ZK Proofs From VOLE and VOLE-in-the-Head

*Carsten Baum and **Peter Scholl***

Foundations of Zero Knowledge Proofs, Edinburgh

3 September 2024

AARHUS UNIVERSITY

Overview of the next sessions

Vector oblivious linear evaluation (VOLE) 

↓

VOLE-based ZK (designated verifier)

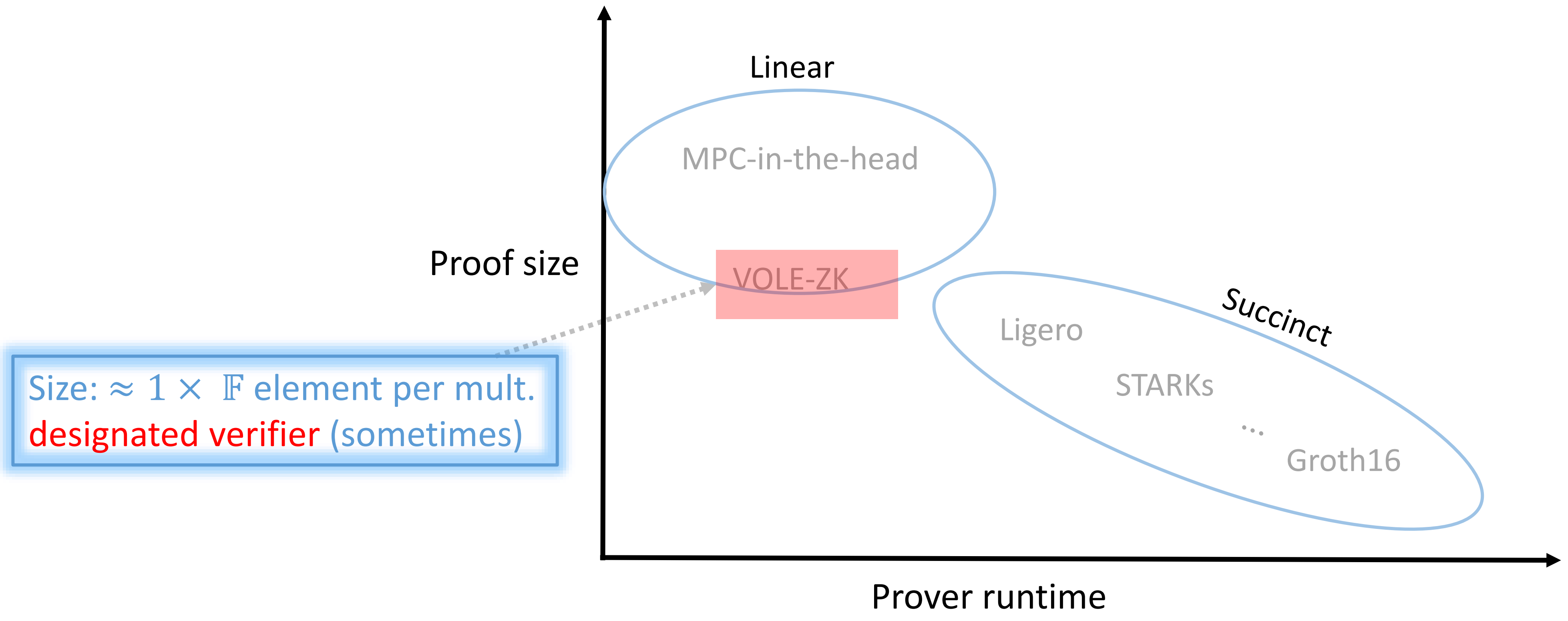
↓

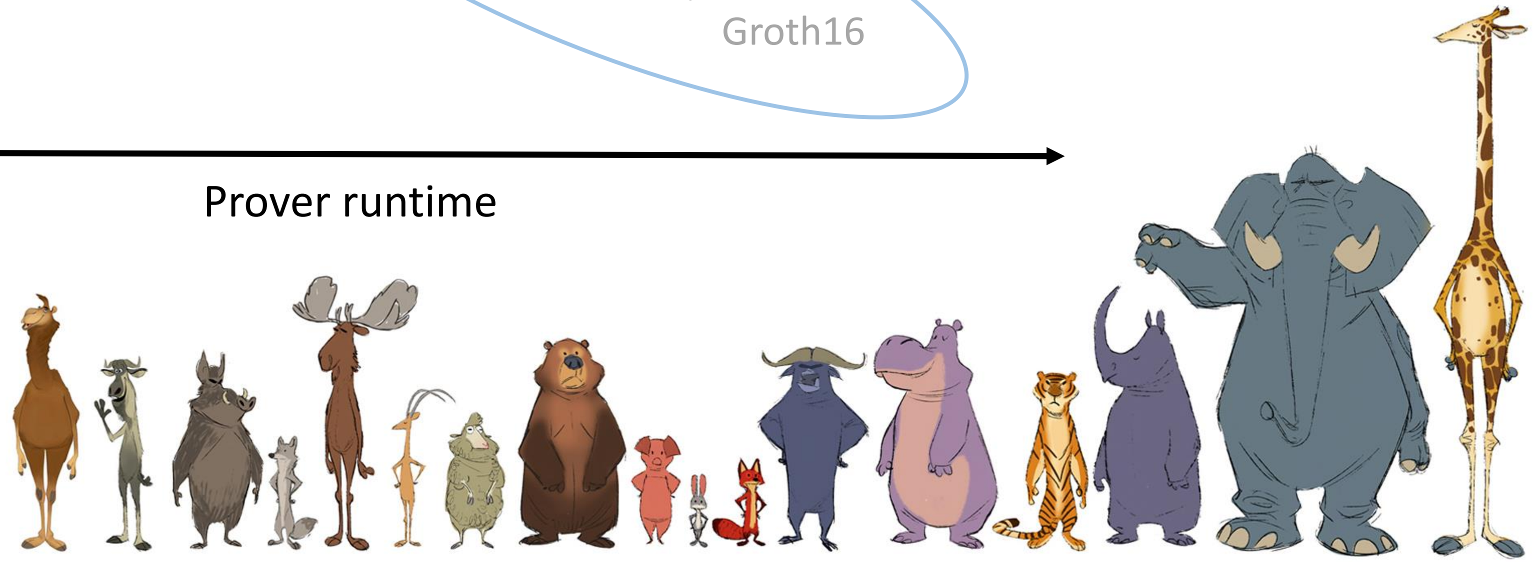
VOLE-in-the-head + NIZK 

↓

FAEST PQ signature

Families of ZK Proofs

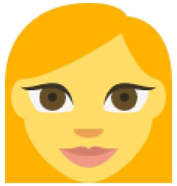


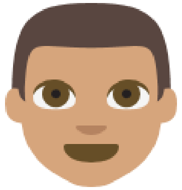


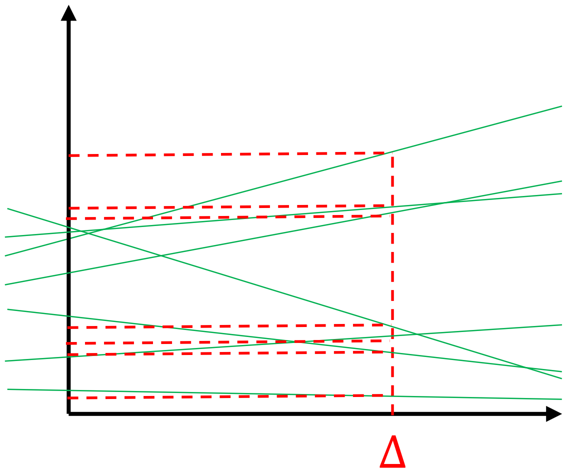
VOLE-ZK

ZK proofs in the designated verifier setting

Vector Oblivious Linear Evaluation: ideal functionality







$\vec{v}, \vec{w} \in \mathbb{F}^n$ → VOLE ← Δ

VOLE → $\vec{q} = \vec{w}\Delta + \vec{v}$

$\Delta \in \mathbb{F}$

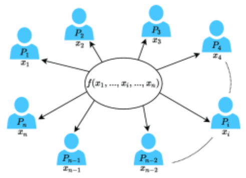
Today: $\vec{v}$ always uniform

Variant: random VOLE where $\vec{w}$ also uniform

What is VOLE good for?

Fundamental building block in many cryptographic protocols:

- General-purpose secure computation



- Oblivious transfer
  - ➢Implied by variant of VOLE

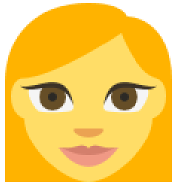
- Private set intersection
  - ➢Contact discovery; online advertising

How do we build VOLE?

