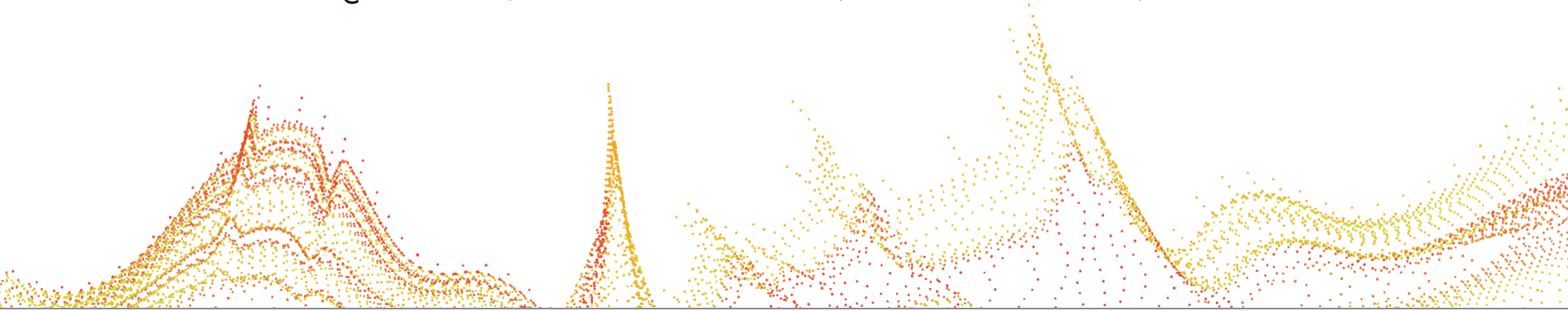
Rackle: Adaptively-Secure Three Round Threshold Schnorr from DL

Guilhem Niot[1,2], Michael Reichle[3], Kaoru Takemure[1,4]
[1]PQShield, [2]Univ Rennes, [3]ETH Zurich, [4]AIST



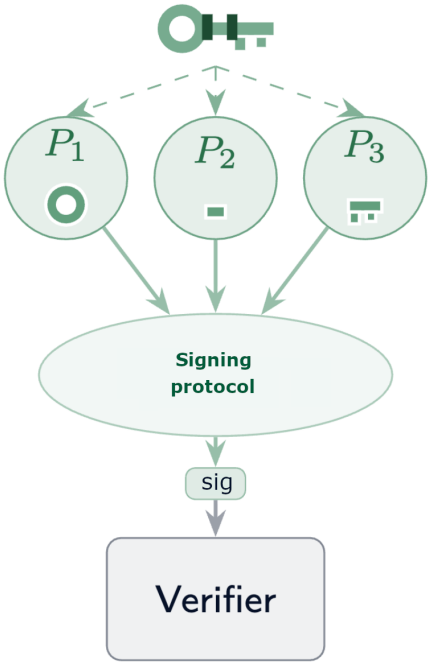
This work

Design a Threshold Schnorr:

- With **adaptive security**, and strongest **TS-SUF-4** security.
- Under most conservative assumptions (e.g. **DL**)
- In a **small number of rounds (3)**.

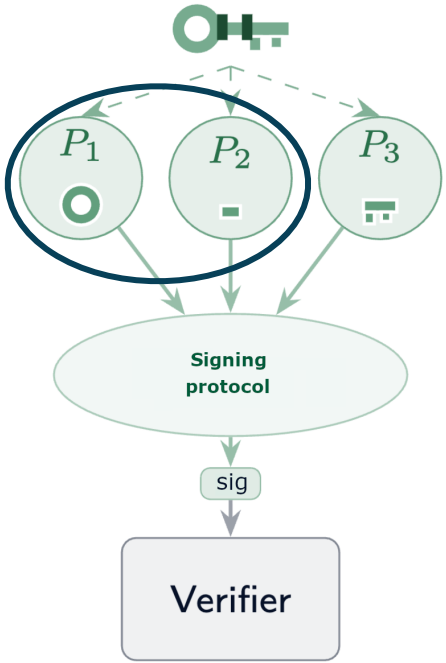
Scheme	Rounds (On+Off)	Assumptions
Rackle	2 + 1	DL

Threshold Signatures



- Distribute the signing key to mitigate single point of failures.

Threshold Signatures



- Distribute the signing key to mitigate single point of failures.
- t-out-of-n threshold signing.
  - Any set of t parties can sign.
  - Any set of fewer than t parties cannot.

Formal Model for Threshold Signatures

- **Correctness:** Just check that the signature is valid when any t signers execute the protocol honestly!

Formal Model for Threshold Signatures

- **Correctness:** Just check that the signature is valid when any t signers execute the protocol honestly!

