# Enforcing honesty of certification authorities: Tagged one-time signature schemes

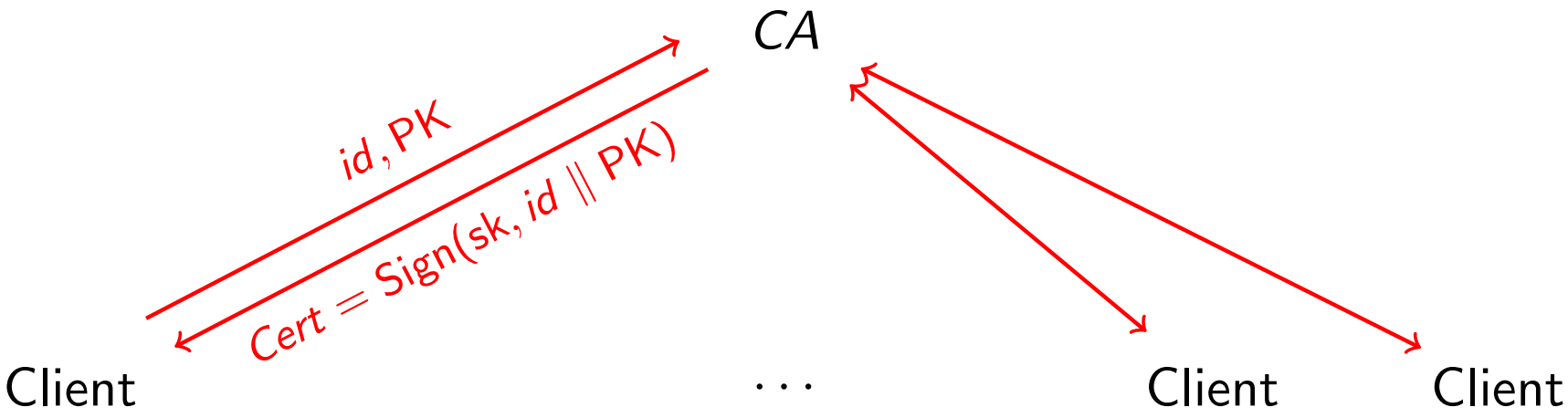
*Bertram Poettering and Douglas Stebila*

**Information Security Group**
Royal Holloway, University of London

bertram.poettering@rhul.ac.uk

Stanford, January 11, 2013

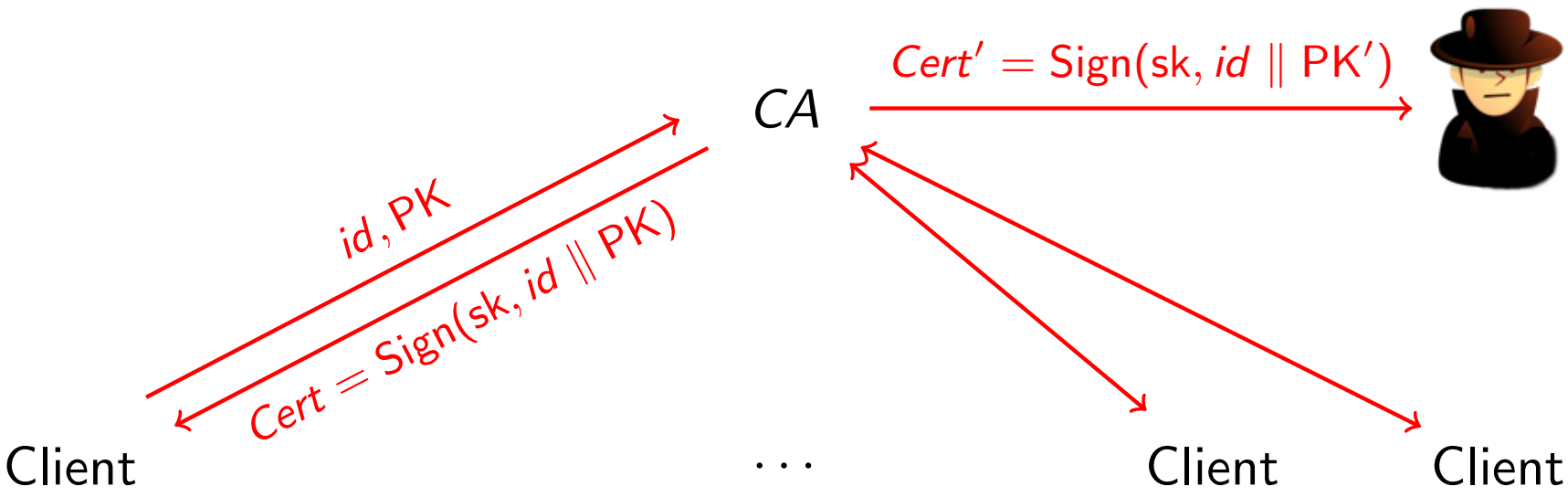
# PKIs and CAs: Current situation




## Signature-based PKIs

- full concentration of trust into CA
- CA has to be absolutely trustworthy

# PKIs and CAs: Current threats




## Malicious CA

- could falsely bind identities in use to auxiliary PKs
- run man-in-the-middle attacks against web sessions
- **ALL SECURITY IS LOST**

# PKIs and CAs: Should we really trust CAs?

Reasons not to trust CAs

- poor management practices
  - we will see examples...
- security breaches
  - we will see examples...
- criminal intention
- coercion
  - by crime organizations
  - legal coercion by law enforcement
  - legal(?) coercion by intelligence services

# CA incidents: A brief history

## Recent security incidents

- DigiNotar in July 2011
  - security breach, malicious certificates for many domains issued
- TURKTRUST in August 2011
  - issued intermediate CA with wildcard signing capabilities
  - later used for man-in-the-middle proxy filtering/scanning
  - no evidence for use in attack
  - detected only in Jan 2013
- Digicert Malaysia in November 2011
  - 22 certificates with weak private keys or missing revocation details issued
- KPN/Getronics in November 2011
  - suspended CA business after detecting infection on its web server
  - no evidence of certificate malfeasance

# CA incidents: Technical countermesures

So far, what helps against malicious CAs?

