HILA5 Pindakaas: On the CCA security of lattice-based encryption with error correction

Daniel J. Bernstein[1] Leon Groot Bruinderink[2]
Tanja Lange[2] Lorenz Panny[2]

[1] University of Illinois at Chicago
[2] Technische Universiteit Eindhoven

Marrakesh, 07 May 2018

Motivation

- HILA5 is a RLWE-based KEM submitted to NISTPQC.

> *This design also provides **IND-CCA secure** KEM-DEM public key encryption if used in conjunction with an appropriate AEAD such as NIST approved AES256-GCM.*
>
> — HILA5 NIST submission document (v1.0)

- Decapsulation much faster than encapsulation (and faster than any other scheme).
- No mention of a CCA transform (e.g. Fujisaki–Okamoto).

Noisy Diffie–Hellman

- Have a ring $R = \mathbb{Z}[x]/(q, \varphi)$ where $q \in \mathbb{Z}$ and $\overset{\text{degree } n}{\varphi} \in \mathbb{Z}[x]$.[1]
- Let χ be a narrow distribution around $0 \in R$.
- Fix some "random" element $g \in R$.

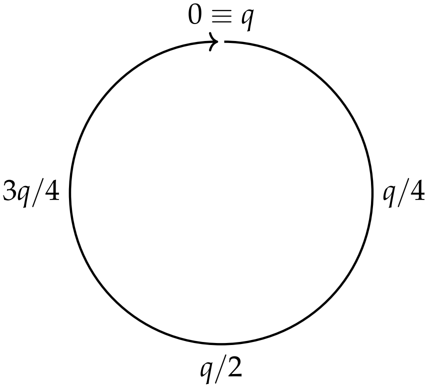
$$\begin{tikzcd}
a, e \leftarrow \chi^n & b, e' \leftarrow \chi^n \\
A = ga + e \arrow[dr] & B = gb + e' \arrow[dl] \\
S = Ba = gab + e'a & S' = Ab = gab + eb
\end{tikzcd}$$

$$\implies S - S' = e'a - eb \underset{\substack{\uparrow \\ \chi \text{ small}}}{\approx} 0$$

[1] There exist other rings that work.

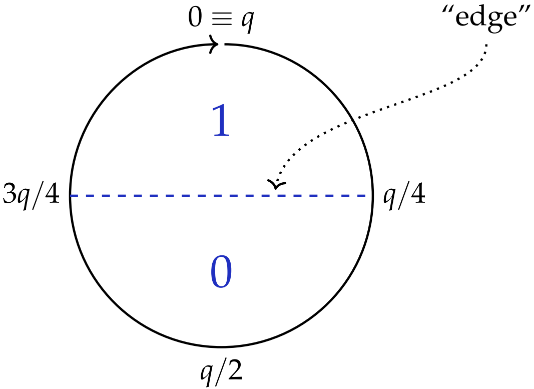
Reconciliation

Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



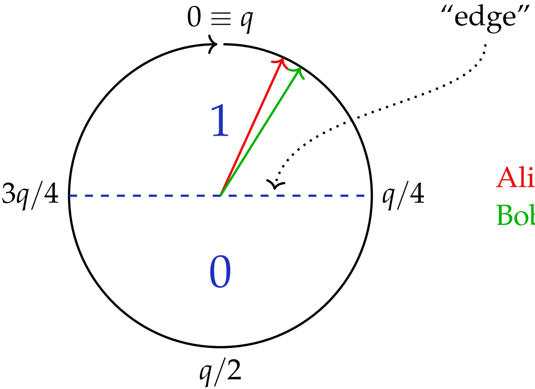
Reconciliation

Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



Reconciliation

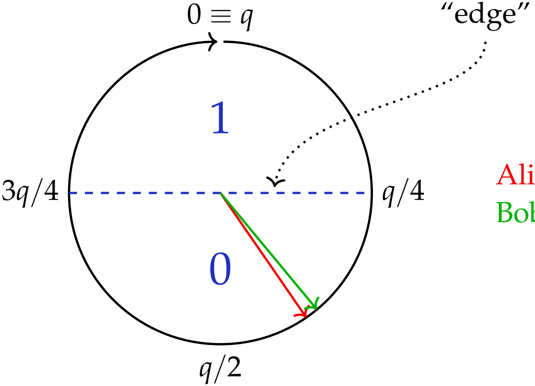
Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



