Key-Homomorphic Signatures

and Applications to Simulation Sound Extractable NIZK

David Derler, Graz University of Technology

March 22, 2017—Paderborn University

Based on joint work with Daniel Slamanig

Key-Homomorphic Signatures

Example 1

- Given two signatures σ_1 and σ_2 on m
- Valid under pk_1 and pk_2

$\Rightarrow$ Publicly compute σ' valid under $\mathsf{pk}' = \mathsf{pk}_1 \circ \mathsf{pk}_2$

Example 2

- Given a signature σ on m valid under pk
- Adapt σ to σ' valid under pk'
- Well defined relationship between pk and pk'

Extremely simple, yet very powerful!

- Never explicitly studied before

Related Work

Implicit usage in signatures schemes:

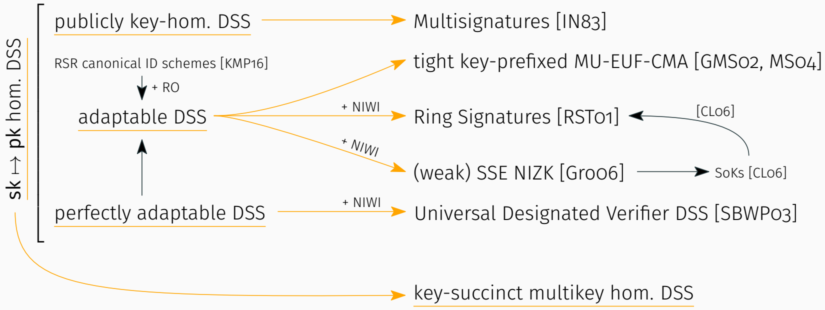
- Security under related (re-randomized) keys [BCM11,BPT12]
- DSS under randomizable keys [FKM+16]
- DSS from canonical identification schemes [FF13,KMP16]

Other directions:

- Key-homomorphic encryption [AHI11, Rot11, BGG+14, DMS16]
- (Constrained) key-homomorphic PRFs [BLMR13, BP14, BFP+15]
- Key-homomorphic projective hashes [BJL16, BJL17]

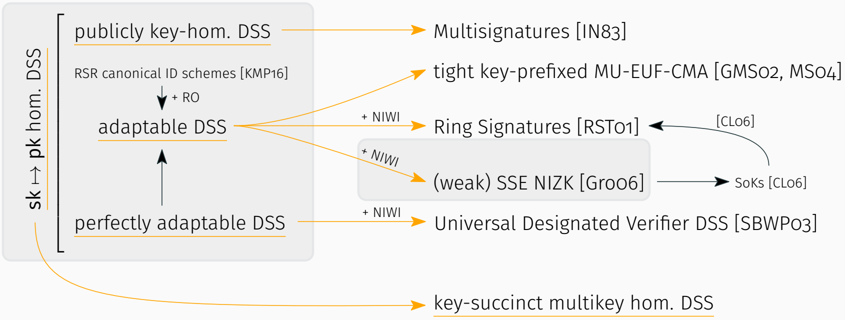
Outline

Results in [DS16]



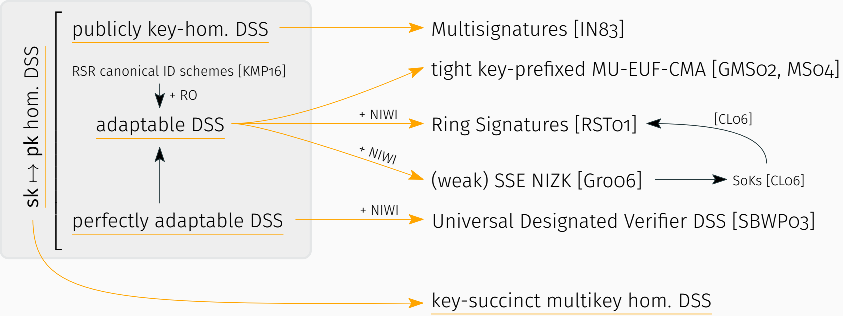
Outline

Covered in this talk



Outline

Variants of key-homomorphisms



Secret-Key-to-Public-Key Homomorphism

Secret keys and public keys live in groups $\mathbb{G}$ and $\mathbb{H}$

- Group homomorphism $\mu : \mathbb{H} \to \mathbb{G}$

$$\forall\, \mathsf{sk}, \mathsf{sk}' \in \mathbb{H} \;:\; \mu(\mathsf{sk} + \mathsf{sk}') = \mu(\mathsf{sk}) + \mu(\mathsf{sk}')$$

- For all $(\mathsf{sk}, \mathsf{pk}) \leftarrow \mathsf{KeyGen}(1^\kappa)$ it must hold that

$$\mathsf{pk} = \mu(\mathsf{sk})$$

Keys may also be vectors:

- Straightforward extension
- Not made explicit for compactness

Φ^+-Key-Homomorphic Signatures

Class of functions Φ^+

- Representing linear shifts
- Functions identified by "shift amount" Δ

Conventional signature scheme

\+ Secret-key-to-public-key homomorphism

\+ Additional PPT algorithm

$$(\mathsf{pk}', \sigma') \leftarrow \mathsf{Adapt}(\mathsf{pk}, m, \sigma, \Delta)$$

$\Rightarrow$ Shift signature from pk to $\mathsf{pk}' = \mathsf{pk} \cdot \mu(\Delta)$

Question: possible to have Adapt' taking $\mu(\Delta)$ instead of Δ?