- Linearly homomorphic encryption (LWE, DCR)
  - High communication and/or computation
- Pseudorandom correlation generators ("Silent" VOLE)
  - Learning parity with noise
  - Random, length-m VOLE: $O(\log m)$ communication [BCGI 18, BCGIKS 19, WYKW20, ….]
- Oblivious transfer extension (SoftSpokenVOLE [Roy 22])
  - Mainly symmetric primitives, fast
  - $O(\log m)$ communication for small fields

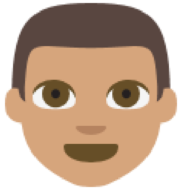
Information-theoretic commitments from VOLE

[CF 13, BMRS 21, WYKW 21]



$v_i, w_i \in \mathbb{F}$ → **VOLE** ← Δ

$q_i = w_i\Delta + v_i$ →

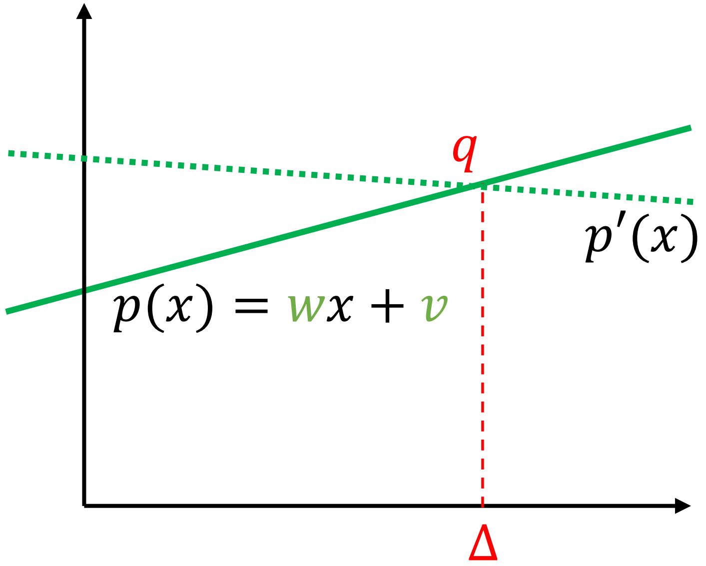


Commit to w_i:

Open:

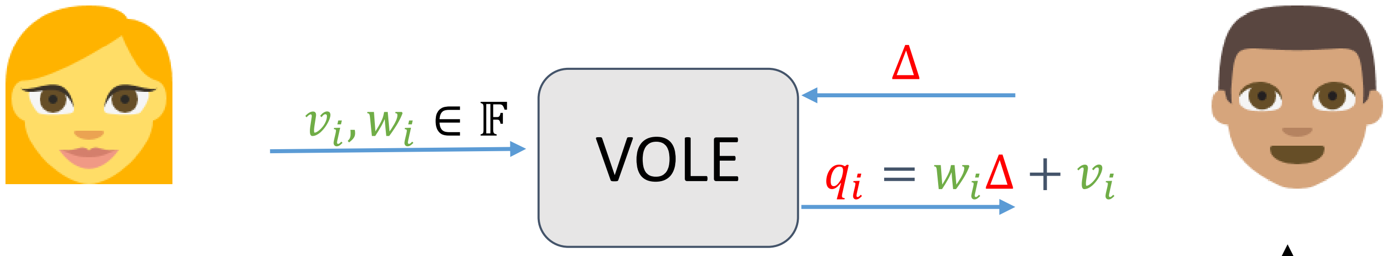
v_i, w_i →

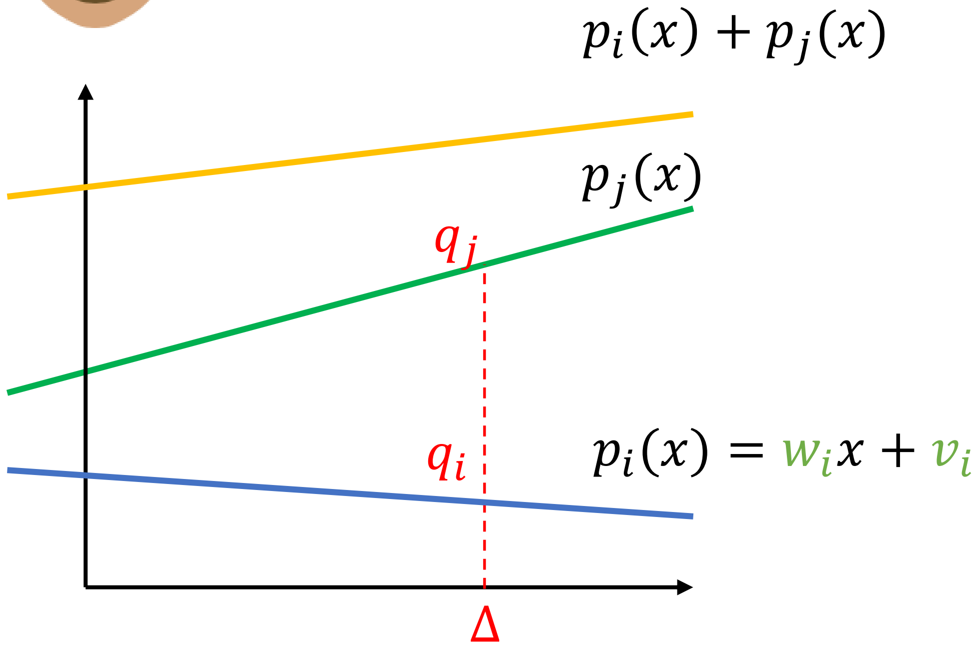
Check $q_i = w_i\Delta + v_i$



- Hiding: since v is random
- Binding: opening to $w' \neq w$ requires guessing Δ, prob. $1/|\mathbb{F}|$

Commitments are linearly homomorphic





Add w_i and w_j:

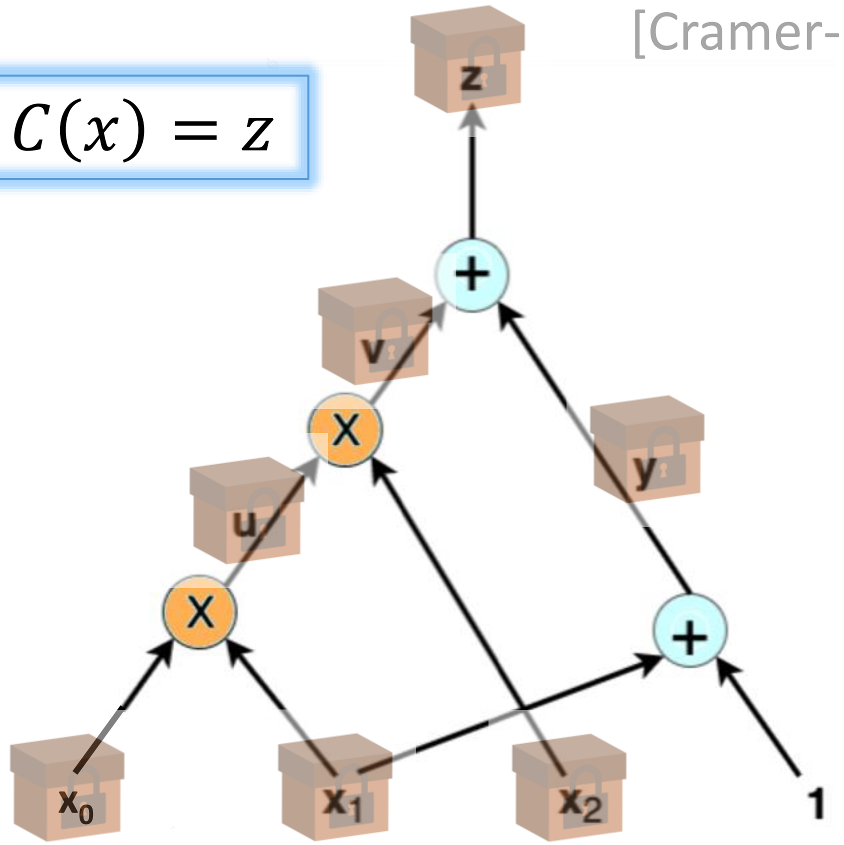
- Alice computes $p_i(x) + p_j(x) = (w_i + w_j)x + \cdots$
- Bob computes $q_i + q_j$

Proving circuits with linear commitments

[Cramer-Damgård 97]

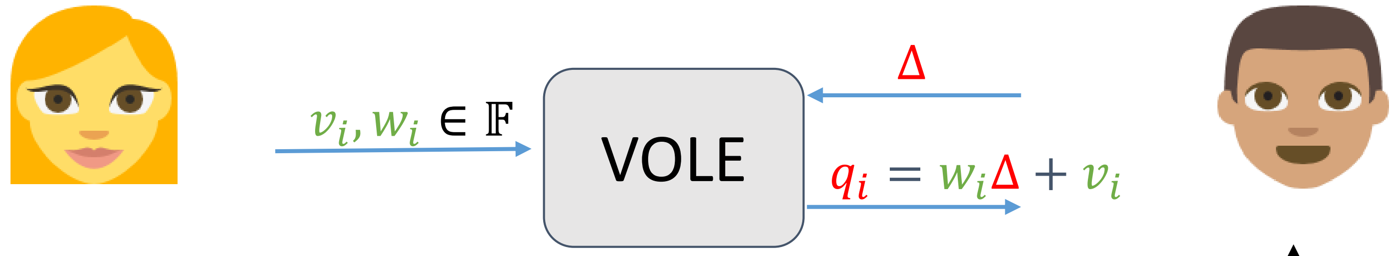
Goal: prove knowledge of x such that $C(x) = z$

- Commit to extended witness $\vec{w}$
  - inputs, + output wire of every mult.
- Evaluate linear gates
  - Using linear homomorphism
- Prove correctness of multiplications



How to prove multiplication gates? Multiplicative homomorphism!

