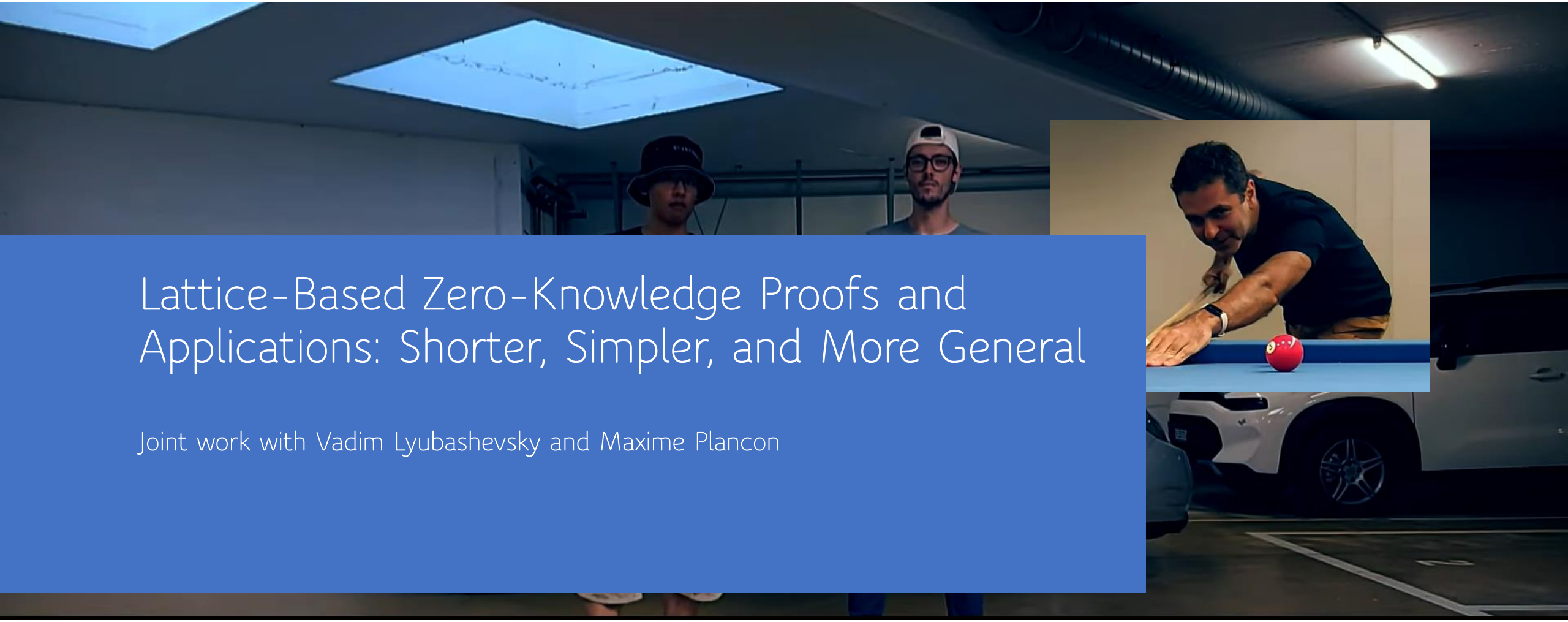
ETH zürich

IBM Research | Zurich



Lattice-Based Zero-Knowledge Proofs and Applications: Shorter, Simpler, and More General

Joint work with Vadim Lyubashevsky and Maxime Plancon

Zero-knowledge proofs

Zero-knowledge proofs

statement

"I got even funnier lines in the full version of the song!"

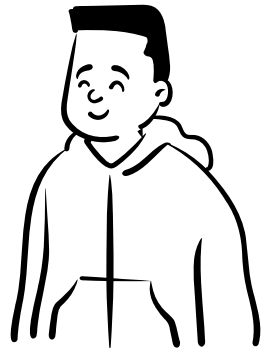
Zero-knowledge proofs

statement

"I got even funnier lines in the full version of the song!"

Prover

Verifier



Zero-knowledge proofs

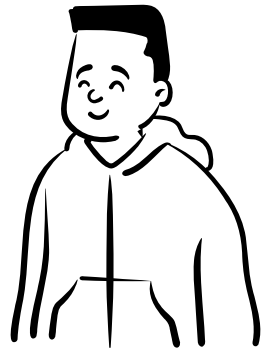
statement

"I got even funnier lines in the full version of the song!"

Witness

Prover

Verifier



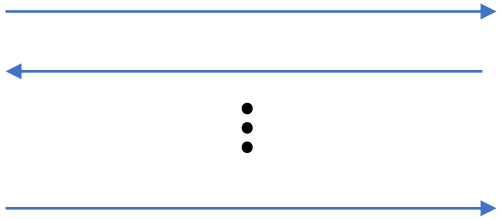
Zero-knowledge proofs

statement

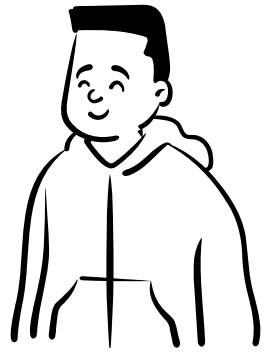
Witness

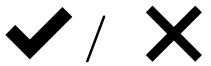
Prover

"I got even funnier lines in the full version of the song!"



Verifier

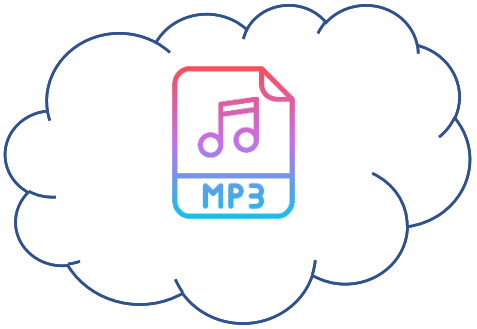




Zero-knowledge proofs

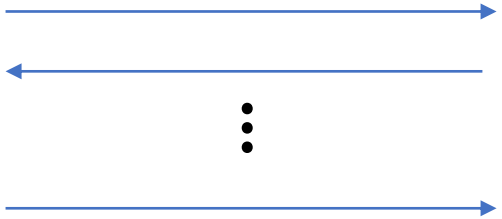
statement

Witness

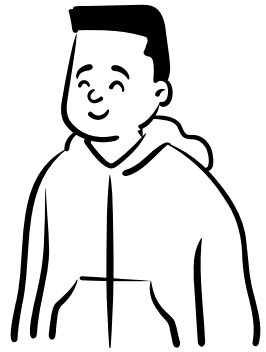


Prover

"I got even funnier lines in the full version of the song!"