Alice: 1
Bob: 1

Reconciliation

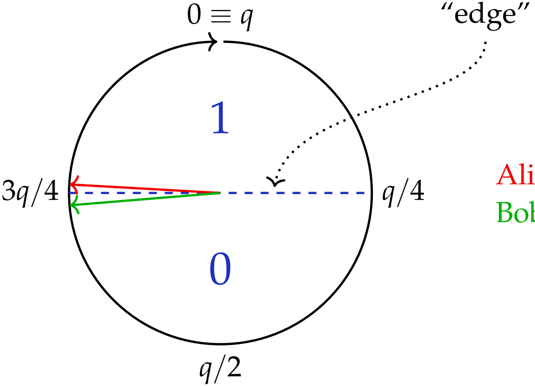
Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



Alice: 0
Bob: 0

Reconciliation

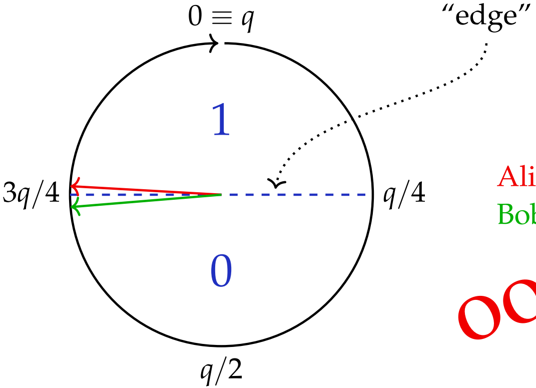
Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



Alice: 1
Bob: 0

Reconciliation

Alice and Bob obtain close secret vectors $S, S' \in (\mathbb{Z}/q)^n$.
How to map coefficients to bits?



Alice: 1
Bob: 0

oops!

Reconciliation

Mapping coefficients to bits using fixed intervals is bad.

Reconciliation

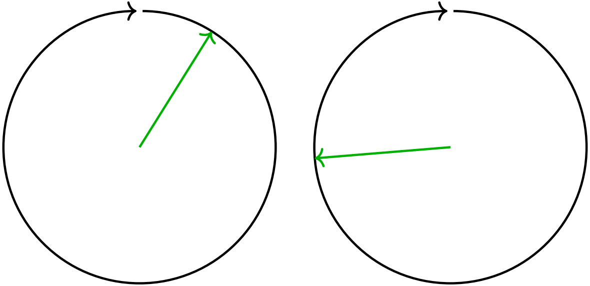
Mapping coefficients to bits using fixed intervals is bad.

Better: Bob chooses a mapping based on his coefficient and tells Alice which mapping he used.

Reconciliation

Mapping coefficients to bits using fixed intervals is bad.

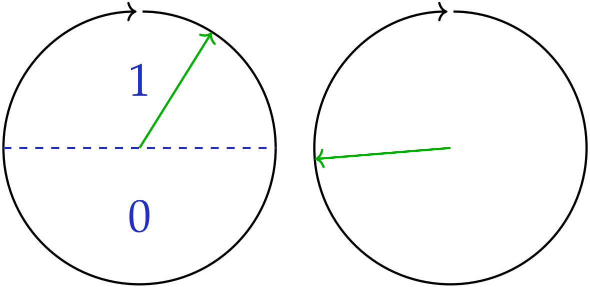
Better: Bob chooses a mapping based on his coefficient and tells Alice which mapping he used.



Reconciliation

Mapping coefficients to bits using fixed intervals is bad.

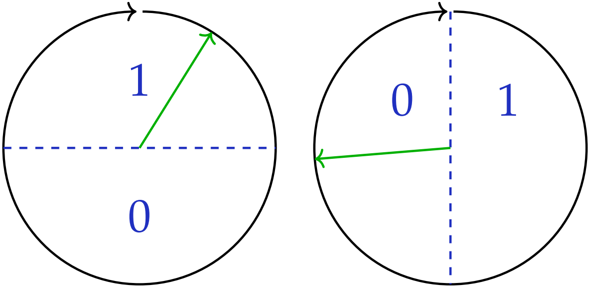
Better: Bob chooses a mapping based on his coefficient and tells Alice which mapping he used.