[CF 13, DIO 21, WSWW 21]

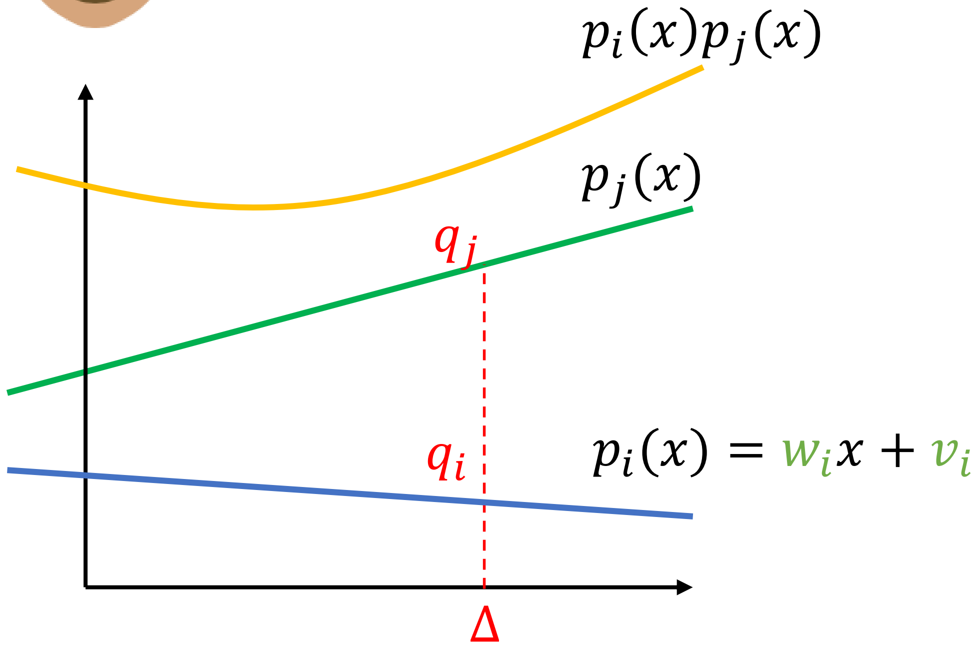


Multiply w_i and w_j:

- Alice computes $p_i(x) \cdot p_j(x) = w_i w_j x^2 + \cdots$
- Bob computes $q_i \cdot q_j$

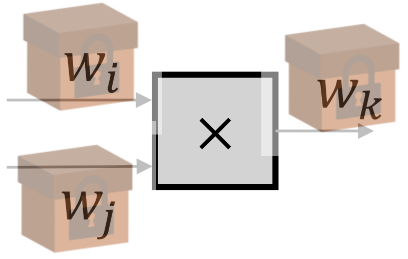
Degree increases!

$\Rightarrow$ opening soundness error grows to $2/|\mathbb{F}|$

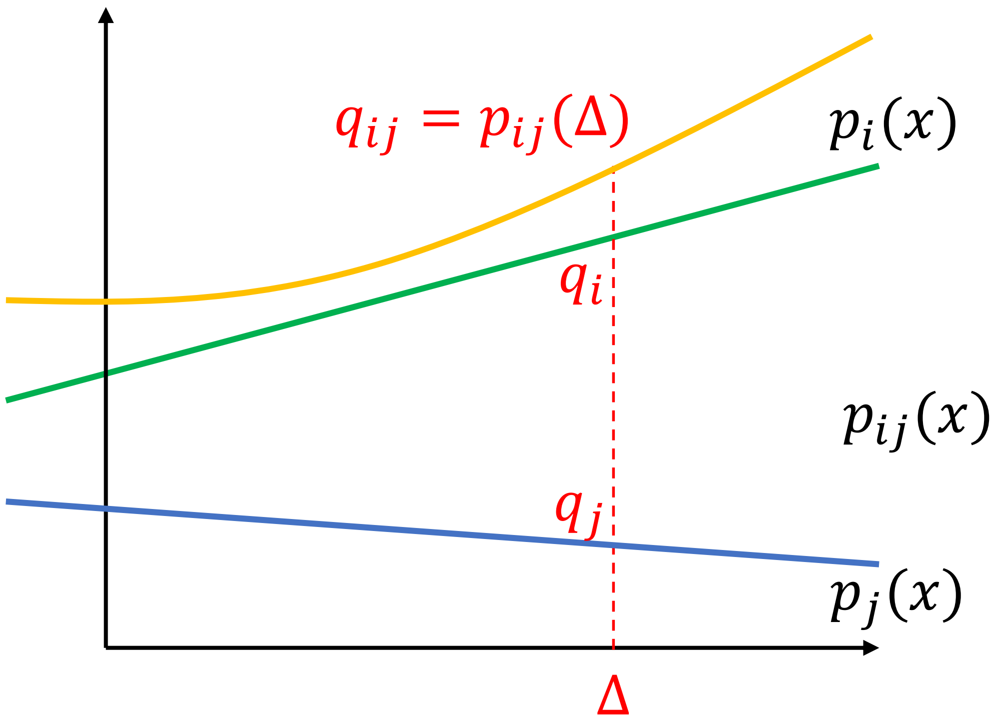


Multiplication gates in VOLE-ZK

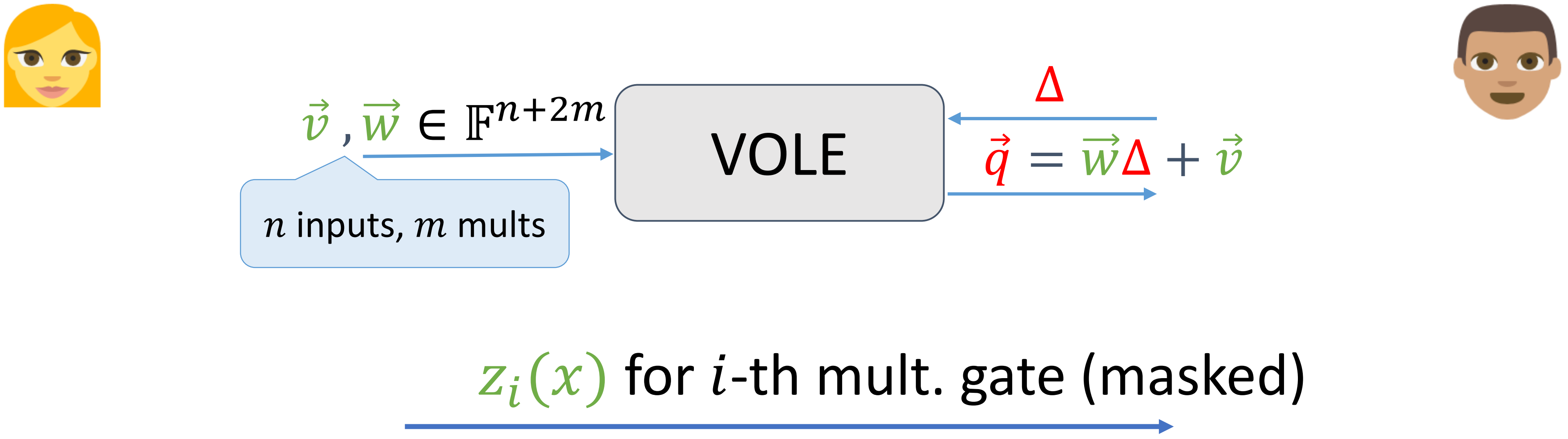
[DIO 21, YSWW 21]



- Multiply commitments to w_i, w_j $\Rightarrow$ quadratic polynomial
  - $p_{ij}(x) = t_0 + t_1 x + w_i w_j x^2$
- Let $z(x) := p_{ij}(x) - x p_k(x)$
  - Should be degree-1
  - Open and check
  - First, mask with random deg-1 commitment



