Sparkle Revisited:
Proving Tight Adaptive Security of a Simple Schnorr Threshold Scheme

January 26, MPTS 2026
NIST Workshop on Multi-Party Threshold Schemes

Ojaswi Acharya (UMass Amherst)
Georg Fuchsbauer (TU Wien)
Adam O'Neill (UMass Amherst)
Marek Sefranek (TU Wien)

"Revisiting the Security of Sparkle"
on ePrint soon…

Threshold Signatures

- t-out-of-n Threshold Signatures (TS):
  - Distribute signing power among n parties
  - Any subset $\geq t$ can jointly generate a standard signature
  - Compatible with a single, aggregate public key

Threshold Signatures

- t-out-of-n Threshold Signatures (TS):
  - Distribute signing power among n parties
  - Any subset $\geq t$ can jointly generate a standard signature
  - Compatible with a single, aggregate public key
- Security models: static vs. adaptive corruptions

Threshold Signatures

- t-out-of-n Threshold Signatures (TS):
  - Distribute signing power among n parties
  - Any subset $\geq t$ can jointly generate a standard signature
  - Compatible with a single, aggregate public key
- Security models: static vs. adaptive corruptions
- Schnorr signatures:

Threshold Signatures

- *t*-out-of-*n* Threshold Signatures (TS):
  - Distribute signing power among *n* parties
  - Any subset $\geq t$ can jointly generate a standard signature
  - Compatible with a single, aggregate public key
- Security models: static vs. adaptive corruptions
- Schnorr signatures:
  - Standardized & widely deployed (e.g., EdDSA, Taproot)
  - Schnorr TS: “out-of-the-box” compatibility with plain Schnorr verification

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm
- However, Bacho et al. [BLT+24] identified a gap in original security proof

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm
- However, Bacho et al. [BLT+24] identified a gap in original security proof
- To address this, Sparkle+ was introduced:

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm
- However, Bacho et al. [BLT+24] identified a gap in original security proof
- To address this, Sparkle+ was introduced:
  - Adds auxiliary signature scheme $\Longrightarrow$ significant overhead

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm
- However, Bacho et al. [BLT+24] identified a gap in original security proof
- To address this, Sparkle+ was introduced:
  - Adds auxiliary signature scheme $\Longrightarrow$ significant overhead
  - Proof of full adaptive security also later invalidated [CS25, CKK+25]

Motivation

- Our focus: Sparkle (Crites, Komlo, and Maller [CKM23])
- Concretely efficient & natural 3-round Schnorr TS scheme
- Follows a commit-reveal-sign paradigm
- However, Bacho et al. [BLT+24] identified a gap in original security proof
- To address this, Sparkle+ was introduced:
  - Adds auxiliary signature scheme $\Longrightarrow$ significant overhead
  - Proof of full adaptive security also later invalidated [CS25, CKK+25]
- These negative results identify proof deficiencies, not practical attacks!

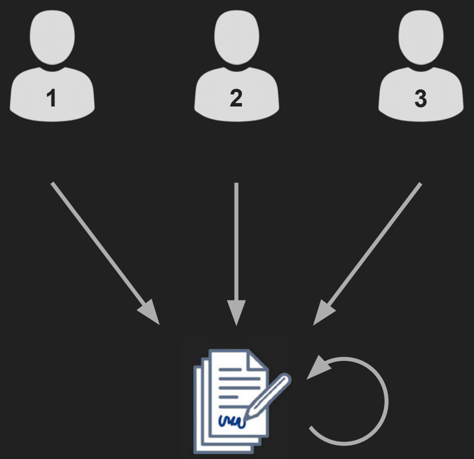
TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Our Question

Can the original Sparkle scheme be proved—statically or even adaptively—secure?

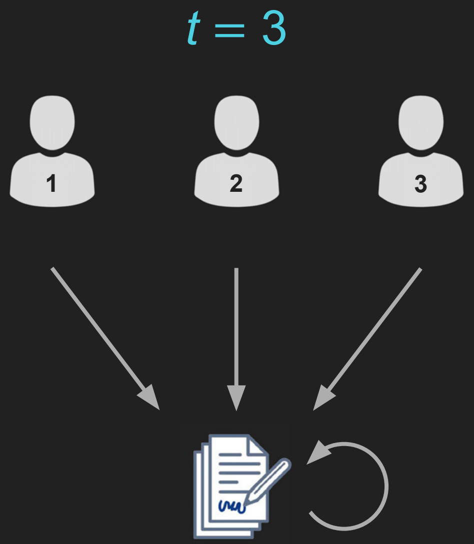
Goal 1: Full Adaptive Security

- Static corruption model



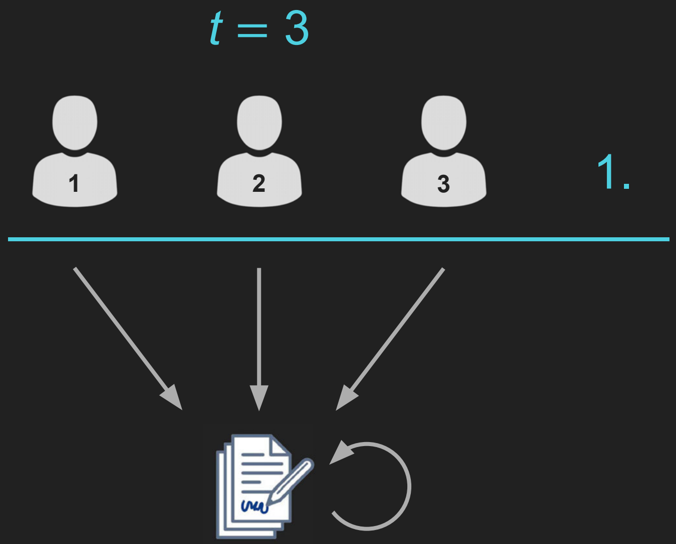
Goal 1: Full Adaptive Security

- Static corruption model



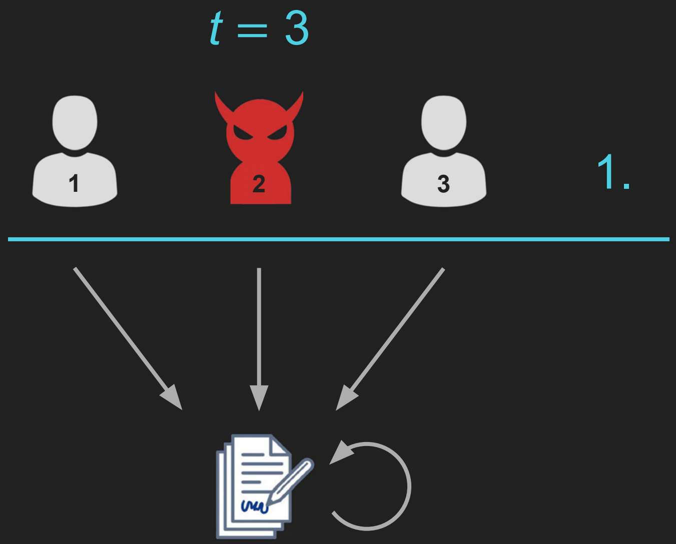
Goal 1: Full Adaptive Security

- Static corruption model



Goal 1: Full Adaptive Security

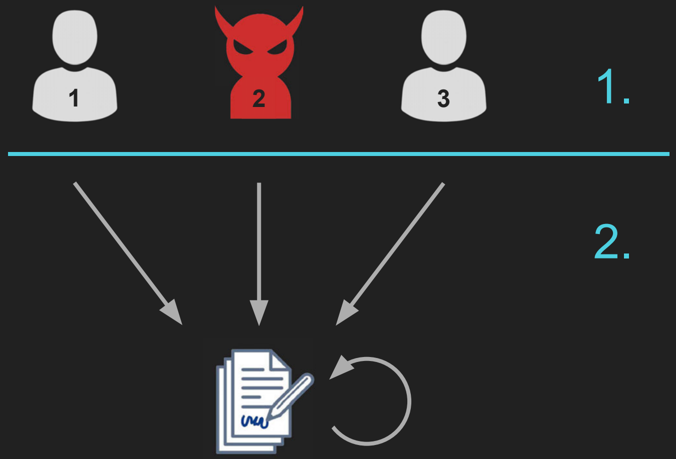
- Static corruption model



Goal 1: Full Adaptive Security

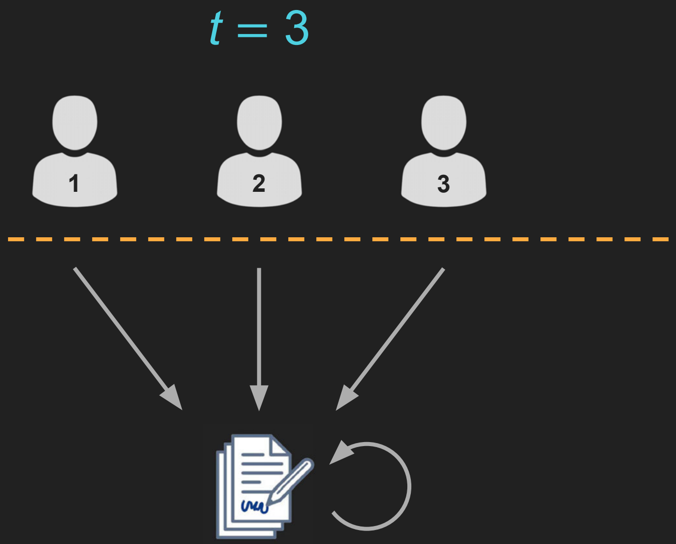
- Static corruption model

$t = 3$



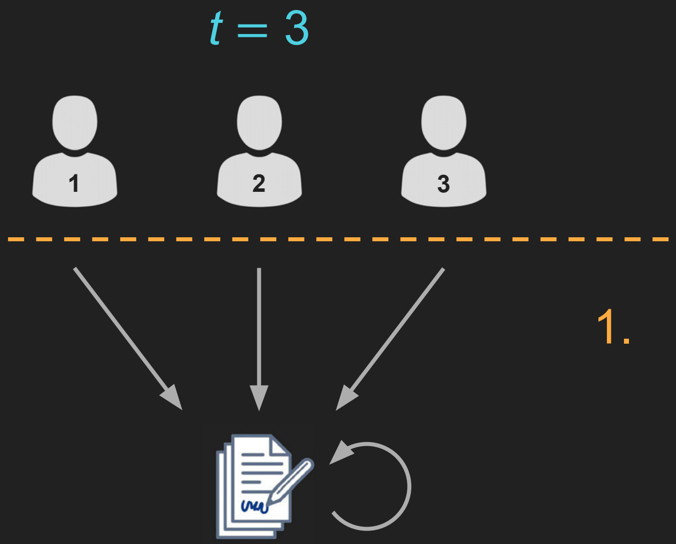
Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:



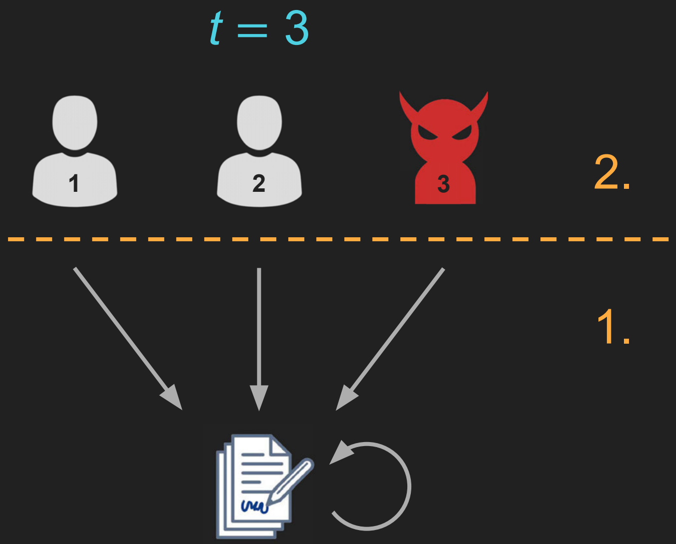
Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:



Goal 1: Full Adaptive Security

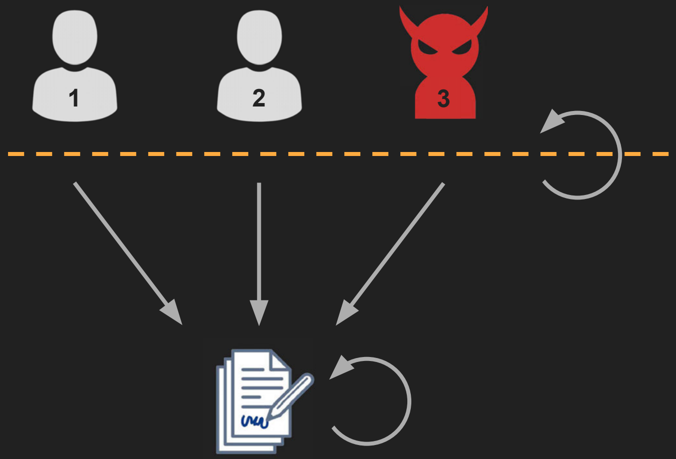
- Static corruption model
- Adaptive corruption model:



Goal 1: Full Adaptive Security

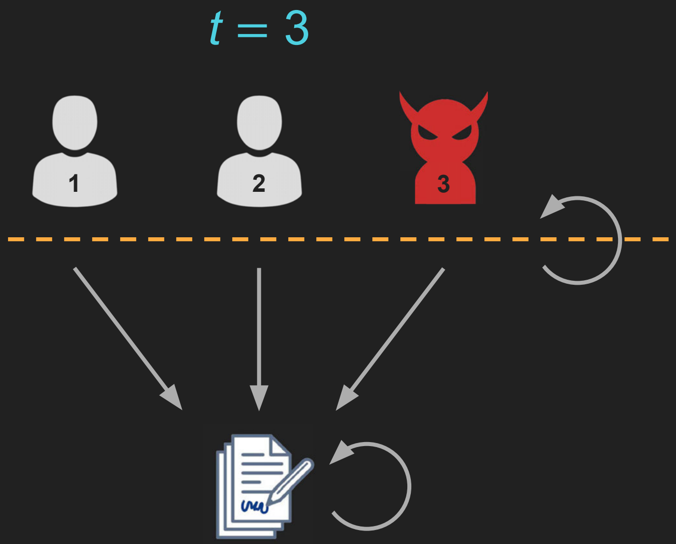
- Static corruption model
- Adaptive corruption model:
  - More realistic & practically relevant

$t = 3$



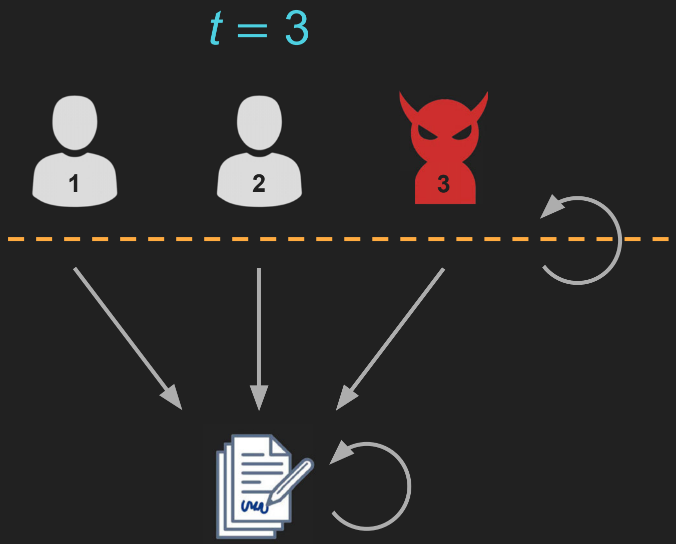
Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:
  - More realistic & practically relevant
- Full adaptive security:



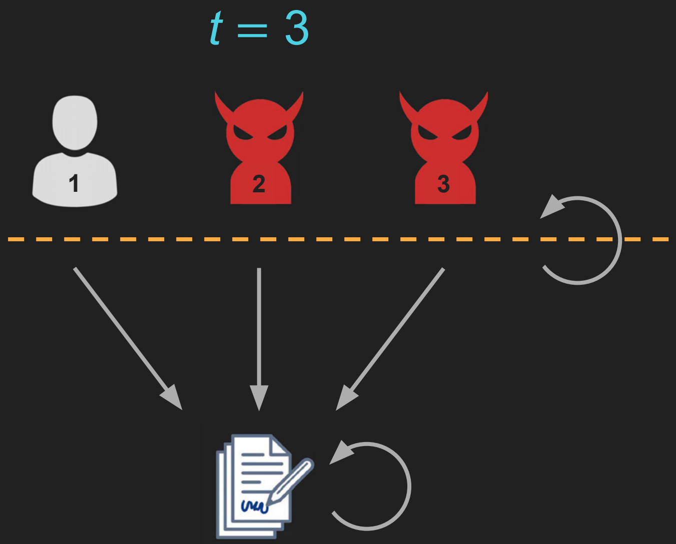
Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:
  - More realistic & practically relevant
- Full adaptive security:
  - Allows (adaptive) corruptions of up to one fewer than the signing threshold t



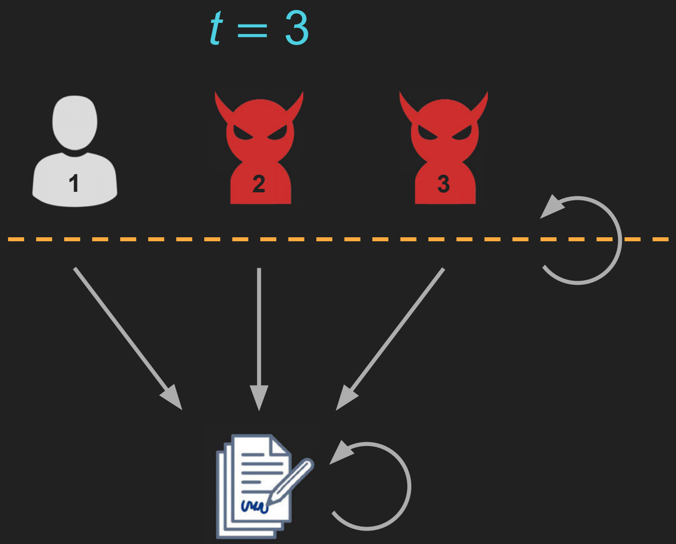
Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:
  - More realistic & practically relevant
- Full adaptive security:
  - Allows (adaptive) corruptions of up to one fewer than the signing threshold t



Goal 1: Full Adaptive Security

- Static corruption model
- Adaptive corruption model:
  - More realistic & practically relevant
- Full adaptive security:
  - Allows (adaptive) corruptions of up to one fewer than the signing threshold t
  - Achieving this notion poses several challenges



Goal 2: “Pure” ROM Security

- Random oracle model (ROM): idealizes hash function as random function

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Goal 2: “Pure” ROM Security

- Random oracle model (ROM): idealizes hash function as random function
- Security proofs for Schnorr-based constructions only known in ROM

Goal 2: “Pure” ROM Security

- Random oracle model (ROM): idealizes hash function as random function
- Security proofs for Schnorr-based constructions only known in ROM
- However, for full adp. security, many Schnorr TS schemes also rely on AGM:

Goal 2: “Pure” ROM Security

- Random oracle model (ROM): idealizes hash function as random function
- Security proofs for Schnorr-based constructions only known in ROM
- However, for full adp. security, many Schnorr TS schemes also rely on AGM:
  - Algebraic group model (AGM) only considers “algebraic” adversaries

Goal 2: "Pure" ROM Security

- Random oracle model (ROM): idealizes hash function as random function
- Security proofs for Schnorr-based constructions only known in ROM
- However, for full adp. security, many Schnorr TS schemes also rely on AGM:
  - Algebraic group model (AGM) only considers "algebraic" adversaries
  - E.g. FROST and its variants and the (flawed) full adp. security proof of Sparkle+

Goal 2: “Pure” ROM Security

- Random oracle model (ROM): idealizes hash function as random function
- Security proofs for Schnorr-based constructions only known in ROM
- However, for full adp. security, many Schnorr TS schemes also rely on AGM:
  - Algebraic group model (AGM) only considers “algebraic” adversaries
  - E.g. FROST and its variants and the (flawed) full adp. security proof of Sparkle+
- Our goal is to avoid additional idealized models

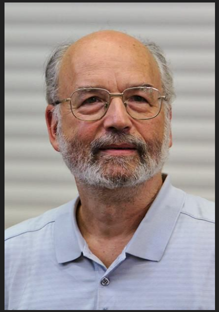
Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with 128-bit security – how large does the group order need to be?

Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with 128-bit security – how large does the group order need to be?

We should use a group order of 256 bits!

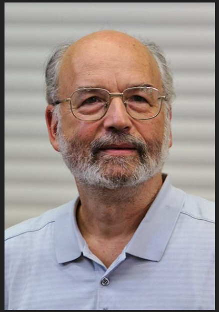


Practitioners

Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with 128-bit security – how large does the group order need to be?

We should use a group order of **256** bits!



Practitioners

Best known attack:
break DL, which takes time $O(\sqrt{|\mathbb{G}|})$

Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with 128-bit security – how large does the group order need to be?

We should use a group order of 768 bits!

Theoreticians

Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with **128-bit security** – how large does the **group order** need to be?

We should use a group order of **768** bits!

[PS96] in **ROM**:

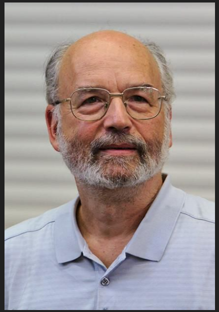
$$\mathsf{Adv}^{\text{euf-cma}}_{\mathsf{Sch}[\mathbb{G}]} \le q_h \cdot \sqrt{\mathsf{Adv}^{\text{dl}}_{\mathbb{G}}} + \dots$$

Theoreticians

Goal 3: Tight Security

- Suppose want to use Schnorr signatures over elliptic curve with 128-bit security – how large does the group order need to be?

We should use a group order of **256** bits!



Practitioners

We should use a group order of **768** bits!

Theoreticians

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Summary of Our Results

1. Introduce new simulation technique $\Longrightarrow$ static security of Sparkle

Summary of Our Results

1. Introduce new simulation technique $\Longrightarrow$ static security of Sparkle
2. Introduce new assumption VCDL $\Longrightarrow$ tight full adaptive security of Sparkle

Summary of Our Results

1. Introduce new simulation technique $\Longrightarrow$ static security of Sparkle
2. Introduce new assumption VCDL $\Longrightarrow$ tight full adaptive security of Sparkle
3. Justify VCDL: reduce VCDL to necessary assumption LDVR [CKK+25] when idealizing the group

Comparison with Selected Schemes

Scheme	Rounds	Model	Comm. / Signer	Comp. / Signer
FROST	2	ROM+AGM	$2\mathbb{G} + \mathbb{Z}_p$	3 Exp
Sparkle	3	ROM*	$\mathbb{G} + 2\mathbb{Z}_p$	1 Exp

* Our new result

Comparison with Selected Schemes

Scheme	Rounds	Model	Comm. / Signer	Comp. / Signer
FROST	2	ROM+AGM	$2\mathbb{G} + \mathbb{Z}_p$	3 Exp
Sparkle	3	ROM*	$\mathbb{G} + 2\mathbb{Z}_p$	1 Exp
Sparkle+	3	ROM+AGM	$2\mathbb{G} + \mathbb{Z}_p + \lvert\text{DS.S}\rvert$	1 Exp + DS.S + t DS.V

* Our new result

DS = auxiliary signature scheme

Comparison with Selected Schemes

Scheme	Rounds	Model	Comm. / Signer	Comp. / Signer
FROST	2	ROM+AGM	$2\mathbb{G} + \mathbb{Z}_p$	3 Exp
Sparkle	3	ROM*	$\mathbb{G} + 2\mathbb{Z}_p$	1 Exp
Sparkle+	3	ROM+AGM	$2\mathbb{G} + \mathbb{Z}_p + \lvert\mathsf{DS.S}\rvert$	1 Exp + DS.S + t DS.V
Gargos	3	ROM	$2\mathbb{G} + 7\mathbb{Z}_p$	8 Exp + NIZK.P + t NIZK.V