Reconciliation

Mapping coefficients to bits using fixed intervals is bad.

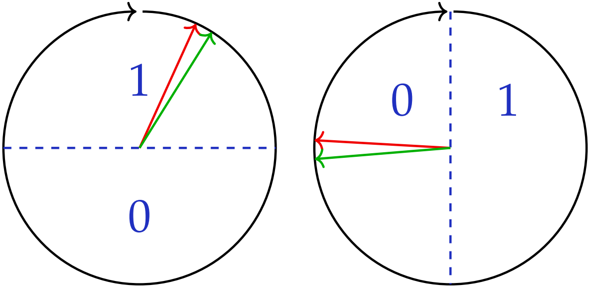
Better: Bob chooses a mapping based on his coefficient and tells Alice which mapping he used.



Reconciliation

Mapping coefficients to bits using fixed intervals is bad.

Better: Bob chooses a mapping based on his coefficient and tells Alice which mapping he used.

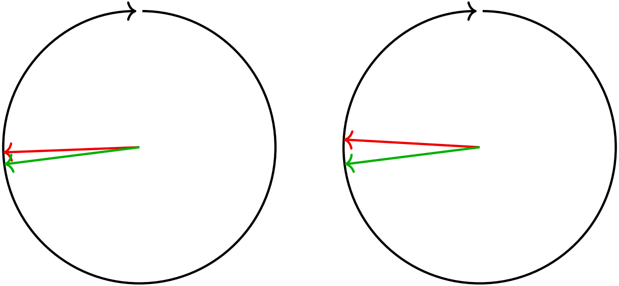


Fluhrer's attack https://ia.cr/2016/085

Problem: Evil Bob can trick Alice into leaking information by deliberately using the wrong mapping for one coefficient.

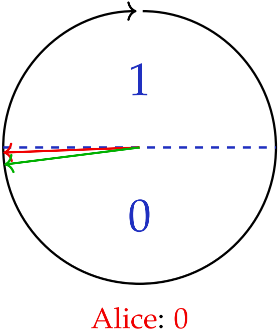
Fluhrer’s attack https://ia.cr/2016/085

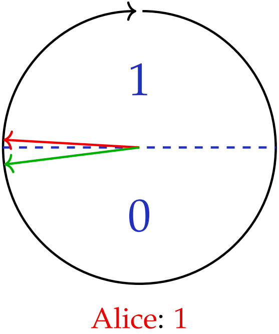
Problem: Evil Bob can trick Alice into leaking information by deliberately using the wrong mapping for one coefficient.



Fluhrer's attack https://ia.cr/2016/085

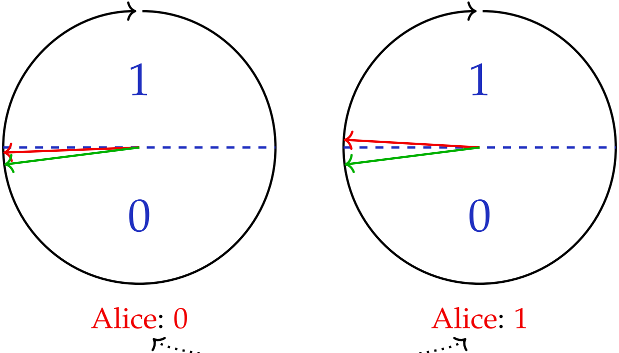
Problem: Evil Bob can trick Alice into leaking information by deliberately using the wrong mapping for one coefficient.





Fluhrer's attack https://ia.cr/2016/085

Problem: Evil Bob can trick Alice into leaking information by deliberately using the wrong mapping for one coefficient.



Alice: 0

Alice: 1

Evil Bob can distinguish these cases!

(He knows all the other key bits.)

Chosen-ciphertext information leaks

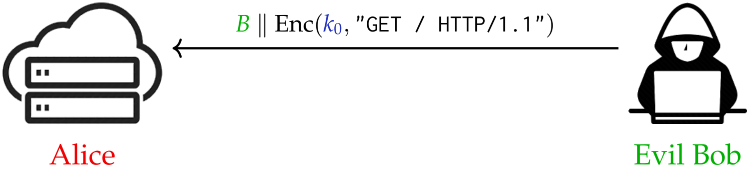
Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.

