Shorter double-authentication preventing signatures for small address spaces

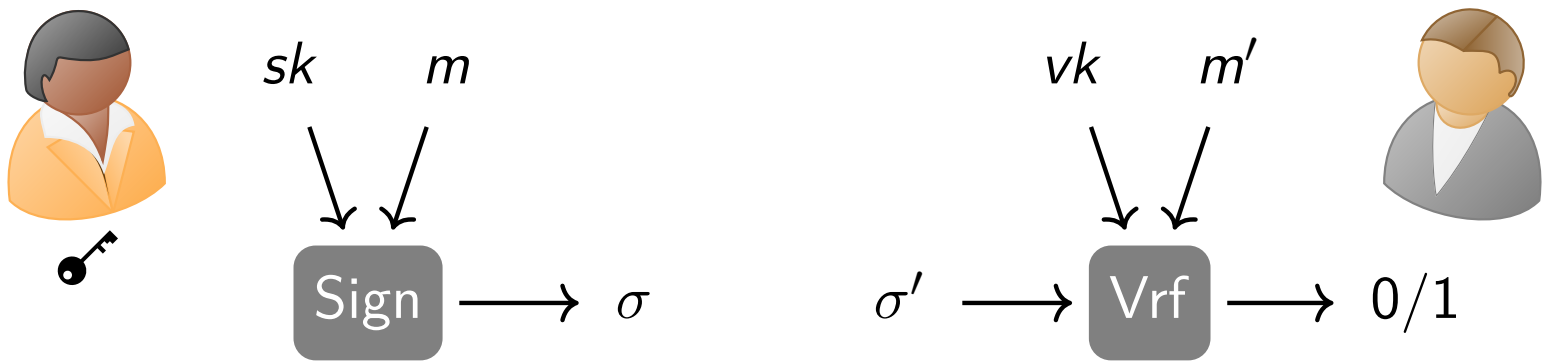
Bertram Poettering
Royal Holloway, University of London

AFRICACRYPT · Marrakesh · 2018-May-09

Signature schemes

In a nutshell

- digital analogue to written signatures
- easy to create and verify
- security goal: unforgeability



Examples and applications

- PKCS#1v1.5, DSA, ECDSA, Schnorr
- message authentication (emails), entity authentication (TLS, . . .)

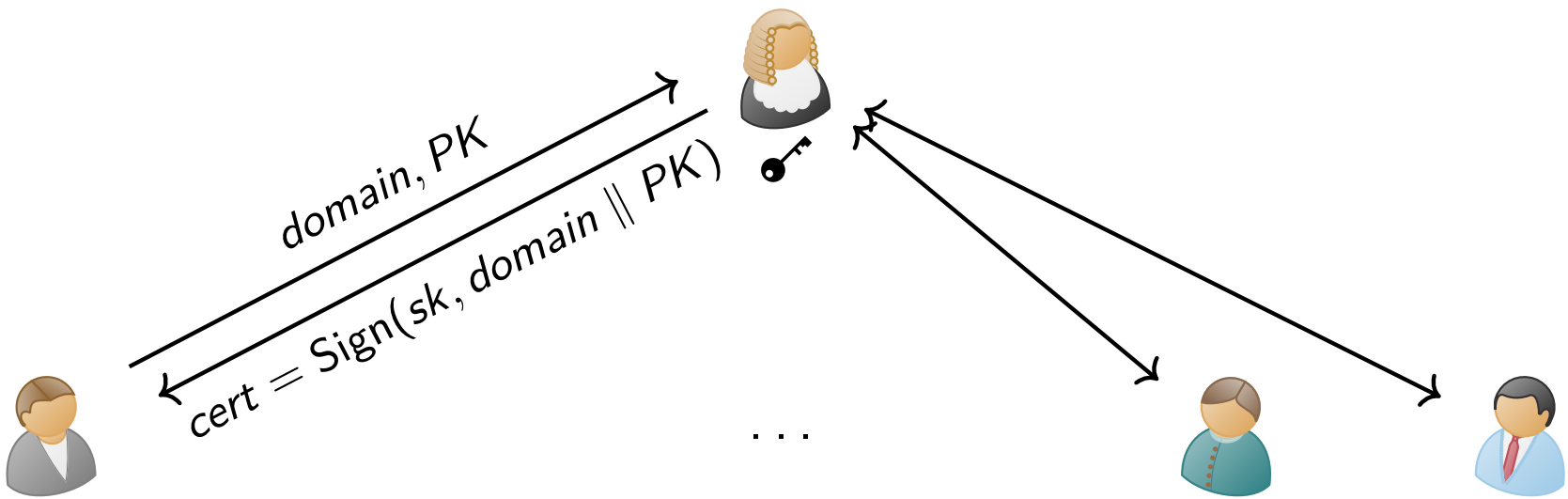
Public-key infrastructure (PKI)

