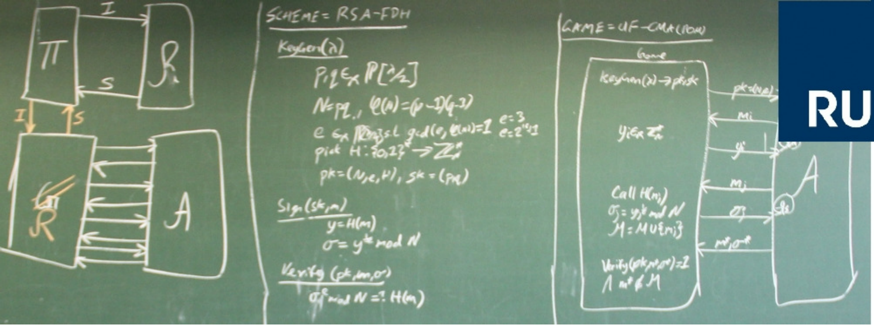


RUB

RUHR-UNIVERSITÄT BOCHUM

Tightly-Secure Signatures from Chameleon Hash Functions

NIST, **Maryland**, PKC 2015

Olivier Blazy[1], Saqib A. Kakvi[2], Eike Kiltz[2],
<u>**Jiaxin Pan**</u>[2]
[1]University of Limoges, France
[2]Ruhr University Bochum, Germany

hgi
Horst Görtz Institut
für IT-Sicherheit

Keywords

1. Signatures
2. Tight Security
3. Chameleon Hash

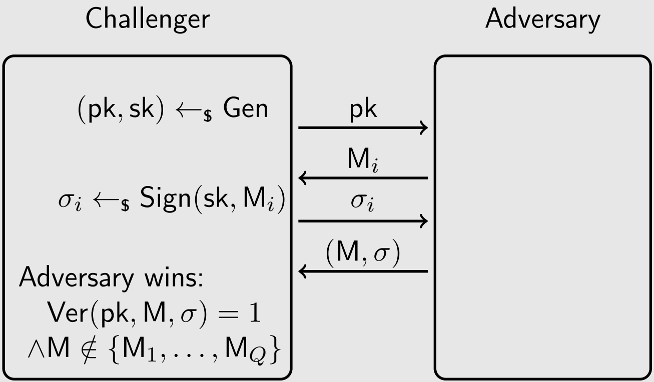
Signature

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Gen}$
- $\sigma \leftarrow_{\$} \mathsf{Sign}(\mathsf{sk}, \mathsf{M})$
- $0/1 \leftarrow \mathsf{Ver}(\mathsf{pk}, \mathsf{M}, \sigma)$

Correctness:

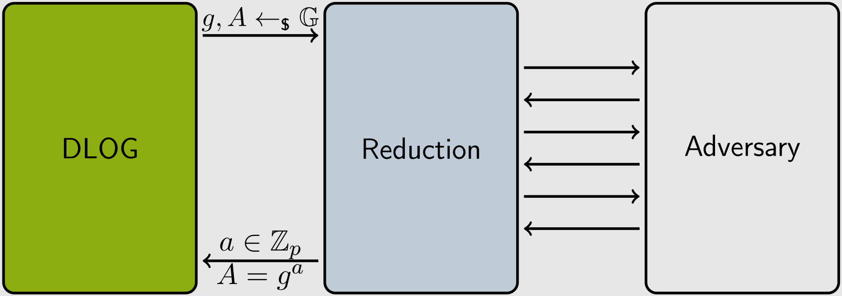
$$\forall (\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Gen},\ \ \mathsf{Ver}(\mathsf{pk}, \mathsf{M}, \mathsf{Sign}(\mathsf{sk}, \mathsf{M})) = 1$$

UF-CMA Security

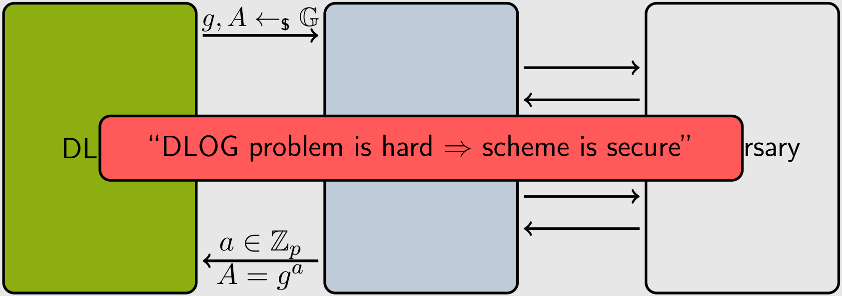


Q is the number of signing queries.