Alice

Evil Bob

Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



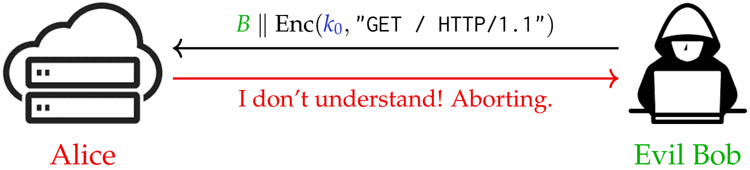
$B \,\|\, \mathrm{Enc}(k_0, \texttt{"GET / HTTP/1.1"})$

Alice

Evil Bob

Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



$B \parallel \mathrm{Enc}(k_0, \texttt{"GET / HTTP/1.1"})$

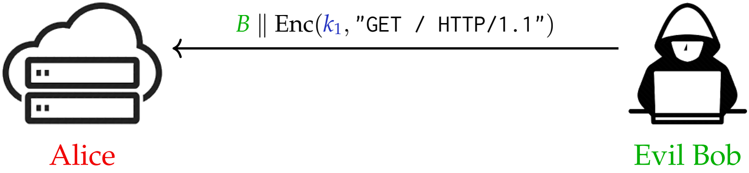
I don't understand! Aborting.

Alice

Evil Bob

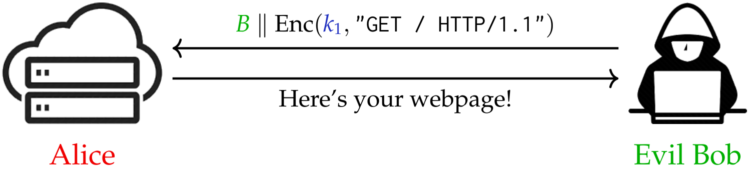
Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.

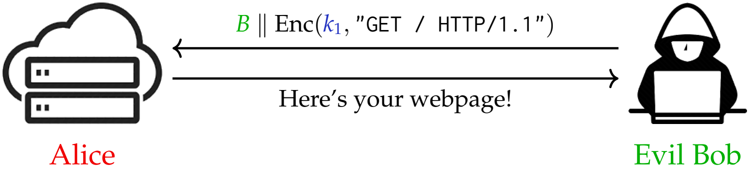


$B \parallel \mathrm{Enc}(k_1, \texttt{"GET / HTTP/1.1"})$

Here's your webpage!

Chosen-ciphertext information leaks

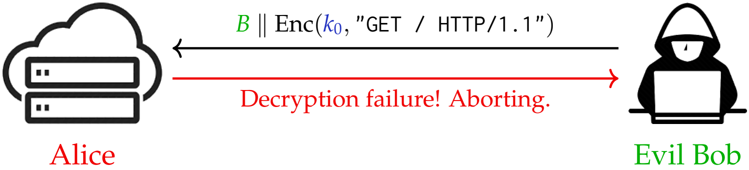
Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



$\implies$ Bob learns that $k = k_1$.

Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



$B \parallel \mathrm{Enc}(k_0, \texttt{"GET / HTTP/1.1"})$

Decryption failure! Aborting.

Alice

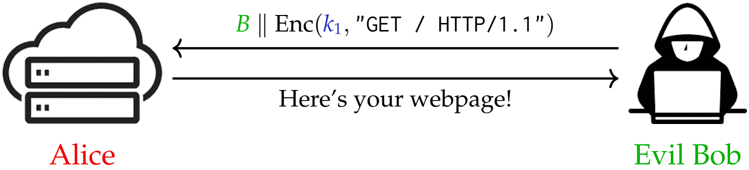
Evil Bob

$\Longrightarrow$ Bob learns that $k = k_1$.

This still works if Enc is an authenticated symmetric cipher!

Chosen-ciphertext information leaks

Evil Bob has two guesses k_0, k_1 for what Alice's key k will be given his manipulated public key B.



$B \,\|\, \mathrm{Enc}(k_1, \texttt{"GET / HTTP/1.1"})$

Here's your webpage!

Alice

Evil Bob

$\Longrightarrow$ Bob learns that $k = k_1$.

This still works if Enc is an authenticated symmetric cipher!

Fluhrer's attack https://ia.cr/2016/085

Adaptive chosen-ciphertext attack against static keys.