- **Security:** *"Any set of fewer than t parties cannot sign."*

Corruption model of up to t-1 parties?

What does it mean to be unable to produce a signature?

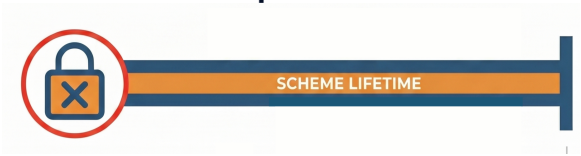
Formal Model for Threshold Signatures

- **Correctness:** Just check that the signature is valid when any t signers execute the protocol honestly!

- **Security:** *"Any set of fewer than t parties cannot sign."*

Corruption model of up to t-1 parties?

Static



Adaptive

What does it mean to be unable to produce a signature?

Formal Model for Threshold Signatures

- **Correctness:** Just check that the signature is valid when any t signers execute the protocol honestly!

- **Security:** *"Any set of fewer than t parties cannot sign."*

Corruption model of up to t-1 parties?

Static

Adaptive

What does it mean to be unable to produce a signature?

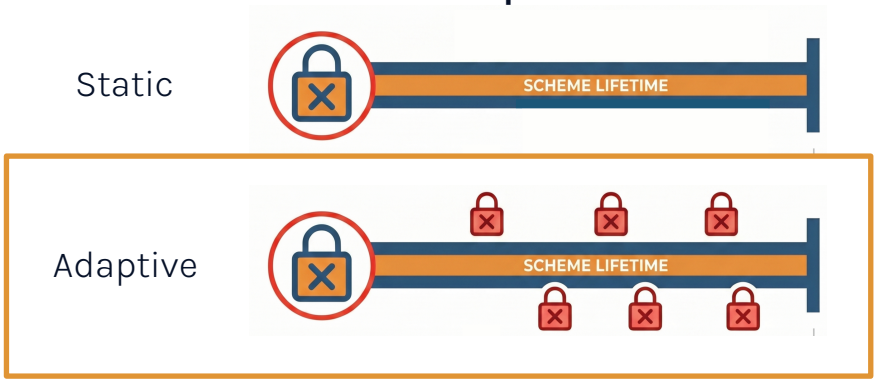
- Honest signers participate, and some signatures are expected to be produced.
- Hierarchy of Bellare et al. (2022)
  - From **TS-UF-0** (one honest participant for a target message = signature produced) to **TS-(S)UF-4** (precise per-session counting of participants).

Formal Model for Threshold Signatures

- **Correctness:** Just check that the signature is valid when any t signers execute the protocol honestly!

- **Security:** *"Any set of fewer than t parties cannot sign."*

Corruption model of up to t-1 parties?



What does it mean to be unable to produce a signature?

- Honest signers participate, and some signatures are expected to be produced.
- Hierarchy of Bellare et al. (2022)
  - From **TS-UF-0** (one honest participant for a target message = signature produced) to **TS-(S)UF-4** (precise per-session counting of participants).

Adaptive Threshold Schnorr

Scheme	Rounds (On+Off)	Assumptions
FROST	1 + 1	AOMDL + LDVR + AGM
Frost-Mask	2 + 1	AOMDL
Glacius	5 + 0	DDH
Gargos	3 + 0	DDH
Crackle	4 + 1	DL
Rackle	2 + 1	DL

Note: next talk, adaptive FROST variant in 1+1 rounds, under AOMDL + AGM.

Our Starting Point: Crackle

Scheme	Rounds (On+Off)	Assumptions
Crackle	4 + 1	DL

Our Starting Point: Crackle

Scheme	Rounds (On+Off)	Assumptions
Crackle	4 + 1	DL

Schnorr Signing

01. $r \leftarrow \mathbb{Z}_p$
02. $R \leftarrow g^r$
03. $c \leftarrow H(R, \mathsf{msg})$
04. $z \leftarrow c \cdot \mathsf{sk} + r$

Static Threshold Schnorr

Round 1

$r_i \leftarrow \mathbb{Z}_p$
$R_i \leftarrow g^{r_i}$
Reveal $H(R_i)$

Round 2

Reveal R_i

Round 3

$R \leftarrow \prod_i R_i$
$c \leftarrow H(R, \mathsf{msg})$
$z_i \leftarrow c \cdot \mathsf{sk}_i + r_i$
Reveal z_i.

Our Starting Point: Crackle

Scheme	Rounds (On+Off)	Assumptions
Crackle	4 + 1	DL

Crackle

ZeroShare = ROM-based tool to non-interactively sample shares of zero from a fixed seed.