Full ZK proof from VOLE: Initial Protocol [DIO 21]



Soundness error:

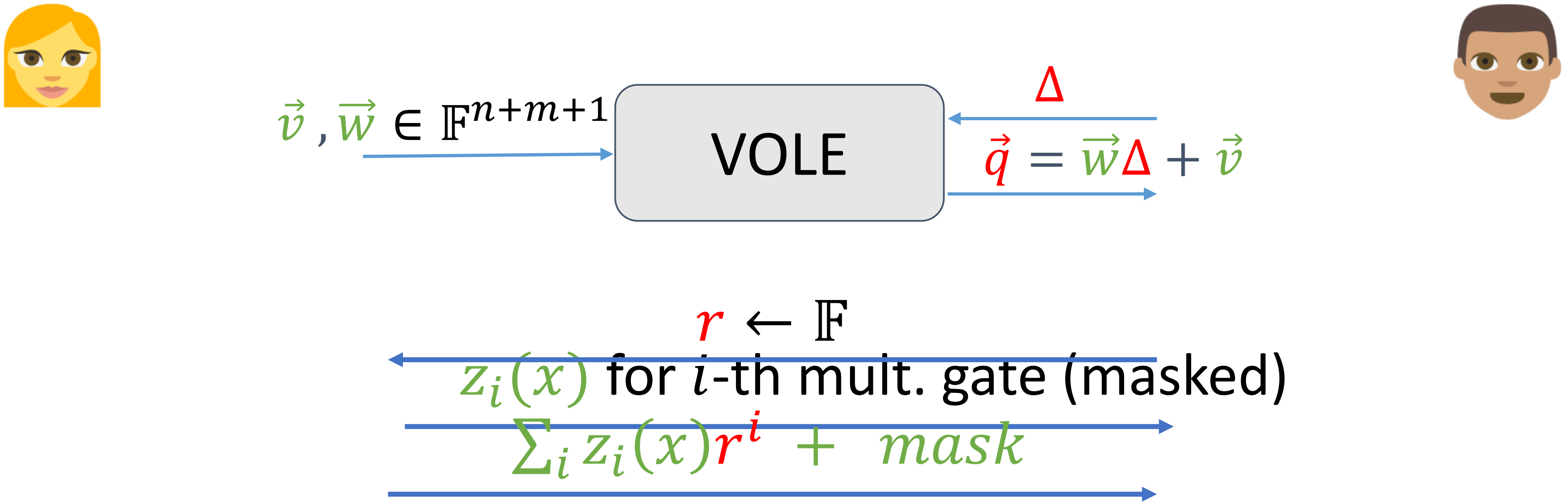
- $2/|\mathbb{F}|$

Cost for m multiplications:

- VOLE + $2m$ field elements

Optimization: batching multiplications

[YSWW 21]



Soundness error:

- $2/|\mathbb{F}| + m/|\mathbb{F}|$

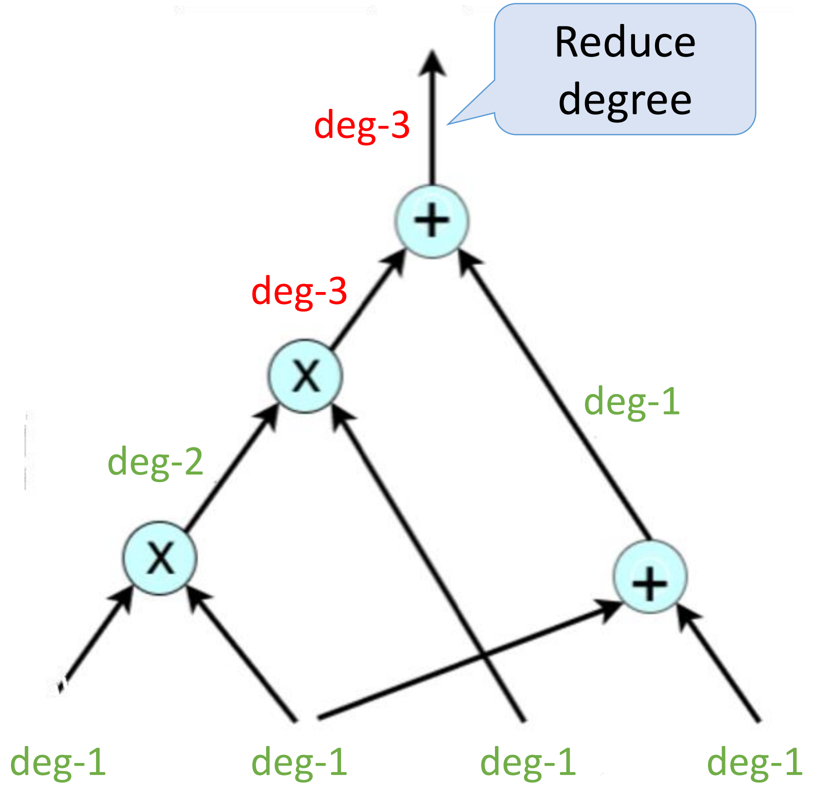
Cost for m multiplications:

- Length-$(n + m + 1)$ VOLE

Exploiting higher-degree multiplicative homomorphism

[YSWW 21, BBGMORRRS 24]

- General degree-reduction gadget:
  - ➢ $p(x) = a_0 + a_1 x + \cdots + w x^d$
  - ➢ Commit to fresh w: $p_w(x) = a' + wx$
  - ➢ Show that $z(x) := p(x) - x^{d-1} p_w(x)$ is deg-$(d-1)$ commitment to zero



- Circuit evaluation:
  - ➢ Lazily reduce degree on-the-fly

Communication complexity of VOLE-ZK with lazy reduction

- Cost per degree reduction:
  - Create fresh commitment: $1 \times$ VOLE element
  - Open masked commitment: send $d - 1$ field elements (amortized via batch check)

- For circuit with m multiplications, using max. degree d:
  - $\leq \frac{m}{\log d} + d$ field elements – sublinear in circuit size!

Improvements/extensions

- Circuits over $\mathbb{F}_2$: [YSWW 21]
  - ➢Let $w \in \mathbb{F}_2$, but use subfield VOLE $q = w\Delta + v$ in $\mathbb{F}_{2^\lambda}$

- Circuits over $\mathbb{Z}_{2^k}$ [BBMS 22]
  - ➢Use VOLE $q = w\Delta + v$ in $\mathbb{Z}_{2^{k+\lambda}}$

- Mixed Boolean/arithmetic circuits [BBMRS 21, YYXKW 21]
  - ➢VOLE in $\mathbb{F}_2$ and $\mathbb{F}_p$, prove consistency

…

Performance of VOLE-ZK

Threads	Boolean circuits	Arithmetic circuits
1	7.6 M gates/s	4.8 M gates/s
4	15.8 M gates/s	8.9 M gates/s

Numbers from QuickSilver [YSWW21]: degree-2 checks over local network, including setup time for LPN-based VOLE

Summary: what's VOLE-ZK good for?

Pros:

- Information-theoretic (after VOLE setup)
- Flexible choice of field/ring
- Low proving time + memory

Cons:

- Proof size: linear-ish (but small constants)
- Designated verifier

Example use-cases:

- Proof of well-formed LWE ciphertexts
- Anonymous credentials
- Ensuring MPC input consistency
- Proof of vulnerability

Credits

[CF 13] Catalano, Fiore
Practical Homomorphic MACs for Arithmetic Circuits
Eurocrypt 2013

[WYKW 21] Weng, Yang, Katz, Wang
Wolverine: Fast, Scalable, and Communication-Efficient Zero-Knowledge Proofs for Boolean and Arithmetic Circuits.
S&P 2021

[BMRS 21] *Baum, Malozemoff, Rosen, Scholl*
Mac'n'Cheese: Zero-Knowledge Proofs for Boolean and Arithmetic Circuits with Nested Disjunctions
Crypto 2021

