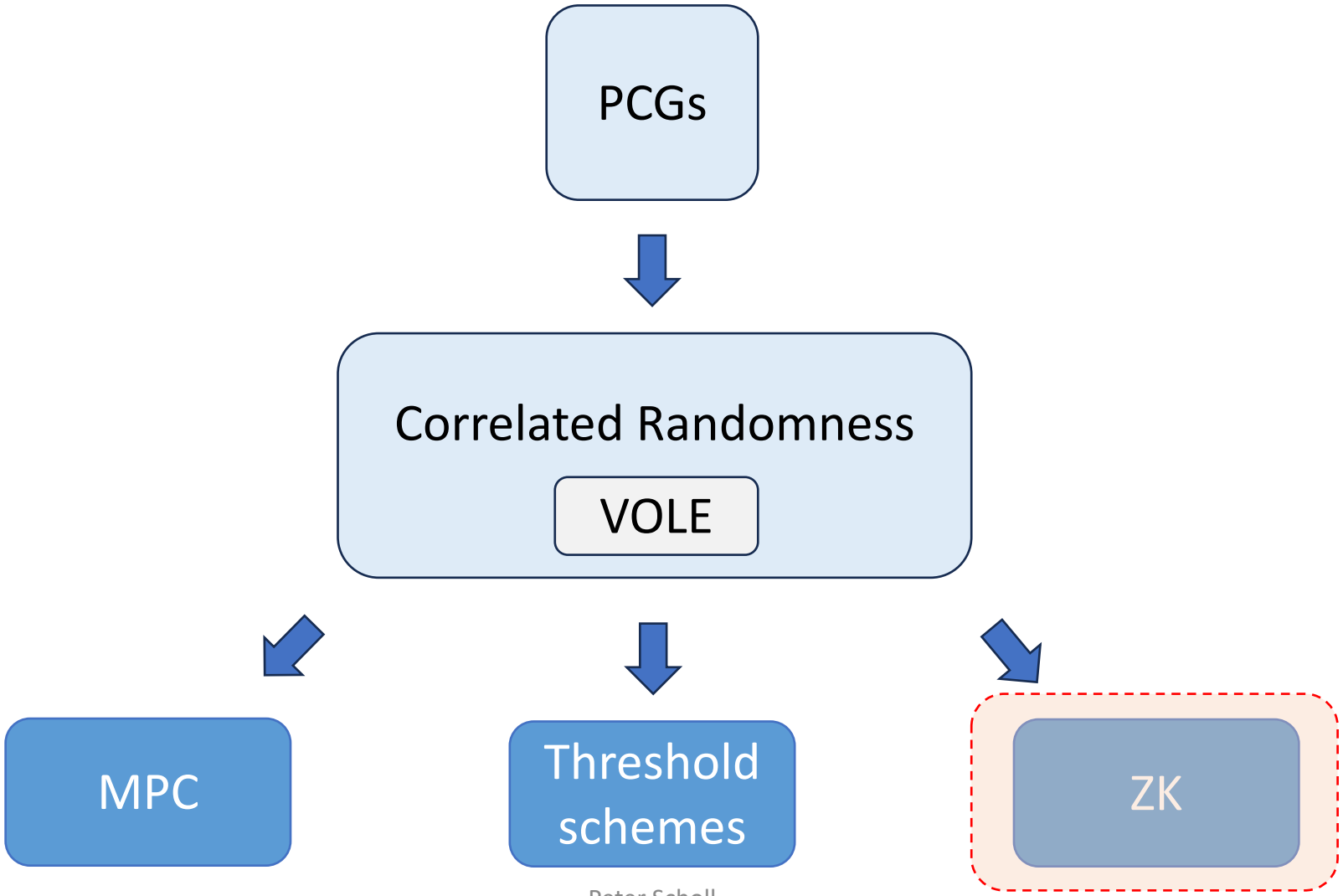
Vector Oblivious Linear Evaluation, PCGs and Correlated Randomness