- Pinning (in HTTP)
  - hosts ask clients to remember PKs that appear in certificate chain
  - identified DigiNotar and TURKTRUST breaches
  - IETF Web Security Internet draft
- Tacking (in TLS)
  - hosts announce that their PK is not going to change for a specified amount of time
  - IETF TLS-WG Internet draft
- DANE (in DNS/TLS)
  - 'DNS-Based Authentication of Named Entities'
  - DNS records announce PKs used within TLS
  - RFC 6698

# Focus of this presentation

In this talk, we want to

- cryptographically enforce a unique binding of *id*s to PKs
  - no such guarantees in (signature-based) PKIs so far
- remain in non-interactive setting
  - no (trusted?) third parties
  - no 'out-of-band' communication
  - preserves robustness of PKIs

# Focus of this presentation

In this talk, we want to

- cryptographically enforce a unique binding of *id*s to PKs
  - no such guarantees in (signature-based) PKIs so far
- remain in non-interactive setting
  - no (trusted?) third parties
  - no 'out-of-band' communication
  - preserves robustness of PKIs

We propose a modified signature scheme for use in certification

# Focus of this presentation

In this talk, we want to

- cryptographically enforce a unique binding of *id*s to PKs
  - no such guarantees in (signature-based) PKIs so far
- remain in non-interactive setting
  - no (trusted?) third parties
  - no 'out-of-band' communication
  - preserves robustness of PKIs

We propose a modified signature scheme for use in certification

Our scheme

- makes misbehaving (cryptographically) fatal
- gives strong incentive to do well with management practices
- puts CAs into strong position against legal coercion

# TOSS: Tagged One-time Signature Scheme

New primitive: tagged one-time signatures (TOSS)

- similar to standard signature schemes
- authentication of tag/message pairs
- adversary cannot forge signatures (akin to EUF-CMA)
- distinguishing property: double-signature forgeability
  - intended security loss if signer 'misbehaves'

Syntax of TOSS

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(1^\lambda)$ outputs signing key and verification key
- $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{tag}, \mathsf{msg})$ signs $\mathsf{tag}, \mathsf{msg} \in \{0,1\}^*$
- $\{0,1\} \leftarrow \mathsf{Ver}(\mathsf{vk}, \mathsf{tag}, \mathsf{msg}, \sigma)$ verifies signatures

Correctness of TOSS

- as usual, with universal quantification over $\mathsf{tag}, \mathsf{msg} \in \{0,1\}^*$

# Security of TOSS: Unforgeability

Security goal: unforgeability (EUF)

- similar to unforgeability of standard signature schemes
- main difference: adversary not allowed to request signatures on different messages for the same tag

**$\mathbf{Exp}^{\mathsf{EUF}}(1^\lambda)$**

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- $(\mathsf{tag}^*, \mathsf{msg}^*, \sigma^*) \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{Sign}}}(\mathsf{vk})$
  - If $\mathcal{A}$ queries $\mathcal{O}_{\mathsf{Sign}}(\mathsf{tag}, \mathsf{msg})$:
    - Append $(\mathsf{tag}, \mathsf{msg})$ to SigList
    - $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, \mathsf{tag}, \mathsf{msg})$
    - Return $\sigma$ to $\mathcal{A}$
- Return 1 iff all the following hold:
  - $\mathsf{Ver}(\mathsf{vk}, \mathsf{tag}^*, \mathsf{msg}^*, \sigma^*) = 1$
  - $(\mathsf{tag}^*, \mathsf{msg}^*) \notin \mathsf{SigList}$
  - $\forall\, \mathsf{tag}, \mathsf{msg}_0, \mathsf{msg}_1$:

$$(\mathsf{tag}, \mathsf{msg}_0), (\mathsf{tag}, \mathsf{msg}_1) \in \mathsf{SigList} \Rightarrow \mathsf{msg}_0 = \mathsf{msg}_1$$

# Security of TOSS: Compromising pair of signatures

Intuition: A TOSS shall be forgeable once signer issued signatures on different messages but the same tag.

We make the condition precise:

## Definition (Compromising pair of signatures)

Fix verification key vk and tag/message/signature triples

$$S_1 = (\mathsf{tag}_1, \mathsf{msg}_1, \sigma_1) \quad \text{and} \quad S_2 = (\mathsf{tag}_2, \mathsf{msg}_2, \sigma_2)$$

such that

$$\mathsf{Ver}(\mathsf{vk}, \mathsf{tag}_1, \mathsf{msg}_1, \sigma_1) = 1 \quad \text{with} \quad \mathsf{Ver}(\mathsf{vk}, \mathsf{tag}_2, \mathsf{msg}_2, \sigma_2) = 1 \ .$$

Pair $(S_1, S_2)$ is compromising if $\mathsf{tag}_1 = \mathsf{tag}_2$ and $\mathsf{msg}_1 \neq \mathsf{msg}_2$.

Note: exactly this condition is excluded in $\mathbf{Exp}^{\mathsf{EUF}}$

# Security of TOSS: Double-signature forgeability

Security goal: double-signature forgeability (DSF)

- Intuition: given a compromising pair $(S_1, S_2)$ issued by a malicious signer, it should be trivial to craft valid signatures
- defined in respect to auxiliary algorithm

$$\sigma^* \leftarrow \mathsf{Forge}(\mathsf{vk}, (S_1, S_2), \mathsf{tag}^*, \mathsf{msg}^*)$$

that computes signatures for arbitrary tags/messages

- two variants: DSF and DSF* (the latter with 'trusted setup')

**$\mathbf{Exp}^{\mathsf{DSF}}(1^\lambda)$**

- $(\mathsf{vk}, (S_1, S_2), \mathsf{tag}^*, \mathsf{msg}^*) \leftarrow \mathcal{A}(1^\lambda)$
- $\sigma^* \leftarrow \mathsf{Forge}(\mathsf{vk}, (S_1, S_2), \mathsf{tag}^*, \mathsf{msg}^*)$
- Return 1 iff all the following hold:
  - $(S_1, S_2)$ is compromising
  - $\mathsf{Ver}(\mathsf{vk}, \mathsf{tag}^*, \mathsf{msg}^*, \sigma^*) \neq 1$

**$\mathbf{Exp}^{\mathsf{DSF}^*}(1^\lambda)$**

- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- $((S_1, S_2), \mathsf{tag}^*, \mathsf{msg}^*) \leftarrow \mathcal{A}(\mathsf{sk}, \mathsf{vk})$
- $\sigma^* \leftarrow \mathsf{Forge}(\mathsf{vk}, (S_1, S_2), \mathsf{tag}^*, \mathsf{msg}^*)$
- Return 1 iff all the following hold:
  - $(S_1, S_2)$ is compromising
  - $\mathsf{Ver}(\mathsf{vk}, \mathsf{tag}^*, \mathsf{msg}^*, \sigma^*) \neq 1$

# Security of TOSS: Double-signature extractability

Security goal: double-signature extractability (DSE)