Round 1

$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$

Reveal str_i

Round 2

$\Delta_i \leftarrow \text{ZeroShare}(\text{str}_1 \,|\, \ldots)$

$r_i \leftarrow \mathbb{Z}_p$

$R_i \leftarrow g^{r_i} \cdot g^{\Delta_i}$

Reveal $H(R_i)$

Round 3

Sign transcript

Round 4

Reveal R_i

Round 5

$R \leftarrow \prod_i R_i$

$c \leftarrow H(R, \text{msg})$

$\Delta_i' \leftarrow \text{ZeroShare}(\text{transcript})$

$z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta_i'$

Reveal z_i.

Our Starting Point: Crackle

Observation 1: ZeroShare randomizes values and avoids revealing individual elements until corruption, e.g. allows to choose sk_i and r_i randomly at corruption time, and program ZeroShare to be consistent.

Our Starting Point: Crackle

Observation 2: The signature in round 3 ensures that at the time R_i is revealed, all honest signers have the same view.

Crackle

unique-aggr-nonce

Round 1	Round 2	Round 3	Round 4	Round 5
$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$ Reveal str_i	$\Delta_i \leftarrow \text{ZeroShare}(\text{str}_1 \mid \ldots)$ $r_i \leftarrow \mathbb{Z}_p$ $R_i \leftarrow g^{r_i} \cdot g^{\Delta_i}$ Reveal $H(R_i)$	Sign transcript	Reveal R_i	$R \leftarrow \prod_i R_i$ $c \leftarrow H(R, \text{msg})$ $\Delta_i' \leftarrow \text{ZeroShare}(\text{transcript})$ $z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta_i'$ Reveal z_i.

Designing Rackle

Round 1

$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$

Reveal str_i

Round 2

$\Delta_i \leftarrow \text{ZeroShare}(\text{str}_1 \mid \ldots)$

$r_i \leftarrow \mathbb{Z}_p$

$R_i \leftarrow g^{r_i} \cdot g^{\Delta_i}$

Reveal $H(R_i)$

Round 3

Sign transcript

Round 4

Reveal R_i

Round 5

$R \leftarrow \prod_i R_i$

$c \leftarrow H(R, \text{msg})$

$\Delta_i' \leftarrow \text{ZeroShare}(\text{transcript})$

$z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta_i'$

Reveal z_i.

Designing Rackle

Homomorphic BCJ commitment + SIM-EXT NIZK

Round 1

$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$

Reveal str_i

Round 2

$s_i \leftarrow \mathbb{Z}_p$

$r_i \leftarrow \mathbb{Z}_p$

$R_i \leftarrow g^{r_i}$

Reveal

$C_i \leftarrow \text{BCJ.Commit}(R_i, s_i)$

+proof well-formed

Round 3

Sign transcript

Round 4

Reveal R_i

Round 5

$R \leftarrow \prod_i R_i$

$c \leftarrow H(R, \text{msg})$

$\Delta_i' \leftarrow \text{ZeroShare(transcript)}$

$z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta_i'$

Reveal z_i.

Designing Rackle

Homomorphic BCJ commitment + SIM-EXT NIZK

Use ZeroShare

function to mask (homomorphic) commitment openings

Round 1	Round 2	Round 3	Round 4	Round 5
~~$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$~~ ~~Reveal str_i~~	$s_i \leftarrow \mathbb{Z}_p$ $r_i \leftarrow \mathbb{Z}_p$ $R_i \leftarrow g^{r_i}$ Reveal $C_i \leftarrow \text{BCJ.Commit}(R_i, s_i)$ +proof well-formed	~~Sign transcript~~	$\Delta_i \leftarrow \text{ZeroShare(transcript)}$ Reveal $s'_i \leftarrow s_i + \Delta_i$	$R \leftarrow \text{Open}(\prod_i C_i, \sum_i s'_i)$ $c \leftarrow H(R, \text{msg})$ $\Delta'_i \leftarrow \text{ZeroShare(transcript)}$ $z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta'_i$ Reveal z_i.

Designing Rackle

Homomorphic BCJ commitment + SIM-EXT NIZK

Use ZeroShare

function to mask (homomorphic) commitment openings

Round 1

$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$

Reveal str_i

Round 2

$s_i \leftarrow \mathbb{Z}_p$

$r_i \leftarrow \mathbb{Z}_p$

$R_i \leftarrow g^{r_i}$

Reveal

$C_i \leftarrow \text{BCJ.Commit}(R_i, s_i)$

+proof well-formed

Round 3

Sign transcript

Round 4

$\Delta_i \leftarrow \text{ZeroShare(transcript)}$

Reveal $s'_i \leftarrow s_i + \Delta_i$

hidden-nonces

Round 5

$R \leftarrow \text{Open}(\prod_i C_i, \sum_i s'_i)$

$c \leftarrow H(R, \text{msg})$

$\Delta'_i \leftarrow \text{ZeroShare(transcript)}$

$z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta'_i$

Reveal z_i.