Fluhrer's attack https://ia.cr/2016/085

Adaptive chosen-ciphertext attack against static keys.

Recall that Alice's "shared" secret is $gab + e'a$.

Fluhrer's attack https://ia.cr/2016/085

Adaptive chosen-ciphertext attack against static keys.

Recall that Alice's "shared" secret is $gab + e'a$.

Suppose Evil Bob knows b_δ such that $gab_\delta[0] = \overset{\text{edge}}{M} + \delta$.
$\Longrightarrow$ Querying Alice with $b = b_\delta$ leaks whether $-e'a[0] > \delta$.

Fluhrer's attack `https://ia.cr/2016/085`

Adaptive chosen-ciphertext attack against static keys.

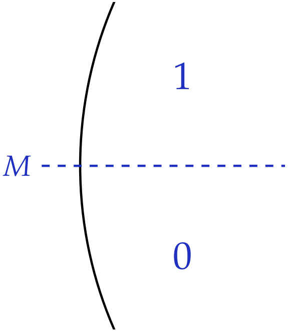
Recall that Alice's "shared" secret is $gab + e'a$.

Suppose Evil Bob knows b_δ such that $gab_\delta[0] = \overset{\text{edge}}{M} + \delta$.
$\Longrightarrow$ Querying Alice with $b = b_\delta$ leaks whether $-e'a[0] > \delta$.

Structure of R
$\rightsquigarrow$ Can choose e' such that $e'a[0] = a[i]$ to recover all of a.

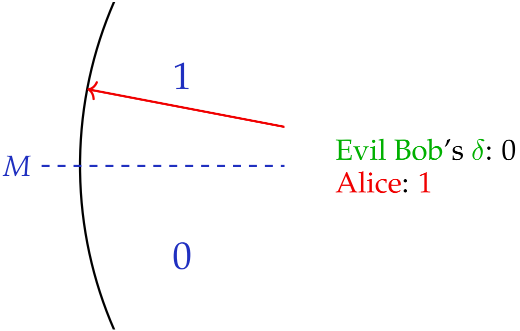
Fluhrer's attack https://ia.cr/2016/085

Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



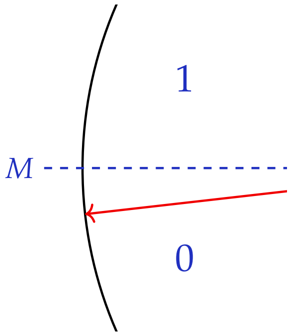
Fluhrer's attack https://ia.cr/2016/085

Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Fluhrer's attack https://ia.cr/2016/085

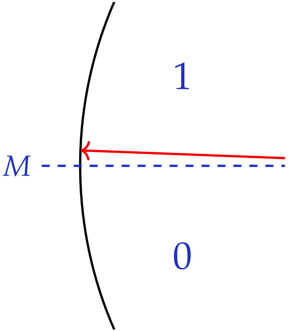
Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Evil Bob's δ: -8
Alice: 0

Fluhrer's attack https://ia.cr/2016/085

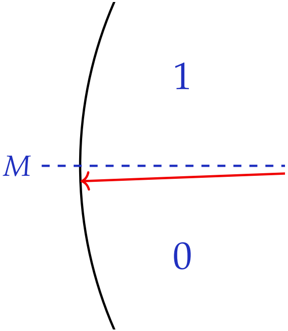
Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Evil Bob's δ: -4
Alice: 1

Fluhrer's attack `https://ia.cr/2016/085`

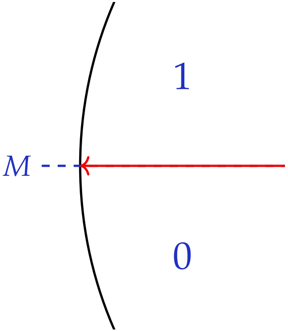
Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Evil Bob's δ: -6
Alice: 0

Fluhrer's attack https://ia.cr/2016/085

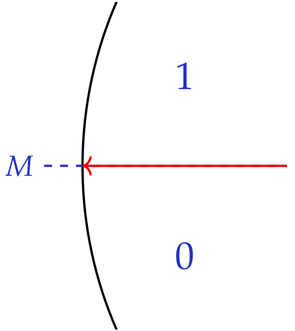
Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Evil Bob's δ: -5
Alice: 1

Fluhrer's attack https://ia.cr/2016/085

Querying Alice with $b = b_\delta$ and $e' = 1$ leaks whether $-a[0] > \delta$.