Verifier



Completeness:
if the witness is valid, the verifier accepts

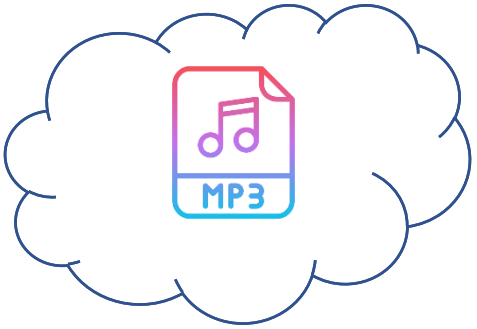
Soundness:
if the witness is invalid, the verifier rejects

Zero-knowledge:
the verifier does not learn anything about the witness from the interaction

Zero-knowledge proofs

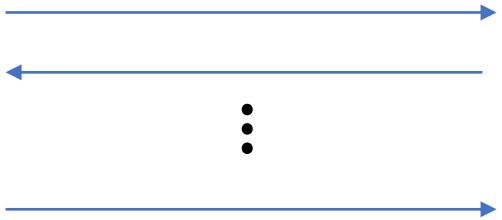
statement

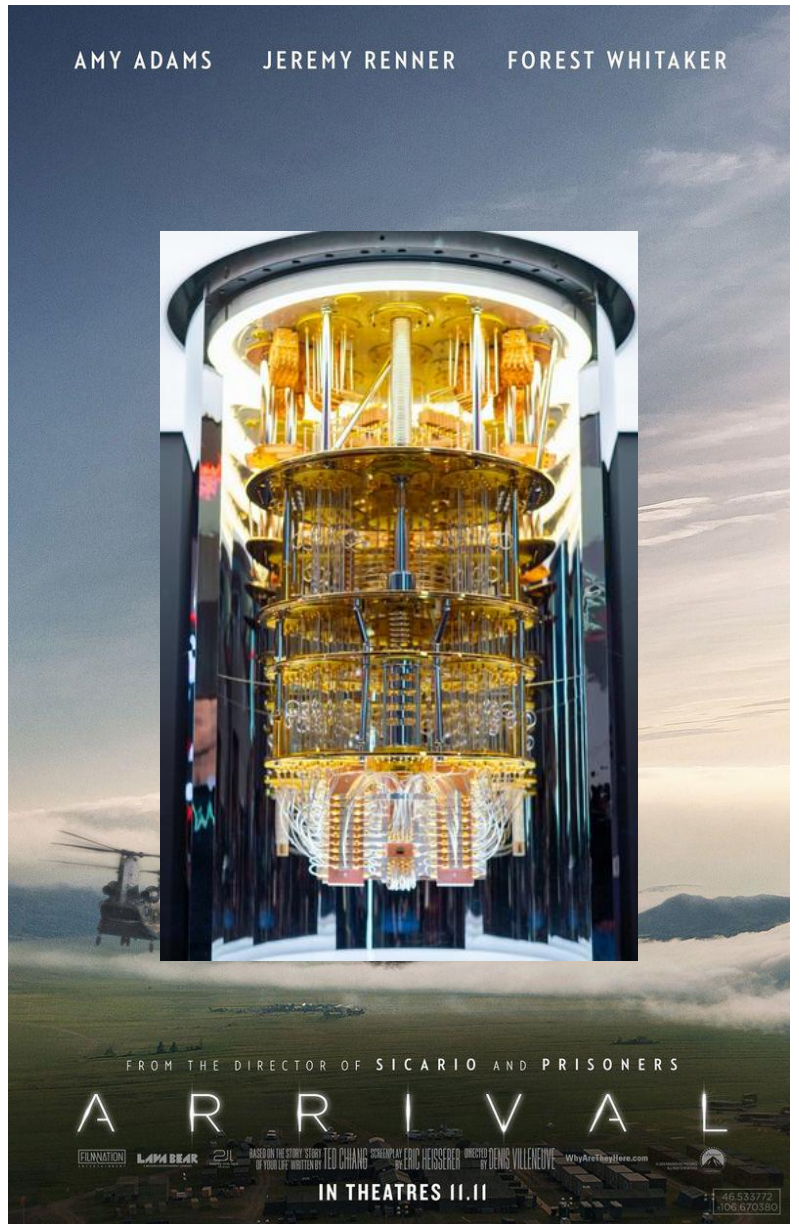
Witness



Prover

"I got even funnier lines in the full version of the song!"





Completeness:
if the witness is valid, the verifier accepts

Soundness:
if the witness is invalid, the verifier rejects

Zero-knowledge:
the verifier does not learn anything about the witness from the interaction

Current state-of-the-art of Quantum-Safe ZK

Scheme	Structure	Asymptotic proof size (witness size = N)	Concrete proof size $N \approx 2^{20}$	Prover runtime
Ligero [AHIV17]	Hash functions	$O(\sqrt{N})$	9MB	38s
Aurora [BCR+19]	Hash functions	$O(\log^2 N)$	170KB	304s
Fractal [COS20]	Hash functions	$O(\log^2 N)$	215KB	184s
[BLNS20]	Lattices	$O(N^{1/d})$	-	-
Lattice Bulletproofs [BLNS20,AL21,ACK21]	Lattices	$O(\log^2 N)$	-	-

Sizes and runtimes taken from [ACMLT22,ISW21, NS22]

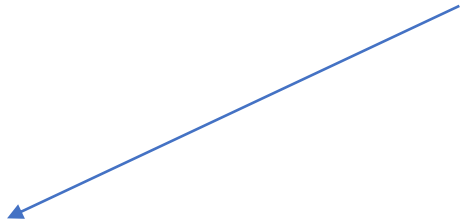
Current state-of-the-art of Quantum-Safe ZK