* Our new result

DS = auxiliary signature scheme

NIZK = non-interactive zero-knowledge proof system

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \rightarrow \mathbb{Z}_p$

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:

  1. Commit to random nonce $R_k \leftarrow g^{r_k}$ for $r_k \xleftarrow{\$} \mathbb{Z}_p$: $\mathsf{cm}_k \leftarrow H_{\mathsf{cm}}(k, R_k)$

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:
  1. Commit to random nonce $R_k \leftarrow g^{r_k}$ for $r_k \xleftarrow{\$} \mathbb{Z}_p$: $\mathsf{cm}_k \leftarrow H_{\mathsf{cm}}(k, R_k)$
  2. Open commitments: reveal R_k

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:
  1. Commit to random nonce $R_k \leftarrow g^{r_k}$ for $r_k \overset{\$}{\leftarrow} \mathbb{Z}_p$: $\mathsf{cm}_k \leftarrow H_{\mathsf{cm}}(k, R_k)$
  2. Open commitments: reveal R_k
  3. Verify openings and compute partial signature:

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:
  1. Commit to random nonce $R_k \leftarrow g^{r_k}$ for $r_k \xleftarrow{\$} \mathbb{Z}_p$: $\mathsf{cm}_k \leftarrow H_{\mathsf{cm}}(k, R_k)$
  2. Open commitments: reveal R_k
  3. Verify openings and compute partial signature:

     $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$, where $c \leftarrow H_{\mathsf{sig}}(X, m, \prod R_i)$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

The Sparkle Scheme

- Group $(\mathbb{G}, p, g)$, hash functions $H_{\mathsf{sig}}, H_{\mathsf{cm}} : \{0,1\}^* \to \mathbb{Z}_p$
- Shamir-secret-shared Schnorr secret key x with verification key shares X_i
- 3 signing rounds:
  1. Commit to random nonce $R_k \leftarrow g^{r_k}$ for $r_k \xleftarrow{\$} \mathbb{Z}_p$: $\mathsf{cm}_k \leftarrow H_{\mathsf{cm}}(k, R_k)$
  2. Open commitments: reveal R_k
  3. Verify openings and compute partial signature:

     $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^S$, where $c \leftarrow H_{\mathsf{sig}}(X, m, \prod R_i)$
- Final Schnorr signature: $(R, z) = (\prod R_i, \sum z_i)$

Problem 1:
How to Simulate Sparkle

Original Simulation Strategy

- Core idea [CKM23]:

Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^S$

Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$
  - Sample $z_k \xleftarrow{\$} \mathbb{Z}_p$ and set $R_k \leftarrow g^{z_k} \cdot X_k^{-c \cdot \lambda_k^{\mathcal{S}}}$ s.t. $r_k = z_k - c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$

TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$
  - Sample $z_k \xleftarrow{\$} \mathbb{Z}_p$ and set $R_k \leftarrow g^{z_k} \cdot X_k^{-c \cdot \lambda_k^{\mathcal{S}}}$ s.t. $r_k = z_k - c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$
- Bacho et al. [BLT+24] identified a gap:

Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^S$
  - Sample $z_k \xleftarrow{\$} \mathbb{Z}_p$ and set $R_k \leftarrow g^{z_k} \cdot X_k^{-c \cdot \lambda_k^S}$ s.t. $r_k = z_k - c \cdot x_k \cdot \lambda_k^S$
- Bacho et al. [BLT+24] identified a gap:
  - Simulation fails when adversary sends inconsistent commitments to different honest parties

Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^S$
  - Sample $z_k \xleftarrow{\$} \mathbb{Z}_p$ and set $R_k \leftarrow g^{z_k} \cdot X_k^{-c \cdot \lambda_k^S}$ s.t. $r_k = z_k - c \cdot x_k \cdot \lambda_k^S$
- Bacho et al. [BLT+24] identified a gap:
  - Simulation fails when adversary sends inconsistent commitments to different honest parties
  - Applies to both static & adaptive security

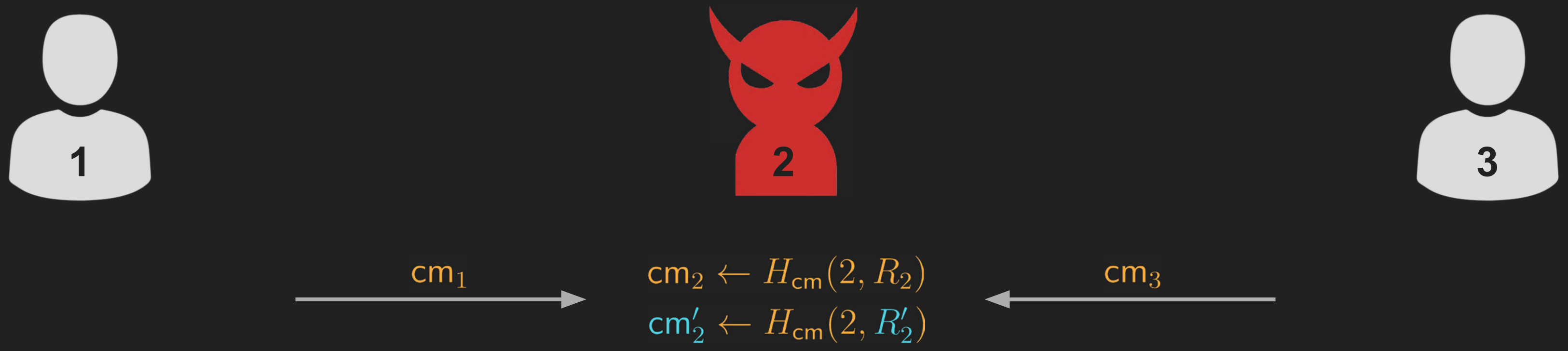
Original Simulation Strategy

- Core idea [CKM23]:
  - Instead of sampling $r_k \xleftarrow{\$} \mathbb{Z}_p$ for $R_k \leftarrow g^{r_k}$ and setting $z_k \leftarrow r_k + c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$
  - Sample $z_k \xleftarrow{\$} \mathbb{Z}_p$ and set $R_k \leftarrow g^{z_k} \cdot X_k^{-c \cdot \lambda_k^{\mathcal{S}}}$ s.t. $r_k = z_k - c \cdot x_k \cdot \lambda_k^{\mathcal{S}}$
- Bacho et al. [BLT+24] identified a gap:
  - Simulation fails when adversary sends inconsistent commitments to different honest parties
  - Applies to both static & adaptive security
  - Fixed in Sparkle+ by having parties sign their local views

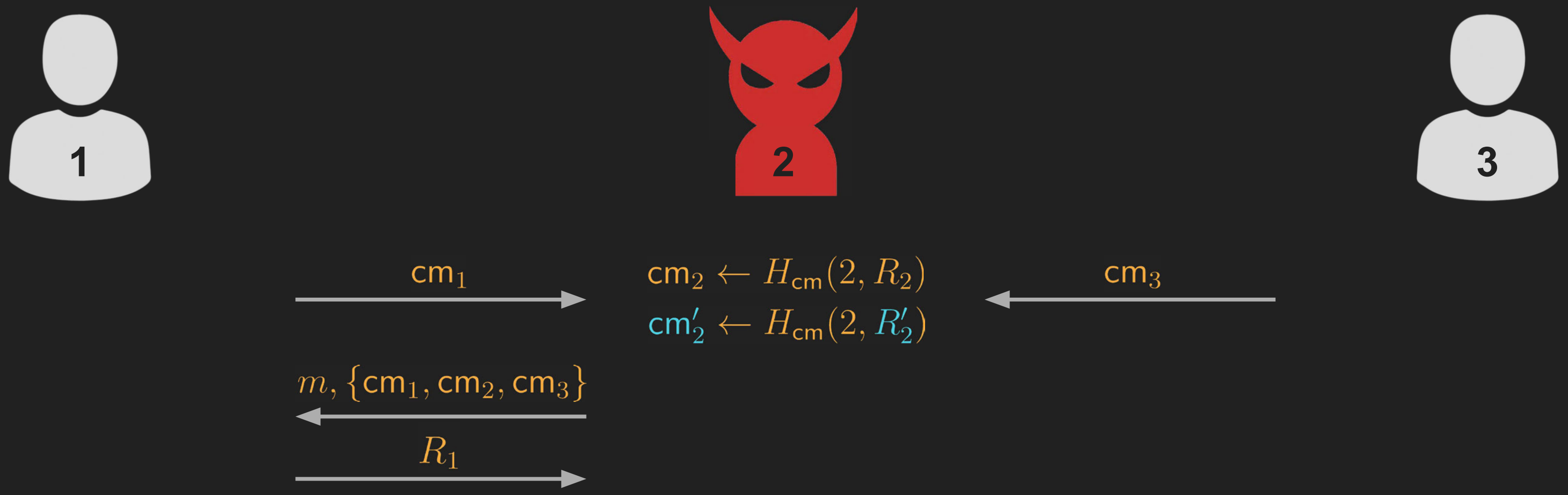
Counterexample: $t = 3, \ S = \{1, 2, 3\}$

Counterexample: $t = 3,\ S = \{1, 2, 3\}$

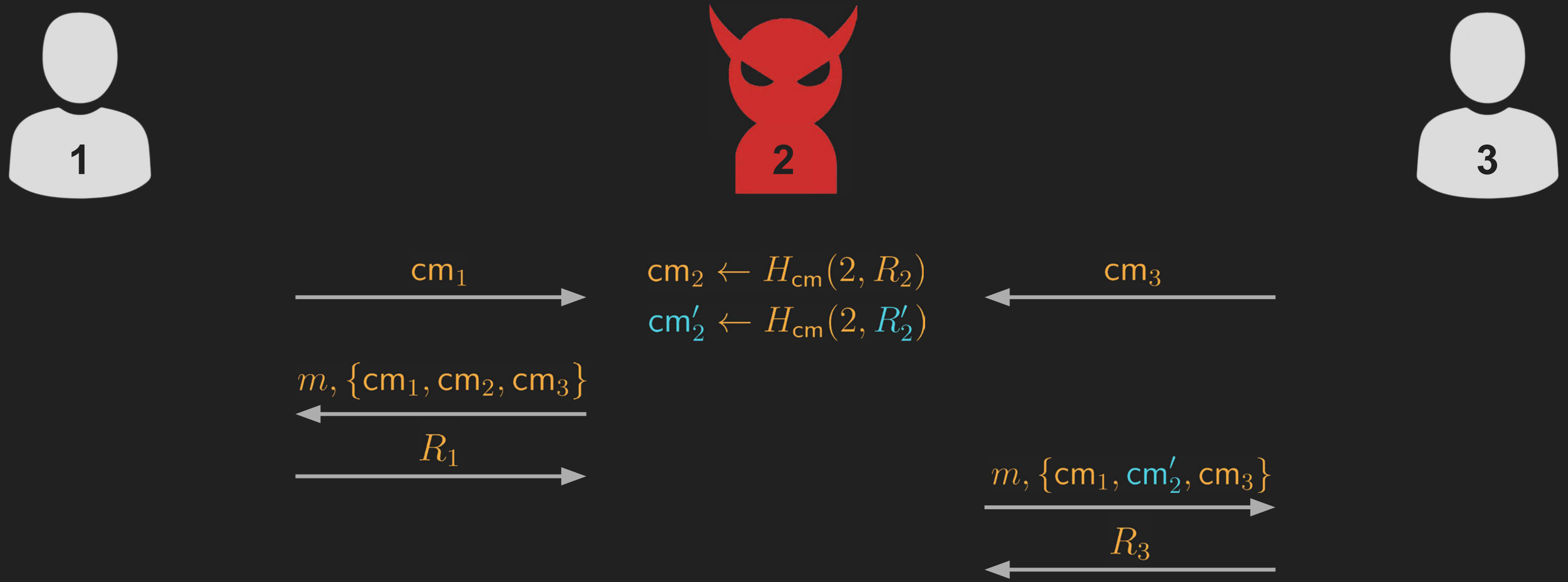
Counterexample: $t = 3, \ S = \{1, 2, 3\}$