- Intuition: given a compromising pair $(S_1, S_2)$ issued by a malicious signer, it should be trivial to compute the signing key
- defined in respect to auxiliary algorithm

$$\mathsf{sk}' \leftarrow \mathsf{Extract}(\mathsf{vk}, (S_1, S_2))$$

  that outputs a signing key
- two variants: DSE and DSE* (the latter with 'trusted setup')

**$\mathbf{Exp}^{\mathsf{DSE}}(1^\lambda)$**

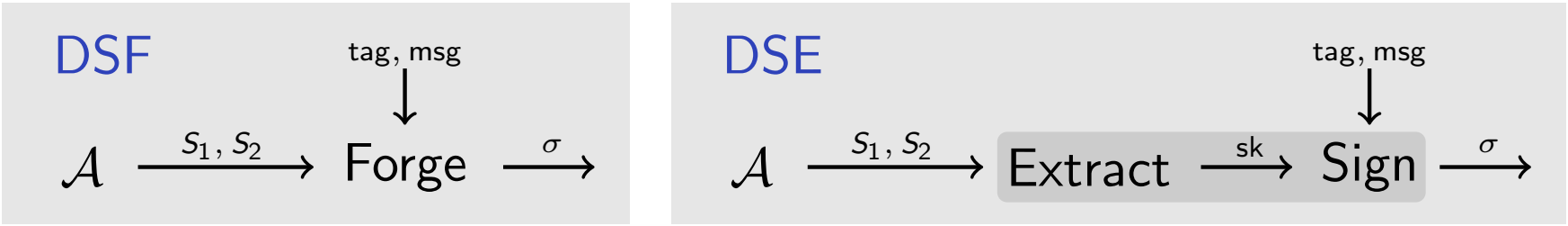
- $(\mathsf{vk}, (S_1, S_2)) \leftarrow \mathcal{A}(1^\lambda)$
- $\mathsf{sk}' \leftarrow \mathsf{Extract}(\mathsf{vk}, (S_1, S_2))$
- Return 1 iff all the following hold:
  - $(S_1, S_2)$ is compromising
  - $\mathsf{sk}'$ is not the signing key corresponding to vk

**$\mathbf{Exp}^{\mathsf{DSE}^*}(1^\lambda)$**

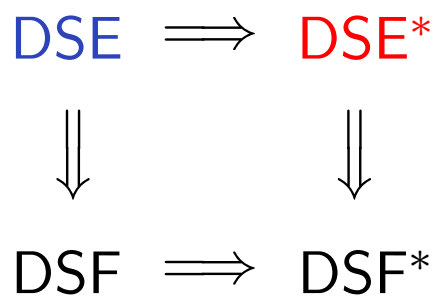
- $(\mathsf{sk}, \mathsf{vk}) \leftarrow \mathsf{KGen}(1^\lambda)$
- $(S_1, S_2) \leftarrow \mathcal{A}(\mathsf{sk}, \mathsf{vk})$
- $\mathsf{sk}' \leftarrow \mathsf{Extract}(\mathsf{vk}, (S_1, S_2))$
- Return 1 iff all the following hold:
  - $(S_1, S_2)$ is compromising
  - $\mathsf{sk}' \neq \mathsf{sk}$

# Double-signature extractability stronger than forgeability

## Comparing DSF and DSE

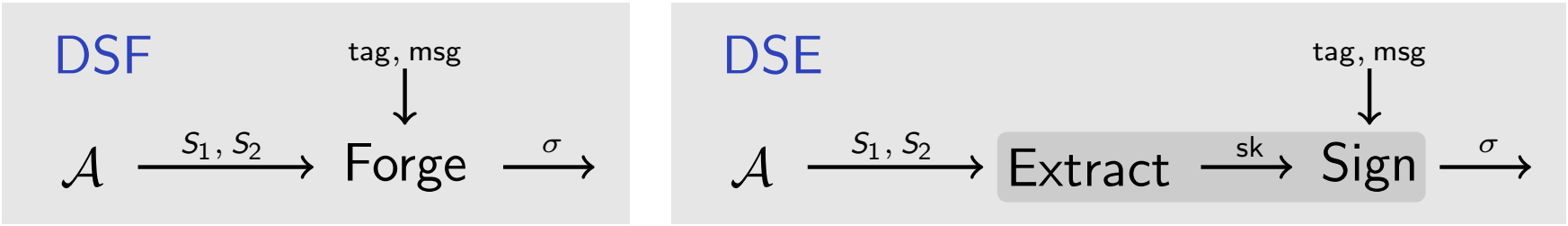



- DSE strictly stronger than DSF
  by Forge := Sign ∘ Extract construction
- DSE natural from engineer's perspective
  - our construction offers DSE*
  - our construction can be extended to DSE



# Double-signature extractability stronger than forgeability

## Comparing DSF and DSE




- DSE strictly stronger than DSF
  by Forge := Sign ∘ Extract construction
- DSE natural from engineer's perspective
  - our construction offers DSE*
  - our construction can be extended to DSE

$$\begin{array}{ccc} \text{DSE} & \Longrightarrow & \text{DSE}^* \\ \Downarrow & & \Downarrow \\ \text{DSF} & \Longrightarrow & \text{DSF}^* \end{array}$$

## Further advantage of DSE

- 'forged' signatures look identical to honest ones
  - relevant feature in practice
  - could be formalized: double-signature indistinguishability
  - counterexamples for DSF exist

# Application of TOSS: Enforcing honesty of CAs in PKIs

**Current PKI certificates**

$$\left\{ \mathsf{id}, \mathsf{PK}, \mathsf{Sign}^{\mathsf{STD}}(\mathsf{sk}, \mathsf{id} \parallel \mathsf{PK}) \right\}$$

where

- id is domain name, email address, ...
- PK is certified public key
- $\mathsf{Sign}^{\mathsf{STD}}$ is standard signature scheme

$\mathsf{id} = \texttt{bank.com}, \mathsf{PK} = 69\,6e\,2c\,20\ldots$
$\mathsf{id} = \texttt{bank.com}, \mathsf{PK} = 72\,20\,64\,61\ldots$

# Application of TOSS: Enforcing honesty of CAs in PKIs

## Current PKI certificates

$$\left\{\mathrm{id}, \mathrm{PK}, \mathrm{Sign}^{\mathrm{STD}}(\mathrm{sk}, \mathrm{id} \parallel \mathrm{PK})\right\}$$

## TOSS-based PKI certificates

$$\left\{\mathrm{id}, \mathrm{PK}, \mathrm{Sign}^{\mathrm{TOSS}}(\mathrm{sk}, \mathrm{id}, \mathrm{PK})\right\}$$

where

- id is domain name, email address, ...
- PK is certified public key
- $\mathrm{Sign}^{\mathrm{STD}}$ is standard signature scheme
- $\mathrm{Sign}^{\mathrm{TOSS}}$ is a tagged one-time signature