Scheme	Structure	Asymptotic proof size (witness size = N)	Concrete proof size $N \approx 2^{20}$	Prover runtime
Ligero [AHIV17]	Hash functions	$O(\sqrt{N})$	9MB	38s
Aurora [BCR+19]	Hash functions	$O(\log^2 N)$	170KB	304s
Fractal [COS20]	Hash functions	$O(\log^2 N)$	215KB	184s
[BLNS20]	Lattices	$O(N^{1/d})$	-	-
Lattice Bulletproofs [BLNS20,AL21,ACK21]	Lattices	$O(\log^2 N)$	-	-
[NS22]	Lattices	$O(\sqrt{N})$	6MB	-
[ACLMT22]	Lattices	$polylog(N)$	32MB	-

Sizes and runtimes taken from [ACMLT22,ISW21, NS22]

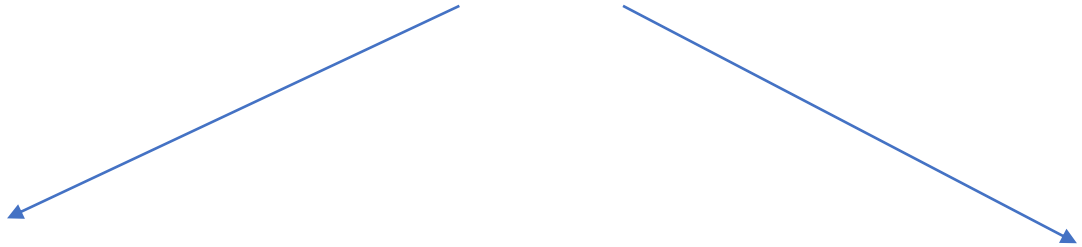
Up next!

Possible research directions (in the lattice world)



Construct succinct (e.g. logarithmic-sized) ZK proofs

Possible research directions (in the lattice world)



Construct succinct (e.g. logarithmic-sized) ZK proofs

Construct **practically** efficient ZK proofs $< 50KB$ for **interesting statements**

Possible research directions (in the lattice world)

Construct succinct (e.g. logarithmic-sized) ZK proofs

Construct **practically** efficient ZK proofs $< 50KB$ for **interesting statements**

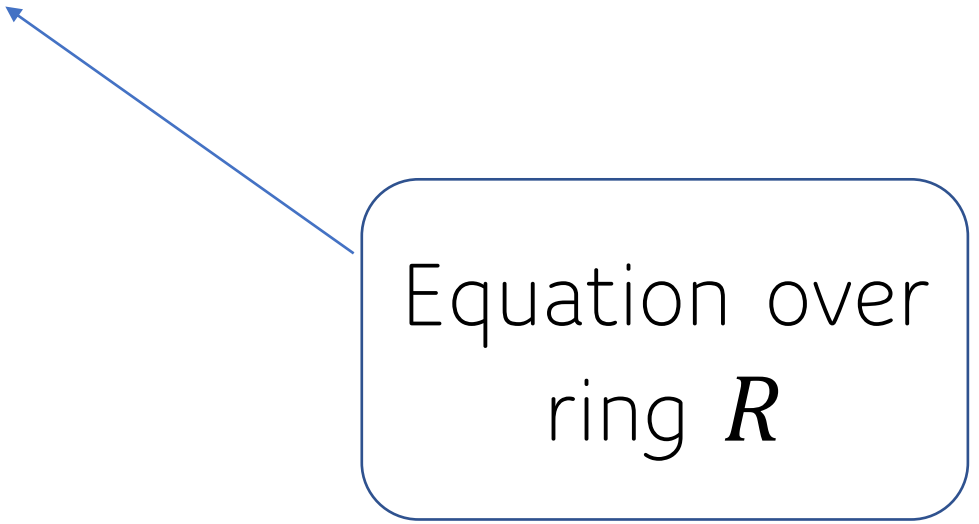
In this talk

What are the interesting statements?

$$\boldsymbol{As = u}$$

What are the interesting statements?

$$\boldsymbol{As = u}$$



What are the interesting statements?

$$\boldsymbol{As = u}$$

Vector $\boldsymbol{s}$ has *small* coefficients e.g. {-1,0,1}

Equation over ring $\boldsymbol{R}$

Exact proofs

Benchmark: prove $\boldsymbol{As} = \boldsymbol{u}$ where $s_i \in \{-1,0,1\}$ and $A \in \mathbb{Z}_q^{1024\times 2048}$ for $q \approx 2^{32}$.

Exact proofs

Benchmark: prove $\boldsymbol{As} = \boldsymbol{u}$ where $s_i \in \{-1,0,1\}$ and $A \in \mathbb{Z}_q^{1024\times 2048}$ for $q \approx 2^{32}$.

Hash-based proof systems

Scheme	Proof size
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB

Sizes taken from [ENS20,LNS21]

Exact proofs

Benchmark: prove $\boldsymbol{As} = \boldsymbol{u}$ where $s_i \in \{-1,0,1\}$ and $A \in \mathbb{Z}_q^{1024\times 2048}$ for $q \approx 2^{32}$.

Hash-based proof systems

Scheme	Proof size
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB

Permutation-based proofs

Scheme	Proof size
Stern proofs (e.g. [Ste93,LNSW13])	3MB
[Beu20]	233KB

Sizes taken from [ENS20,LNS21]

Exact proofs

Benchmark: prove $\boldsymbol{As} = \boldsymbol{u}$ where $s_i \in \{-1,0,1\}$ and $A \in \mathbb{Z}_q^{1024\times 2048}$ for $q \approx 2^{32}$.