In a nutshell

- solution to key distribution problem of public key cryptography
- move trust from many to one/few entities
- organize entities hierarchically in tree/forest
- top + intermediate nodes: **certificate authority** (CA)

Applications

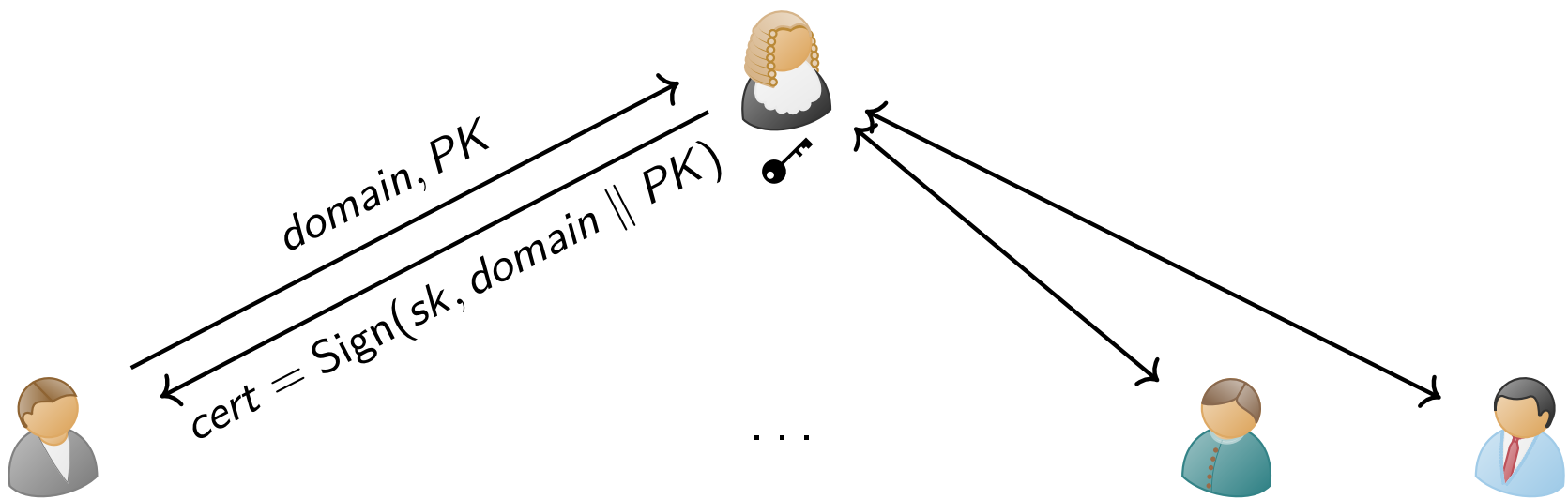
- X.509, email security, web security, business networks



Public-key infrastructure (PKI)

Security

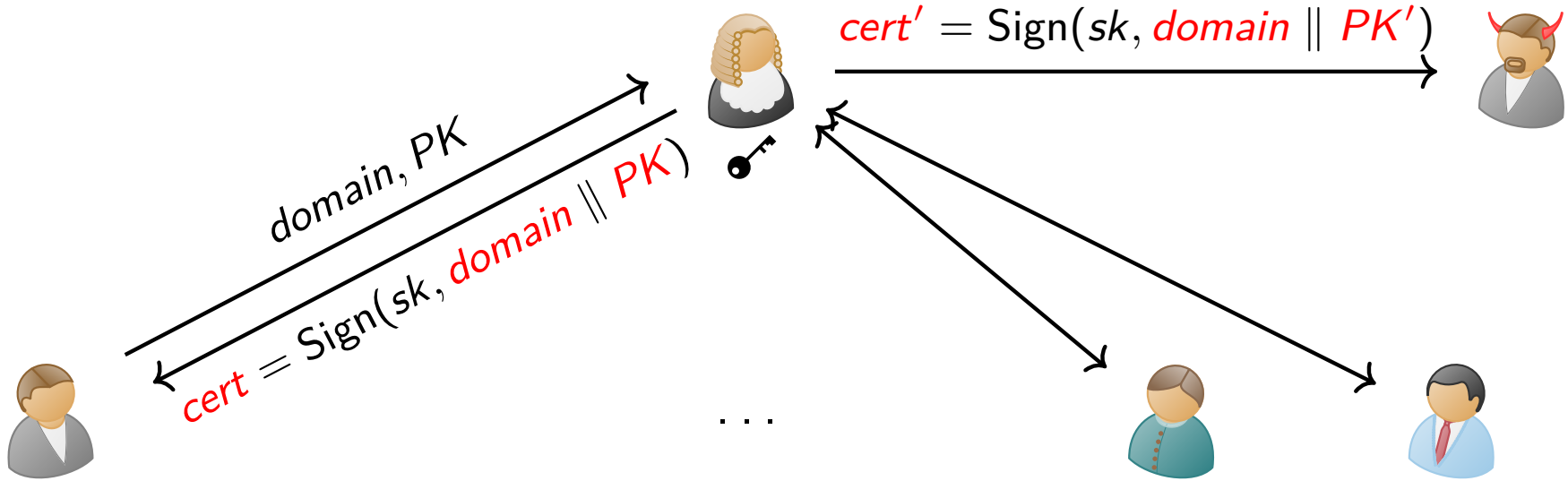
- in theory: yes; in practice: almost
- concentration of trust implies single point of failure
- CA corruption absolutely fatal
- after corruption: email forgery, decryption of TLS traffic (MITM)