Provable Security



Provable Security



- Let k be the security parameter,

  $\mathsf{Adv}[\mathsf{Sig}] < f(k) \cdot \mathsf{Adv}[\mathsf{DLOG}]$

Tight Security

$\mathsf{Adv}[\mathsf{Sig}] < f(k) \cdot \mathsf{Adv}[\mathsf{DLOG}]$

- "Tight" if
  $$f(k) = O(1)$$
- "Loose" if
  $$f(k) = O(Q)$$

Why “tight”?

- In practice:
  - We want efficient schemes!
  - Smaller security parameters!

For example

- We want 80-bit security and $Q = 2^{40}$

Tight scheme

- $\mathsf{Adv}[\mathsf{Sig}] < \mathsf{Adv}[\mathsf{DLOG}] < 2^{-80}$

  $\Longrightarrow$ We need DLOG problem with 80-bit security

  $\Longrightarrow |p| = \boxed{160}$ (by the best DLOG attack)

Loose Scheme

- $\mathsf{Adv}[\mathsf{Sig}] < 2^{40} \cdot \mathsf{Adv}[\mathsf{DLOG}] < 2^{-80}$

  $\Longrightarrow \mathsf{Adv}[\mathsf{DLOG}] < 2^{-120}$

  $\Longrightarrow$ We need DLOG problem with 120-bit security

  $\Longrightarrow |p| = \boxed{240}$ (by the best DLOG attack)

Signatures in the Standard Model

- Loose Reduction
  - e.g. Waters '05
- Non-standard/"Q-Type" Assumptions
  - e.g. Boneh-Boyen '04
- Exceptions: ...

Tight Signatures from Standard Assumptions

- CRYPTO '96 Cramer-Damgård: RSA
- PKC '05 Catalano-Gennaro: Factoring
- CRYPTO '12 Hofheinz-Jager: DLIN

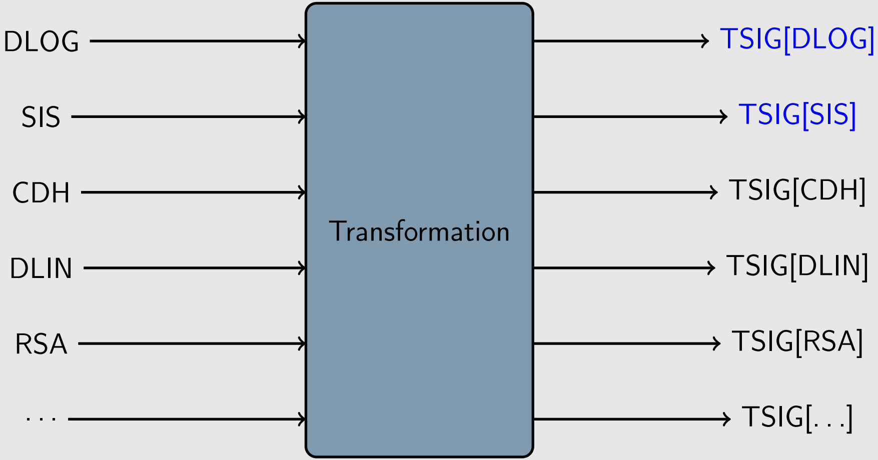
Tight Signatures from Standard Assumptions

- CRYPTO '96 Cramer-Damgård: RSA
- PKC '05 Catalano-Gennaro: Factoring
- CRYPTO '12 Hofheinz-Jager: DLIN

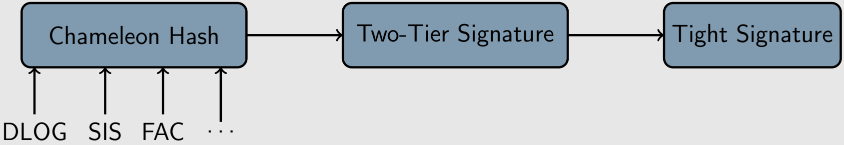
Question

Generic constructions for tight signatures?