Hash-based proof systems

Scheme	Proof size
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB

Permutation-based proofs

Scheme	Proof size
Stern proofs (e.g. [Ste93,LNSW13])	3MB
[Beu20]	233KB

NTT/CRT-packing proofs

Scheme	Proof size
[BLS19,YAZ+19]	384KB
[ALS20,ENS20]	47KB
[LNS21]	33KB

Sizes taken from [ENS20,LNS21]


it's time for some technical overview

Exact proofs via NTT packing

- Consider the standard polynomial ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two and $q = 1 \, (mod \, 2d)$.
- Then, we can write:

$$X^d + 1 = (X - r_1)(X - r_2) \ldots (X - r_d) \, mod \, q.$$

Exact proofs via NTT packing

- Consider the standard polynomial ring $R_q = \mathbb{Z}_q[X]/(X^d + 1)$ where d is a power-of-two and $q = 1\ (mod\ 2d)$.
- Then, we can write:

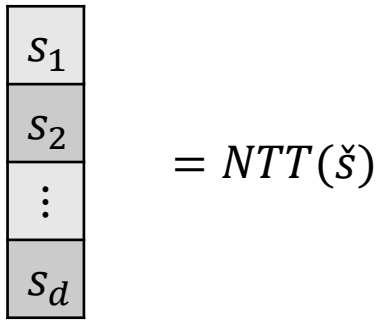
$$X^d + 1 = (X - r_1)(X - r_2) \ldots (X - r_d)\ mod\ q.$$

Given a polynomial $a = a_0 + a_1 X + \ \ldots + a_{d-1} X^{d-1} \in R_q$, define $\mathbf{NTT}(a)$ as the vector $\hat{a} = \left(a(r_1), \ldots, a(r_d)\right) \in \mathbb{Z}_q^d$.

By definition, $\mathbf{NTT}(ab) = \mathbf{NTT}(a) \circ \mathbf{NTT}(b)$.

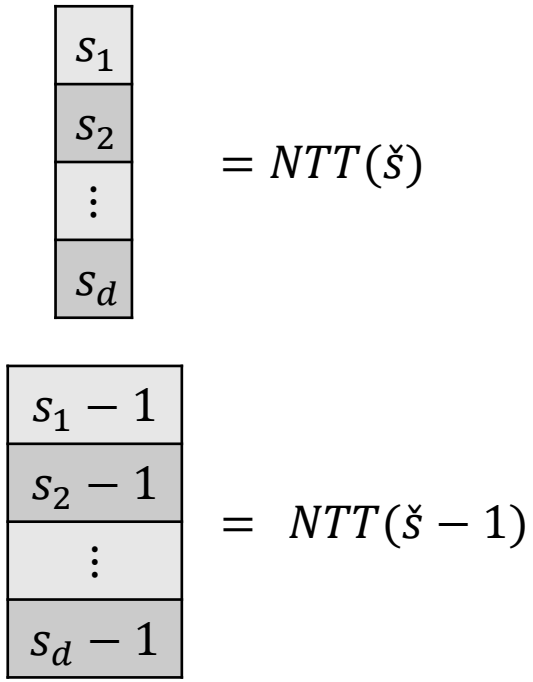
Exact proofs via NTT packing

We want to prove $\boldsymbol{s} \in \{0,1\}^d$.



Exact proofs via NTT packing

We want to prove $\boldsymbol{s} \in \{0,1\}^d$.



Exact proofs via NTT packing

We want to prove $\boldsymbol{s} \in \{0,1\}^d$.

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} s_1 - 1 \\ s_2 - 1 \\ \vdots \\ s_d - 1 \end{bmatrix} \circ \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_d \end{bmatrix} \quad = NTT(\check{s}) \circ NTT(\check{s} - 1) = NTT(\check{s}(\check{s} - 1))$$

Exact proofs via NTT packing

We want to prove $\check{s}(\check{s} - 1) = 0$.

Exact proofs via NTT packing

We want to prove $\check{s}(\check{s}-1)=0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.

Exact proofs via NTT packing

We want to prove $\check{s}(\check{s}-1)=0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.

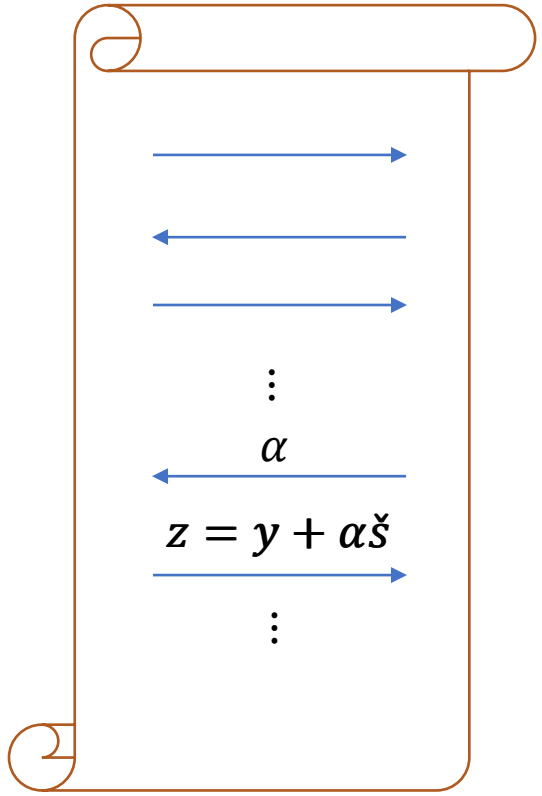
Note that $\check{s}$ has large coefficients.

Hence, we commit to it using the [BDLOP18] homomorphic commitment.

Exact proofs via NTT packing

We want to prove $\check{s}(\check{s}-1)=0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.



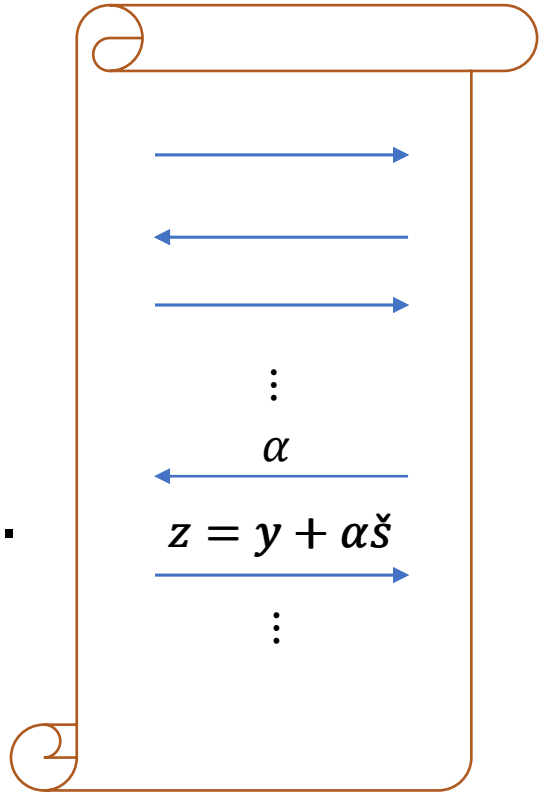
Exact proofs via NTT packing

We want to prove $\check{s}(\check{s}-1)=0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.