Public-key infrastructure (PKI)

Security

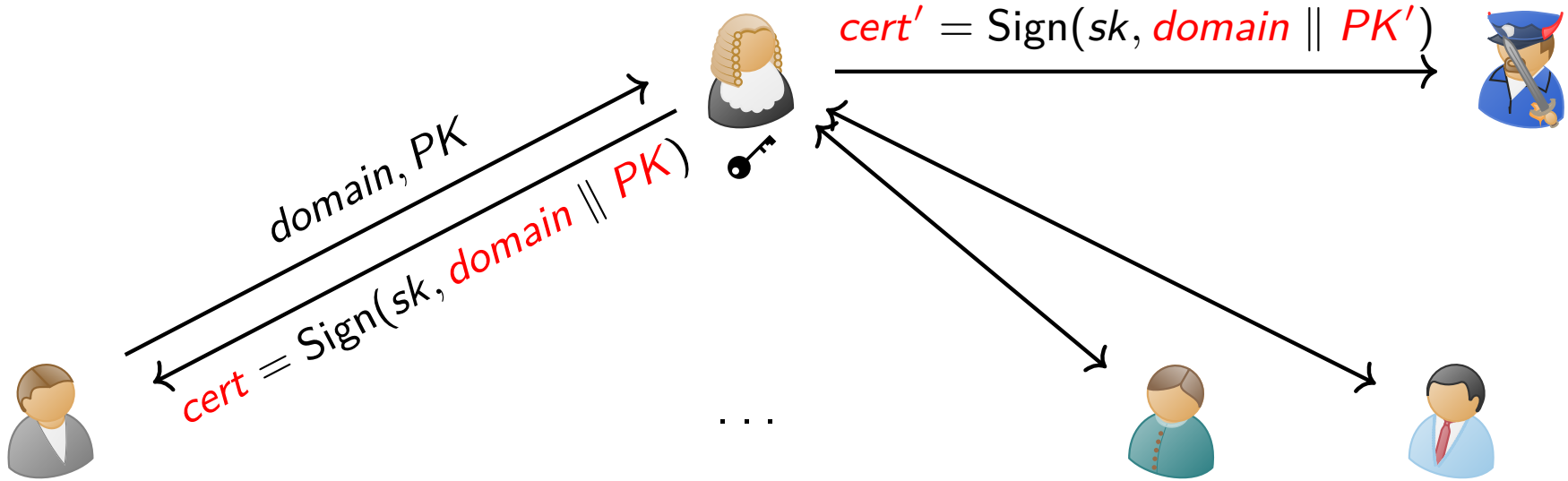
- in theory: yes; in practice: almost
- concentration of trust implies single point of failure
- CA corruption absolutely fatal
- after corruption: email forgery, decryption of TLS traffic (MITM)
- weaker attack with similar consequences: **double-signing**



Public-key infrastructure (PKI)

Security

- in theory: yes; in practice: almost
- concentration of trust implies single point of failure
- CA corruption absolutely fatal
- after corruption: email forgery, decryption of TLS traffic (MITM)
- weaker attack with similar consequences: **double-signing**
  e.g. after coercion by law enforcement, intelligence services, . . .



How can we protect PKIs against CA coercion?

And can we do this without trusted parties,
without added interactivity?

How can we protect PKIs against CA coercion?

And can we do this without trusted parties,
without added interactivity?