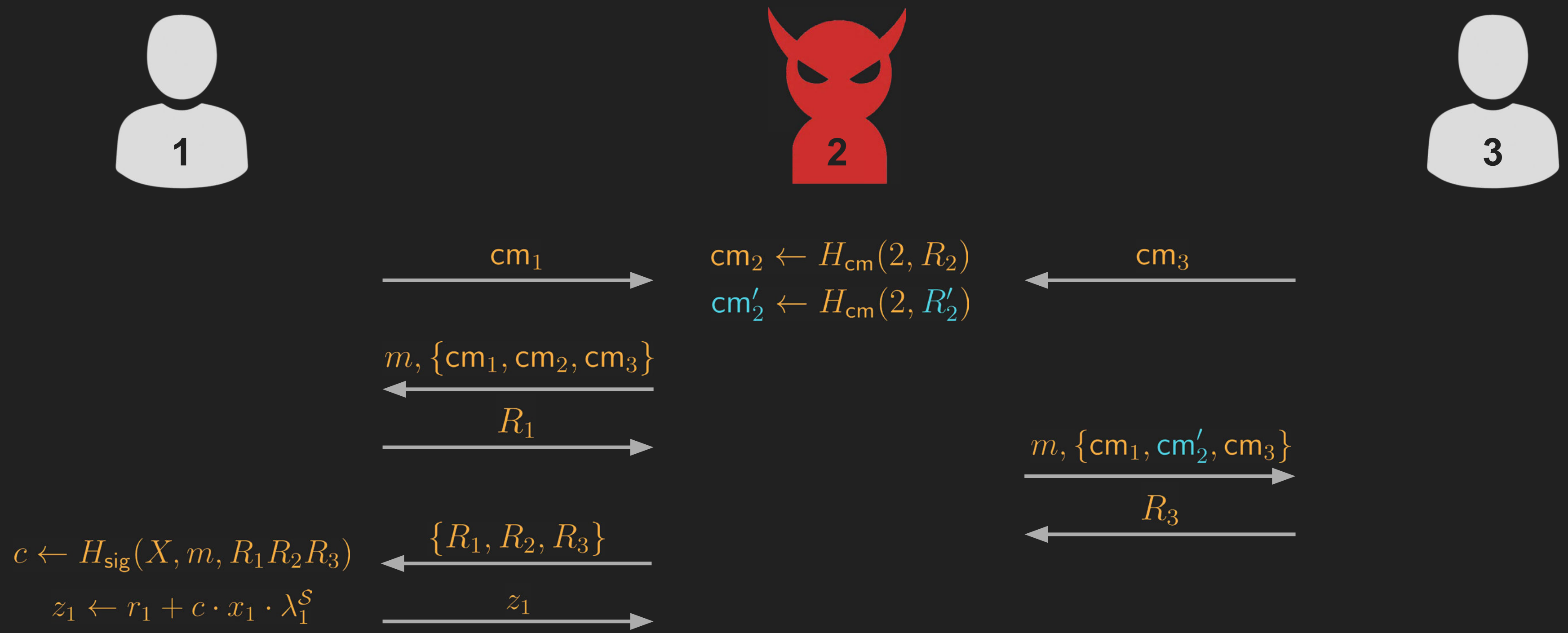
Counterexample: $t = 3, \ S = \{1, 2, 3\}$