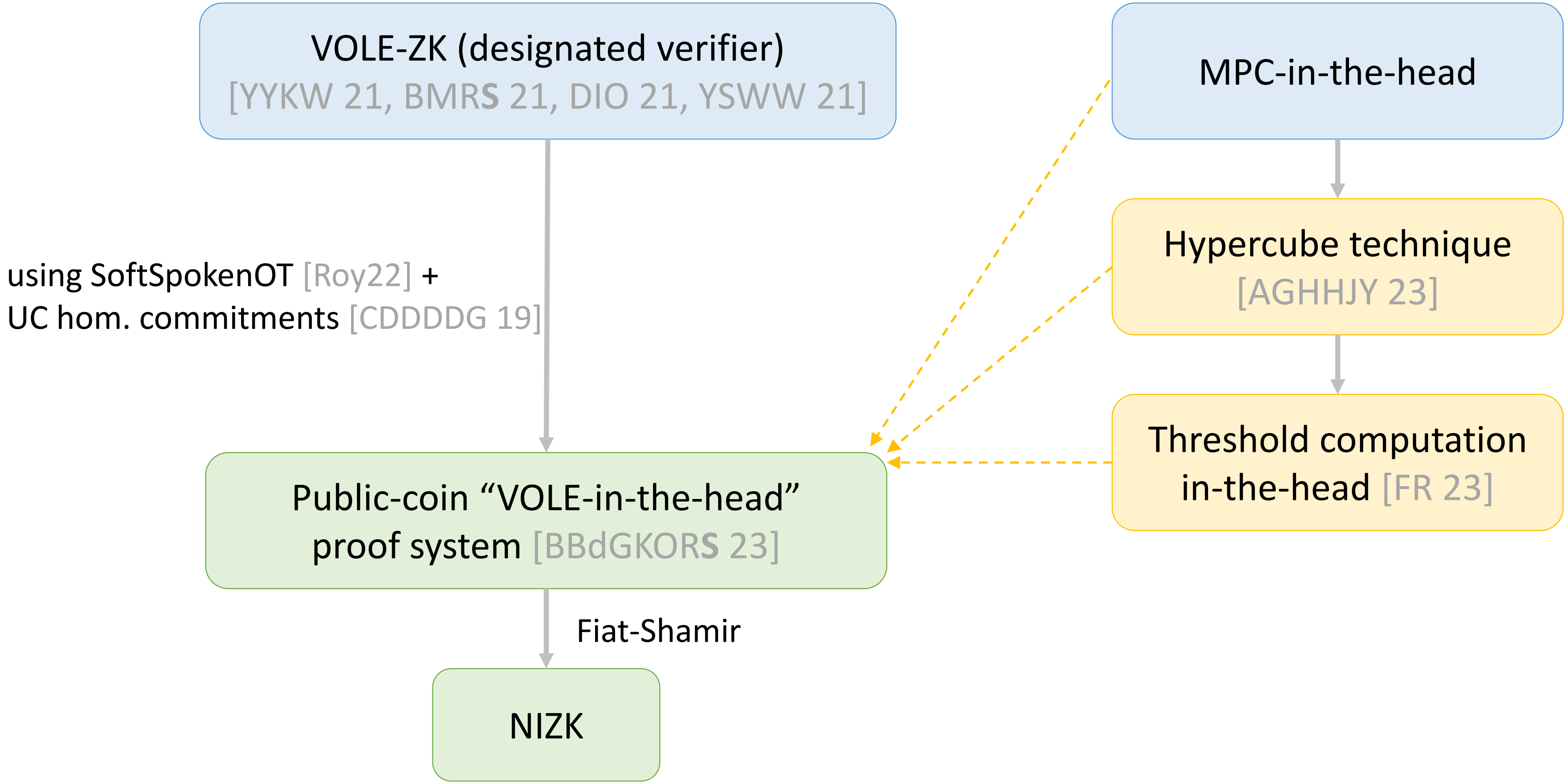
[DIO 21] *Dittmer, Ishai, Ostrovsky*
Line-Point Zero Knowledge and its Applications
ITC 2021

[YSWW 21] Kang Yang, Pratik Sarkar, Chenkai Weng, Xiao Wang
QuickSilver: Efficient and Affordable Zero-Knowledge Proofs for Circuits and Polynomials over Any Field.
CCS 2021