Peter Scholl

NIST MPTC Workshop, 28 September 2023

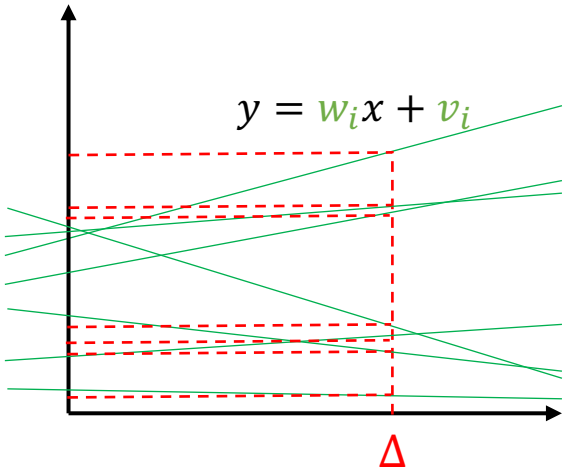
AARHUS UNIVERSITY

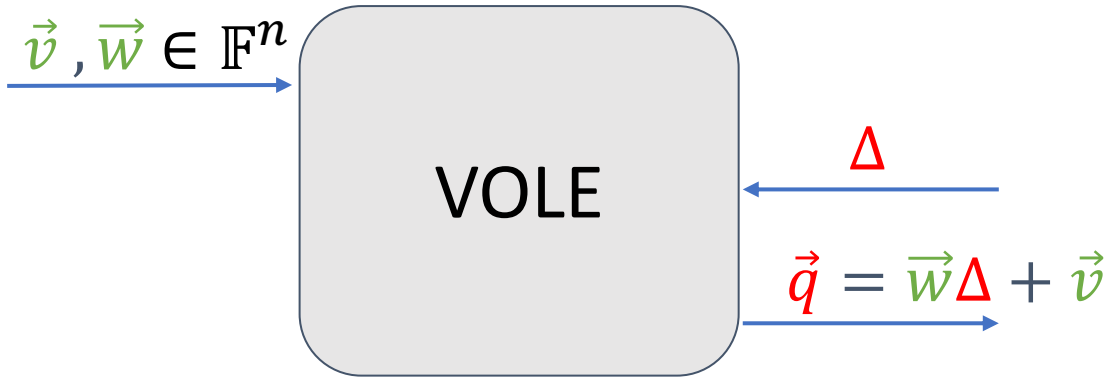
Overview



Vector Oblivious Linear Evaluation 🦫

[ADINZ 17, BCGI 18, Roy 22]



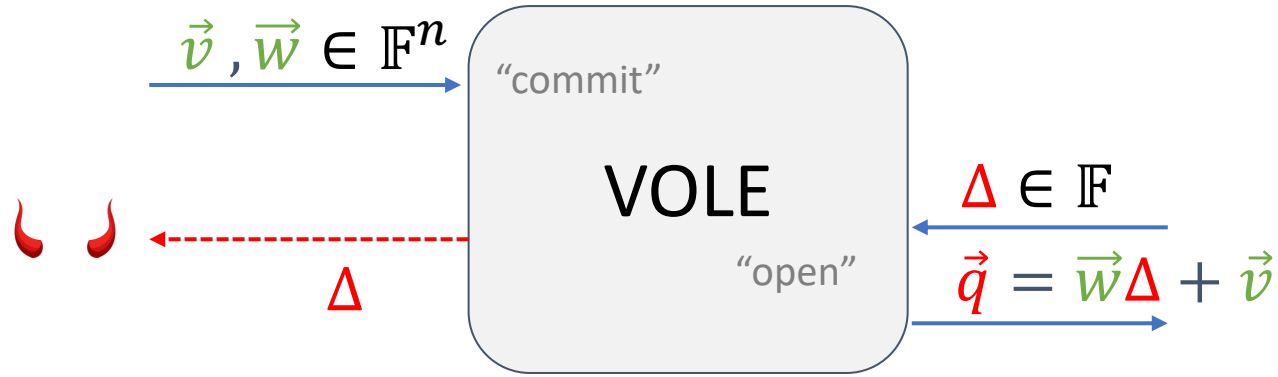


Variants:

- Subfield VOLE (e.g. correlated OT)
- Subspace VOLE

Sender-Private VOLE (aka VOLE-in-the-head)

[BBdSKORS 23]