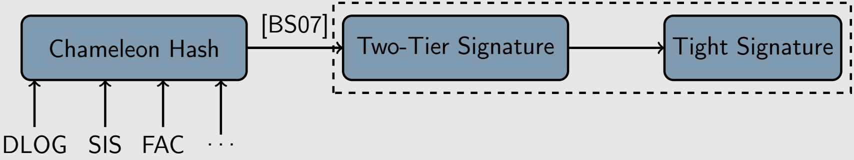
Our Contribution



Our Contribution



Our Contribution



Two-Tier Signature

- Proposed by Bellare and Shoup at PKC '07

Two-Tier Signature

Signature

- $(\mathsf{pk}, \mathsf{sk}) \leftarrow_{\$} \mathsf{Gen}$
- $\sigma \leftarrow_{\$} \mathsf{Sign}(\mathsf{sk}, \mathsf{M})$
- $0/1 \leftarrow \mathsf{Ver}(\mathsf{pk}, \mathsf{M}, \sigma)$

Two-Tier Signature

- $(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PrimaryGen}$
- $(\mathsf{spk}, \mathsf{ssk}) \leftarrow_{\$} \mathsf{SecondaryGen}$
- $\sigma \leftarrow_{\$} \mathsf{TTSign}(\mathsf{sk}, \mathsf{ssk}, \mathsf{M})$
- $0/1 \leftarrow \mathsf{TTVer}(\mathsf{pk}, \mathsf{spk}, \mathsf{M}, \sigma)$

Security of two-tier signature

Challenger | **Adversary**

$(\mathsf{ppk}, \mathsf{psk}) \leftarrow_\$ \mathsf{PrimaryGen}$

Challenger → Adversary: pk

Adversary → Challenger: M_i

$(\mathsf{spk}_i, \mathsf{ssk}_i) \leftarrow_\$ \mathsf{SecondaryGen}$

$\sigma_i \leftarrow_\$ \mathsf{TTSign}(\mathsf{sk}, \mathsf{ssk}_i, \mathsf{M}_i)$

Challenger → Adversary: $(\sigma_i, \mathsf{spk}_i)$

Adversary → Challenger: $(\mathsf{M}, \sigma, \mathsf{spk})$

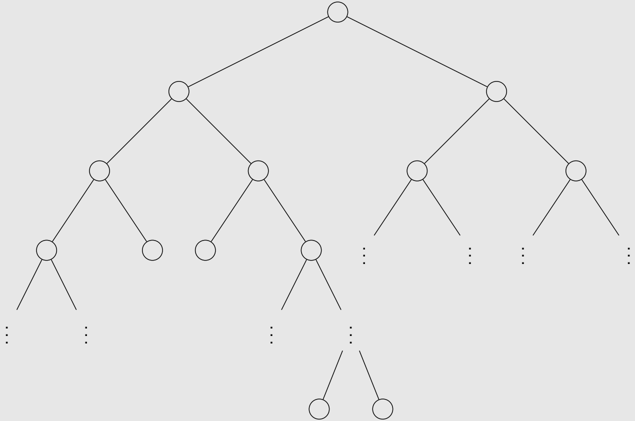
Adversary wins:

$\mathsf{TTVer}(\mathsf{ppk}, \mathsf{spk}, \mathsf{M}, \sigma) = 1$

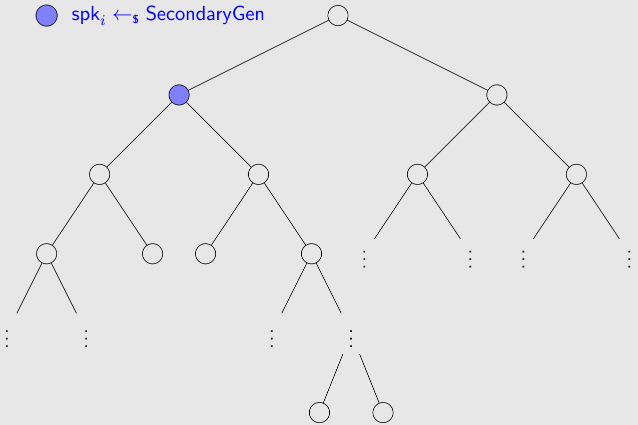
$\wedge\ \mathsf{M} \notin \{\mathsf{M}_1, \ldots, \mathsf{M}_Q\}$