Counterexample: $t = 3,\ \mathcal{S} = \{1, 2, 3\}$

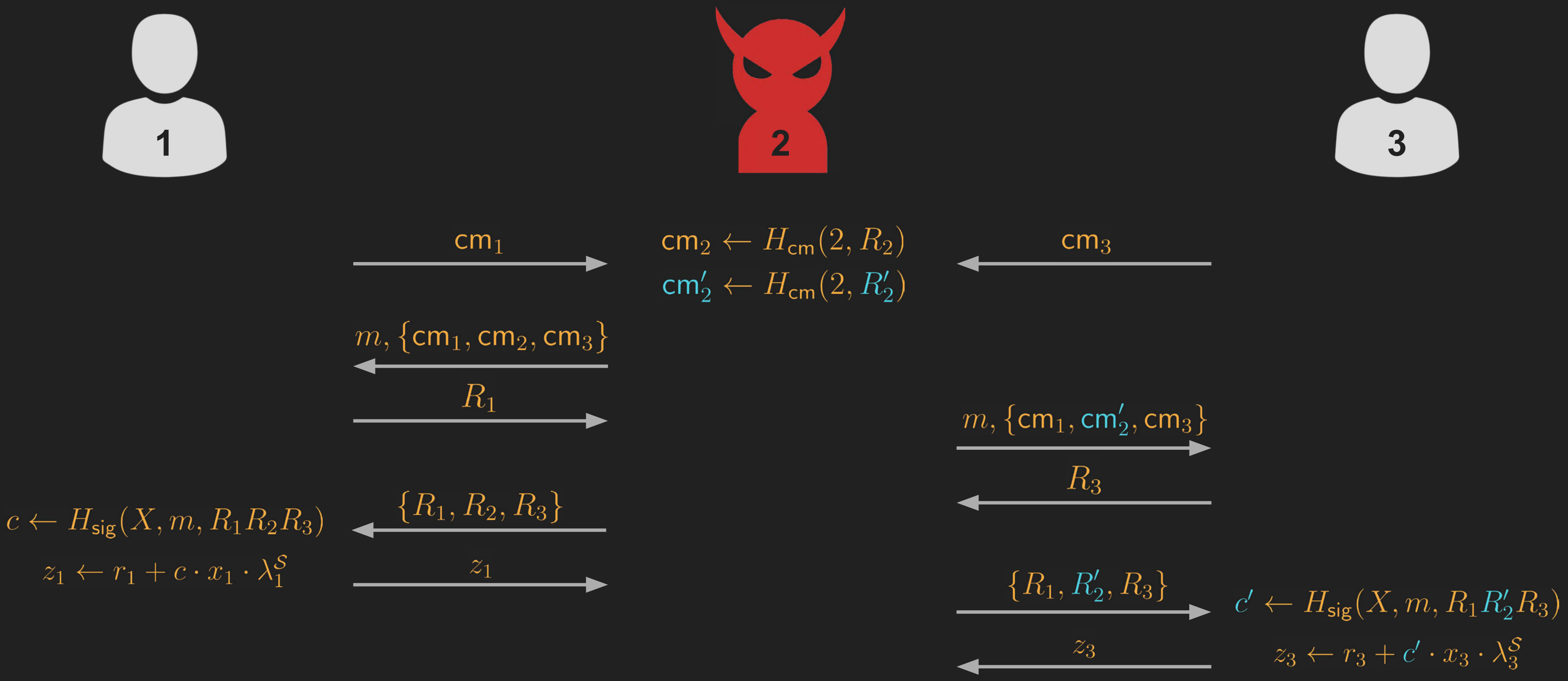


Counterexample: $t = 3,\ \mathcal{S} = \{1, 2, 3\}$

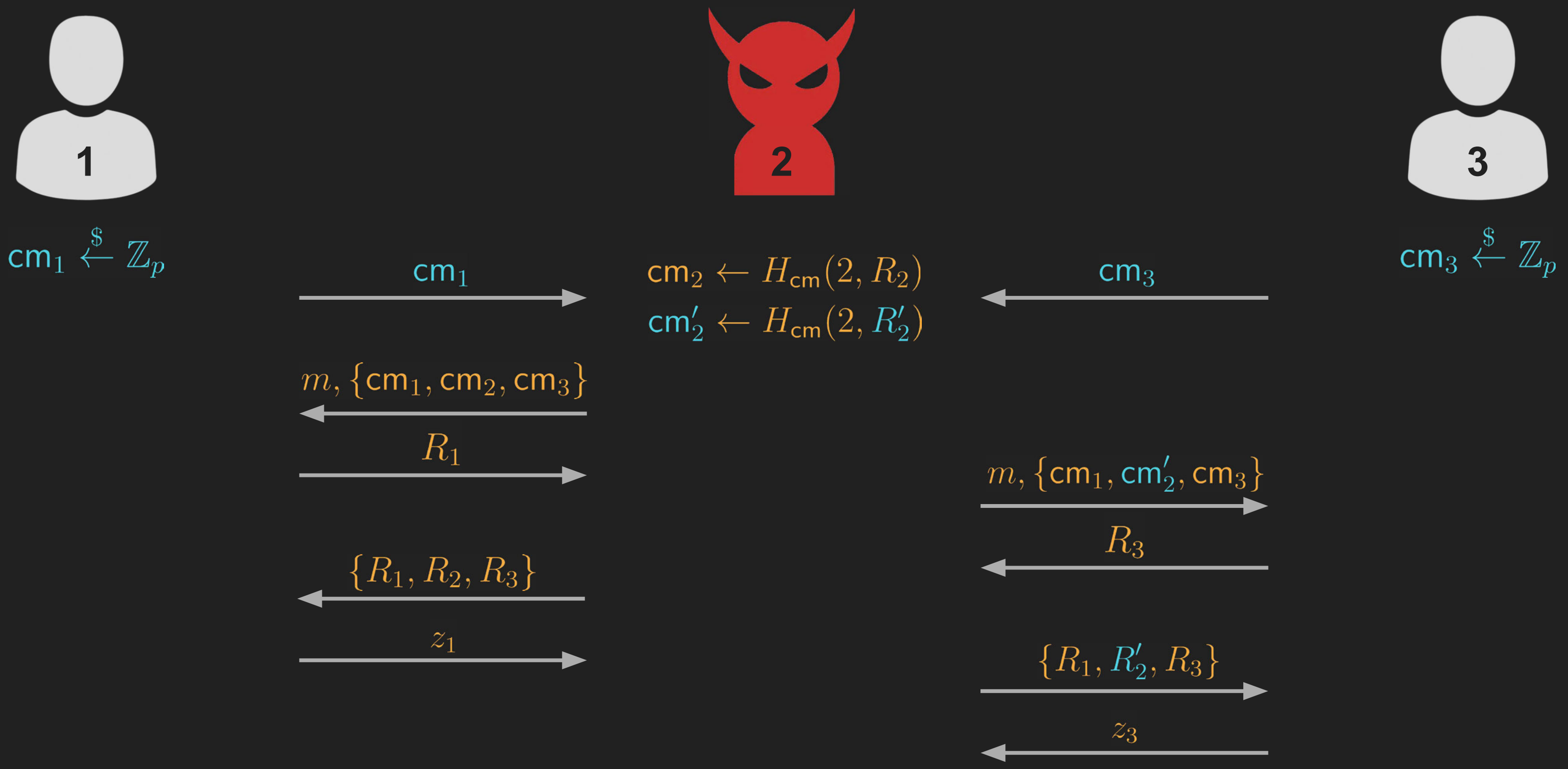


TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

Counterexample: $t = 3,\ \mathcal{S} = \{1, 2, 3\}$



Counterexample: Original Simulation Strategy

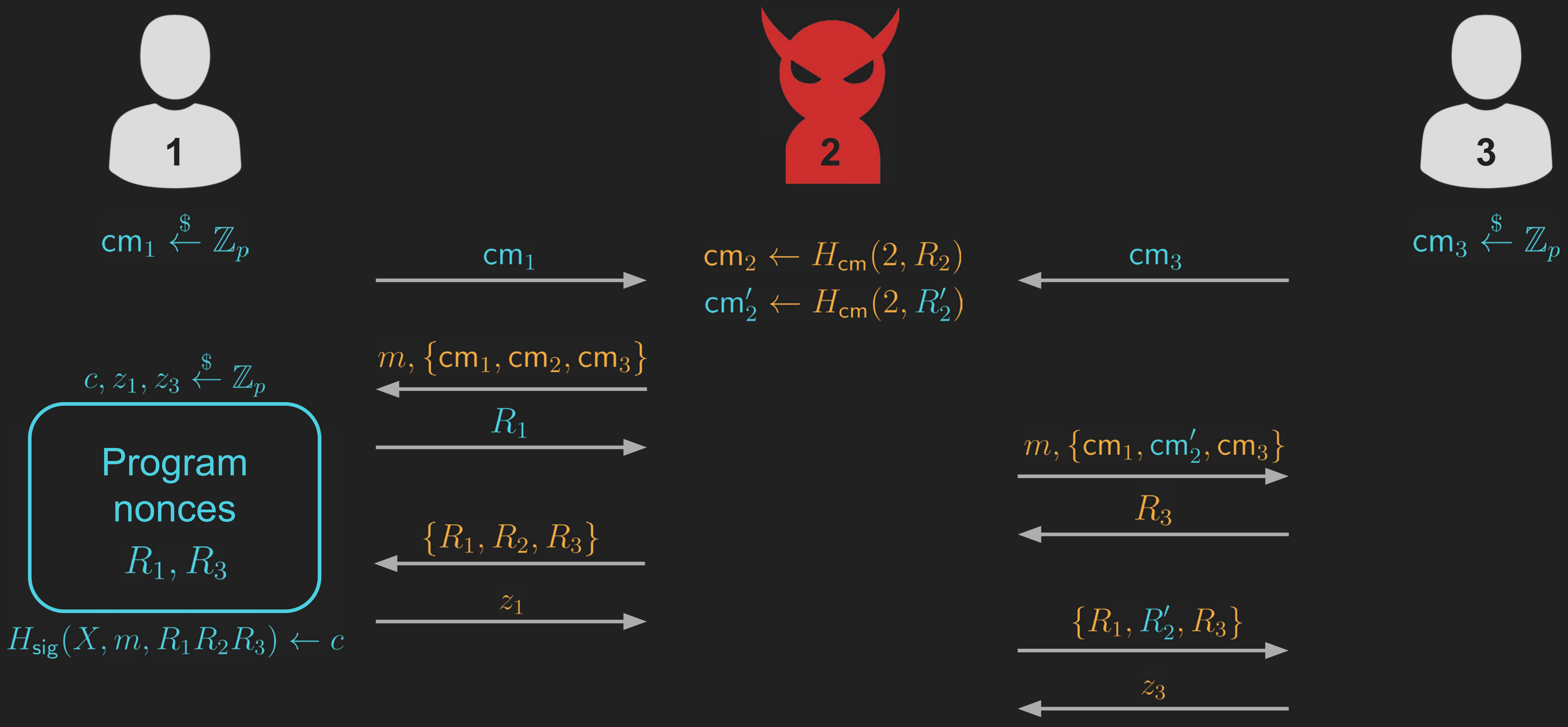


TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

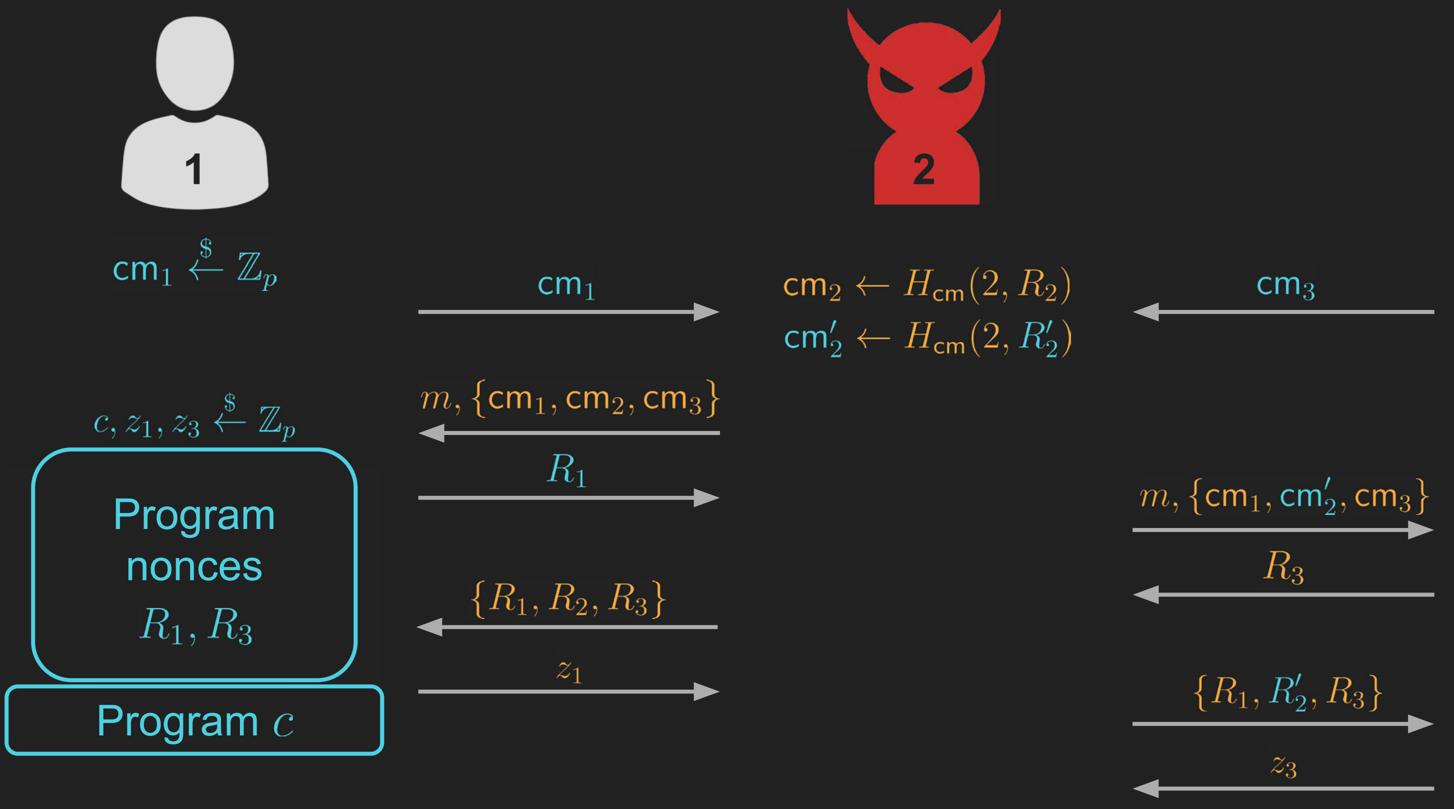
Counterexample: Original Simulation Strategy

1		2		3
$\mathsf{cm}_1 \xleftarrow{\$} \mathbb{Z}_p$	$\mathsf{cm}_1 \rightarrow$	$\mathsf{cm}_2 \leftarrow H_{\mathsf{cm}}(2, R_2)$ $\mathsf{cm}'_2 \leftarrow H_{\mathsf{cm}}(2, R'_2)$	$\leftarrow \mathsf{cm}_3$	$\mathsf{cm}_3 \xleftarrow{\$} \mathbb{Z}_p$
$c, z_1, z_3 \xleftarrow{\$} \mathbb{Z}_p$	$\leftarrow m, \{\mathsf{cm}_1, \mathsf{cm}_2, \mathsf{cm}_3\}$			
$R_1 \leftarrow g^{z_1} \cdot X_1^{-c \cdot \lambda_1^S}$ $H_{\mathsf{cm}}(1, R_1) \leftarrow \mathsf{cm}_1$	$R_1 \rightarrow$		$m, \{\mathsf{cm}_1, \mathsf{cm}'_2, \mathsf{cm}_3\} \rightarrow$	
			$\leftarrow R_3$	
$R_3 \leftarrow g^{z_3} \cdot X_3^{-c \cdot \lambda_3^S}$ $H_{\mathsf{cm}}(3, R_3) \leftarrow \mathsf{cm}_3$	$\leftarrow \{R_1, R_2, R_3\}$			
$H_{\mathsf{sig}}(X, m, R_1 R_2 R_3) \leftarrow c$	$z_1 \rightarrow$		$\{R_1, R'_2, R_3\} \rightarrow$	
			$\leftarrow z_3$	

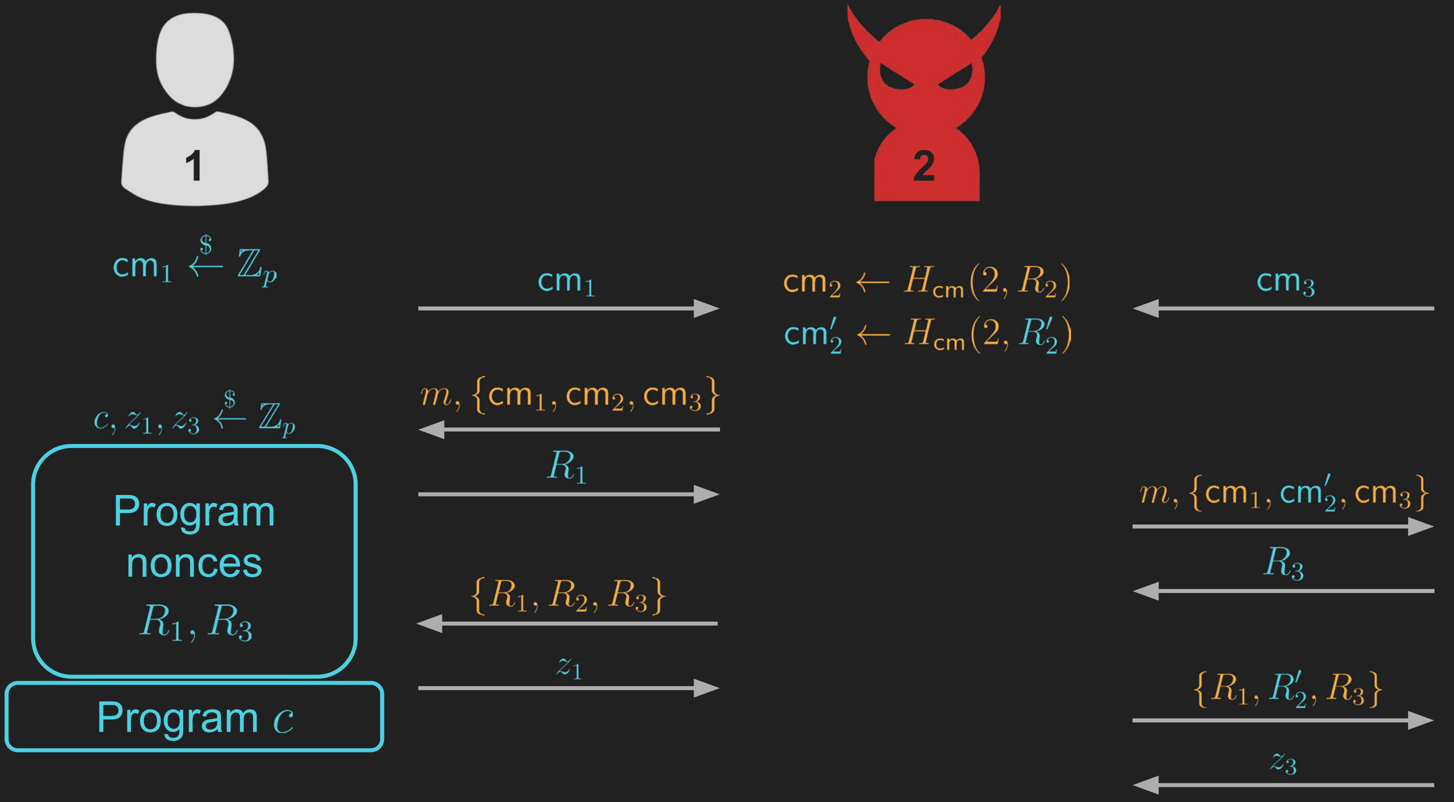
Counterexample: Original Simulation Strategy