$\mathrm{id} = \texttt{bank.com}, \mathrm{PK} = 69\,6e\,2c\,20\ldots$
$\mathrm{id} = \texttt{bank.com}, \mathrm{PK} = 72\,20\,64\,61\ldots$

$\mathrm{id} = \texttt{bank.com}, \mathrm{PK} = 69\,6e\,2c\,20\ldots$
$\mathrm{id} = \texttt{bank.com}, \mathrm{PK} = 72\,20\,64\,61\ldots$

**New property:**

CA looses sk when certifying different PKs for same id

# Application of TOSS: Internet timestamping

## Internet timestamping service

- use current time epoch as tag
- use digest of current documents as msg
- publish $\mathrm{Sign}^{\mathrm{TOSS}}(sk, tag, msg)$
- DSF guarantees: timestamping service cannot 'rewind history'

Time: 8234098324 - Document: "This patent covers a beer umbrella"
Time: 8234098324 - Document: "This patent covers a life expectancy watch"

# Application of TOSS: Digital notaries

## Digital notary service

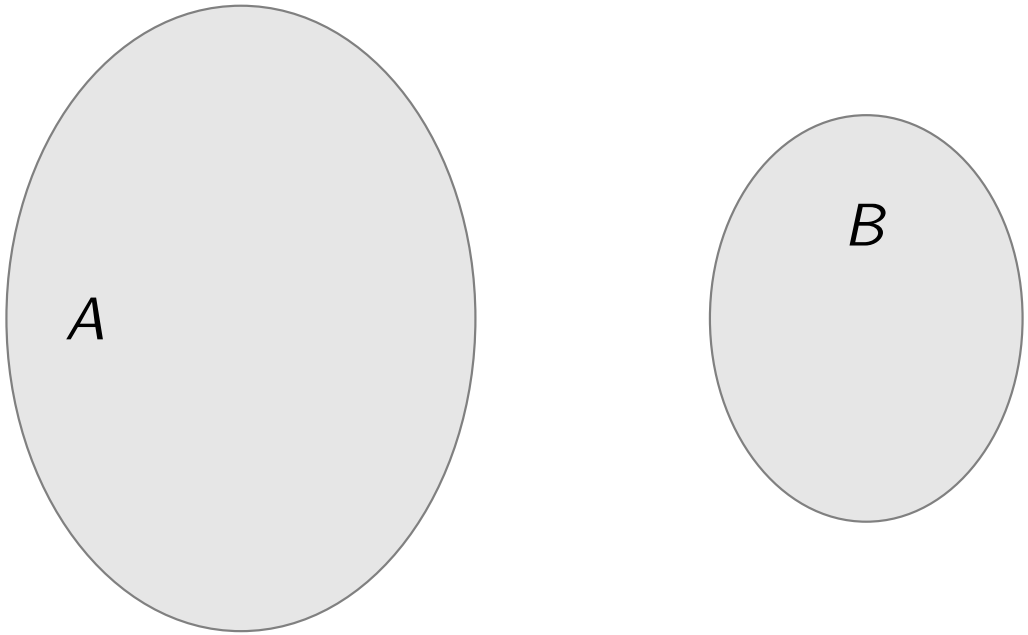
- use subject of contract as tag
- use affected bodies as msg
- publish $\mathrm{Sign}^{\mathrm{TOSS}}(sk, tag, msg)$
- DSF guarantees: contract can be signed only 'once'

Subject: 'Real property #94794 is sold to $\langle \ldots \rangle$.' - Body: Alice
Subject: 'Real property #94794 is sold to $\langle \ldots \rangle$.' - Body: Bob

# 2:1-TDF: Two-to-one trapdoor functions

New primitive: two-to-one trapdoor function (2:1-TDF)

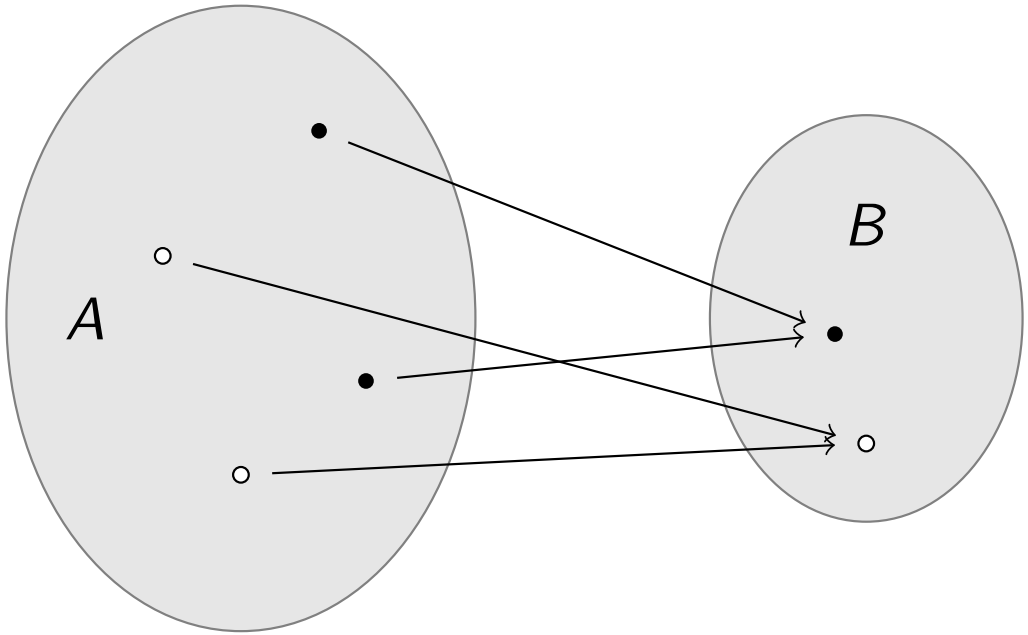
- finite sets $A, B$ such that $|A| = 2 \cdot |B|$



# 2:1-TDF: Two-to-one trapdoor functions

New primitive: two-to-one trapdoor function (2:1-TDF)

