lambda' \subset \mathbb{Z}^n$ of rank $\leq \frac{n}{2}$, with a vector $\mathbf{v}$ of norm $\leq \sqrt{2}$.
2. We gain a *quadratic speedup* by solving SVP in Λ' instead of $\mathbb{Z}^n$.
3. One vector $\mathbf{v} \in \mathbb{Z}^n$ with $\|\mathbf{v}\| \in \{1, \sqrt{2}\}$ solves SMLIP.

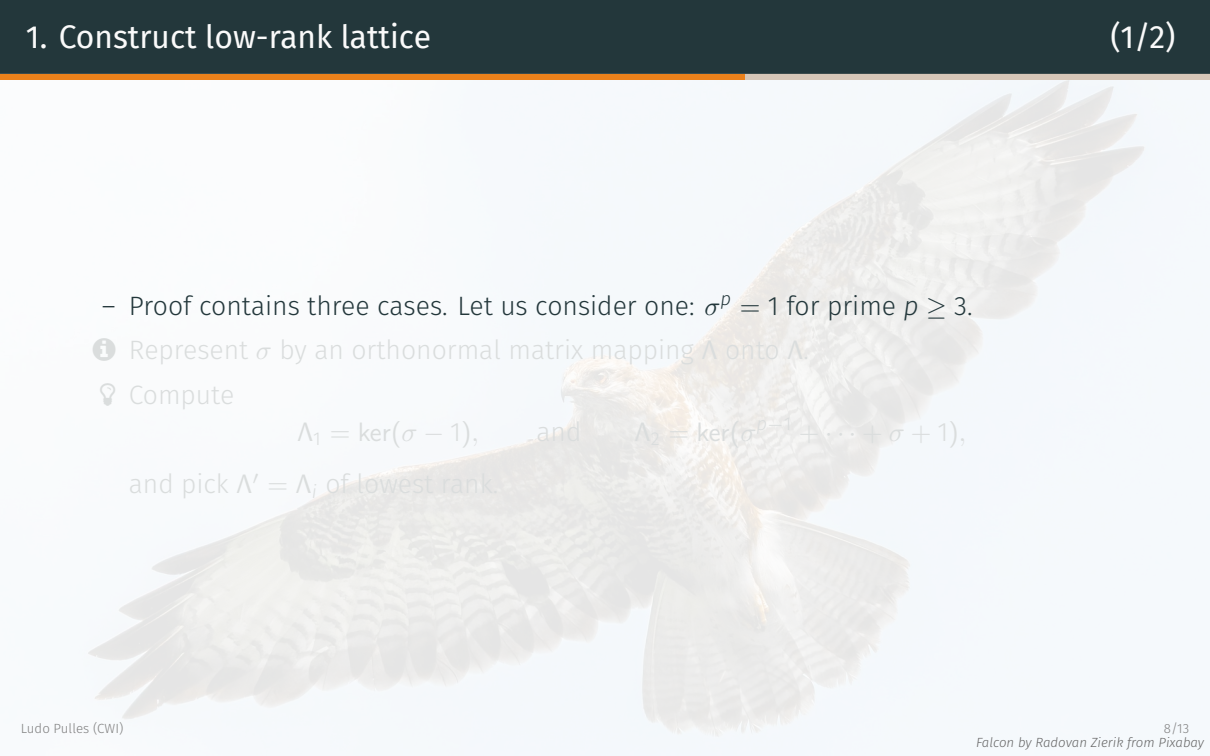
+

Having $\sigma \notin G$, reduces bit security of SMLIP by *at least 50%*.

Bonus: one uniformly sampled σ solves SMLIP in poly(n) time.

Red hawk by VIT DUCKEN from Pixabay

1. Construct low-rank lattice (1/2)



- Proof contains three cases. Let us consider one: $\sigma^p = 1$ for prime $p \geq 3$.

Represent σ by an orthonormal matrix mapping Λ onto Λ.

Compute

$$\Lambda_1 = \ker(\sigma - 1), \quad \text{and} \quad \Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1),$$

and pick $\Lambda' = \Lambda_i$ of lowest rank.

Falcon by Radovan Zierik from Pixabay

1. Construct low-rank lattice (1/2)

- Proof contains three cases. Let us consider one: $\sigma^p = 1$ for prime $p \geq 3$.

Represent σ by an orthonormal matrix mapping Λ onto Λ.

Compute

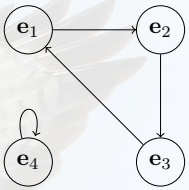
$$\Lambda_1 = \ker(\sigma - 1), \qquad \text{and} \qquad \Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1),$$

and pick $\Lambda' = \Lambda_i$ of lowest rank.