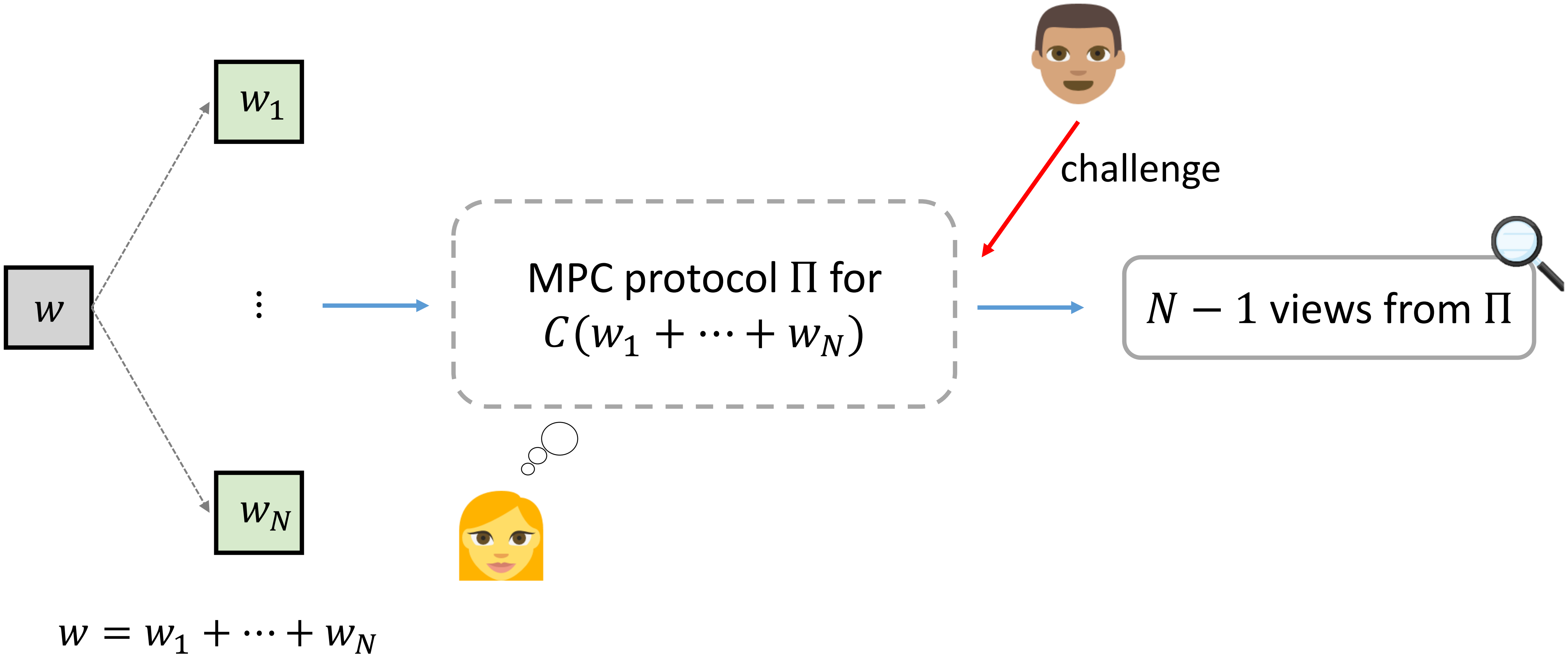
VOLE-in-the-Head

Adding public verifiability

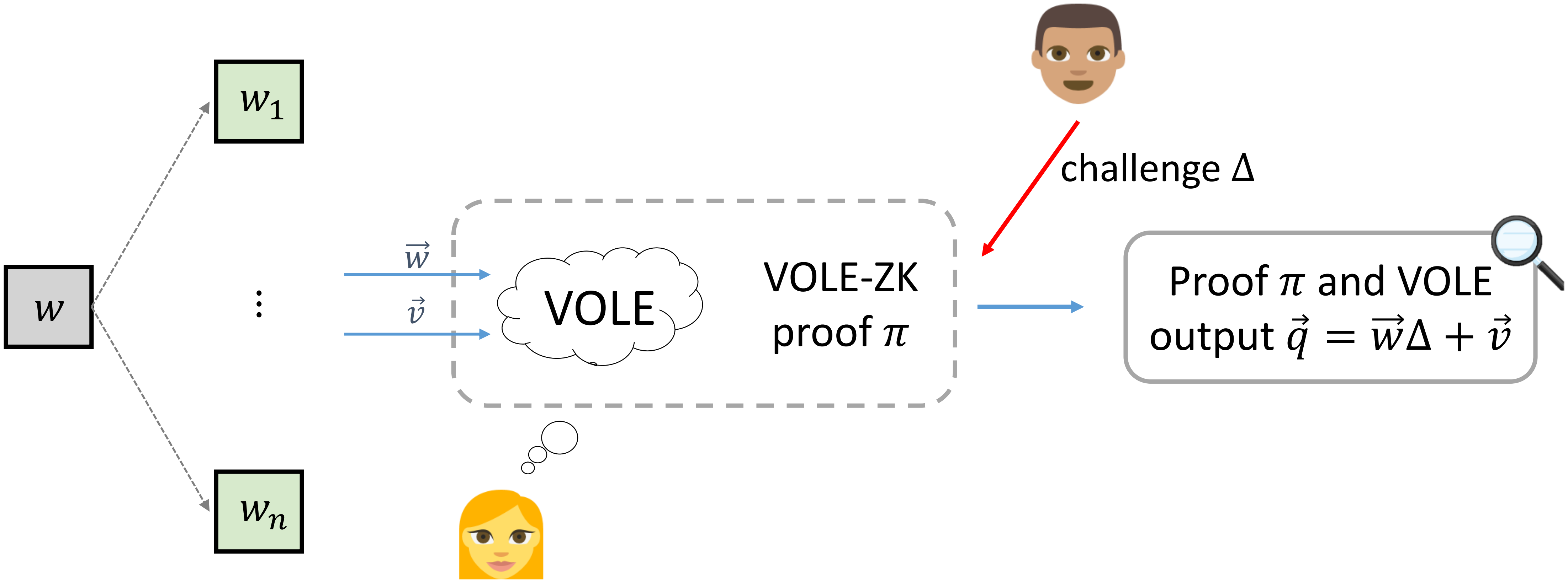
Story of VOLE-in-the-head



Recap: MPC-in-the-Head

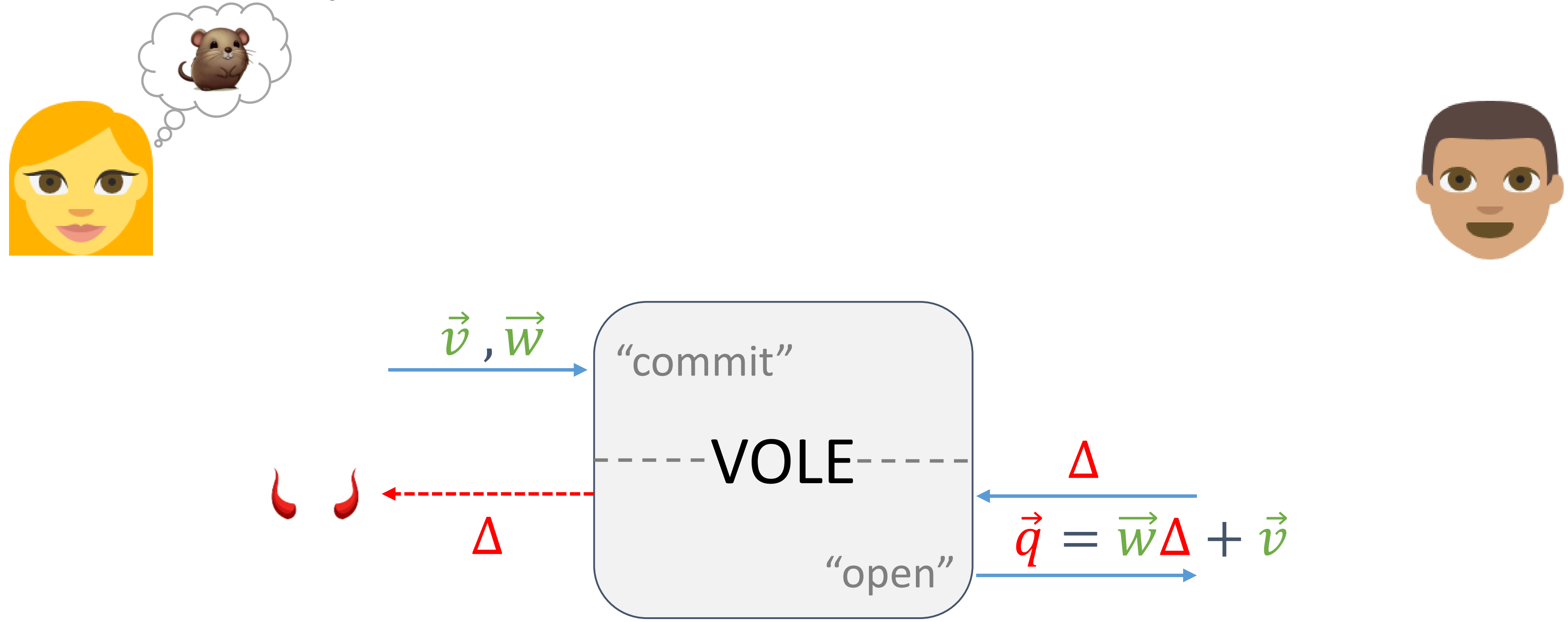


VOLE-in-the-head: high-level overview

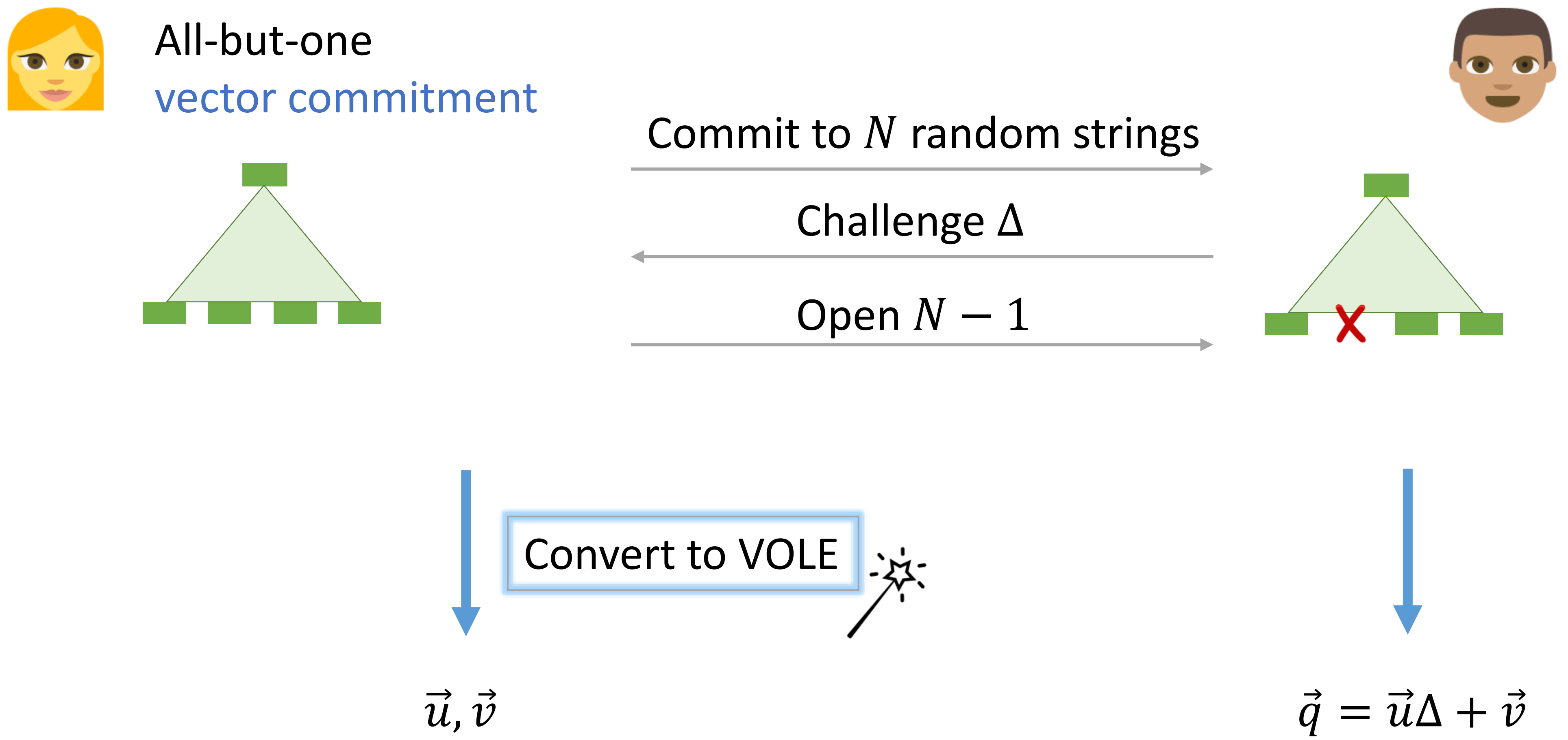


$$w = w_1 + \cdots + w_n$$

Goal: implement public-receiver VOLE functionality



Building Public-Receiver VOLE



Conversion to VOLE

Key observation: $(N-1)$-out-of-N commitment $\Rightarrow$ VOLE in $\mathbb{F}_N$