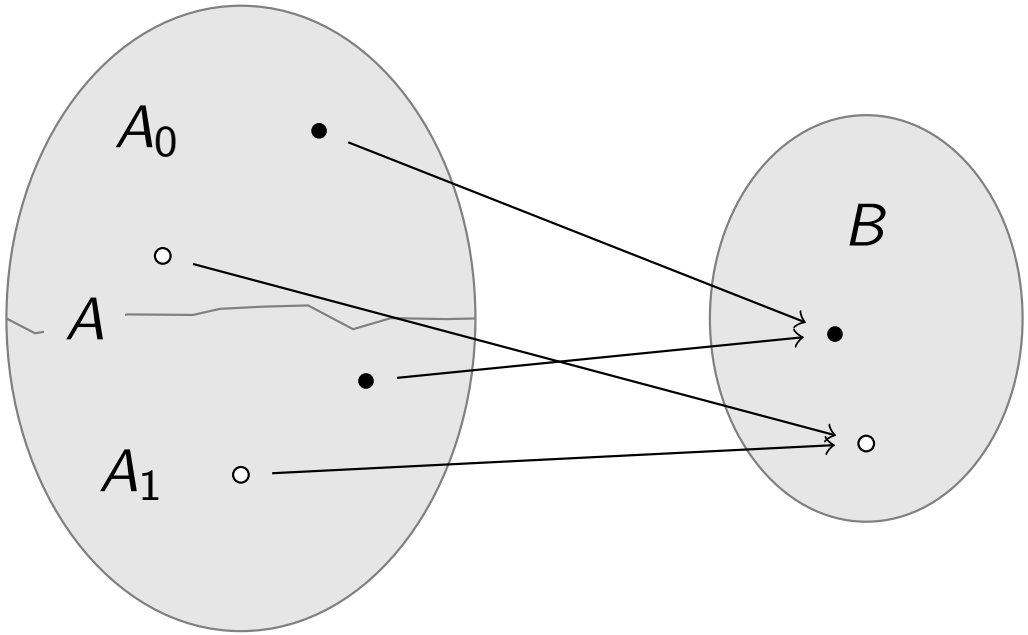
- finite sets $A, B$ such that $|A| = 2 \cdot |B|$
- surjective 2:1 function $f : A \to B$



# 2:1-TDF: Two-to-one trapdoor functions

New primitive: two-to-one trapdoor function (2:1-TDF)

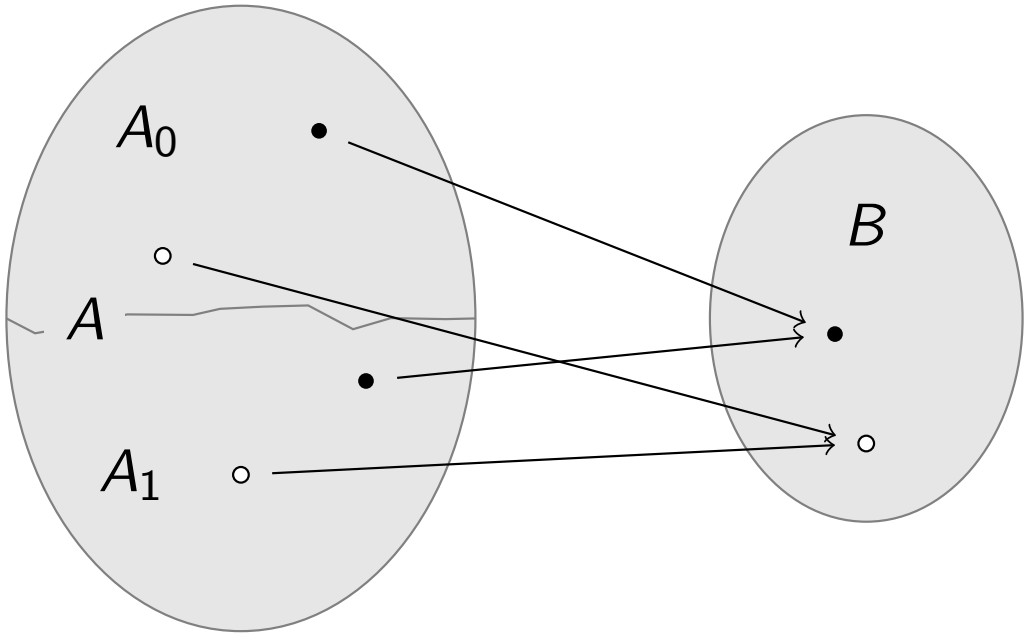
- finite sets $A, B$ such that $|A| = 2 \cdot |B|$
- surjective 2:1 function $f : A \to B$
- if $f^{-1}(b, 0)$ and $f^{-1}(b, 1)$ denote the two preimages of $b \in B$, define $A_0 = f^{-1}(B, 0)$ and $A_1 = f^{-1}(B, 1)$



# 2:1-TDF: Two-to-one trapdoor functions

New primitive: two-to-one trapdoor function (2:1-TDF)

- finite sets $A, B$ such that $|A| = 2 \cdot |B|$
- surjective 2:1 function $f : A \to B$
- if $f^{-1}(b, 0)$ and $f^{-1}(b, 1)$ denote the two preimages of $b \in B$, define $A_0 = f^{-1}(B, 0)$ and $A_1 = f^{-1}(B, 1)$
- $f$ efficient, but $f^{-1}$ hard without trapdoor
- define relation $a_0 \overset{x}{\sim} a_1 \Leftrightarrow a_0 \neq a_1 \wedge f(a_0) = f(a_1)$



# 2:1-TDF: One-wayness

## Technical requirement

- $A_0, A_1, B$ shall be efficiently publicly samplable and decidable

# 2:1-TDF: One-wayness

## Technical requirement

- $A_0, A_1, B$ shall be efficiently publicly samplable and decidable

## One-wayness

- preimage resistance (INV-1)
- second preimage resistance (INV-2)

$\mathbf{Exp}^{\text{INV-1}}_{\mathcal{A}}(1^\lambda)$

- $(\text{td}, \text{par}) \leftarrow \text{2:1-Gen}(1^\lambda)$
- $b \leftarrow_R B(\text{par})$
- $a \leftarrow \mathcal{A}(\text{par}, b)$
- Return 1 iff $f(a) = b$

$\mathbf{Exp}^{\text{INV-2}}_{\mathcal{B}}(1^\lambda)$

- $(\text{td}, \text{par}) \leftarrow \text{2:1-Gen}(1^\lambda)$
- $a \leftarrow_R A(\text{par})$
- $a' \leftarrow \mathcal{B}(\text{par}, a)$
- Return 1 iff $a \overset{x}{\sim} a'$

# 2:1-TDF: Extractability

Extractability (optional)

- defined in respect to auxiliary algorithm

$$\mathsf{td}' \leftarrow \mathsf{Extract}(\mathsf{par}, a, a')$$

that computes $\mathsf{td}' = \mathsf{td}$ from all $a, a' \in A$ with $a \overset{\times}{\approx} a'$