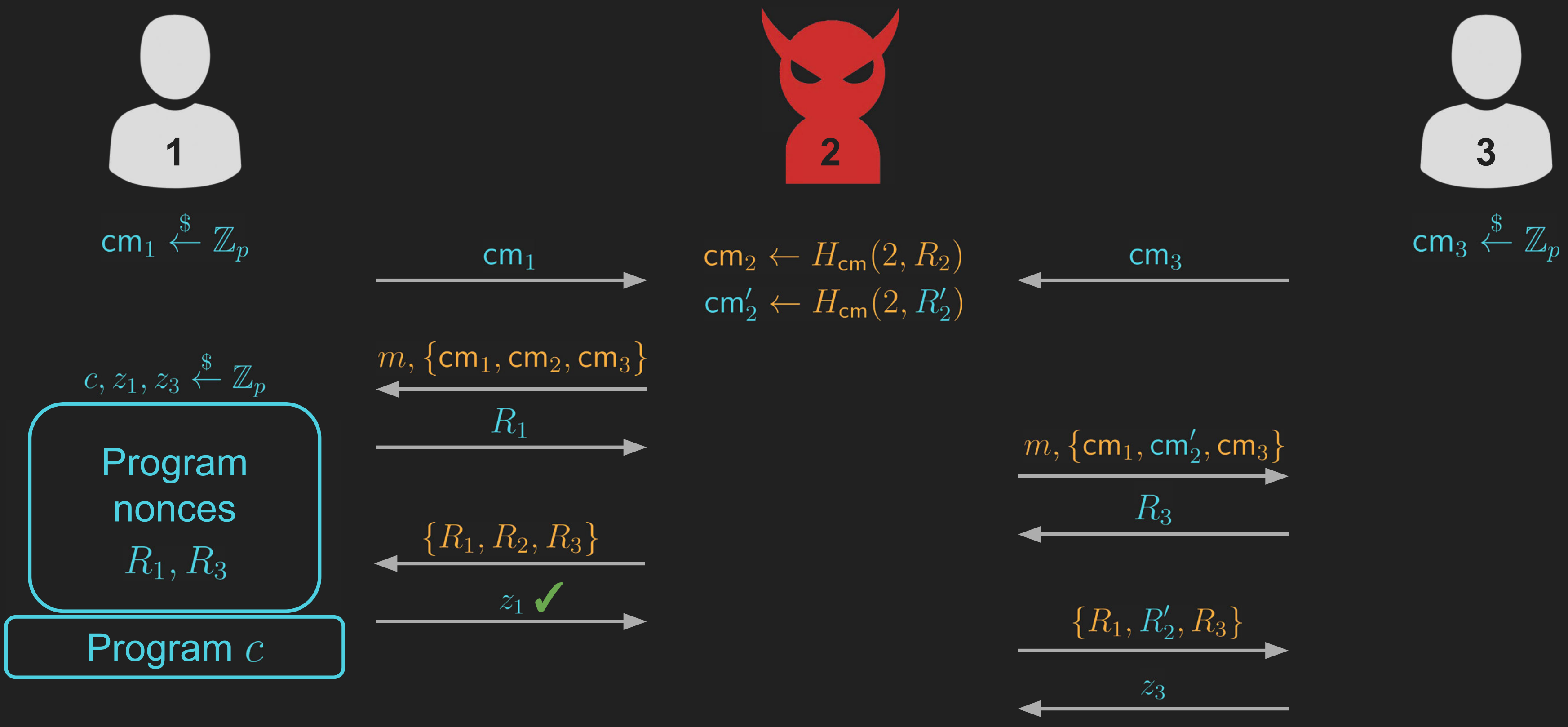
Counterexample: Original Simulation Strategy



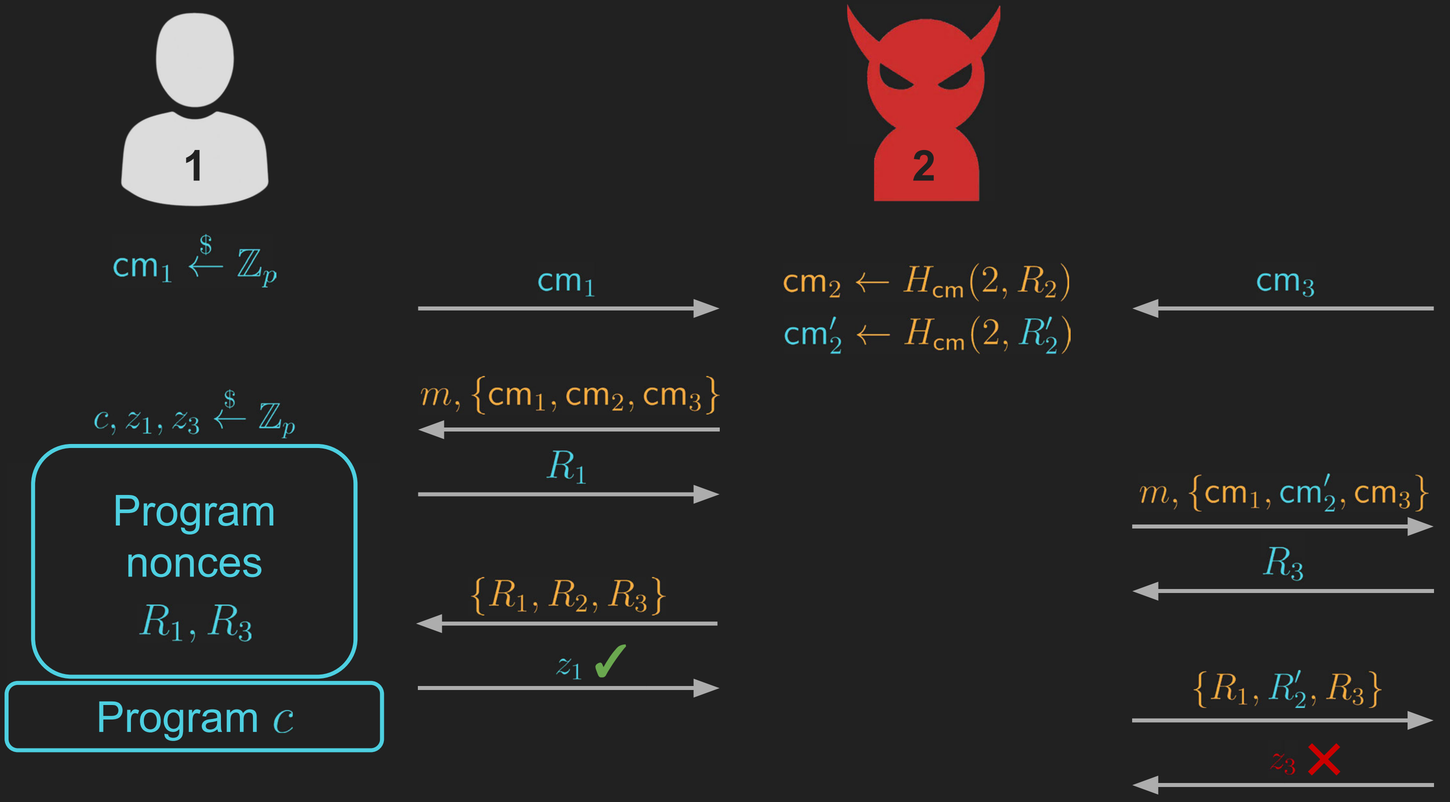
Counterexample: Original Simulation Strategy



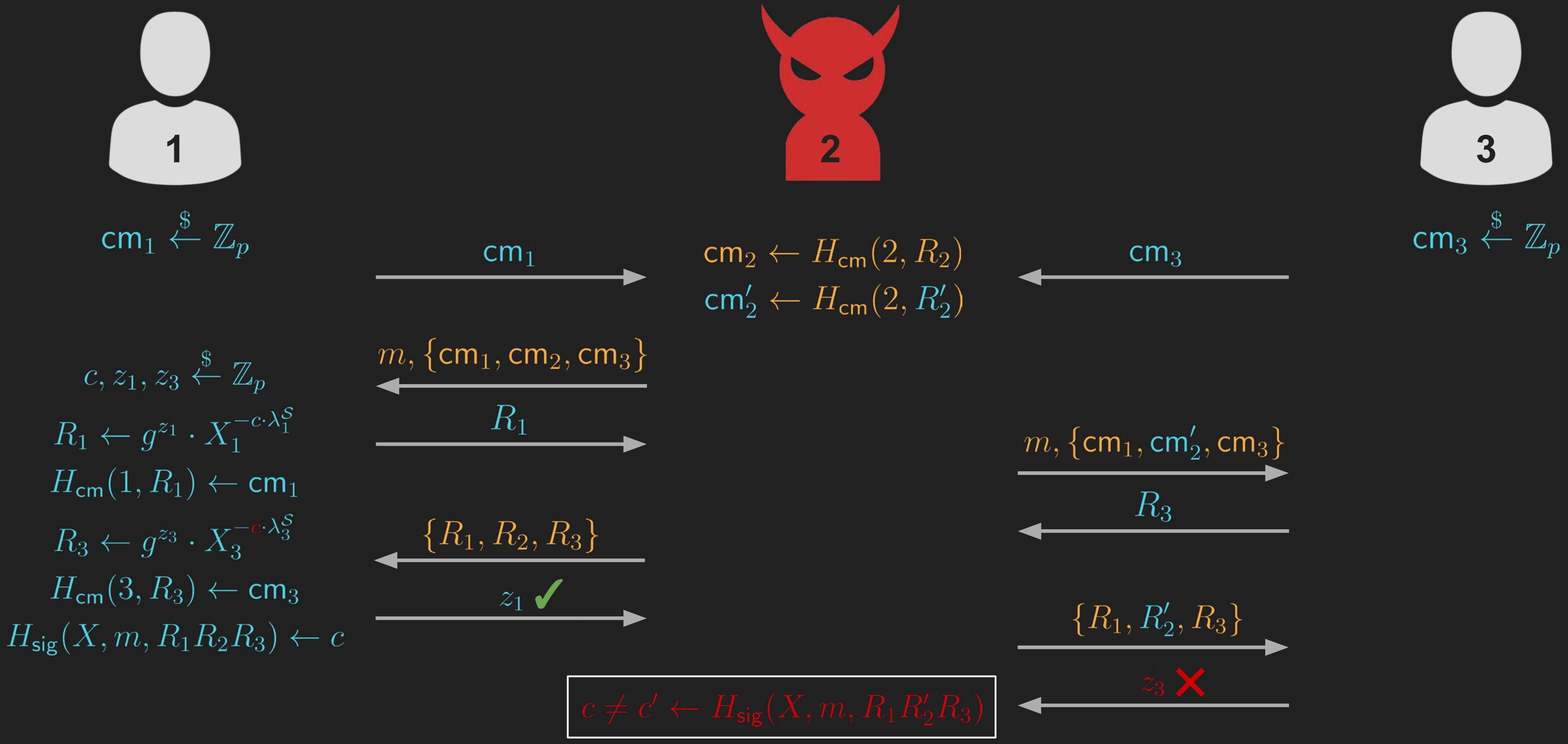
Counterexample: Original Simulation Strategy



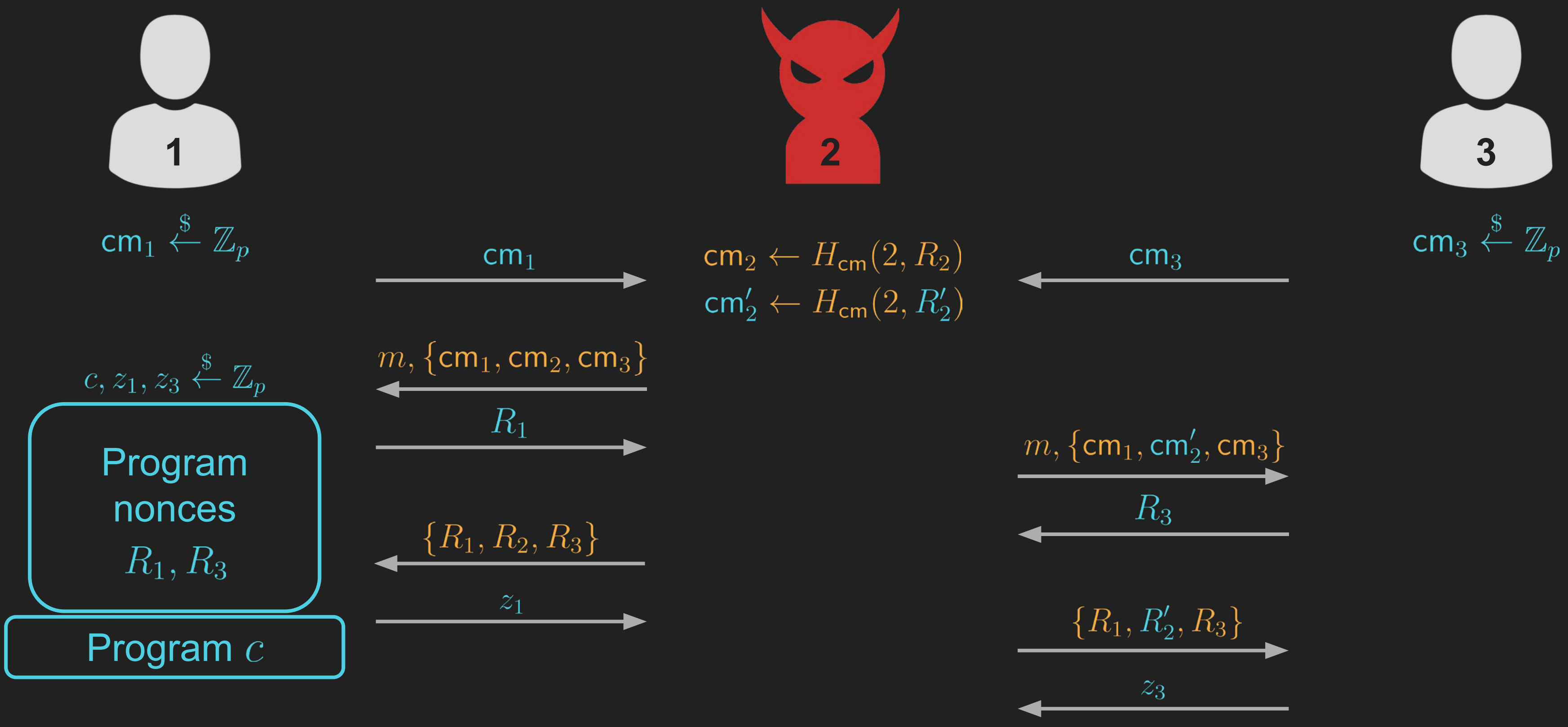
Counterexample: Original Simulation Strategy