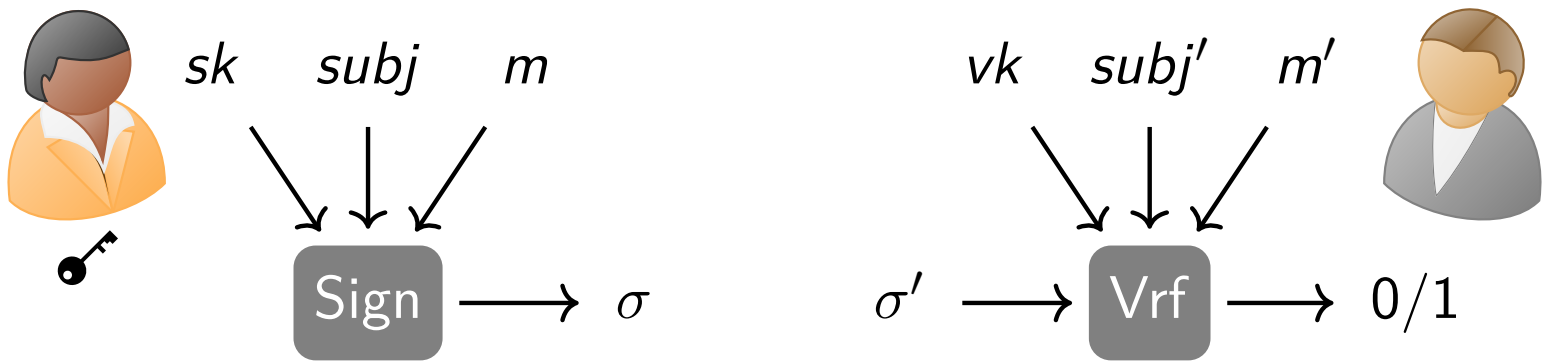
DAPS:

Double-authentication preventing signatures

Double-authentication preventing signatures (DAPS)

In a nutshell

- special signature scheme dedicated for use in CAs
- cryptographic enforcement of **binding uniqueness**
- double-signing becomes fatal: reveals signing key



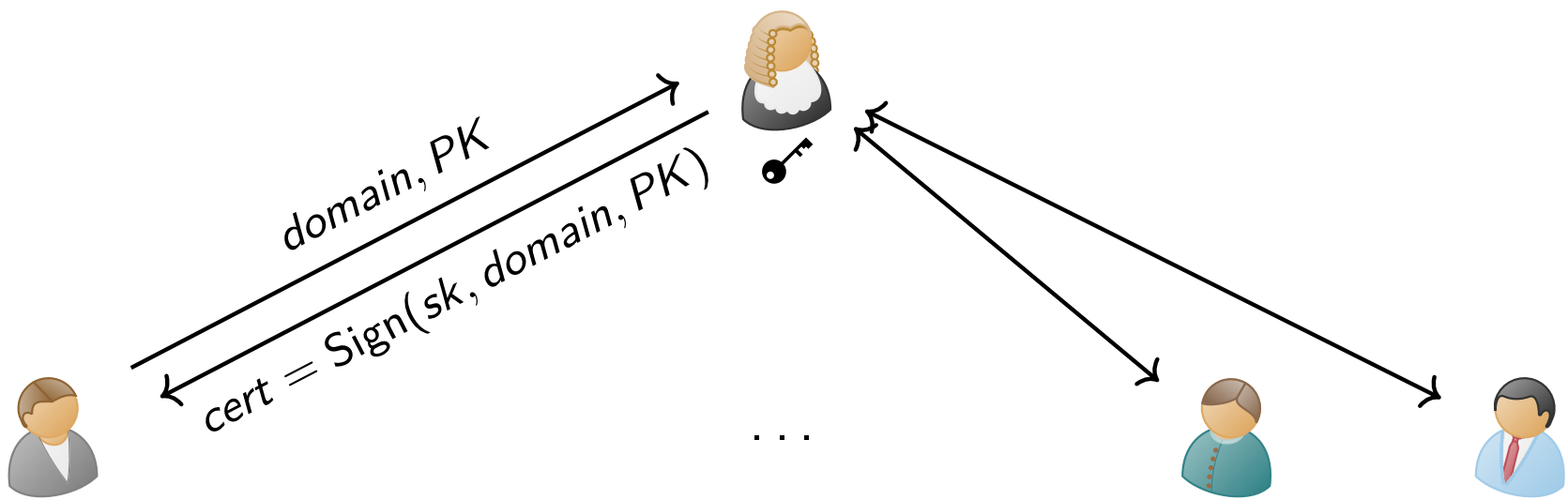
Properties

- **unforgeability**: cannot forge on fresh *subj*, any *m*
- **extractability**: signature pair on $(subj, m), (subj, m')$ reveals *sk*