The verifier can thus compute:

$$z(z-\alpha) = y^2 + \alpha\cdot(2\check{s}y - y) + \alpha^2\cdot\check{s}(\check{s}-1).$$

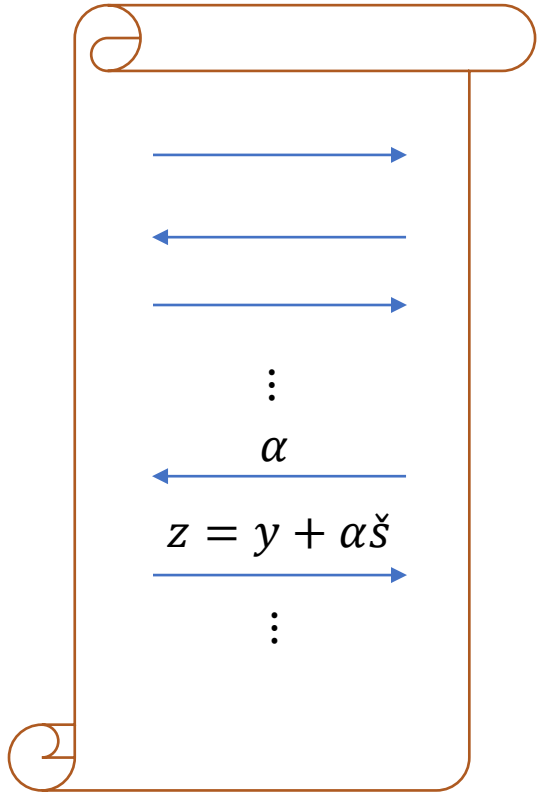


Exact proofs via NTT packing

$$z(z-\alpha) = y^2 + \alpha \cdot (2\check{s}y - y) + \alpha^2 \cdot \check{s}(\check{s}-1).$$

We want to prove $\check{s}(\check{s}-1) = 0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.
2. Send $t_y = Com(y)$.



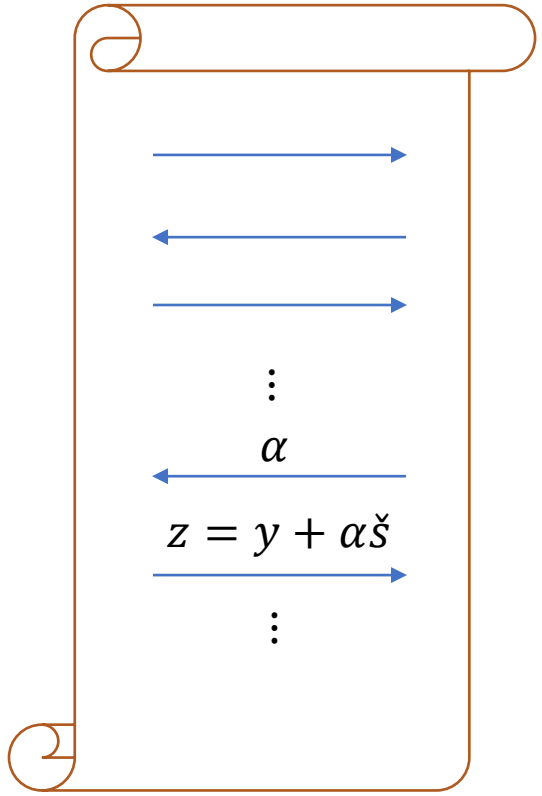
Exact proofs via NTT packing

$$z(z-\alpha) = y^2 + \alpha \cdot (2\check{s}y - y) + \alpha^2 \cdot \check{s}(\check{s}-1).$$

We want to prove $\check{s}(\check{s}-1) = 0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.
2. Send $t_y = Com(y)$.
3. Send commitments

   $t_0 = Com(y^2)$ and $t_1 = Com(2\check{s}y - y)$.



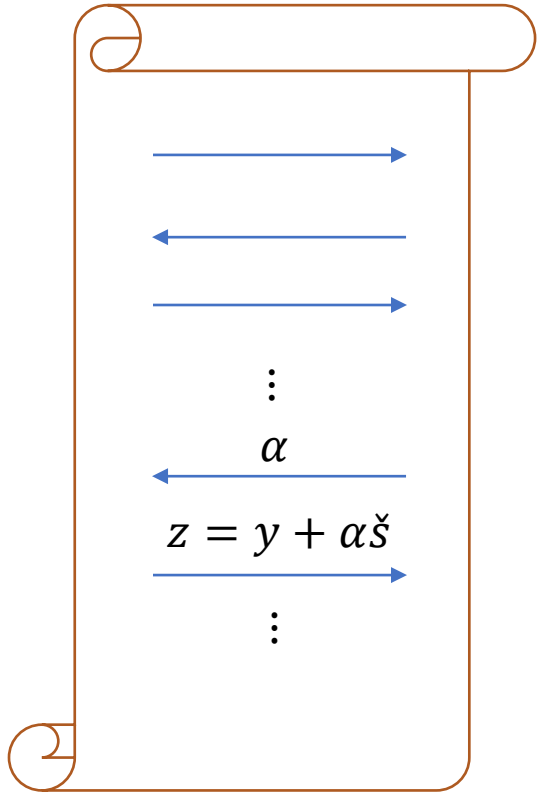
Exact proofs via NTT packing

$$z(z-\alpha) = y^2 + \alpha \cdot (2\check{s}y - y) + \alpha^2 \cdot \check{s}(\check{s}-1).$$

We want to prove $\check{s}(\check{s}-1) = 0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.
2. Send $t_y = Com(y)$.
3. Send commitments

   $t_0 = Com(y^2)$ and $t_1 = Com(2\check{s}y - y)$.
4. Given a challenge α, output $z = y + \alpha\check{s}$.