Evil Bob's δ: -5
Alice: 1

$\implies$ Evil Bob learns that $a[0] = 5$.

Our work

Adaption of Fluhrer's attack to HILA5 and analysis

HILA5 https://ia.cr/2017/424 https://github.com/mjosaarinen/hila5

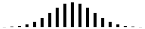
- Standard noisy Diffie–Hellman with new reconciliation.

HILA5 https://ia.cr/2017/424 https://github.com/mjosaarinen/hila5

- Standard noisy Diffie–Hellman with new reconciliation.
- Ring: $\mathbb{Z}[x]/(q, x^{1024}+1)$ where $q = 12289$.[1]
- Noise distribution χ: Ψ_{16}.[1] on $\{-16, \ldots, 16\}$

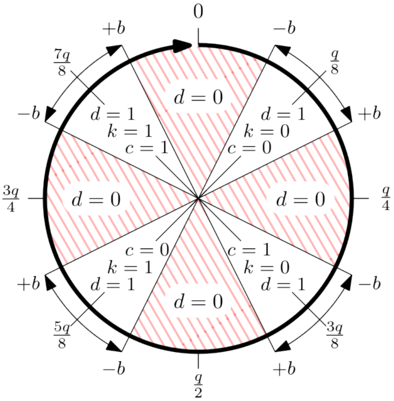
[1] same as New Hope.

HILA5 https://ia.cr/2017/424 https://github.com/mjosaarinen/hila5

- Standard noisy Diffie–Hellman with new reconciliation.
- Ring: $\mathbb{Z}[x]/(q, x^{1024}+1)$ where $q = 12289$.[1]
- Noise distribution χ: Ψ_{16}.[1]  on $\{-16, \ldots, 16\}$
- New reconciliation mechanism:
  - Only use "safe bits" that are far from an edge.
  - Additionally apply an error-correcting code.

[1] same as New Hope.

HILA5's reconciliation



(picture: HILA5 documentation)

For each coefficient:

$d = 0$: Discard coefficient.

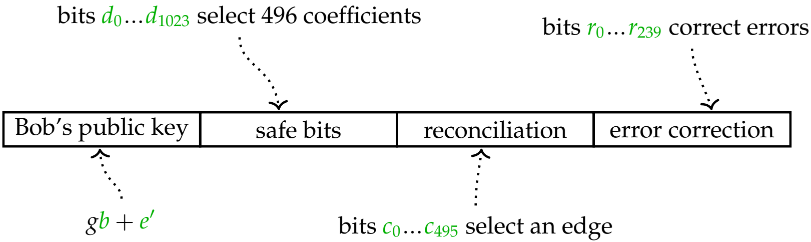
$d = 1$: Send reconciliation information c; use for key bit k.

Edges:

$c = 0$: $\lceil 3q/8 \rfloor \ldots \lceil 7q/8 \rfloor \rightsquigarrow k = 0.$
$\lceil 7q/8 \rfloor \ldots \lceil 3q/8 \rfloor \rightsquigarrow k = 1.$

$c = 1$: $\lceil q/8 \rfloor \ldots \lceil 5q/8 \rfloor \rightsquigarrow k = 0.$
$\lceil 5q/8 \rfloor \ldots \lceil\ q/8 \rfloor \rightsquigarrow k = 1.$

HILA5's packet format



bits $d_0 \ldots d_{1023}$ select 496 coefficients

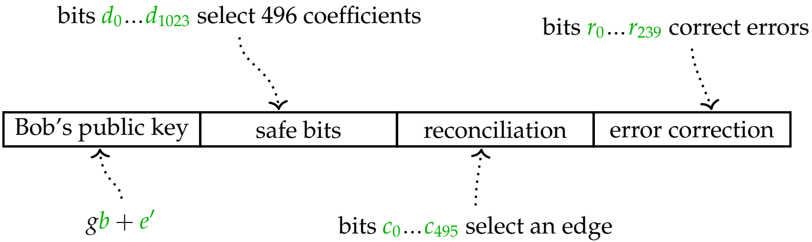
bits $r_0 \ldots r_{239}$ correct errors