DAPS in PKI

Application for CAs

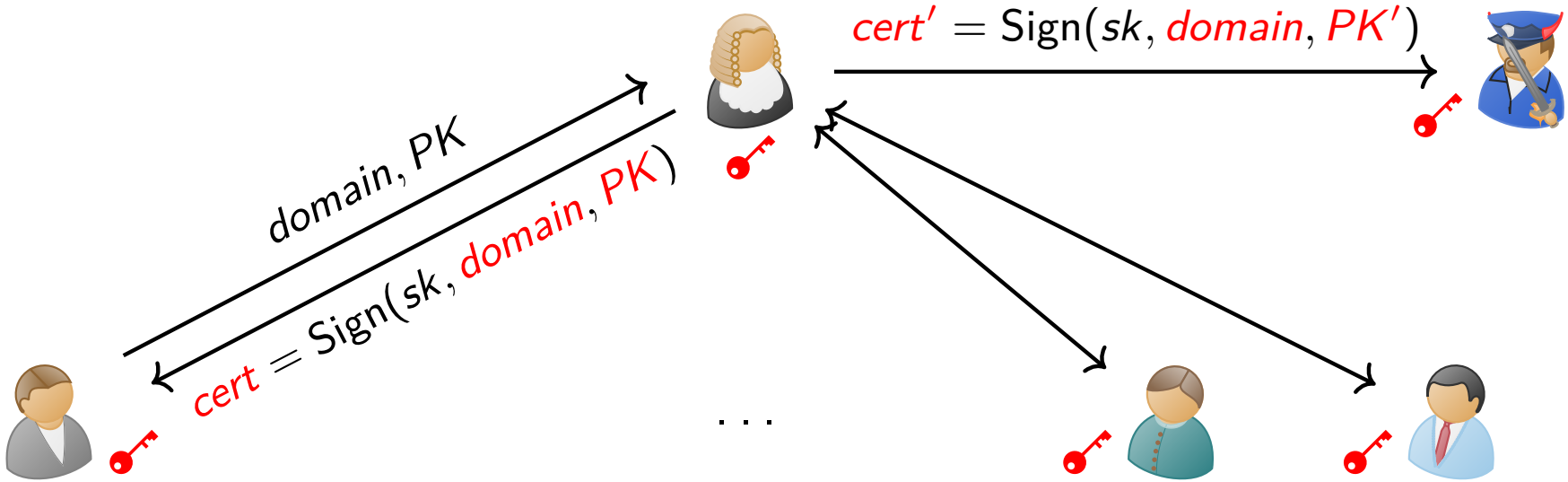
- using a double-issued certificate leaks CA's signing key
- brings victim to power, corruptor has to take responsibility
- makes signer corruption pointless
- no (trusted?) third parties, remains in non-interactive setting



DAPS in PKI

Application for CAs

- using a double-issued certificate leaks CA's signing key
- brings victim to power, corruptor has to take responsibility
- makes signer corruption pointless
- no (trusted?) third parties, remains in non-interactive setting



DAPS in the literature

(all figures for 128 bit security level)

Constructions of PoSt14, PoSt16

- ad-hoc from factoring
- huge signatures ($\approx$530.000 bit)

Construction of RuKaSc15

- from chameleon hash function and Merkle tree
- large signatures ($\approx$130.000 bit with ECC)

Construction of BePoSt17

- ad-hoc from factoring
- compact signatures (2048 bit)

Construction of DeRaSI18

- DL-based, e.g. from DSA + ElGamal + secret sharing
- small signatures (1280 bit)

DAPS in the literature

(all figures for 128 bit security level)

Constructions of PoSt14, PoSt16

- ad-hoc from factoring
- huge signatures ($\approx$530.000 bit)

Construction of RuKaSc15

- from chameleon hash function and Merkle tree
- large signatures ($\approx$130.000 bit with ECC)

Construction of BePoSt17

- ad-hoc from factoring
- compact signatures (2048 bit)

tight

Construction of DeRaSI18

- DL-based, e.g. from DSA + ElGamal + secret sharing
- small signatures (1280 bit)

untight

DAPS in the literature

(all figures for 128 bit security level)

Constructions of PoSt14, PoSt16

- ad-hoc from factoring
- huge signatures ($\approx$530.000 bit)

Construction of RuKaSc15

- from chameleon hash function and Merkle tree
- large signatures ($\approx$130.000 bit with ECC)

Construction of BePoSt17

- ad-hoc from factoring
- compact signatures (2048 bit)

Construction of DeRaSI18

- DL-based, e.g. from DSA + ElGamal + secret sharing
- small signatures (1280 bit)