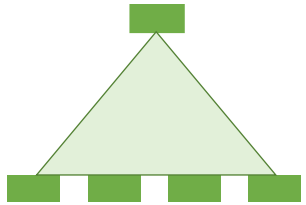


- Especially useful if protocol is public-coin
- Note: VOLE ⇒ sender-private VOLE

Building Sender-Private VOLE (in small fields)

[BBdSKORS 23]

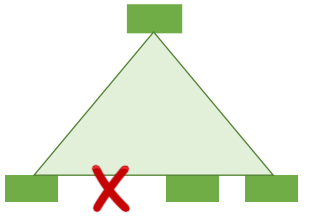
All-but-one
vector commitment



Commit to n random strings

Challenge Δ

Open $n-1$



Convert to VOLE
(in $\mathbb{F}_n$)

[Roy 22]

$\vec{u}, \vec{v}$

$\vec{q} = \vec{u}\Delta + \vec{v}$

Case Study: Zero-Knowledge from VOLE

VOLE-ZK: fast, linear-size proofs for Boolean and/or arithmetic statements

Designated Verifier

[DIO 21, WYKW 21, …]

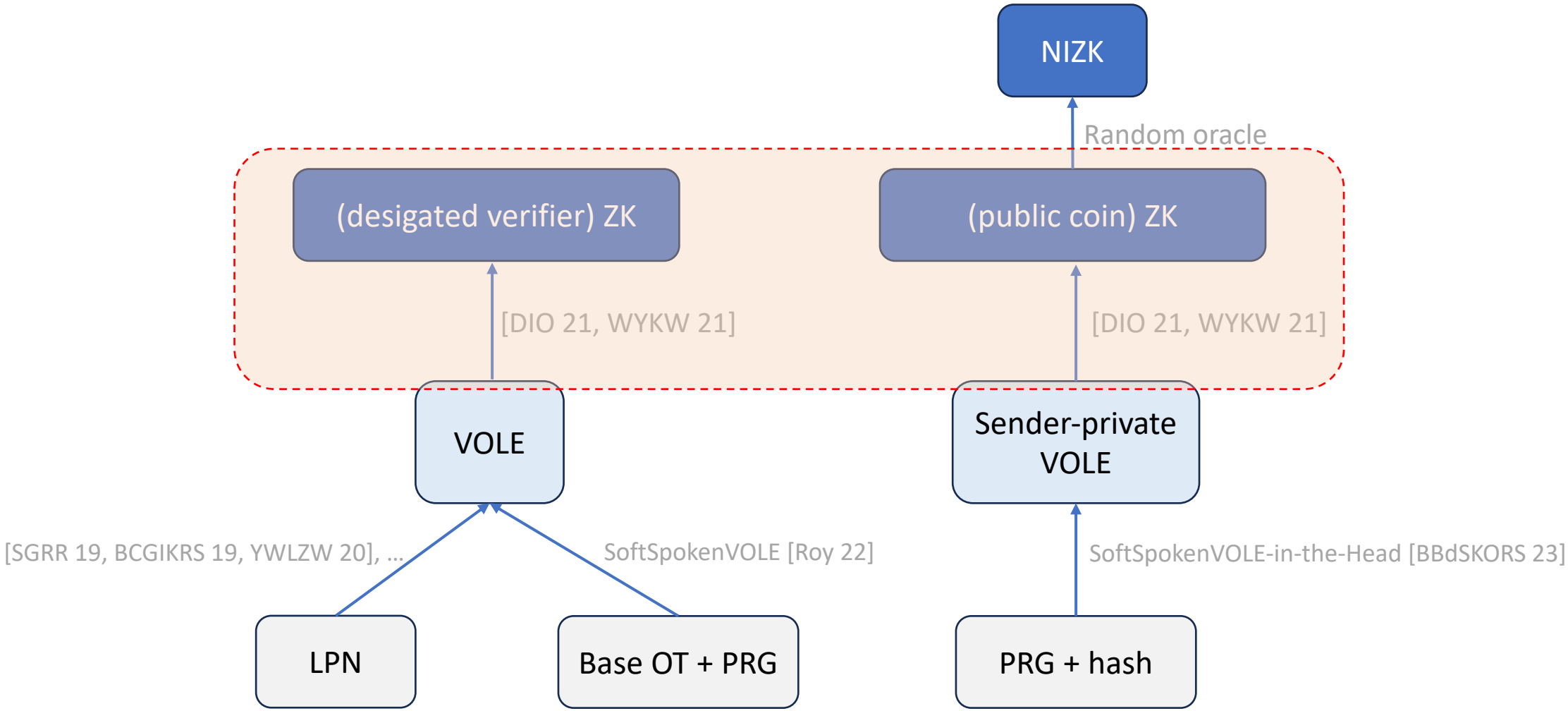
- More efficient
- Good enough for many threshold protocols