Bob's public key	safe bits	reconciliation	error correction

$gb + e'$

bits $c_0 \ldots c_{495}$ select an edge

HILA5's packet format



bits $d_0 \ldots d_{1023}$ select 496 coefficients

bits $r_0 \ldots r_{239}$ correct errors

Bob's public key | safe bits | reconciliation | error correction

$gb + e'$

bits $c_0 \ldots c_{495}$ select an edge

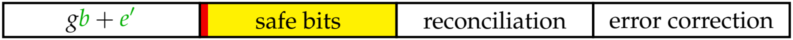
We're going to manipulate each of these parts.

Unsafe bits

$gb + e'$	safe bits	reconciliation	error correction

We want to attack the first coefficient.

Unsafe bits



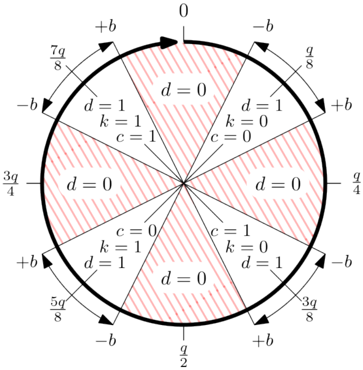
$gb + e'$ | safe bits | reconciliation | error correction

We want to attack the first coefficient.
$\Longrightarrow$ Force $d_0 = 1$ to make Alice use it.

Living on the edge

$gb + e'$	safe bits	reconciliation	error correction

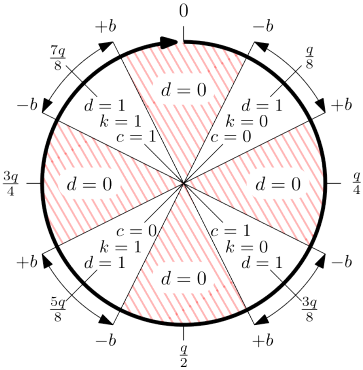
We want to attack the edge at $M = \lceil q/8 \rfloor$.



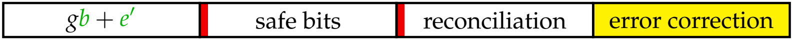
Living on the edge

$gb + e'$	safe bits	reconciliation	error correction

We want to attack the edge at $M = \lceil q/8 \rfloor$. $\implies$ Force $c_0 = 1$.



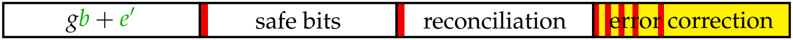
Making errors



- HILA5 uses a custom linear error-correcting code XE5.
- Encrypted (XOR) using part of Bob's shared secret S'.
- Ten variable-length codewords $R_0 \dots R_9$.
- Alice corrects $S[0]$ using the first bit of each R_i.
- Capable of correcting (at least) 5-bit errors.

We want to keep errors in $S[0]$.

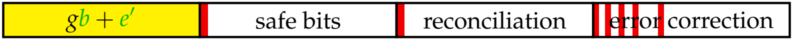
Making errors



- HILA5 uses a custom linear error-correcting code XE5.
- Encrypted (XOR) using part of Bob's shared secret S'.
- Ten variable-length codewords $R_0 \ldots R_9$.
- Alice corrects $S[0]$ using the first bit of each R_i.
- Capable of correcting (at least) 5-bit errors.

We want to keep errors in $S[0]$. $\implies$ Flip the first bit of $R_0 \ldots R_4$!

All coefficients for the price of one



$gb + e'$

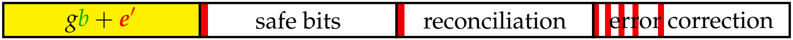
safe bits

reconciliation

error correction

Our binary search recovers $e'a[0]$ from $gab_\delta + e'a$ by varying δ.
How to get $a[1]$, $a[2]$, ..?

All coefficients for the price of one



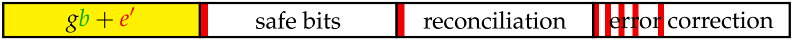
$gb + e'$ | safe bits | reconciliation | error correction

Our binary search recovers $e'a[0]$ from $gab_\delta + e'a$ by varying δ.
How to get $a[1]$, $a[2]$, ..?