DAPS in the literature

(all figures for 128 bit security level)

Constructions of PoSt14, PoSt16

- ad-hoc from factoring
- huge signatures ($\approx$530.000 bit)

Construction of RuKaSc15

- from chameleon hash function and Merkle tree
- large signatures ($\approx$130.000 bit with ECC)

Construction of BePoSt17

- ad-hoc from factoring
- compact signatures (2048 bit)

Construction of DeRaSI18

- DL-based, e.g. from DSA + ElGamal + secret sharing
- small signatures (1280 bit)
- **supports only small number of subjects**

Details of 'best' prior DAPS

(in the small subject-space setting)

Construction of Derler, Ramacher, Slamanig (EuroS&P 2018)

- fix signing key *sk* of any DL-based signature scheme
- for each *subj*: create individual secret sharing instance for *sk*
- to DAPS-sign a pair $(subj, m)$:
  - sign m with DL-based scheme
  - create m-specific share of *subj*-specific secret sharing instance
  - DAPS signature: DL-based signature + share + ZK proof
- auxiliary values of secret sharing instances included in DAPS sk, vk

Signature and key sizes

size of ...	in **DeRaSl18**
signatures	1280 bit
signing keys	cardinality of subj space
verification keys	cardinality of subj space

Details of 'best' prior DAPS

(in the small subject-space setting)

Construction of Derler, Ramacher, Slamanig (EuroS&P 2018)

- fix signing key *sk* of any DL-based signature scheme
- for each *subj*: create individual secret sharing instance for *sk*
- to DAPS-sign a pair $(subj, m)$:
  - sign m with DL-based scheme
  - create m-specific share of *subj*-specific secret sharing instance
  - DAPS signature: DL-based signature + share + ZK proof
- auxiliary values of secret sharing instances included in DAPS sk, vk

Signature and key sizes

size of ...	in **DeRaSI18**	in our construction
signatures	1280 bit	256 bit
signing keys	cardinality of subj space	256 bit
verification keys	cardinality of subj space	cardinality of subj space

Details of 'best' prior DAPS

(in the small subject-space setting)

Construction of Derler, Ramacher, Slamanig (EuroS&P 2018)

- fix signing key *sk* of any DL-based signature scheme
- for each *subj*: create individual secret sharing instance for *sk*
- to DAPS-sign a pair $(subj, m)$:
  - sign m with DL-based scheme
  - create m-specific share of *subj*-specific secret sharing instance
  - DAPS signature: DL-based signature + share + ZK proof
- auxiliary values of secret sharing instances included in DAPS sk, vk

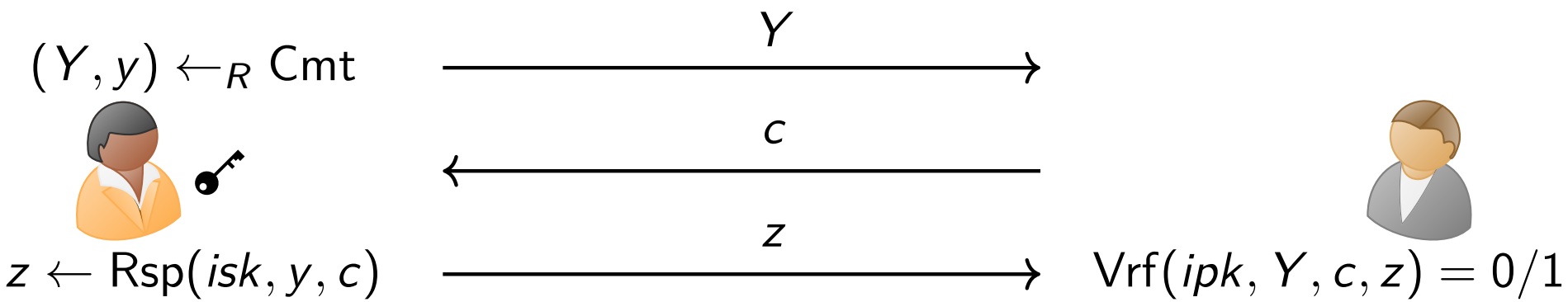
Signature and key sizes

size of ...	in **DeRaSI18**	in our construction
signatures	1280 bit	256 bit
signing keys	cardinality of subj space	256 bit
verification keys	cardinality of subj space	cardinality of subj space

ID schemes

Sigma protocols

- three messages: commitment Y, challenge c, response z
- constructions known from factoring, DLP, ...