hidden-nonces

Designing Rackle

Homomorphic BCJ commitment + SIM-EXT NIZK

Use ZeroShare

function to mask (homomorphic) commitment openings

Round 1

$\text{str}_i \leftarrow \{0,1\}^{2\lambda}$

Reveal str_i

Round 2

$s_i \leftarrow \mathbb{Z}_p$

$r_i \leftarrow \mathbb{Z}_p$

$R_i \leftarrow g^{r_i}$

Reveal

$C_i \leftarrow \text{BCJ.Commit}(R_i, s_i)$

+proof well-formed

hidden-nonces

Round 3

Sign transcript

Round 4

unique-aggr-nonce

$\Delta_i \leftarrow \text{ZeroShare(transcript)}$

Reveal $s'_i \leftarrow s_i + \Delta_i$

hidden-nonces

Round 5

$R \leftarrow \text{Open}(\prod_i C_i, \sum_i s'_i)$

$c \leftarrow H(R, \text{msg})$

$\Delta'_i \leftarrow \text{ZeroShare(transcript)}$

$z_i \leftarrow c \cdot \text{sk}_i + r_i + \Delta'_i$

Reveal z_i.

Pillar 1: BCJ commitment with product equivocability

- BCJ commitments (Bagherzandi et al., CCS'08) were already known to be homomorphic, binding and allows to equivocate for Schnorr nonces R
  - e.g. one can sample a commitment, and find an acceptable opening only after selecting R.

Pillar 1: BCJ commitment with product equivocability

- BCJ commitments (Bagherzandi et al., CCS'08) were already known to be homomorphic, binding and allows to equivocate for Schnorr nonces R
  - e.g. one can sample a commitment, and find an acceptable opening only after selecting R.
- In this work, we reveal an aggregate opening, and up to t-1 openings for corrupted users (adaptively).
  - We show that we can equivocate all these openings.

Pillar 2: NIZK with Simulation-Extractability with Labeled Fischlin

- For the proof to go through, we need to extract nonces from malicious commitments.
  - Add **extractable** NIZK to them as BCJ commitments are not extractable.
  - BUT, proofs for honest signers need to be **simulated** as there is no witness. And **explainable** in case of later corruption.

Fischlin transform can be both set up in extractable and simulation mode

→ We separate the space of extractable and simulated proofs with labels.

Pillar 3: Masking-based Aggregated Openings

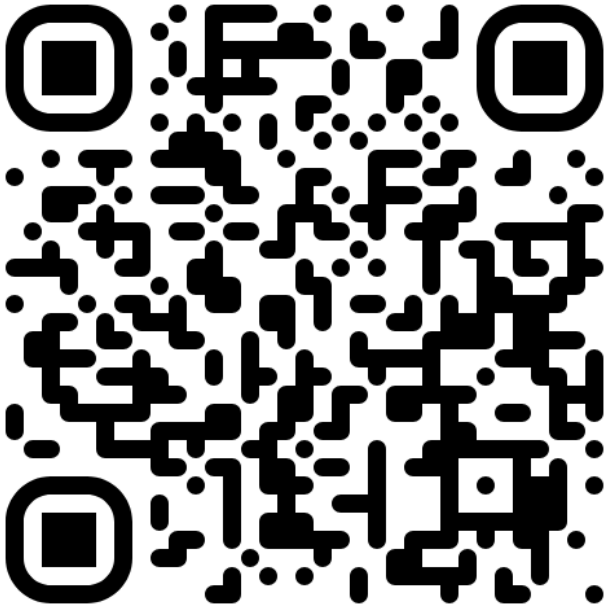
- The ZeroShare function allows to only open the sum of the openings, thus keeping individual nonces hidden (**hidden-nonces** property).
- Further, ZeroShare implicitly authenticates its input, here the transcript. Hence, if two honest users disagree on their view, the commitments cannot be opened (under binding of the commitment), enforcing **unique-aggr-nonce** implicitly.

Conclusion

Scheme	Rounds (On+Off)	Assumptions
Rackle	2 + 1	DL

- Rackle achieves the strongest security notions under DL in only 3 rounds!
- Along the way, other interesting techniques
  - Restricted product equivocability of BCJ commitments.
  - Simulation-Equivocability of Labeled Fischlin.
  - Masked Aggregated Openings.

Questions?



Rackle: Adaptively-Secure Three Round Threshold Schnorr from DL

Guilhem Niot, Michael Reichle, Kaoru Takemure

https://eprint.iacr.org/2025/1941