Falcon by Radovan Zierik from Pixabay

1. Construct low-rank lattice — odd case (2/3)

- σ permutes $\mathbf{e}_1, \ldots, \mathbf{e}_n$ (mod sign) giving cycles of size 1 or p.
- σ has at least one p-cycle ($\mathsf{ord}(\sigma) = p$), and one fixed point ($p \nmid n$).



p-cycle, say $\sigma\colon \mathbf{e}_1 \mapsto \mathbf{e}_2 \mapsto \cdots \mapsto \mathbf{e}_p \mapsto \mathbf{e}_1$:

$$\Lambda_1 = \ker(\sigma - 1) = \{(x, \ldots, x) \in \mathbb{Z}^p \mid x \in \mathbb{Z}\} \cong \sqrt{p}\,\mathbb{Z},$$
$$\Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1) = \{(x_1, \ldots, x_p) \in \mathbb{Z}^p \mid x_1 + \cdots + x_p = 0\} \cong A_{p-1}.$$

fixed point, say $\sigma(\mathbf{e}_1) = \mathbf{e}_1$:

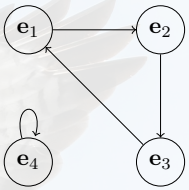
$$\Lambda_1 = \mathbb{Z}, \qquad \Lambda_2 = \{0\}.$$

$$\Lambda_1 \cong \mathbb{Z}^a \oplus \sqrt{p}\,\mathbb{Z}^c, \qquad \Lambda_2 \cong A_{p-1}^c.$$

Observe: (1) $\mathrm{rk}(\Lambda_1) + \mathrm{rk}(\Lambda_2) = n$, (2) $\mathbf{e}_i \in \Lambda_1$, (3) $\mathbf{e}_i - \mathbf{e}_j \in \Lambda_2$.

Falcon by Radovan Zierik from Pixabay

- σ permutes $\mathbf{e}_1, \ldots, \mathbf{e}_n$ (mod sign) giving cycles of size 1 or p.
- σ has at least one p-cycle ($\mathsf{ord}(\sigma) = p$), and one fixed point ($p \nmid n$).



p-cycle, say $\sigma : \mathbf{e}_1 \mapsto \mathbf{e}_2 \mapsto \cdots \mapsto \mathbf{e}_p \mapsto \mathbf{e}_1$:

$$\Lambda_1 = \ker(\sigma - 1) = \{(x, \ldots, x) \in \mathbb{Z}^p | x \in \mathbb{Z}\} \cong \sqrt{p}\,\mathbb{Z},$$
$$\Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1) = \{(x_1, \ldots, x_p) \in \mathbb{Z}^p | x_1 + \cdots + x_p = 0\} \cong A_{p-1}.$$

fixed point, say $\sigma(\mathbf{e}_1) = \mathbf{e}_1$:

$$\Lambda_1 = \mathbb{Z}, \qquad \Lambda_2 = \{0\}.$$

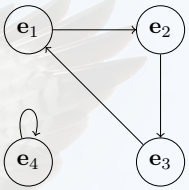
+

$$\Lambda_1 \cong \mathbb{Z}^a \oplus \sqrt{p}\,\mathbb{Z}^c, \qquad \Lambda_2 \cong A_{p-1}^c.$$

Observe: (1) $\mathsf{rk}(\Lambda_1) + \mathsf{rk}(\Lambda_2) = n$, (2) $\mathbf{e}_i \in \Lambda_1$, (3) $\mathbf{e}_i - \mathbf{e}_j \in \Lambda_2$.

Falcon by Radovan Zierik from Pixabay

- σ permutes $\mathbf{e}_1, \ldots, \mathbf{e}_n$ (mod sign) giving cycles of size 1 or p.
- σ has at least one p-cycle ($\mathsf{ord}(\sigma) = p$), and one fixed point ($p \nmid n$).



p-cycle, say $\sigma : \mathbf{e}_1 \mapsto \mathbf{e}_2 \mapsto \cdots \mapsto \mathbf{e}_p \mapsto \mathbf{e}_1$:

$$\Lambda_1 = \ker(\sigma - 1) = \{(x, \ldots, x) \in \mathbb{Z}^p | x \in \mathbb{Z}\} \cong \sqrt{p}\,\mathbb{Z},$$
$$\Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1) = \{(x_1, \ldots, x_p) \in \mathbb{Z}^p | x_1 + \cdots + x_p = 0\} \cong A_{p-1}.$$

fixed point, say $\sigma(\mathbf{e}_1) = \mathbf{e}_1$:

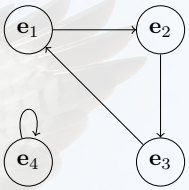
$$\Lambda_1 = \mathbb{Z}, \qquad \Lambda_2 = \{\mathbf{0}\}.$$

$$\Lambda_1 \cong \mathbb{Z}^a \oplus \sqrt{p}\,\mathbb{Z}^c, \qquad \Lambda_2 \cong A_{p-1}^c.$$

Observe: (1) $\mathsf{rk}(\Lambda_1) + \mathsf{rk}(\Lambda_2) = n$, (2) $\mathbf{e}_i \in \Lambda_1$, (3) $\mathbf{e}_i - \mathbf{e}_j \in \Lambda_2$.