Formal properties

- **zero-knowledge**: simulate (Y, c, z) without knowing isk
- **special soundness**: recover isk from $(Y, c, z), (Y, c', z')$

(Strictly one-time) signatures from ID schemes

Fiat–Shamir transform

- generic construction: unforgeable signatures from ID scheme

FS-Sign(sk, m)

$isk \leftarrow sk$

$(Y, y) \leftarrow_R \mathsf{Cmt}$

$c \leftarrow H(Y, m)$

$z \leftarrow \mathsf{Rsp}(isk, y, c)$

$\sigma \leftarrow (Y, z)$

(Strictly one-time) signatures from ID schemes

Fiat–Shamir transform

- generic construction: unforgeable signatures from ID scheme

Fixed-Commitment transform

- commitment establishment moved to key generation
- by special soundness: double-signing leaks signing key
  - also observe: half-size signatures

FS-Sign(sk, m)

$isk \leftarrow sk$

$(Y, y) \leftarrow_R \mathrm{Cmt}$

$c \leftarrow H(Y, m)$

$z \leftarrow \mathrm{Rsp}(isk, y, c)$

$\sigma \leftarrow (Y, z)$

In **KGen**: $(Y, y) \leftarrow_R \mathrm{Cmt}$

FC-Sign(sk, m)

$(isk, Y, y) \leftarrow sk$

$c \leftarrow H(Y, m)$

$z \leftarrow \mathrm{Rsp}(isk, y, c)$

$\sigma \leftarrow z$

DAPS from strictly one-time signatures

Our DAPS construction

- associate one strictly one-time signature key pair with each *subj*
  - to DAPS-sign $(subj, m)$, sign m with key corresponding to *subj*
  - double-signing immediately reveals *subj*-specific signing key
- multi-encrypt DAPS signing key using one-time *sk*'s as keys
  - include ciphertexts in DAPS verification key
  - consequence: DAPS signing key recoverable from any double-signature
- compress DAPS signing key to a short value using PRF
  - instead of storing a long secret key, re-generate it from a short seed!

DAPS from strictly one-time signatures

Our DAPS construction

- associate one strictly one-time signature key pair with each *subj*
  - to DAPS-sign $(subj, m)$, sign m with key corresponding to *subj*
  - double-signing immediately reveals *subj*-specific signing key
- multi-encrypt DAPS signing key using one-time *sk*'s as keys
  - include ciphertexts in DAPS verification key
  - consequence: DAPS signing key recoverable from any double-signature
- compress DAPS signing key to a short value using PRF
  - instead of storing a long secret key, re-generate it from a short seed!

Signature and key sizes

size of ...	in **DeRaSI18**	in our construction
signatures	1280 bit	256 bit
signing keys	cardinality of subj space	256 bit
verification keys	cardinality of subj space	cardinality of subj space

Our DAPS in practice

Proc gen
00 $k \leftarrow_{\$} \{0,1\}^{\kappa}$
01 $vk[\cdot] \leftarrow \perp$; $K[\cdot] \leftarrow \perp$
02 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
03 For all $a \in \mathcal{A}$:
04 $\quad r \leftarrow F(k, a)$; $R \leftarrow g^r$
05 $\quad vk[a] \leftarrow R$
06 $\quad K[a] \leftarrow k + h(a, x, r)$
07 $sk \leftarrow k$
08 $vk \leftarrow (X, vk[\cdot], K[\cdot])$
09 Return (sk, vk)

Proc sgn(sk, m)
10 $k \leftarrow sk$; $(a, p) \leftarrow m$
11 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
12 $r \leftarrow F(k, a)$; $R \leftarrow g^r$
13 $c \leftarrow H(X, R, m)$
14 $\sigma \leftarrow r + xc$
15 Return σ

Proc vfy(vk, m, σ)
16 $(X, vk[\cdot], \_) \leftarrow vk$
17 $(a, p) \leftarrow m$
18 $R \leftarrow vk[a]$
19 $c \leftarrow H(X, R, m)$
20 Return $[RX^c \stackrel{?}{=} g^{\sigma}]$

Proc ext$(vk, m_1, \sigma_1, m_2, \sigma_2)$
21 $(X, vk[\cdot], K[\cdot]) \leftarrow vk$
22 $(a_1, p_1) \leftarrow m_1$
23 $(a_2, p_2) \leftarrow m_2$
24 Require $a_1 = a_2 \wedge p_1 \neq p_2$
25 $R \leftarrow vk[a_1]$
26 $c_1 \leftarrow H(X, R, m_1)$
27 $c_2 \leftarrow H(X, R, m_2)$
28 Require $c_1 \neq c_2$
29 $x \leftarrow (\sigma_1 - \sigma_2)/(c_1 - c_2)$
30 $r \leftarrow \sigma_1 - xc_1$
31 $k \leftarrow K[a_1] - h(a_1, x, r)$
32 $sk \leftarrow k$
33 Return sk