# 2:1-TDF: Extractability

Extractability (optional)

- defined in respect to auxiliary algorithm

$$\mathsf{td}' \leftarrow \mathsf{Extract}(\mathsf{par}, a, a')$$

that computes $\mathsf{td}' = \mathsf{td}$ from all $a, a' \in A$ with $a \overset{\times}{\sim} a'$

INV-1 vs. INV-2

- INV-2 $\Rightarrow$ INV-1 (as expected)
- if extractable: INV-1 $\Leftrightarrow$ INV-2

# 2:1-TDF: Extractability

Extractability (optional)

- defined in respect to auxiliary algorithm

$$\mathsf{td}' \leftarrow \mathsf{Extract}(\mathsf{par}, a, a')$$

that computes $\mathsf{td}' = \mathsf{td}$ from all $a, a' \in A$ with $a \overset{\times}{\sim} a'$

INV-1 vs. INV-2

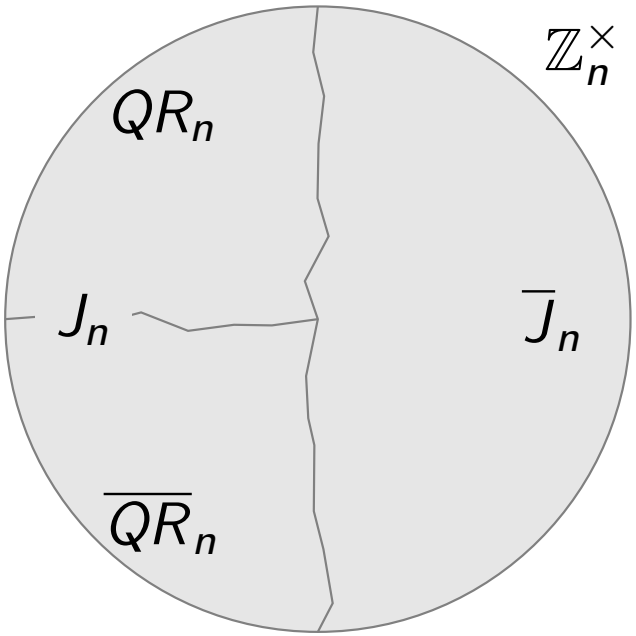
- INV-2 $\Rightarrow$ INV-1 (as expected)
- if extractable: INV-1 $\Leftrightarrow$ INV-2

2:1-TDF vs. CFP (claw-free permutation)

- CFPs imply 2:1-TDFs, other direction unclear
- CFPs have no (formalized) extraction capability

# 2:1-TDF: Factoring-based construction I

Let $n = pq$ be Blum integer.

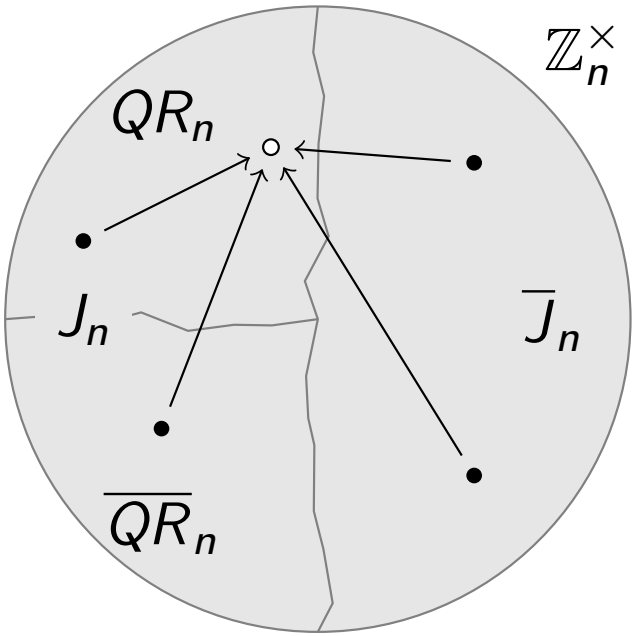



## Known facts

- $QR_n$ not decidable, not directly samplable

# 2:1-TDF: Factoring-based construction I

Let $n = pq$ be Blum integer.

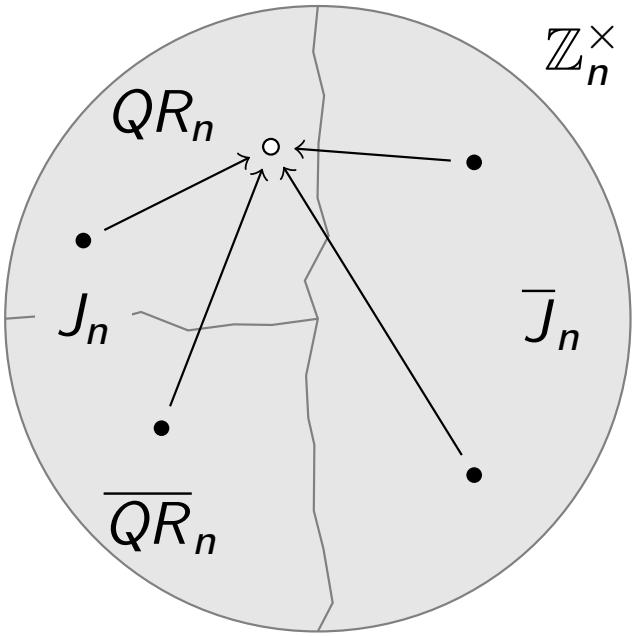


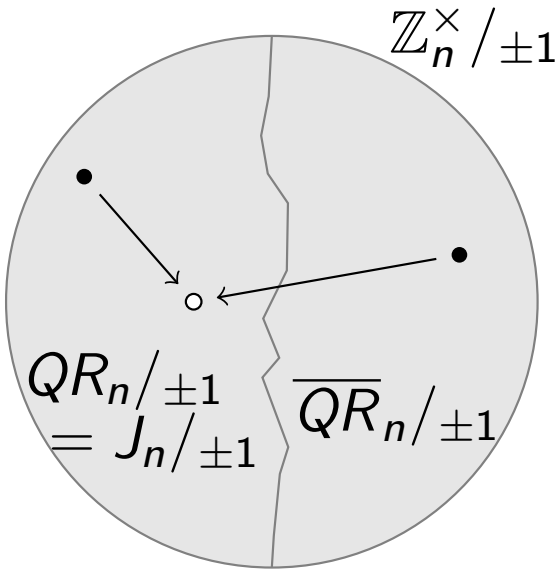

## Known facts

- $QR_n$ not decidable, not directly samplable
- squaring operation $\mathbb{Z}_n^\times(J_n, QR_n) \to QR_n$ is 4:1 (2:1, 1:1)
- computing square roots as hard as factoring
- $n$ can be factored from $x_0 \in J_n, x_1 \in \overline{J}_n$ with $x_0^2 = x_1^2$

# 2:1-TDF: Factoring-based construction II

Let $n = pq$ be Blum integer. The following bases on [GMR88,HK09].