Exact proofs via NTT packing

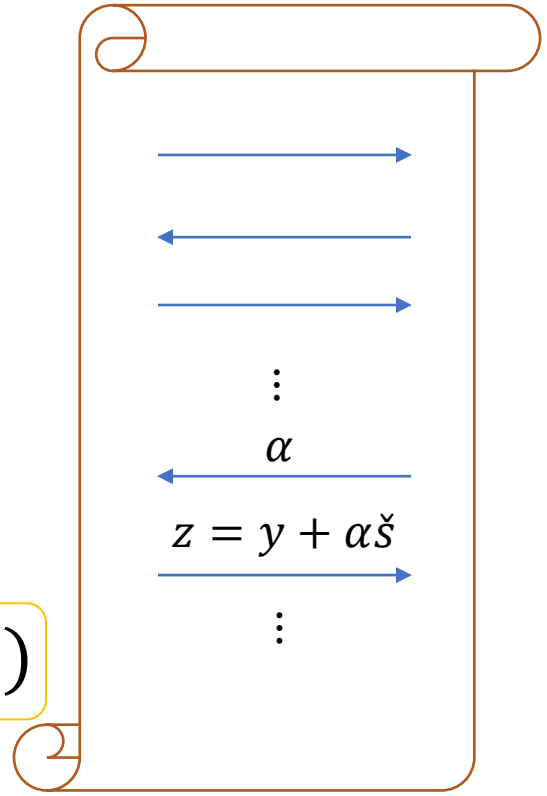
$$z(z-\alpha) = y^2 + \alpha \cdot (2\check{s}y - y) + \alpha^2 \cdot \check{s}(\check{s}-1).$$

We want to prove $\check{s}(\check{s}-1) = 0$.

1. Commit to $\check{s}$. Send $t_s = Com(\check{s})$.
2. Send $t_y = Com(y)$.
3. Send commitments

   $t_0 = Com(y^2)$ and $t_1 = Com(2\check{s}y - y)$.
4. Given a challenge α, output $z = y + \alpha\check{s}$.
5. Prove: (i) $z - (t_y + \alpha t_s)$ and (ii) $z(z-\alpha) - (t_0 + \alpha t_1)$

are commitments to zero.



Exact proofs via NTT packing

Proving linear relations, i.e. $\boldsymbol{As = u}$:

$$z = y + \alpha\check{s}$$

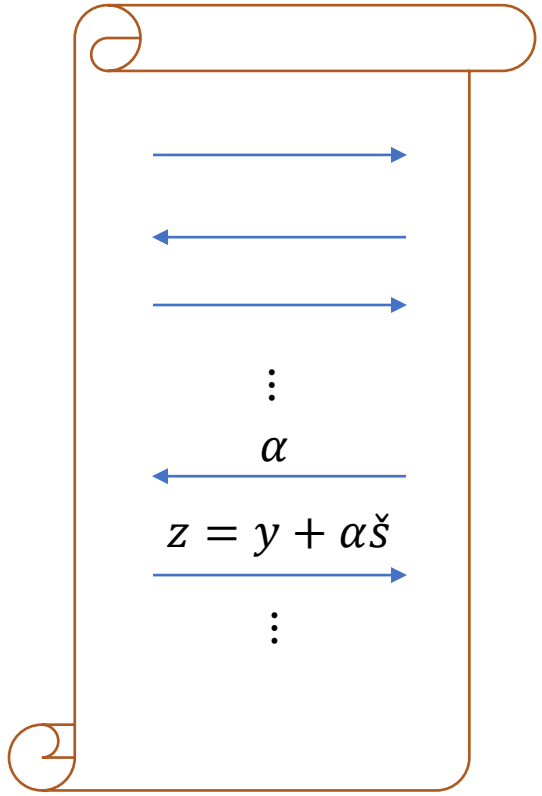
$$\downarrow \quad \alpha \in \mathbb{Z}_q$$

$$\hat{z} = \hat{y} + \alpha\boldsymbol{s}$$

$$\downarrow$$

$$A\hat{z} = A\hat{y} + \alpha\boldsymbol{u}$$

Here, $\hat{x} = NTT(x)$.



Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.

Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{2d}$.

Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{2d}$.
- Prover needs to send $\boldsymbol{z}$ in the clear, which is of the same length as $\boldsymbol{s}$.

Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{2d}$.
- Prover needs to send $\boldsymbol{z}$ in the clear, which is of the same length as $\boldsymbol{s}$.

- If one wants to prove a degree $\boldsymbol{k}$ equation, the prover sends $\boldsymbol{k}$ garbage commitments $\boldsymbol{t_i}$.

Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{q}$ [ALS20].
- Prover needs to send $\boldsymbol{z}$ in the clear, which is of the same length as $\boldsymbol{s}$.
  - New product proof which does not require sending $\boldsymbol{z}$ [ALS20]

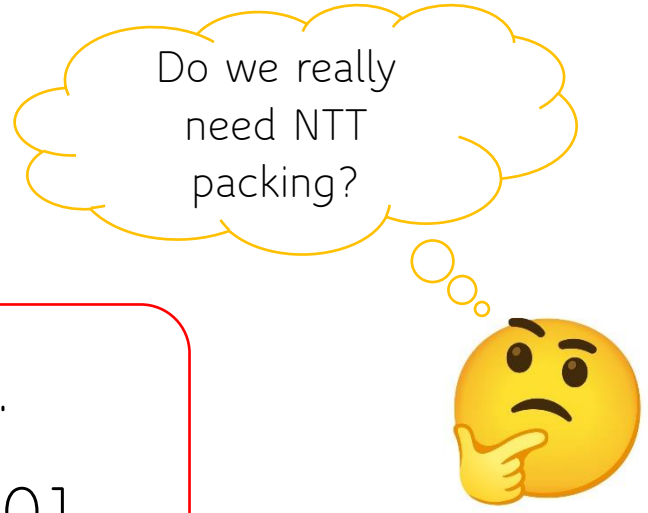
- If one wants to prove a degree $\boldsymbol{k}$ equation, the prover sends $k-1$ garbage commitments $\boldsymbol{t_i}$ [ALS20].

Bottlenecks of [BLS19,YAZ+19]

- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{q}$ [ALS20].

- Prover needs to send $\boldsymbol{z}$ in the clear, which is of the same length as $\boldsymbol{s}$.
  - New product proof which does not require sending z [ALS20]
  - New linear proof which does not require sending z [ENS20]
- If one wants to prove a degree $\boldsymbol{k}$ equation, the prover sends $k-1$ garbage commitments $\boldsymbol{t_i}$ [ALS20].