$\wedge\ \mathsf{spk} = \mathsf{spk}_i$ for some i

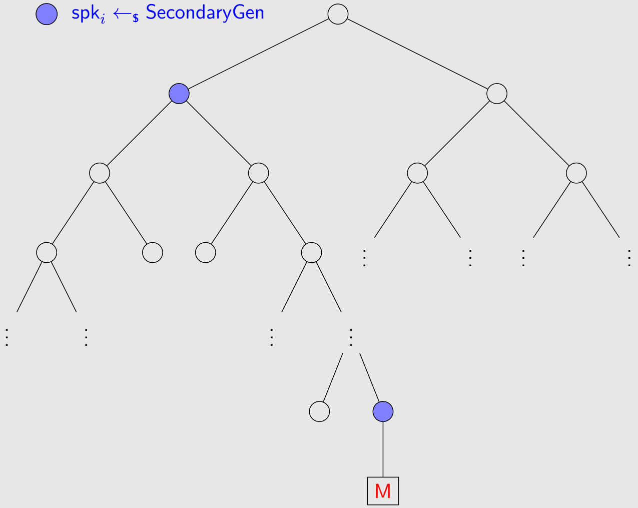
Two-Tier Signature → Standard Signature



Two-Tier Signature → Standard Signature



Two-Tier Signature → Standard Signature

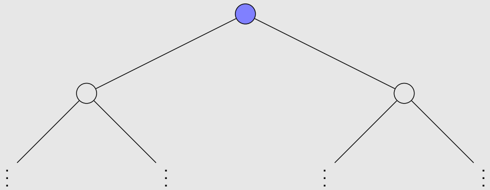


Gen of Tree Signature

- $(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PrimaryGen}$

Gen of Tree Signature

- $(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PrimaryGen}$
- $(\mathsf{spk}_{root}, \mathsf{ssk}_{root}) \leftarrow_{\$} \mathsf{SecondaryGen}$



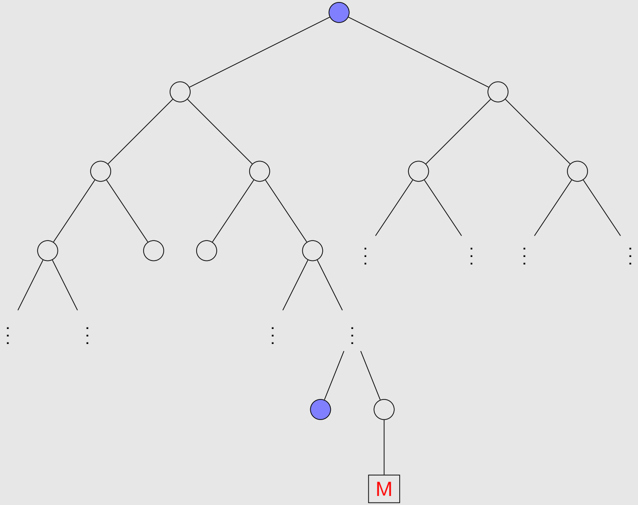
Gen of Tree Signature

- $(\mathsf{ppk}, \mathsf{psk}) \leftarrow_{\$} \mathsf{PrimaryGen}$
- $(\mathsf{spk}_{root}, \mathsf{ssk}_{root}) \leftarrow_{\$} \mathsf{SecondaryGen}$
- $\mathsf{PK} = (\mathsf{ppk}, \mathsf{spk}_{root}), \mathsf{sk} = (\mathsf{psk}, \mathsf{ssk}_{root})$

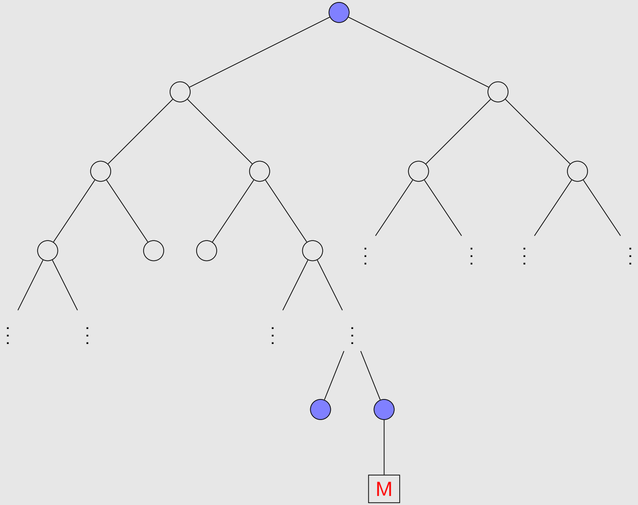
Sign(sk,M)