Counterexample: Original Simulation Strategy

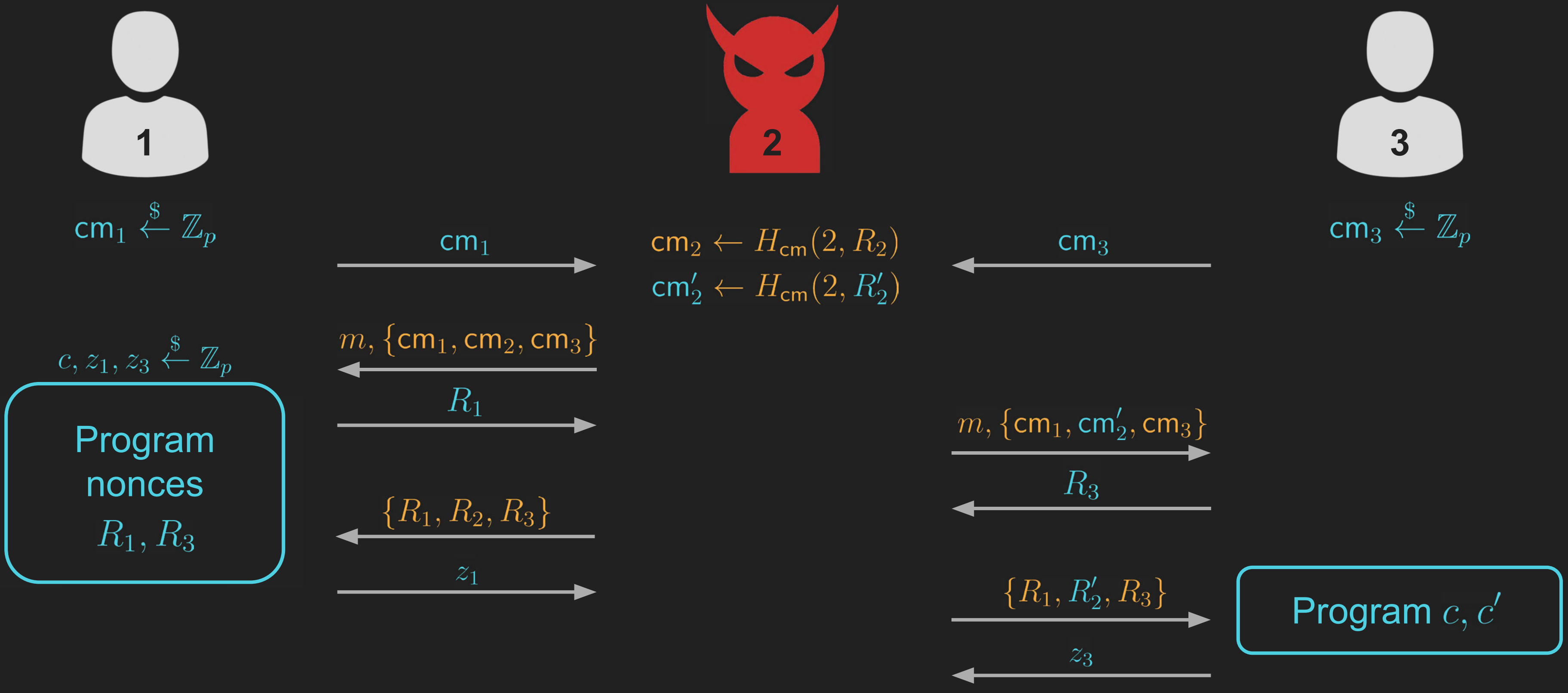


Counterexample: The New Simulation Strategy

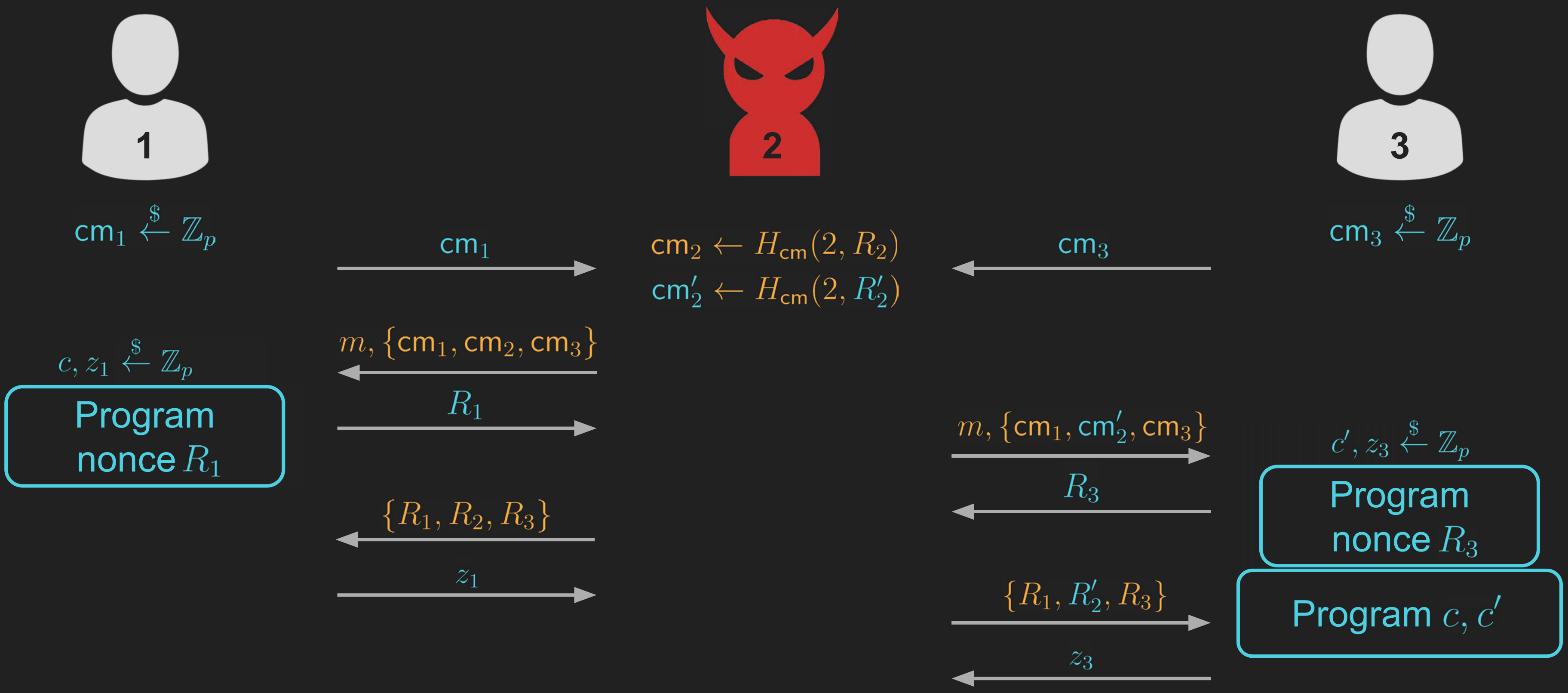


TU WIEN CYSEC CYBERSECURITYCENTER S&P Security & Privacy

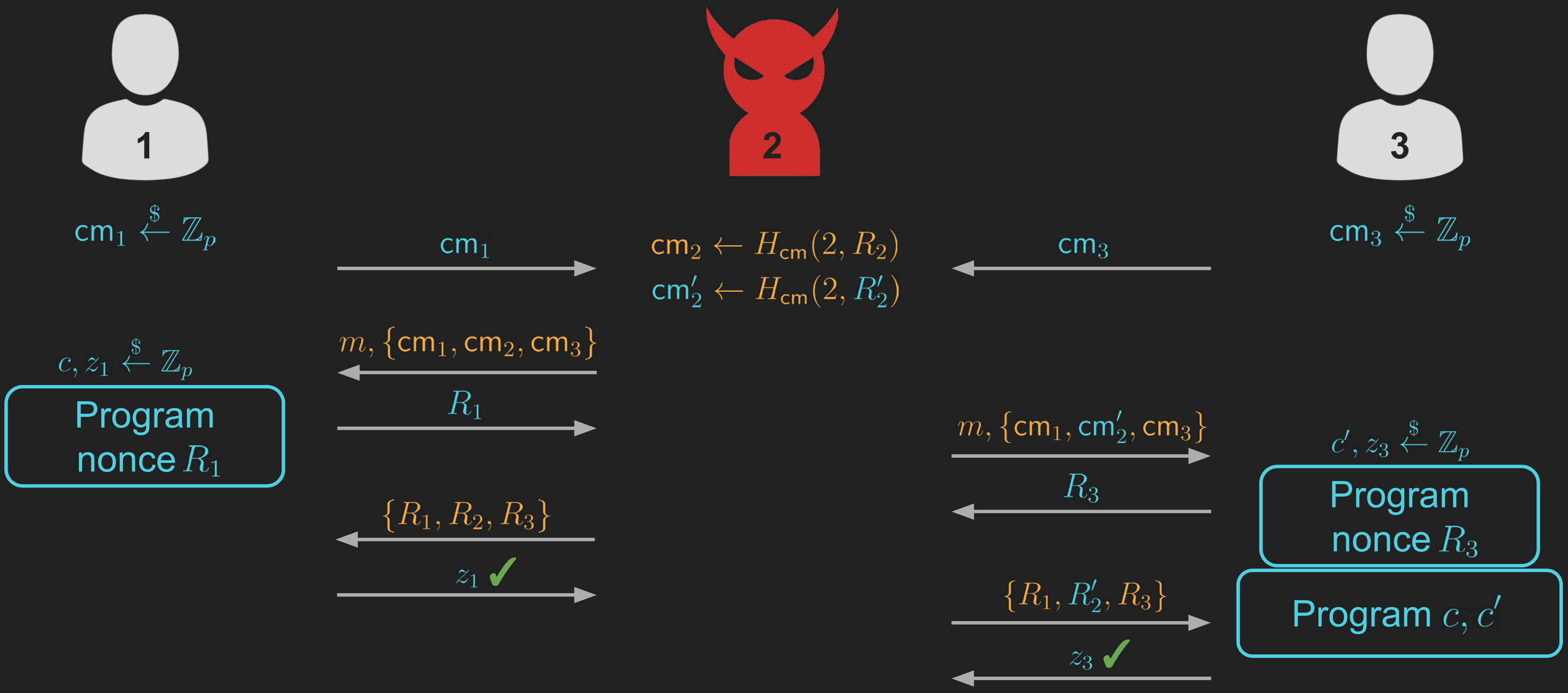
Counterexample: The New Simulation Strategy