By construction of $R = \mathbb{Z}[x]/(q, x^{1024} + 1)$,
Evil Bob can rotate $a[i]$ into $e'a[0]$ by setting $e' = -x^{1024-i}$.

Running the search for all i yields all coefficients of a.

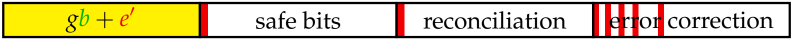
Evil Bob needs evil b_δ



$gb + e'$ | safe bits | reconciliation | error correction

Recall that Evil Bob needs b_δ such that $gab_\delta[0] = M + \delta$.
How to obtain b_δ without knowing a?

Evil Bob needs evil b_δ

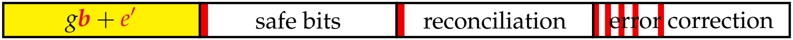


$gb + e'$ safe bits reconciliation error correction

Recall that Evil Bob needs b_δ such that $gab_\delta[0] = M + \delta$.
How to obtain b_δ without knowing a?

$\Longrightarrow$ Guess b_0 based on Alice's public key $A = ga + e$:

Evil Bob needs evil b_δ



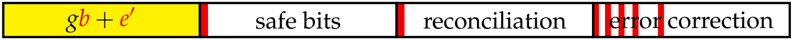
Recall that Evil Bob needs b_δ such that $gab_\delta[0] = M + \delta$.
How to obtain b_δ without knowing a?

$\Longrightarrow$ Guess b_0 based on Alice's public key $A = ga + e$:

If b_0 has two entries ±1 and $(Ab_0)[0] = M$, then

$$\Pr_{e \leftarrow \chi^n}[gab_0[0] = M] = \Pr_{x,y \leftarrow \Psi_{16}}[x + y = 0] \approx 9.9\%.$$

Evil Bob needs evil b_δ



$gb + e'$ | safe bits | reconciliation | error correction

Recall that Evil Bob needs b_δ such that $gab_\delta[0] = M + \delta$.
How to obtain b_δ without knowing a?

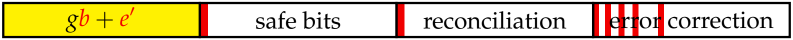
$\Longrightarrow$ Guess b_0 based on Alice's public key $A = ga + e$:

If b_0 has two entries ±1 and $(Ab_0)[0] = M$, then

$$\Pr_{e \leftarrow \chi^n}[gab_0[0] = M] = \Pr_{x,y \leftarrow \Psi_{16}}[x + y = 0] \approx 9.9\%.$$

For all other δ, set $b_\delta := (1 + \delta M^{-1} \bmod q) \cdot b_0$.
This works because $M^{-1} \bmod q = -8$ is small here.

Evil Bob needs evil b_δ



$gb + e'$

safe bits

reconciliation

error correction

Recall that Evil Bob needs b_δ such that $gab_\delta[0] = M + \delta$.
How to obtain b_δ without knowing a?

$\Longrightarrow$ Guess b_0 based on Alice's public key $A = ga + e$:

If b_0 has two entries ±1 and $(Ab_0)[0] = M$, then

$$\Pr_{e \leftarrow \chi^n}[gab_0[0] = M] = \Pr_{x,y \leftarrow \Psi_{16}}[x + y = 0] \approx 9.9\%.$$

For all other δ, set $b_\delta := (1 + \delta M^{-1} \bmod q) \cdot b_0$.
This works because $M^{-1} \bmod q = -8$ is small here.

If b_0 was wrong, the recovered coefficients are all 0 or -1.
$\Longrightarrow$ easily detectable.

Implementation

- Our code[2] attacks the HILA5 reference implementation.
- 100% success rate in our experiments.
- Less than 6000 queries (virtually always).

  (Note: Evil Bob could recover fewer coefficients and compute the rest by solving a lattice problem of reduced dimension.)

[2] `https://helaas.org/hila5-20171218.tar.gz`

Thank you!

Bonus slide in case somebody asks...

TU/E KWALITEITSGARANTIE
NOF
SMEUÏG TOT OP DE BODEM
BRON VAN BOUWSTOFFEN*
BRON VAN VITAMINEN &
NOF
HILA5
PINDAKAAS
Per
Per 100 g:
STUKJES PINDA