- Step 1: Nodes Generation
- Step 2: Path Authentication

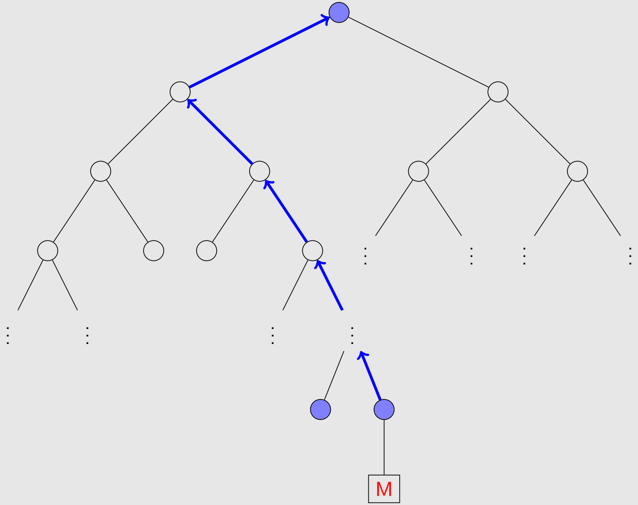
Step 1: Node Generation



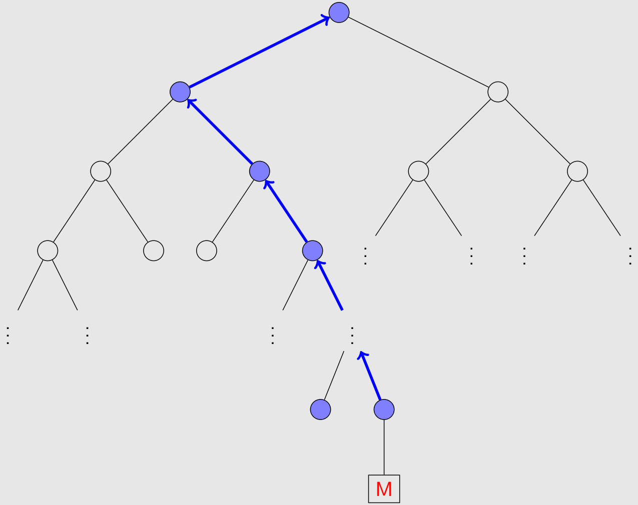
Step 1: Node Generation



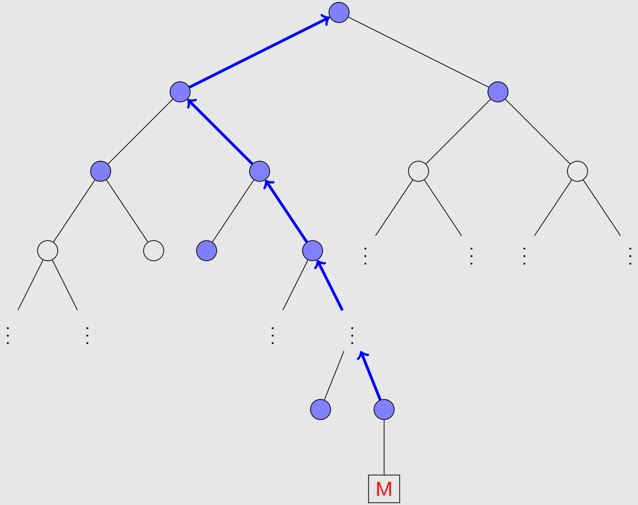
Step 1: Node Generation



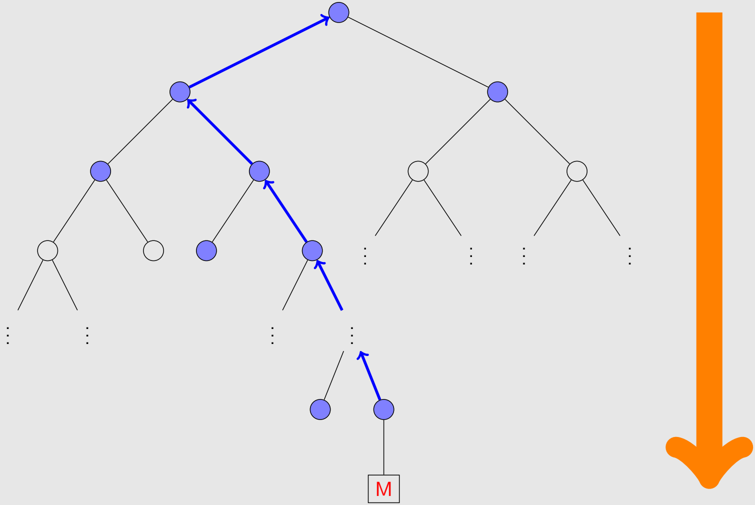
Step 1: Node Generation



Step 1: Node Generation

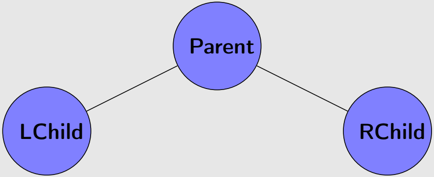


Step 2: Path Authentication



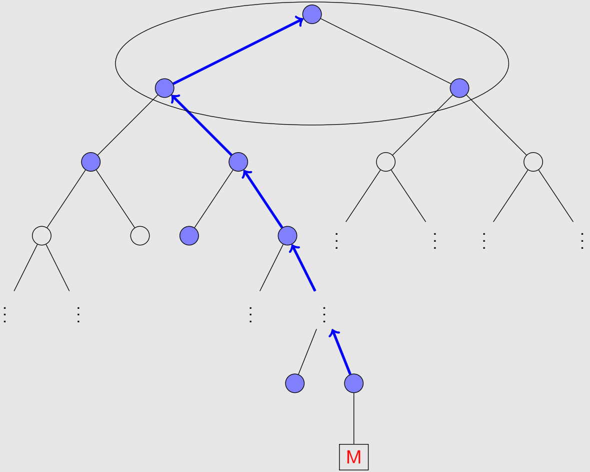
Step 2: Path Authentication