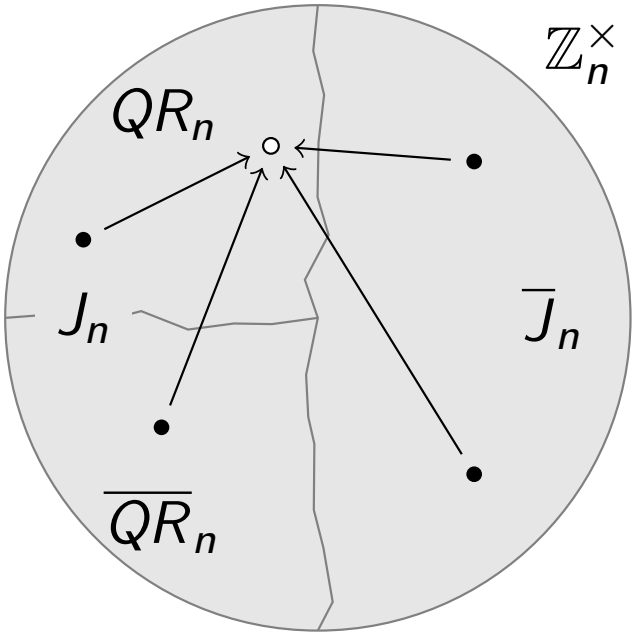



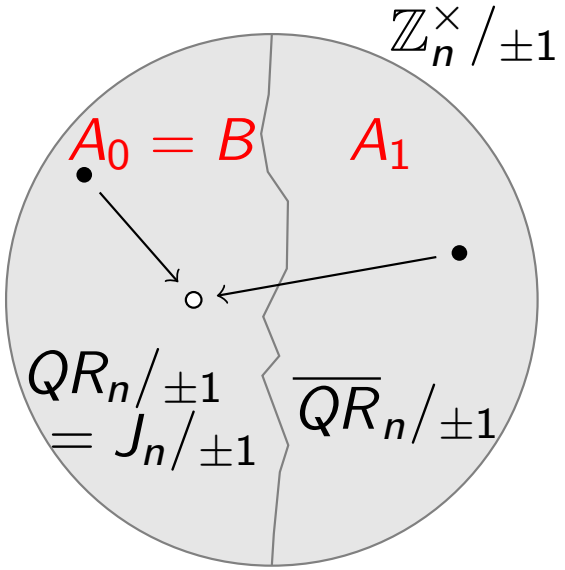

## Some number theory

- $\{\pm 1\}$ normal in $\mathbb{Z}_n^\times$, induces homomorphism $\psi : \mathbb{Z}_n^\times \to \mathbb{Z}_n^\times/_{\pm 1}$
- define groups $QR_n/_{\pm 1} = \psi(QR_n)$ and $J_n/_{\pm 1} = \psi(J_n)$
- computing 'square roots' as hard as factoring
- $n$ can be factored from $x_0 \in QR_n/_{\pm 1}, x_1 \in \overline{QR}_n/_{\pm 1}$ with $x_0^2 = x_1^2$

# 2:1-TDF: Factoring-based construction III

Let $n = pq$ be Blum integer. The following bases on [GMR88,HK09].




## Constructing a 2:1-TDF

- set $A_0 = B = QR_n/_{\pm 1}$ and $A_1 = \overline{QR}_n/_{\pm 1}$
- $A_0$ and $A_1$ and $B$ are efficiently samplable
- 'squaring' operation $A \to B$ is 2:1-TDF
- any $a, a' \in A$ with $a \overset{\times}{\sim} a'$ leak factorization

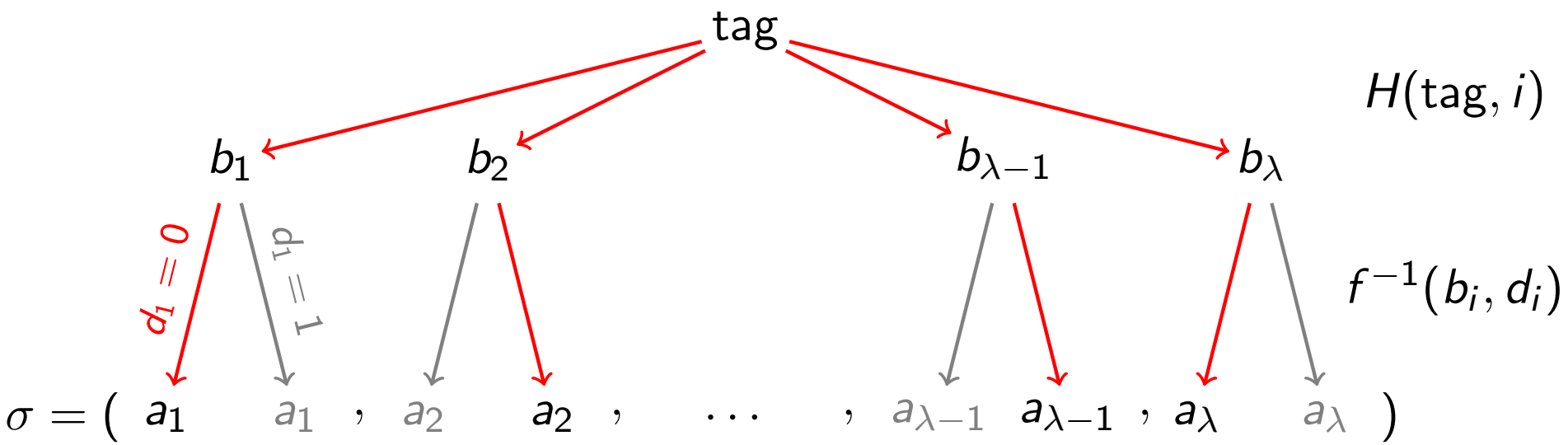
# Our TOSS construction (simplified)

## TOSS construction

- KGen $\equiv$ 2:1-Gen
- Sign(sk, tag, msg)
  - $\forall i : b_i = H(\text{tag}, i)$
  - $d_1, \ldots, d_\lambda \leftarrow H^{\#}(\text{tag}, \text{msg})$
  - $\forall i : a_i = f^{-1}(b_i, d_i)$
  - $\sigma = (a_1, \ldots, a_\lambda)$
- Ver(vk, tag, msg) clear

$H : \{0,1\}^* \to B$ random oracle
$H^{\#} : \{0,1\}^* \to \{0,1\}^\lambda$ CRHF
$f$ extractable 2:1-TDF

(requires decidability $A_0 \leftrightarrow A_1$)



# Our TOSS construction (full)

The scheme is simple and elegant.