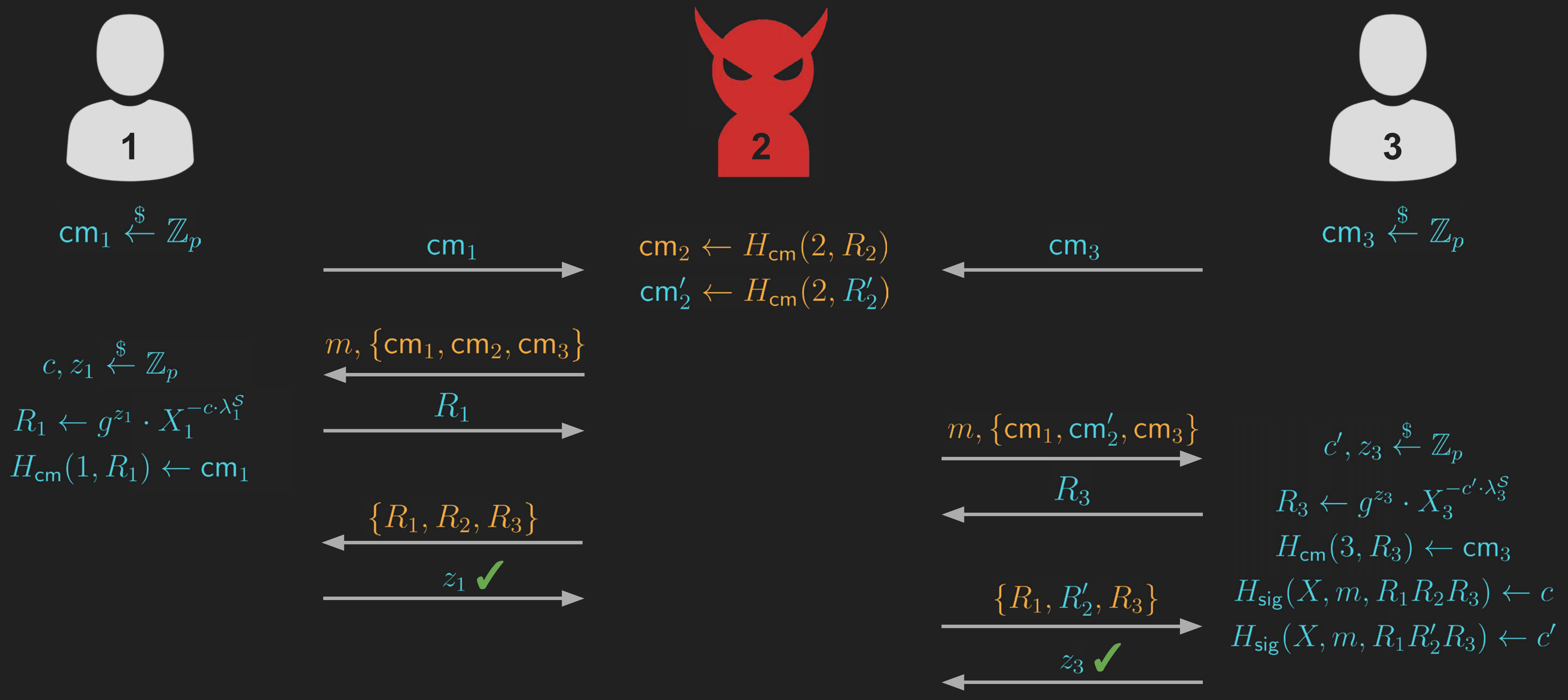
Counterexample: The New Simulation Strategy



Counterexample: The New Simulation Strategy



Counterexample: The New Simulation Strategy



Problem 2:
Avoiding Rewinding

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem
  - Any such proof must rely on hardness of LDVR

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem
  - Any such proof must rely on hardness of LDVR
- Our starting point – Circular Discrete-Logarithm (CDL) assumption:

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem
  - Any such proof must rely on hardness of LDVR
- Our starting point – Circular Discrete-Logarithm (CDL) assumption:
  - Variant of DL we introduced to give first tight proof of basic Schnorr in ROM [CFOS25]

Achieving Full Adaptive Security

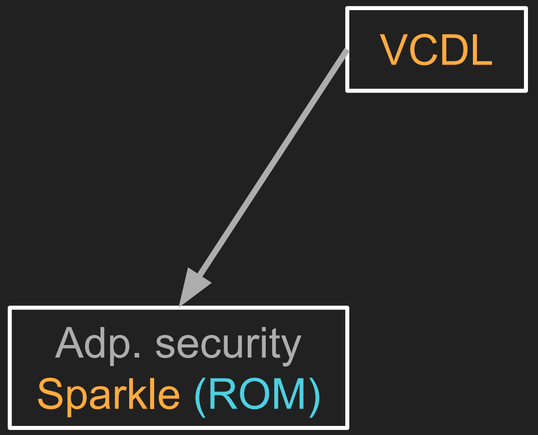
- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem
  - Any such proof must rely on hardness of LDVR
- Our starting point – Circular Discrete-Logarithm (CDL) assumption:
  - Variant of DL we introduced to give first tight proof of basic Schnorr in ROM [CFOS25]
  - Crucially, CDL avoids rewinding (and AGM)

Achieving Full Adaptive Security

- Our new simulation technique immediately yields static security of Sparkle
- Sparkle+ originally claimed fully adaptively secure [CKM23]
- However, Crites et al. [CS25, CKK+25] invalidate the proof:
  - Introduce LDVR (low-dimensional vector representation) problem
  - Any such proof must rely on hardness of LDVR
- Our starting point – Circular Discrete-Logarithm (CDL) assumption:
  - Variant of DL we introduced to give first tight proof of basic Schnorr in ROM [CFOS25]
  - Crucially, CDL avoids rewinding (and AGM)
- We strengthen CDL to interactive variant: “Vandermonde” CDL (VCDL)

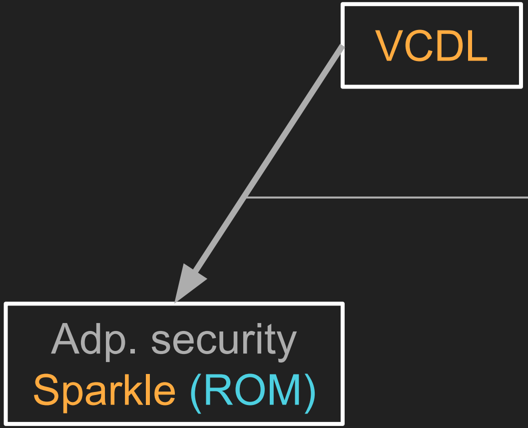
Main Results

- Tight proof of full adaptive security of Sparkle under VCDL in the ROM



Main Results

- Tight proof of full adaptive security of Sparkle under VCDL in the ROM

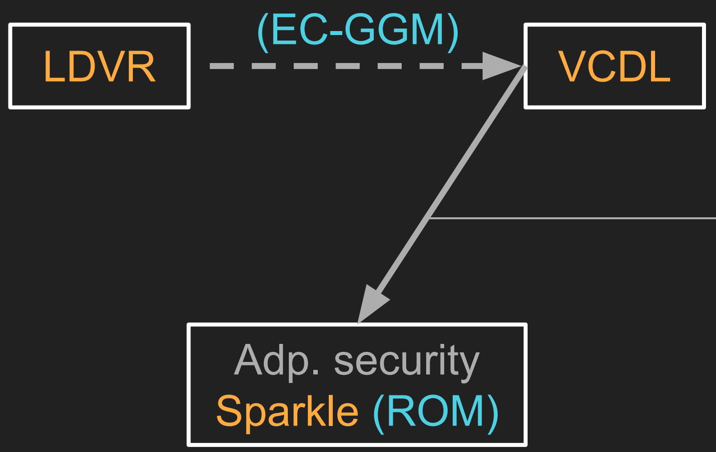


Theorem (in ROM):

$$\mathsf{Adv}^{\text{adp-ts-uf}}_{\mathsf{Sparkle}} \leq \mathsf{Adv}^{\text{vcdl}} + \mathsf{negl}$$

Main Results

- Tight proof of full adaptive security of Sparkle under VCDL in the ROM
- Justify VCDL: proof from LDVR in the elliptic-curve GGM [GS22]



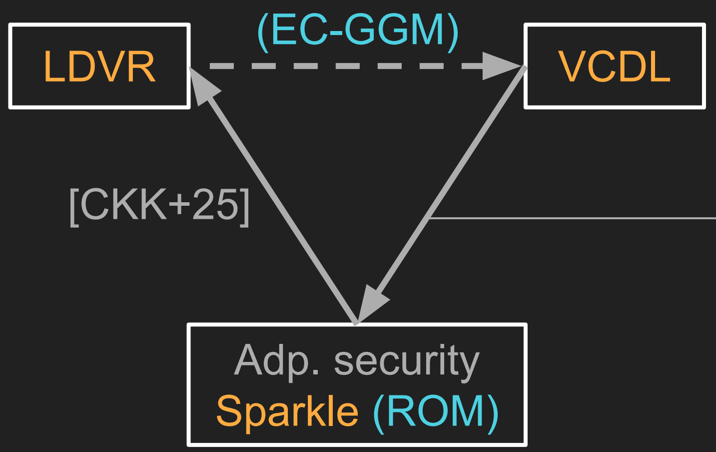
Theorem (in ROM):

$$\mathsf{Adv}^{\text{adp-ts-uf}}_{\mathsf{Sparkle}} \leq \mathsf{Adv}^{\text{vcdl}} + \mathsf{negl}$$

CYBERSECURITYCENTER

Main Results

- Tight proof of full adaptive security of Sparkle under VCDL in the ROM
- Justify VCDL: proof from LDVR in the elliptic-curve GGM [GS22]
  (which is necessary)



Theorem (in ROM):

$$\mathsf{Adv}^{\text{adp-ts-uf}}_{\mathsf{Sparkle}} \leq \mathsf{Adv}^{\text{vcdl}} + \mathsf{negl}$$

More in the Full Paper…

“Revisiting the Security of Sparkle”

on ePrint soon…

Thanks!

Questions?

References

[BLT+24] Renas Bacho, Julian Loss, Stefano Tessaro, Benedikt Wagner, and Chenzhi Zhu. Twinkle: Threshold Signatures from DDH with Full Adaptive Security. In Advances in Cryptology – EUROCRYPT 2024, volume 14651 of LNCS, pages 429–459. Springer, 2024. https://doi.org/10.1007/978-3-031-58716-0_15.

[CFOS25] Gavin Cho, Georg Fuchsbauer, Adam O'Neill, and Marek Sefranek. Schnorr Signatures are Tightly Secure in the ROM Under a Non-interactive Assumption. In Advances in Cryptology – CRYPTO 2025, volume 16005 of LNCS, pages 223–255. Springer, 2025. https://doi.org/10.1007/978-3-032-01887-8_8.

[CKK+25] Elizabeth Crites, Jonathan Katz, Chelsea Komlo, Stefano Tessaro, and Chenzhi Zhu. On the Adaptive Security of FROST. In Advances in Cryptology – CRYPTO 2025, volume 16005 of LNCS, pages 480–511. Springer, 2025. https://doi.org/10.1007/978-3-032-01887-8_16.

[CKM23] Elizabeth Crites, Chelsea Komlo, and Mary Maller. Fully Adaptive Schnorr Threshold Signatures. In Advances in Cryptology – CRYPTO 2023, volume 14081 of LNCS, pages 678–709. Springer, 2023. https://doi.org/10.1007/978-3-031-38557-5_22.

[CS25] Elizabeth Crites and Alistair Stewart. A Plausible Attack on the Adaptive Security of Threshold Schnorr Signatures. In Advances in Cryptology – CRYPTO 2025, volume 16005 of LNCS, pages 457–479. Springer, 2025. https://doi.org/10.1007/978-3-032-01887-8_15.

[GS22] Jens Groth and Victor Shoup. On the Security of ECDSA with Additive Key Derivation and Presignatures. In Advances in Cryptology – EUROCRYPT 2022, volume 13275 of LNCS, pages 365–396. Springer, 2022. https://doi.org/10.1007/978-3-031-06944-4_13.

[PS96] David Pointcheval and Jacques Stern. Security Proofs for Signature Schemes. In Advances in Cryptology – EUROCRYPT 1996, volume 1070 of LNCS, pages 387–398. Springer, 1996. https://doi.org/10.1007/3-540-68339-9_33.