- $\sigma = \mathsf{TTSign}(\mathsf{psk}, \mathsf{ssk}_{\mathsf{parent}}, (\mathsf{LChild}||\mathsf{RChild}))$



Step 2: Path Authentication

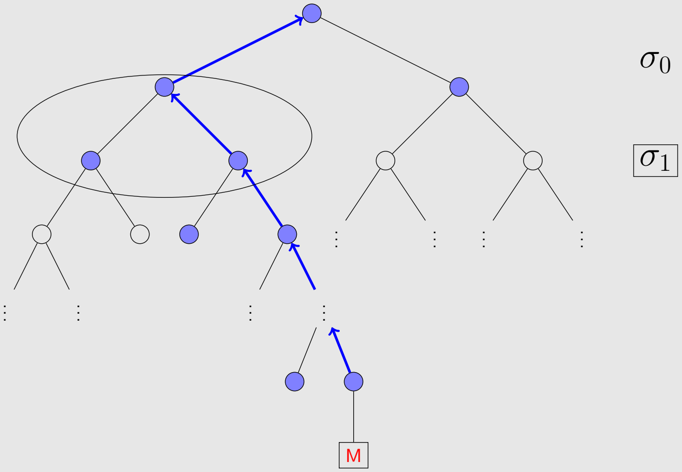
Use Two-Tier Sig to authenticate the path



σ_0

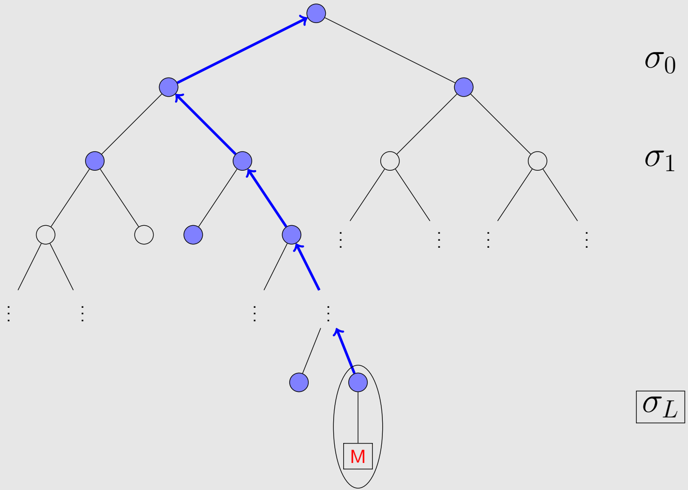
Step 2: Path Authentication

Use Two-Tier Sig to authenticate the path



Step 2: Path Authentication

Use Two-Tier Sig to authenticate the path



Signatures

- Define signature := (path, $\sigma_1, \ldots, \sigma_L$)
- Verify:
  - Check if $(\sigma_1, \ldots, \sigma_L)$ are valid two-tier signatures on path