- No: efficient Adapt' would imply an UUF-NMA adversary

Adaptability of Signatures

Identical distribution of fresh and adapted signatures

- "Initial" signature on m under sk not revealed

$$\boxed{\mathsf{Adapt}(\mathsf{pk}, m, \mathsf{Sign}(\mathsf{sk}, m), \Delta)} \qquad \boxed{(\mathsf{pk} \cdot \mu(\Delta), \mathsf{Sign}(\mathsf{sk} + \Delta, m))}$$

Perfect Adaption of Signatures

Identical distribution of fresh and adapted signatures

- Even when seeing the initial signatures $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}, m)$

$$\boxed{(\sigma, \mathsf{Adapt}(\mathsf{pk}, m, \sigma, \Delta))} \qquad \boxed{(\sigma, \mathsf{pk} \cdot \mu(\Delta), \mathsf{Sign}(\mathsf{sk} + \Delta, m))}$$

Publicly Key-Homomorphic Signatures

Conventional signature scheme with

- Secret-key-to-public-key homomorphism
- \+ Additional algorithm **Combine**

$$(\hat{\mathsf{pk}}, \hat{\sigma}) \leftarrow \mathsf{Combine}((\mathsf{pk}_i)_{i=1}^n, m, (\sigma_i)_{i=1}^n)$$

- Combine multiple signatures on m under distinct keys
- Combined key $\hat{\mathsf{pk}} = \prod_{i=1}^n \mathsf{pk}_i$

Examples of Key-Homomorphic Schemes

Publicly key-homomorphic schemes

- BLS signatures [BLS04]
- CL signature variant [CHP12]
- Waters' signatures with shared params [Wat05, BFG13]

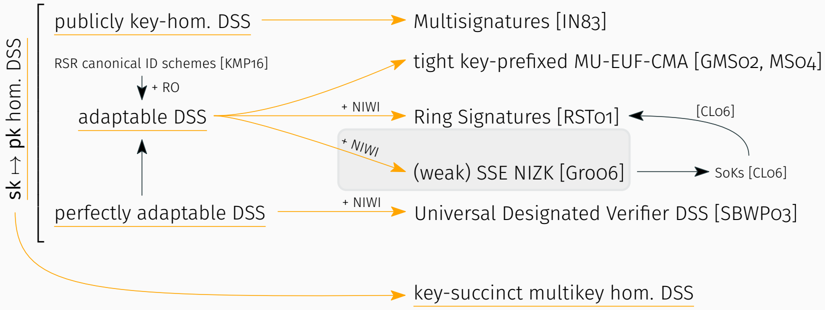
Adaptable schemes

- Schnorr signatures [Sch91]
- Katz-Wang signatures [KW03, GJKW07]

Perfectly adaptable schemes

- BLS signatures [BLS04]
- CL signature variant [CHP12]
- Waters' signatures with shared params [Wat05, BFG13]
- Pointcheval-Sanders signatures [PS16]

Application to SSE NIZK



Non-Interactive Proof Systems

NP-language L w.r.t. relation R

- $x \in L \iff \exists\, w \,:\, (x, w) \in R$

Non-interactive proof system

$(x, w) \in R$

$\pi \leftarrow \mathsf{Proof}(x, w)$

Proof π $\longrightarrow$

$x \stackrel{?}{\in} L$

$\checkmark / \times \leftarrow \mathsf{Verify}(x, \pi)$

Non-Interactive Proof Systems - Properties

Completeness

- Honestly computed proof for $(x, w) \in R$ will always verify

Soundness

- Infeasible to produce valid proof for $x \notin L$

Extractability

- Stronger variant of soundness
- Extract witness from valid proof (using trapdoor)

Non-Interactive Proof Systems - Properties contd.