Falcon by Radovan Zierik from Pixabay

- σ permutes $\mathbf{e}_1, \ldots, \mathbf{e}_n$ (mod sign) giving cycles of size 1 or p.
- σ has at least one p-cycle ($\mathsf{ord}(\sigma) = p$), and one fixed point ($p \nmid n$).



p-cycle, say $\sigma : \mathbf{e}_1 \mapsto \mathbf{e}_2 \mapsto \cdots \mapsto \mathbf{e}_p \mapsto \mathbf{e}_1$:

$$\Lambda_1 = \ker(\sigma - 1) = \{(x, \ldots, x) \in \mathbb{Z}^p | x \in \mathbb{Z}\} \cong \sqrt{p}\,\mathbb{Z},$$
$$\Lambda_2 = \ker(\sigma^{p-1} + \cdots + \sigma + 1) = \{(x_1, \ldots, x_p) \in \mathbb{Z}^p | x_1 + \cdots + x_p = 0\} \cong A_{p-1}.$$

fixed point, say $\sigma(\mathbf{e}_1) = \mathbf{e}_1$:

$$\Lambda_1 = \mathbb{Z}, \qquad \Lambda_2 = \{\mathbf{0}\}.$$

+

$$\Lambda_1 \cong \mathbb{Z}^a \oplus \sqrt{p}\,\mathbb{Z}^c, \qquad \Lambda_2 \cong A_{p-1}^c.$$

Observe: (1) $\mathsf{rk}(\Lambda_1) + \mathsf{rk}(\Lambda_2) = n$, (2) $\mathbf{e}_i \in \Lambda_1$, (3) $\mathbf{e}_i - \mathbf{e}_j \in \Lambda_2$.

Falcon by Radovan Zierik from Pixabay

2. Unusual-SVP heuristics

- The BKZ algorithm with *block size* β solves SVP on Λ' if β large enough.

BKZ runtime is $\approx 2^{c\beta}$.

- Short vector $\mathbf{v} \in \Lambda'$ with $\|\mathbf{v}\| \in \{1, \sqrt{2}\} \implies \beta \approx k/2$ solves SVP on Λ'. (heuristically)
- Similarly, $\beta \approx n/2$ solves SVP on $\mathbb{Z}^n$.

Runtime has quadratic speedup:
from $2^{cn/2}$ for $\mathbb{Z}^n$
to $\leq 2^{cn/4}$ for Λ'

Northern harrier by Iulian Ursache from Pixabay

3. Near-shortest vector solves smLIP

Case 1: $\|\mathbf{v}\| = 1$

Lemma. Group $G = \langle \zeta, \omega \rangle$ acts transitively on all the $2n$ shortest vectors of $\mathbb{Z}^n$.

$\implies$ G maps $\mathbf{v}$ to all n elementary vectors of $\mathbb{Z}^n$.

Case 2: $\|\mathbf{v}\|^2 = 2$

Reduce to case 1, see paper.

Bonus: random automorphism

Osprey & fish by Andy Morffew from PxHere

Bonus: uniformly random automorphism

$$\sigma \text{ (modulo signs)} \sim \mathrm{U}(S_n).$$

- A random permutation in S_n has $\approx \log(n)$ cycles in expectation.
  $\implies \Lambda_1 = \ker(\sigma - 1)$ is of expected rank $\approx \log(n)$ (ignoring cycles $\mathbf{e}_1 \mapsto \cdots \mapsto -\mathbf{e}_1$).
- With prob. $1 - 1/e \approx 63\%$, $\pm\sigma$ has a fixed point $e_i \mapsto e_i$,
  $\implies \mathbf{e}_i \in \ker(\pm\sigma - 1)$ with norm 1.

For a constant fraction of σ, we can solve SMLIP in poly(n) time,
by solving SVP on $\ker(\sigma - 1)$.

Bonus: uniformly random automorphism

$$\sigma \text{ (modulo signs)} \sim \mathrm{U}(S_n).$$

- A random permutation in S_n has $\approx \log(n)$ cycles in expectation.
  $\implies \Lambda_1 = \ker(\sigma - 1)$ is of expected rank $\approx \log(n)$ (ignoring cycles $\mathbf{e}_1 \mapsto \cdots \mapsto -\mathbf{e}_1$).
- With prob. $1 - 1/e \approx 63\%$, $\pm\sigma$ has a fixed point $e_i \mapsto e_i$,
  $\implies \mathbf{e}_i \in \ker(\pm\sigma - 1)$ with norm 1.

For a constant fraction of σ, we can solve SMLIP in poly(n) time,
by solving SVP on $\ker(\sigma - 1)$.

Conclusion

> “ Finding an automorphism $\sigma \notin \langle \zeta, \omega_Q \rangle$ is almost as hard as smLIP. ”

👍 omSVP using $G = \langle \zeta, \omega_Q \rangle$, cannot be broken by a ‘missed’ automorphism.

Conclusion

❝ Finding an automorphism $\sigma \notin \langle \zeta, \omega_Q \rangle$
is almost as hard as smLIP. ❞

👍 omSVP using $G = \langle \zeta, \omega_Q \rangle$, cannot be broken by a 'missed' automorphism.

Thank you! Questions?

CiC.iacr.org/p/2/2/20

lnp@cwi.nl