Bottlenecks of [BLS19,YAZ+19]



- Using a BDLOP commitment is relatively expensive.
- Small challenge space, soundness error $\approx \frac{1}{q}$ [ALS20].
- Prover needs to send $\boldsymbol{z}$ in the clear, which is of the same length as $\boldsymbol{s}$.
  - New product proof which does not require sending z [ALS20]
  - New linear proof which does not require sending z [ENS20]
- If one wants to prove a degree $\boldsymbol{k}$ equation, the prover sends $k-1$ garbage commitments $\boldsymbol{t_i}$ [ALS20].

Lattice-based zero-knowledge proofs aye!

- Does not rely on NTT packing
- (Almost) one-shot
- Compressing commitment

Roadmap

ABDLOP commitment

- It combines the Ajtai [Ajt96] and BDLOP [BDLOP18] commitments into one.
- It puts the long commitment to $\boldsymbol{s}$ into the "Ajtai" part of the commitment scheme.
- The BDLOP part of the commitment scheme is then used for low-dimensional auxiliary elements that will need to be committed to later in the protocol.

Roadmap

ABDLOP commitment

"are you still with me?"

- It combines the Ajtai [Ajt96] and BDLOP [BDLOP18] commitments into one.
- It puts the long commitment to $\boldsymbol{s}$ into the "Ajtai" part of the commitment scheme.
- The BDLOP part of the commitment scheme is then used for low-dimensional auxiliary elements that will need to be committed to later in the protocol.

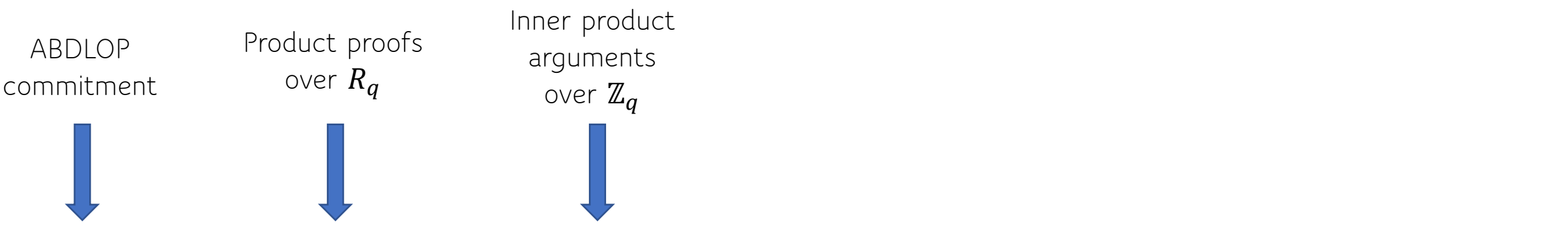
Roadmap

ABDLOP commitment

Product proofs over R_q

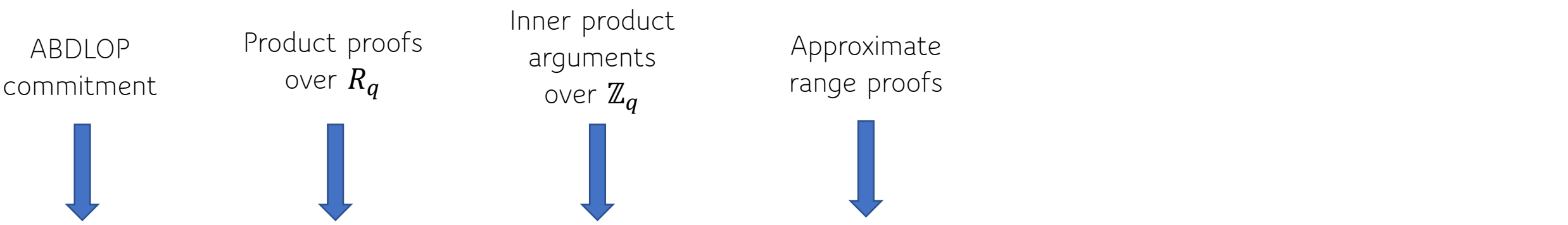
- Simple adaption of the [ALS20] protocol.
- It can be used to prove product relations, e.g. $\boldsymbol{s^T s = 0}$.
- Extended to also prove quadratic equations involving R_q-automorphisms, e.g. $\boldsymbol{s^T \sigma(s) = 0}$.

Roadmap



- We prove inner products over $\mathbb{Z}_q$, e.g. $\langle \boldsymbol{s}, \boldsymbol{v} \rangle = 0$ or $\langle \boldsymbol{s}, \boldsymbol{s} \rangle = B$.
- Fact: There is an automorphism σ such that $\langle \boldsymbol{x}, \boldsymbol{y} \rangle \in \mathbb{Z}_q$ is the constant coefficient of the polynomial $\boldsymbol{x}^T \sigma(\boldsymbol{y})$.
- Proving the const. coeff. of a polynomial is zero: [ENS20] + Product proof (with automorphisms) over R_q.

Roadmap



- We know how to prove equations modulo q.
- But how to prove relations **over integers**?
- $||s||^2 = \langle s, s \rangle = B \ (mod\ q)$ and s *approximately* small coefficients $\Longrightarrow$ $||s||^2 = B$ over integers!

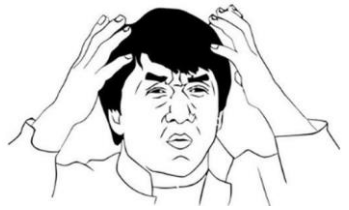
Roadmap

ABDLOP commitment

Product proofs over R_q

Inner product arguments over $\mathbb{Z}_q$

Approximate range proofs



Applications

- Proving knowledge of a Module-LWE sample

Scheme	Proof size
Stern proofs (e.g. [Ste93,LNSW13])	3MB
[Beu20]	233KB
[BLS19,YAZ+19]	384KB
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB
[ALS20,ENS20]	47KB
[LNS21]	33KB
This work	14KB

"Proof size so small, made my mom impressed!"