[Roy 22, BBdGKORS 23, CDI 05]

w_1

$\vdots$

w_N

Commit to $\overrightarrow{w_i} \in \mathbb{F}_N^k$ →

← $\Delta \leftarrow \mathbb{F}_N$

Open $\overrightarrow{w_i}$, for $i \neq \Delta$ →

$\vec{w} = \overrightarrow{w_1} + \cdots + \overrightarrow{w_N}$

$v = -1 \cdot \overrightarrow{w_1} - \cdots - N \cdot \overrightarrow{w_N}$ (over $\mathbb{F}_N$)

$$\vec{q} = \sum_{i=1}^{N} \overrightarrow{w_i} \cdot (\Delta - i)$$
$$= \vec{w}\Delta + \vec{v}$$

Public-Receiver VOLE: Summary

- If $\vec{w}$ is random, can succinctly commit to arbitrarily long VOLE
  - Commit to N seeds, expand to $\vec{w_i}$'s with PRG
- Cost for $\vec{w} \in \mathbb{F}_N^\ell$:
  - Communication: $\mathrm{O}(\log N)$ seeds
  - Computation: $O(N)$
- For non-random w:
  - Send extra $|w|$ field elements

Simplified VOLE-in-the-head: 3-round sigma protocol for arithmetic circuit satisfiability

Extended witness shares: $(w_1, \dots, w_N)$

Convert to VOLE

Commit to $\{w_i\}_i$

+ VOLE-ZK mult. checks

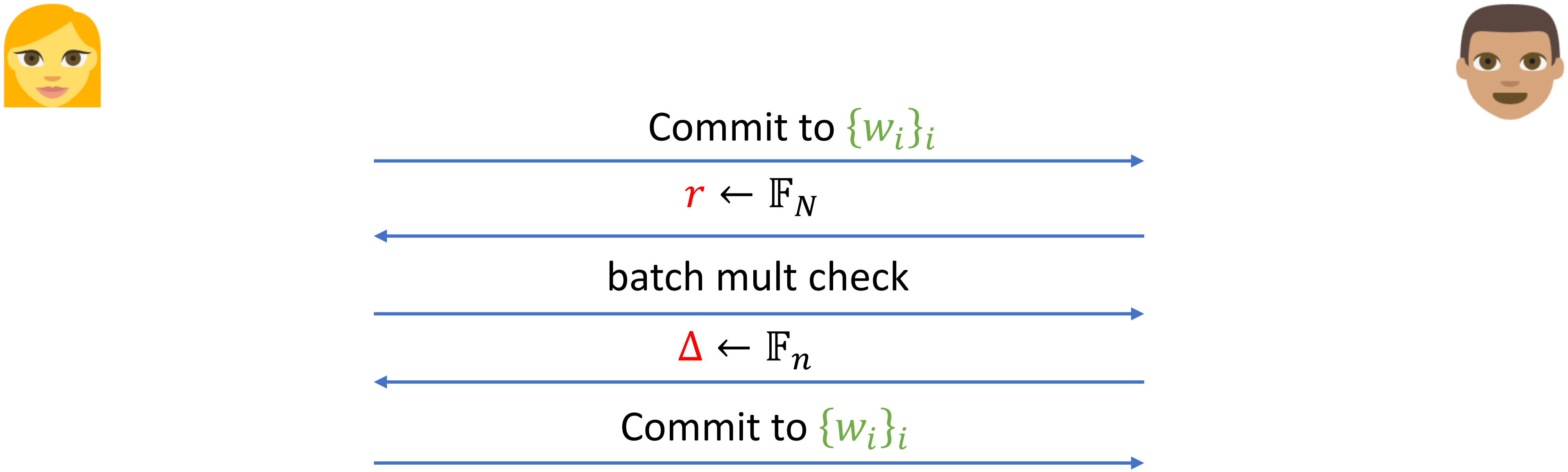
$\Delta \leftarrow \mathbb{F}_N$

Open $\{w_i\}_{i \neq \Delta}$

Soundness error:

- $2/N$
- Shrink via parallel repetition

5-round VOLE-in-the-head: batching multiplications



Soundness error:

- $3/N$

The Curse of Parallel Repetitions with >3 Rounds

- Problem: Fiat-Shamir can worsen security for >3-round protocols
  - ➢Adversary can attack each round independently

- **Solution**: more rounds!

VOLE-in-the-head, last optimization: avoiding parallel repetition

- Naïve repetition:
  - τ sets of VOLEs $\overrightarrow{q_i} = \vec{w}\Delta_i + \vec{v}_i$ in $\mathbb{F}_q^\ell$, with same witness
  - τ independent VOLE-ZK checks in $\mathbb{F}_q$

- **Idea:** pack into a single VOLE and run one check
  - Combine and lift VOLEs into $\mathbb{F}_{q^\tau}$
  - Gives subfield VOLE $\vec{q} = \vec{w}\Delta + \vec{v}$, where $\Delta = \sum_i \alpha^i \Delta_i$ in $\mathbb{F}_{q^\tau}$ (α: generator of $\mathbb{F}_q$ over $\mathbb{F}_{q^\tau}$)

Challenge: need to prove τ witnesses are consistent

- When repeating τ times:
  - Ensure prover uses consistent w

- Check consistency via [Roy 22]:

Linear, universal hash H

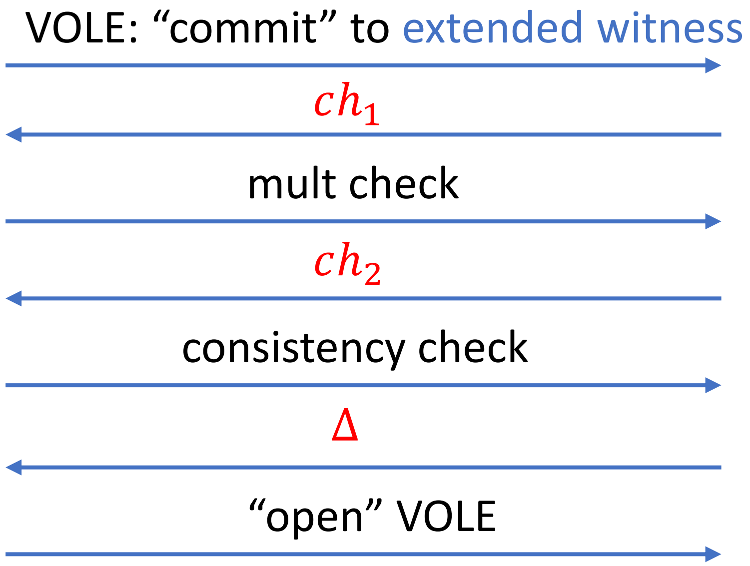
$\widetilde{w} = H(\vec{w}),\ \widetilde{v_i} = H(\vec{v_i})$

Check $H(\vec{q_i}) = \widetilde{w}\Delta + \widetilde{v_i}$

- Security:
  - Intricate analysis, esp. to prove compatibility with Fiat-Shamir

Final Protocol for $\mathbb{F}_2$: Overview

[BBdGKORS 23]



Communication cost:

- $\mathbb{F}_2$: ≈10-16 bits per AND
- $\mathbb{F}_p$ variant: 1-2 field elements per mult

Application to Post-Quantum Signatures

Call for Additional Digital Signature Schemes

Standardization of Post-Quantum Signatures

	Dilithium	Falcon	SPHINCS+	FAEST
Security:	Structured lattices	Structured lattices	Hash-based	AES/hash-based
Speed:	Fast	Fast	Slow signing	Fast-ish
Size:	2.4 kB	0.7 kB	8-17 kB	4.2-6 kB

2023: new algorithms submitted to diversify candidates

Paradigm for ZK-based signatures

- Keypair $sk, pk = (x, \mathrm{Enc}_{sk}(x))$, for symmetric Enc
- Signature:
  - ➢NIZK proof of knowledge of sk
  - ➢With Fiat-Shamir transform

- Challenge: finding a ZK-friendly Enc
  - ➢Custom cipher designs with few AND gates: e.g. LowMC (Picnic)
  - ➢Code-based: syndrome decoding, MinRank

AES: a ZK-friendly OWF?