Publicly Verifiable

[BBdSKORS 23]

- Non-interactive
  - Good for PQ signatures, e.g. FAEST
- Can help with public verifiability/identifiable abort

Case Study: Zero-Knowledge from VOLE



Other Applications of VOLE

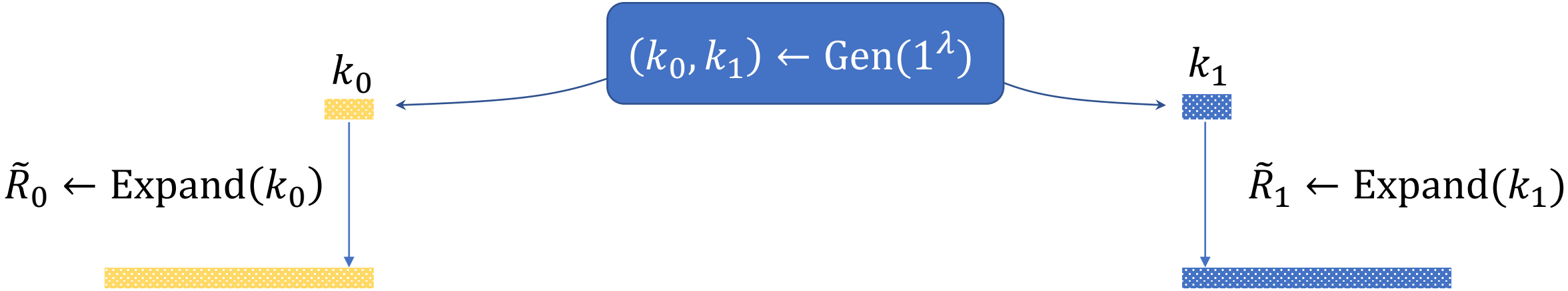
- Authenticated garbling [WRK 17, …]
  - Correlated OT ($\mathbb{F}_2$)
- Threshold ECDSA [DKLS, …]
  - Scalar-vector multiplication in $\mathbb{F}_p$
- General-purpose MPC in large fields [DPSZ 12, …]
  - SPDZ etc.

Pseudorandom Correlation Generators (PCGs)

[BCGI 18, BCGIKS 19]

- Target correlation: (R_0, R_1)

- Algorithms Gen, Expand:



Security: $(\tilde{R}_0, \tilde{R}_1) \approx (R_0, R_1)$, when given k_b

Example PCG Constructions

- VOLE/OT:
  - Via learning parity with noise (LPN) [BCGI 18, BCGIKS 19]
- Multiplication triples
  - Ring LPN [BCGIKS 20]
- Also: pseudorandom correlation **functions** (PCF) for VOLE
  - Unbounded # of outputs
  - Paillier [OSY 21], or variants of LPN [BCGIKRS 22]

Is a trusted setup acceptable?

- PCG Gen algorithm: inherent trusted setup
  - Maybe OK if trusted client can generate keys
  - What happens when correlated randomness is used up?
    - PCF avoids this issue

- Distributed setup protocol
  - Securely set-up keys with multi-party protocol
  - Analogous to DKG

Distributed setup protocols: a definitional challenge

- Naïve solution: Π securely realizes Gen functionality
  - OK for passive security
  - Active security: not always practical!

- Current practice: Π securely computes correlated randomness (R_0, R_1)
  - With succinct communication

- Is there a better definition?
  - If not, distributed PCG protocols are just a special case of correlated randomness protocols

Conclusion

VOLE is an important form of correlated randomness

- Seems useful to consider for standardization
- Possible submission from FAEST team:
  - VOLE-ZK/VOLE-in-the-head

- PCGs/PCFs
  - Useful tool for saving bandwidth
  - Harder to standardize as a gadget (application-dependent)