Our DAPS in practice

Proc gen
00 $k \leftarrow_{\$} \{0,1\}^{\kappa}$
01 $vk[\cdot] \leftarrow \perp$; $K[\cdot] \leftarrow \perp$
02 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
03 For all $a \in \mathcal{A}$:
04 $\quad r \leftarrow F(k,a)$; $R \leftarrow g^r$
05 $\quad vk[a] \leftarrow R$
06 $\quad K[a] \leftarrow k + h(a,x,r)$
07 $sk \leftarrow k$
08 $vk \leftarrow (X, vk[\cdot], K[\cdot])$
09 Return (sk, vk)

Proc sgn(sk, m)
10 $k \leftarrow sk$; $(a,p) \leftarrow m$
11 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
12 $r \leftarrow F(k,a)$; $R \leftarrow g^r$
13 $c \leftarrow H(X,R,m)$
14 $\sigma \leftarrow r + xc$
15 Return σ

Proc vfy(vk, m, σ)
16 $(X, vk[\cdot], \_) \leftarrow vk$
17 $(a,p) \leftarrow m$
18 $R \leftarrow vk[a]$
19 $c \leftarrow H(X,R,m)$
20 Return $[RX^c \stackrel{?}{=} g^{\sigma}]$

Proc ext$(vk, m_1, \sigma_1, m_2, \sigma_2)$
21 $(X, vk[\cdot], K[\cdot]) \leftarrow vk$
22 $(a_1, p_1) \leftarrow m_1$
23 $(a_2, p_2) \leftarrow m_2$
24 Require $a_1 = a_2 \wedge p_1 \neq p_2$
25 $R \leftarrow vk[a_1]$
26 $c_1 \leftarrow H(X,R,m_1)$
27 $c_2 \leftarrow H(X,R,m_2)$
28 Require $c_1 \neq c_2$
29 $x \leftarrow (\sigma_1 - \sigma_2)/(c_1 - c_2)$
30 $r \leftarrow \sigma_1 - xc_1$
31 $k \leftarrow K[a_1] - h(a_1,x,r)$
32 $sk \leftarrow k$
33 Return sk

Full paper at `https://eprint.iacr.org/2018/223`

Our DAPS in practice

Proc gen
00 $k \leftarrow_{\$} \{0,1\}^{\kappa}$
01 $vk[\cdot] \leftarrow \perp$; $K[\cdot] \leftarrow \perp$
02 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
03 For all $a \in \mathcal{A}$:
04 $\quad r \leftarrow F(k,a)$; $R \leftarrow g^r$
05 $\quad vk[a] \leftarrow R$
06 $\quad K[a] \leftarrow k + h(a,x,r)$
07 $sk \leftarrow k$
08 $vk \leftarrow (X, vk[\cdot], K[\cdot])$
09 Return (sk, vk)

Proc sgn(sk, m)
10 $k \leftarrow sk$; $(a,p) \leftarrow m$
11 $x \leftarrow F(k, \texttt{"x"})$; $X \leftarrow g^x$
12 $r \leftarrow F(k,a)$; $R \leftarrow g^r$
13 $c \leftarrow H(X,R,m)$
14 $\sigma \leftarrow r + xc$
15 Return σ

Proc vfy(vk, m, σ)
16 $(X, vk[\cdot], \_) \leftarrow vk$
17 $(a,p) \leftarrow m$
18 $R \leftarrow vk[a]$
19 $c \leftarrow H(X,R,m)$
20 Return $[RX^c \stackrel{?}{=} g^{\sigma}]$

Proc ext$(vk, m_1, \sigma_1, m_2, \sigma_2)$
21 $(X, vk[\cdot], K[\cdot]) \leftarrow vk$
22 $(a_1, p_1) \leftarrow m_1$
23 $(a_2, p_2) \leftarrow m_2$
24 Require $a_1 = a_2 \wedge p_1 \neq p_2$
25 $R \leftarrow vk[a_1]$
26 $c_1 \leftarrow H(X,R,m_1)$
27 $c_2 \leftarrow H(X,R,m_2)$
28 Require $c_1 \neq c_2$
29 $x \leftarrow (\sigma_1 - \sigma_2)/(c_1 - c_2)$
30 $r \leftarrow \sigma_1 - xc_1$
31 $k \leftarrow K[a_1] - h(a_1,x,r)$
32 $sk \leftarrow k$
33 Return sk

Full paper at `https://eprint.iacr.org/2018/223`