ShiftRows, MixColumns, AddRoundKey:

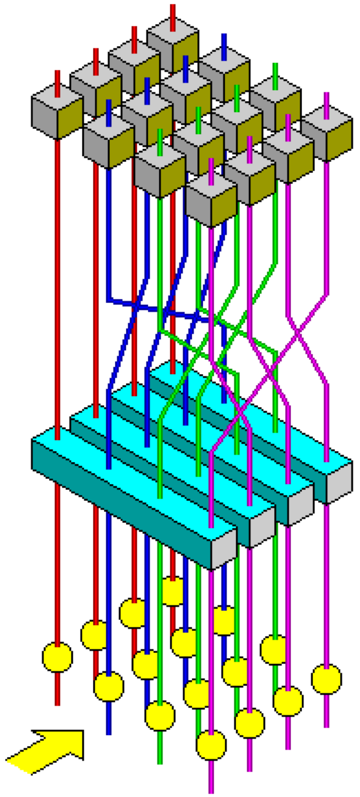
- All linear over $\mathbb{F}_2$

SubBytes:

- Nice representation in $\mathbb{F}_{2^8}$

Approach for ZK:

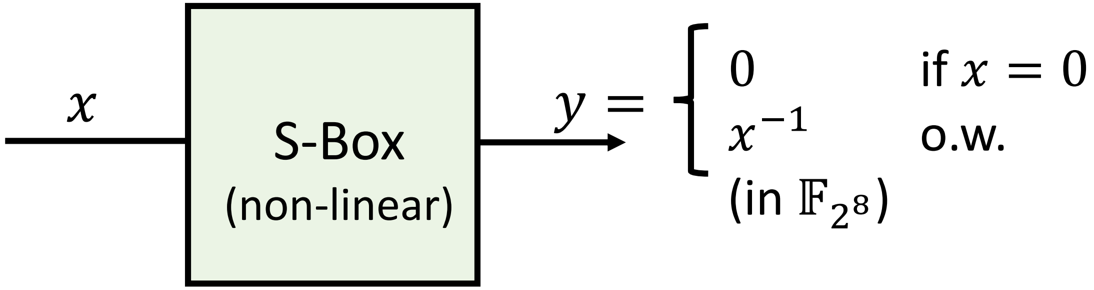
- Commit to state after each round
- Prove consistency of rounds $i, i+1$



Proving the AES S-Box, v1

- Given commitments to (bits of) x, y over $\mathbb{F}_2$
- Lift to x, y in $\mathbb{F}_{2^8}$
- Verify S-Box with

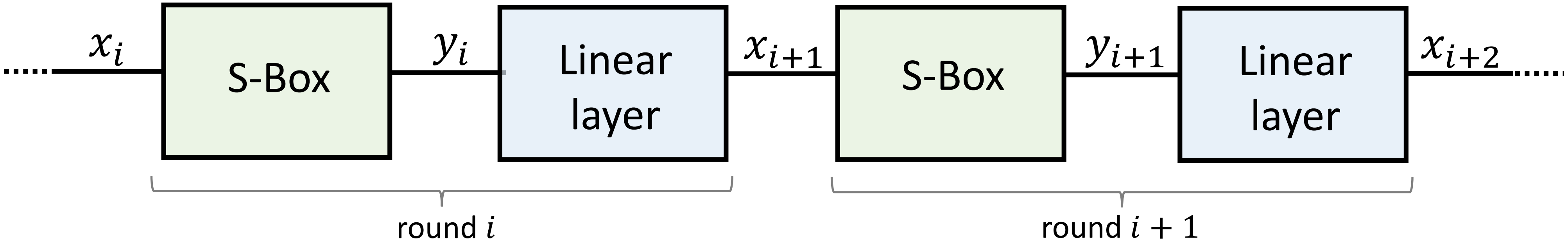
$$xy = 1$$



What if $x = 0$?

- ➢ Sample key such that this never happens
- ➢ 1-2 bits less security

Proving the AES S-Box, v2



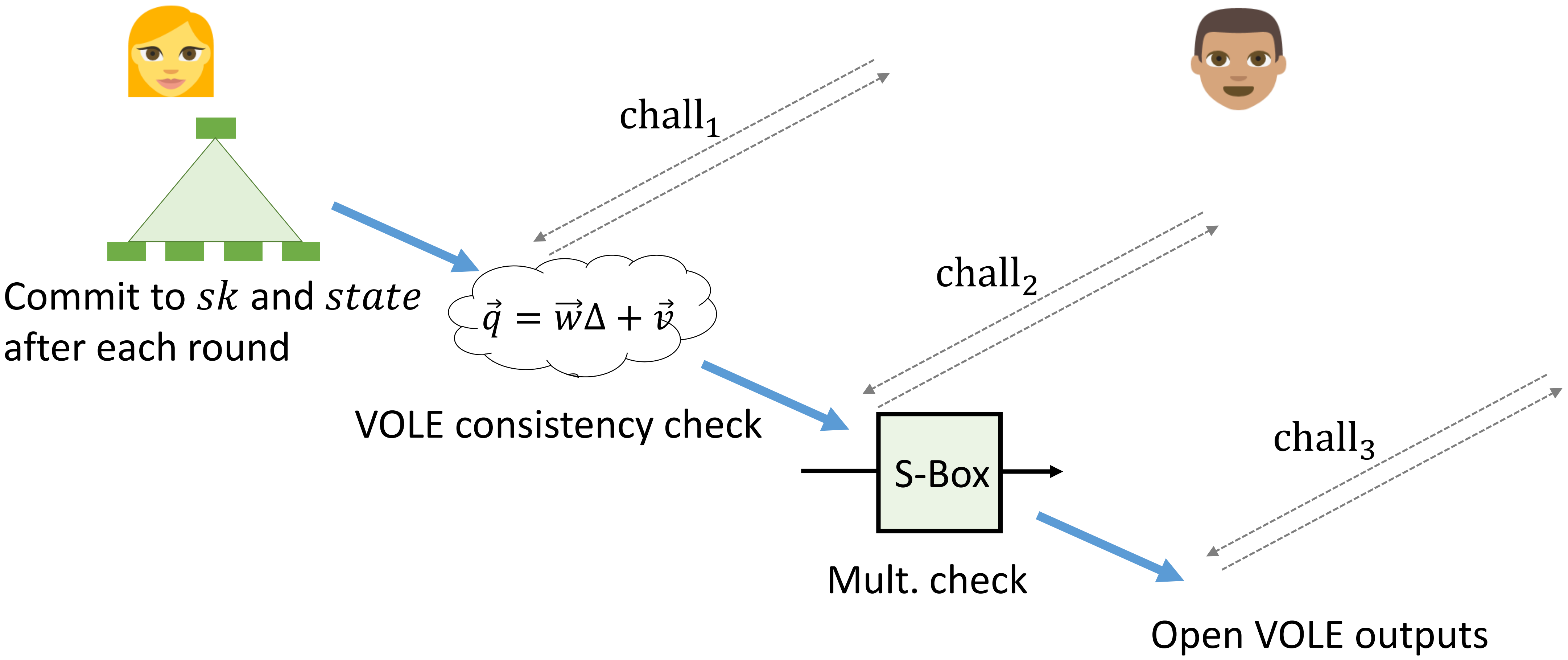
- Observation: S-box and its inverse are degree-7 functions over $\mathbb{F}_2$
- Verify two S-Boxes at once by checking:

$$\mathrm{Linear}(\mathrm{SBox}(x_i)) = \mathrm{SBox}^{-1}(\mathrm{Linear}^{-1}(x_{i+2}))$$

  - Only need to commit to every other x_i value!
  - Drawback: degree-7 check instead of degree-2

Impact: 5-10% smaller signatures [BBMORRRS 24]

FAEST summary: proving $pk = AES_{sk}(x)$



FAEST performance

	Sign (ms)	Verify (ms)	|sig| (bytes)
FAEST-128s	4.4	4.1	5 006
FAEST-128f	0.4	0.4	6 336
FAEST-256s	14.4	14.4	22 100
FAEST-256f	1.6	1.6	28 400

- Signature sizes:
  - ➢Smaller than SPHINCS+ and most MPCitH-based candidates
  - ➢Faster signing, slower verification vs SPHINCS+

- Latest optimizations/variants: 10-20% smaller for same/faster signing

Conclusion

VOLE-ZK proofs:

- Simple proof systems for circuit satisfiability
- Fast prover, flexible, linear-ish size
- VOLE-in-the-head: publicly verifiable
    - ➢ Useful for PQ signatures

Resources:

https://ia.cr/2023/996

https://faest.info

Credits

[Roy 22] *Roy*
SoftSpokenOT: Communication-Computation Tradeoffs in OT Extension
Crypto 2022

[AGHHJY 22] *Aguilar-Melchor, Gama, Howe, Hülsing, Joseph, Yue*
The Return of the SDitH
Eurocrypt 2022

[BBdGKOR**S** 23] *Baum, Braun, de Saint Guilhem, Klooß, Orsini, Roy, Scholl*
Publicly Verifiable Zero-Knowledge and Post-Quantum Signatures From VOLE-in-the-Head
CRYPTO 2023

FAEST Digital Signature Scheme
+ Majenz, Mukherjee, Ramacher, Rechberger
Submission to NIST PQC call

[FR 23] Fenuil, Rivain
Threshold Computation in the Head: Improved Framework for Post-Quantum Signatures and Zero-Knowledge Arguments

[BBMORRRS 24] *Baum, Beullens, Mukherjee, Orsini, Ramacher, Rechberger, Roy, Scholl*
One Tree to Rule Them All: Optimizing GGM Trees and OWFs for Post-Quantum Signatures
Asiacrypt 2024