# Our TOSS construction (full)

The scheme is simple and elegant.

But it is unclear how to do the security reduction...

# Our TOSS construction (full)

The scheme is simple and elegant.
But it is unclear how to do the security reduction...

## 'Repaired' TOSS construction

- KGen $\equiv$ 2:1-Gen
- Sign(sk, tag, msg)
  - $s \leftarrow f^{-1}(H(\text{tag}), 0)$
  - $\forall i : b_i = H(s, \text{tag}, i)$
  - $d_1, \ldots, d_\lambda \leftarrow H^{\#}(s, \text{tag}, \text{msg})$
  - $\forall i : a_i = f^{-1}(b_i, d_i)$
  - $\sigma = (a_1, \ldots, a_\lambda)$
- Ver(vk, tag, msg) clear

$H : \{0,1\}^* \to B$ random oracle
$H^{\#} : \{0,1\}^* \to \{0,1\}^\lambda$ CRHF
$f$ extractable 2:1-TDF

(requires decidability $A_0 \leftrightarrow A_1$)

# Our TOSS construction (full)

The scheme is simple and elegant.
But it is unclear how to do the security reduction...

## 'Repaired' TOSS construction

- KGen $\equiv$ 2:1-Gen
- Sign(sk, tag, msg)
  - $s \leftarrow f^{-1}(H(\text{tag}), 0)$
  - $\forall i : b_i = H(s, \text{tag}, i)$
  - $d_1, \ldots, d_\lambda \leftarrow H^{\#}(s, \text{tag}, \text{msg})$
  - $\forall i : a_i = f^{-1}(b_i, d_i)$
  - $\sigma = (a_1, \ldots, a_\lambda)$
- Ver(vk, tag, msg) clear

$H : \{0,1\}^* \to B$ random oracle
$H^{\#} : \{0,1\}^* \to \{0,1\}^\lambda$ CRHF
$f$ extractable 2:1-TDF

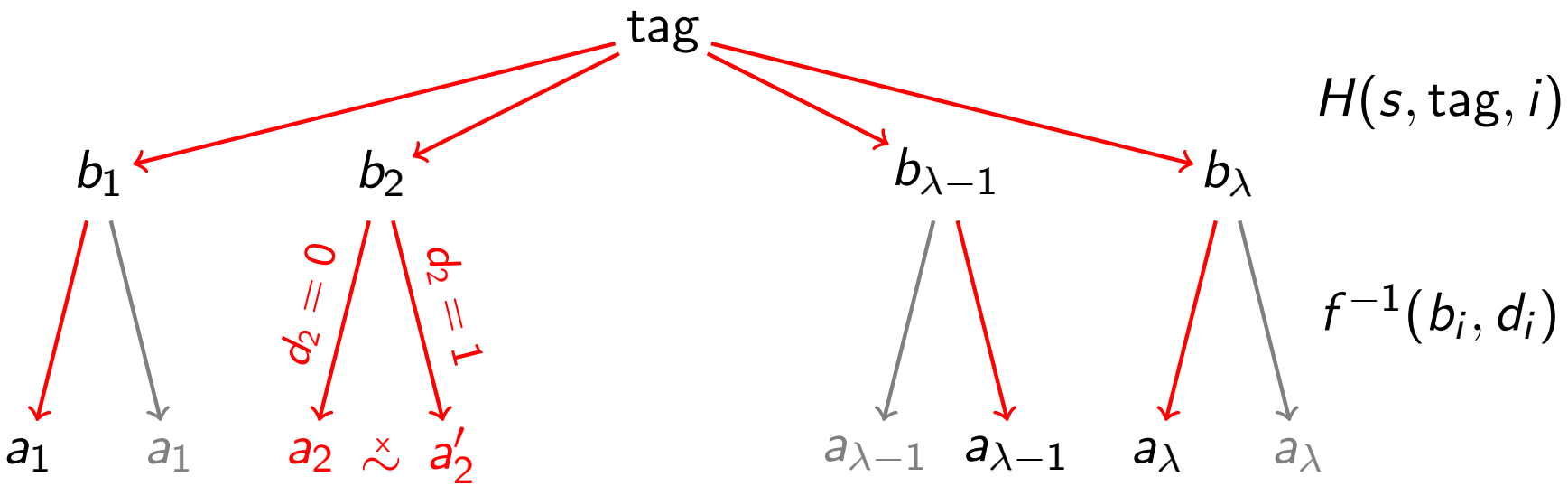
(requires decidability $A_0 \leftrightarrow A_1$)

## Theorem (Unforgeability of TOSS)

If $H$ is RO, $H^{\#}$ is CRHF, and $f$ is 2:1-TDF, then TOSS provides EUF.

Note: TOSS even strongly unforgeable (and unique)

# Our TOSS construction: DSE*




Theorem (Double-signature extractability of TOSS)
If $H^{\#}$ is CRHF and $f$ is extractable 2:1-TDF, then TOSS provides DSE*.

Note: Can be strengthened to DSE. Really relevant?

# Our TOSS construction: Practical aspects

## Security requirements

- tolerated forging probability $2^{-80}$
- $2^{25}$ signature queries allowed
- ECRYPT recommendations

## Derived key/signature sizes

- moduli of 2432 bits
- TOSS signature size: 48 kB

## Efficiency of signature verification

- $\lambda + 1$ squarings
- $\lambda$ Jacobi symbol evaluations ($A_0 \leftrightarrow A_1$)
- $\lambda$ Jacobi symbol evaluations (sampling of $b_i$ in RO $H$)

# Conclusion

## Tagged one-time signature schemes (TOSS)

- violation of rules always catastrophic (for signers)
- enforcement of honesty of signers?

## Security of TOSS

- notions of DSF, DSF*, DSE, DSE* and their relations

## Extractable 2:1 trapdoor functions (2:1-TDF)

- '2:1' version of TDPs, more general than CFPs
- extractability: colliding preimages reveal trapdoor
- construction based on factorization

## 2:1-TDF-based TOSS

- achieves EUF, DSE* and DSF* (DSE and DSF feasible)
- efficient verification
- signature size not prohibitively large