Security

Theorem 1

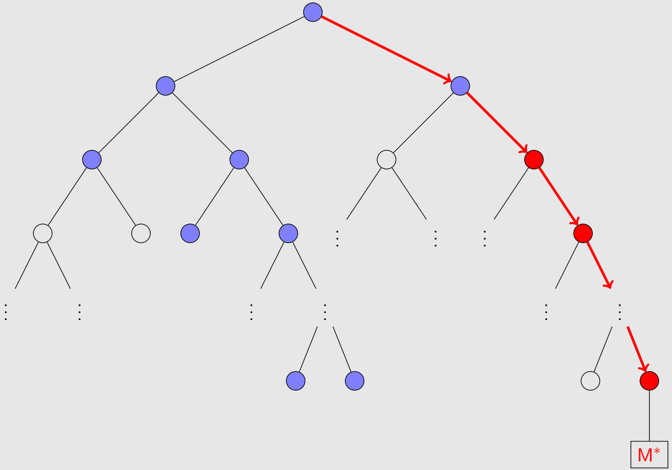
Our construction is tightly secure, if the underlying two-tier signature is tightly-secure. Particularly,

- *Adv[TreeSig] = Adv[Two-TierSig]*

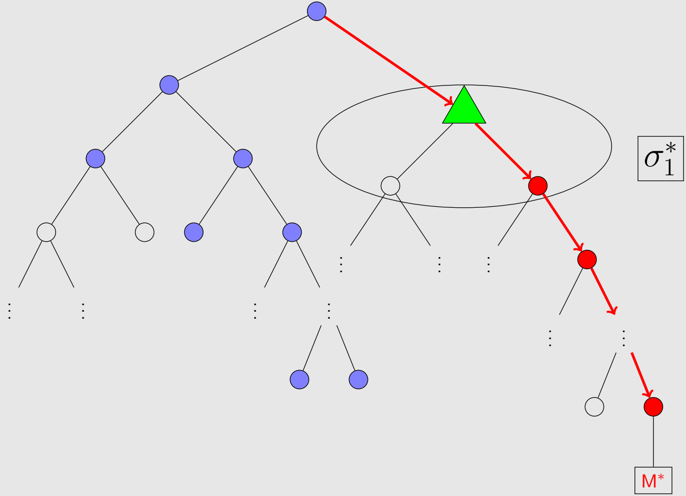
Proof Idea

- Simulate the signature without sk:
  - Use two-tier signing oracle
- Tightly extract the two-tier forgery:
  - Observation:
    - Forgery path differs from signing paths
  - "Splitting" node: the valid two-tier forgery

“Splitting” Node



“Splitting” Node



Differences to Merkle trees

- Our tree node only contains "half" of the PK
  - Merkle: the whole PK
- We have a tight reduction
  - Merkle: loose reduction, guessing

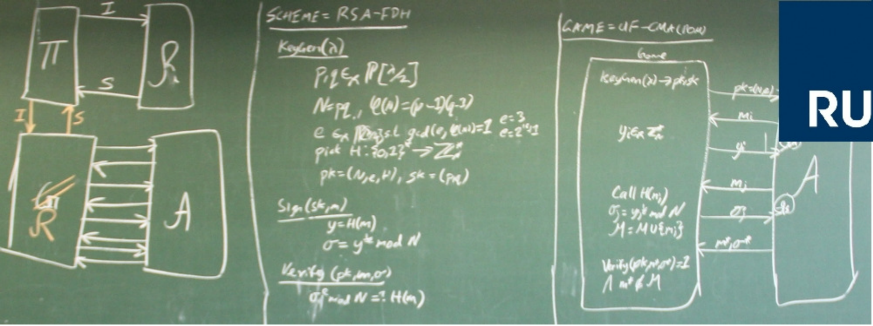
Summary

Our Contributions

- Generic framework, new constructions
- Extensions: flat-tree signatures, ssNIZK, multi-challenge PKE
- Shortcoming: linear signature size

Open Problems

- Reducing signature size
  - For DLIN, it is already solved by [CW13], [BKP14];
  - Tight and constant size signatures based on DLOG, RSA, SIS?



RUHR-UNIVERSITÄT BOCHUM

Many thanks for your attention!

QUESTIONS?

Horst Görtz Institut für IT-Sicherheit