Witness Indistinguishability (WI)

- Distinguish proofs for same x w.r.t. different w, w'

Zero-Knowledge (ZK)

- Stronger variant of witness indistinguishability
- Simulate proofs without knowing w (using trapdoor)

Simulation Sound Extractability (SSE)

Very strong notion

- Combination of zero-knowledge and extractability
- Adversary sees simulated proofs for arbitrary x
- Still infeasible to forge proof for new $x \notin L$

$\Rightarrow$ Can extract witness for any proof with "new" x

Requires non-malleability

weak SSE: non-malleability w.r.t. proven statement

SSE: non-malleability w.r.t. proof and proven statement

Construction Idea - weak SSE

Extend proof with adaptable signature

- Sign proof under a random key pk'
- Include signature on proven statement under pk'

Extend Language L to L'

- Include pk in CRS

$$x \in L \iff \exists\, w : (x, w) \in R$$

$$(x, \mathsf{pk}, \mathsf{pk}') \in L' \iff \exists\, w : (x, w) \in R \;\vee\; \exists\, \Delta : \mathsf{pk} = \mathsf{pk}' \cdot \mu(\Delta)$$

$\Rightarrow$ Shift amount allows to extract valid signature under pk

Can construct an SSE NIZK proof system for L

- From extractable WI proof system for L'
- \+ Adaptable signature scheme

Construction Idea - weak SSE - Security

Observations

- Either need witness for 1st or 2nd literal in OR clause

$$(x, w) \in R \;\vee\; \exists\, \Delta \;:\; \mathsf{pk} = \mathsf{pk}' \cdot \mu(\Delta)$$

- Prover has to use w s.t. $(x, w) \in R$

$\Rightarrow$ Otherwise needs signature under pk

Zero-Knowledge

- Set up CRS so that sk for pk is known
- Create signature σ on proof under sk
- Shift σ to σ' under random key pk′
- Create proof using “shift amount” Δ

$\Rightarrow$ Cannot be detected under WI

Weak Simulation Sound Extractability

- Simulate as before
  - Use pk from EUF-CMA challenger
  - Obtain signatures via Sign oracle
- Use extractor from underlying proof system
- EUF-CMA implies extraction of w s.t. $(x, w) \in R$

Simulation Sound Extractability

- Additionally use strong one-time DSS

Instantiation Example

Recall Waters' variant in SXDH setting [BFG13]

- Public parameters $U = (u_0, \dots, u_n) \xleftarrow{R} \mathbb{G}_1^n$
- Waters' hash $H(m) := u_0 \prod_{i \in [n]} u_i^{m_i}$ where $m \in \{0,1\}^m$
- Public key: $\mathsf{sk} \leftarrow g^x \in \mathbb{G}_1$, $\mathsf{pk} \leftarrow \hat{g}^x \in \mathbb{G}_2$, with $x \xleftarrow{R} \mathbb{Z}_p$
- Signature: $\sigma \leftarrow (g^x \cdot H(m)^r, g^r, \hat{g}^r)$, with $r \xleftarrow{R} \mathbb{Z}_p$
- Verification via pairing

Perfect adaptability

- Signature $\sigma = (\sigma_1, \sigma_2, \sigma_3)$
- Shift amount $\Delta = (\Delta_1, \Delta_2) \in \mathbb{G}_1 \times \mathbb{G}_2$
- With $e(\Delta_1, \hat{g}) = e(g, \Delta_2)$
- Shift $(\mathsf{pk}', \sigma') \leftarrow (\hat{g}^x \cdot \Delta_2, (\sigma_1 \cdot \Delta_1 \cdot H(m)^{r'}, \sigma_2 \cdot g^{r'}, \sigma_3 \cdot \hat{g}^{r'}))$

Combination with Groth-Sahai proofs

- Only requires to prove knowledge of single element in $\mathbb{G}_1$!

Conclusions

Generic compilers for various signature variants

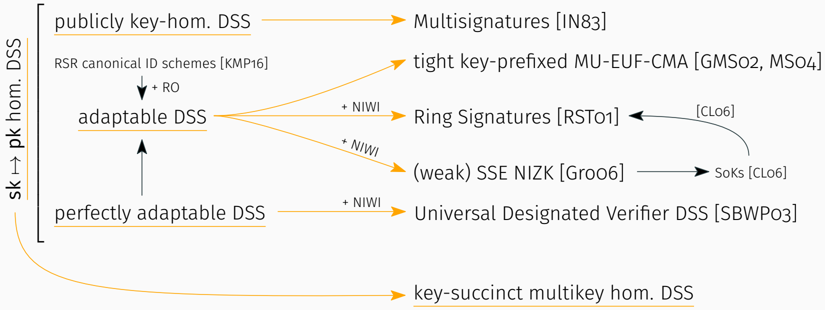
- Applicable to large classes of schemes
- Strong security guarantees from very mild requirements
- Extremely simple
- Favorable regarding efficiency w.r.t. previous schemes
- Deeper understanding of construction paradigms

Directly yields novel instantiations

- Comparing favorably to existing work
- Standard model & assumptions: Waters' sigs + GS proofs

Conclusions

Results in [DS16]



Thank you.

Preprint available as IACR ePrint Archive Report 2016/792

david.derler@iaik.tugraz.at @dderler