Applications

- Proving knowledge of a Module-LWE sample

Scheme	Proof size
Stern proofs (e.g. [Ste93,LNSW13])	3MB
[Beu20]	233KB
[BLS19,YAZ+19]	384KB
Ligero [AHIV17]	157KB
Aurora [BCR+19,BCOS20]	72KB
[ALS20,ENS20]	47KB
[LNS21]	33KB
This work	14KB

More applications in the paper…

ETH zürich

IBM Research | Zurich

Thank you!

References

[Ajt96] Miklós Ajtai. Generating Hard Instances of Lattice Problems (Extended Abstract).In STOC 1996.

[ACLMT22] Martin R. Albrecht, Valerio Cini, Russell W. F. Lai, Giulio Malavolta, Sri AravindaKrishnan Thyagarajan. Lattice-Based SNARKs: Publicly Verifiable, Preprocessing, and Recursively Composable. In CRYPTO 2022.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight Sublinear Arguments Without a Trusted Setup.. In ACM CCS 2017.

[AL21] Martin R. Albrecht and Russell W. F. Lai. Subtractive Sets over Cyclotomic Rings: Limits of Schnorr-like Arguments over Lattices. In CRYPTO 2021.

[ALS20] Thomas Attema, Vadim Lyubashevsky, and Gregor Seiler. Practical Product Proofs for Lattice Commitments. In CRYPTO 2020.

[AHIV17] Scott Ames, Carmit Hazay, Yuval Ishai, and Muthuramakrishnan Venkitasubramaniam. Ligero: Lightweight sublinear arguments without a trusted setup. In ACM CCS 2017.

[ACK21] A Compressed **Σ**-Protocol Theory for Lattices. Thomas Attema, Ronald Cramer, and Lisa Kohl. In CRYPTO 2021.

[AKSY21] Shweta Agrawal and Elena Kirshanova and Damien Stehle and Anshu Yadav. Practical, Round-Optimal Lattice-Based Blind Signatures. IACR Cryptol. ePrint Arch., 2021:1565

[BCOS20] Cecilia Boschini, Jan Camenisch, Max Ovsiankin, and Nicholas Spooner . Efficient Post-Quantum SNARKs for RSIS and RLWE and their Applications to Privacy. In PQCrypto 2020.

[BCR+19] Eli Ben-Sasson, Alessandro Chiesa, Michael Riabzev, Nicholas Spooner, Madars Virza, and Nicholas P. Ward. Aurora: Transparent succinct arguments for R1CS. In EUROCRYPT 2019.

[Beu20] Ward Beullens. Sigma protocols for mq, PKP and sis, and fishy signature schemes. In EUROCRYPT 2020.

[BKP20] Ward Beullens, Shuichi Katsumata, and Federico Pintore. Calamari and Falafl: Logarithmic (Linkable) Ring Signatures from Isogenies and Lattices. In ASIACRYPT 2020.

[BLNS20] Jonathan Bootle, Vadim Lyubashevsky, Ngoc Khanh Nguyen, and Gregor Seiler. A non-PCP approach to succinct quantum-safe zero-knowledge. In CRYPTO 2020.

[BLS19] Jonathan Bootle, Vadim Lyubashevsky, and Gregor Seiler. Algebraic techniques for short(er) exact lattice- based zero-knowledge proofs. In CRYPTO 2019.

[COS20] Alessandro Chiesa, Dev Ojha, and Nicholas Spooner. Fractal: Post-quantum and transparent recursive proofs from holography. In EUROCRYPT 2020.

[ENS20] Muhammed F. Esgin, Ngoc Khanh Nguyen, and Gregor Seiler. Practical Exact Proofs from Lattices: New Techniques to Exploit Fully-Splitting Rings. In ASIACRYPT 2020.

[ESZ22] Muhammed F. Esgin, Ron Steinfeld, and Raymond K. Zhao. MatRiCT+: More Efficient Post-Quantum Private Blockchain Payments. In IEEE S&P 2022.

References

[ISW21] Yuval Ishai and Hang Su and David J. Wu. Shorter and Faster Post-Quantum Designated-Verifier zkSNARKs from Lattices. In ACM CCS 2021.

[LN17] Vadim Lyubashevsky and Gregory Neven. One-Shot Verifiable Encryption from Lattices.

[LNPS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Maxime Plancon, Gregor Seiler. Shorter Lattice-Based Group Signatures via "Almost Free'' Encryption and Other Optimizations. In ASIACRYPT 2021.

[LNS20] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Gregor Seiler. Practical Lattice-Based Zero-Knowledge Proofs for Integer Relations. In ACM CCS 2020.

[LNS21] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Gregor Seiler. Shorter Lattice-Based Zero-Knowledge Proofs via One-Time Commitments. In PKC 2021.

[LNS21b] Vadim Lyubashevsky, Ngoc Khanh Nguyen, Gregor Seiler. SMILE: Set Membership from Ideal Lattices with Applications to Ring Signatures and Confidential Transactions. In CRYPTO 2021.

[LNSW13] San Ling, Khoa Nguyen, Damien Stehle, and Huaxiong Wang. Improved zero-knowledge proofs of knowledge for the ISIS problem, and applications. In PKC 2013.

[Lyu09] Vadim Lyubashevsky. Fiat-Shamir with aborts: Applications to lattice and factoring-based signatures. In ASIACRYPT 2009.

[Lyu12] Vadim Lyubashevsky. Lattice signatures without trapdoors. In EUROCRYPT 2012.

[NS22] Ngoc Khanh Nguyen and Gregor Seiler. Practical Sublinear Proofs for R1CS from Lattices. In CRYPTO 2022.

[PLS18] Rafael del Pino, Vadim Lyubashevsky, and Gregor Seiler. Lattice-Based Group Signatures and Zero-Knowledge Proofs of Automorphism Stability. In ACM CCS 2018.

[Sta21] StarkWare Team. ethSTARK documentation. IACR Cryptol. ePrint Arch., 2021:582, 2021

[Ste93] Jacques Stern. A new identification scheme based on syndrome decoding. In CRYPTO 1993.

[YAZ+19] Rupeng Yang, Man Ho Au, Zhenfei Zhang, Qiuliang Xu, Zuoxia Yu, and William Whyte. Efficient lattice- based zero-knowledge arguments with standard soundness: Construction and applications. In CRYPTO 2019.