Rethinking Privacy for Extended Sanitizable Signatures

and a Black-Box Construction of Strongly Private Schemes

David Derler and Daniel Slamanig,
IAIK, Graz University of Technology, Austria

November 26, 2015

Supported by EU H2020 Project prisma cloud

Outline

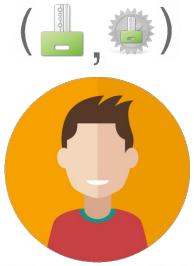
1. Introduction
2. Revisiting Privacy
3. Generic Construction
4. Conclusions

Outline

1. Introduction
2. Revisiting Privacy
3. Generic Construction
4. Conclusions

Sanitizable Signature Schemes (SSS) [ACdMT05]

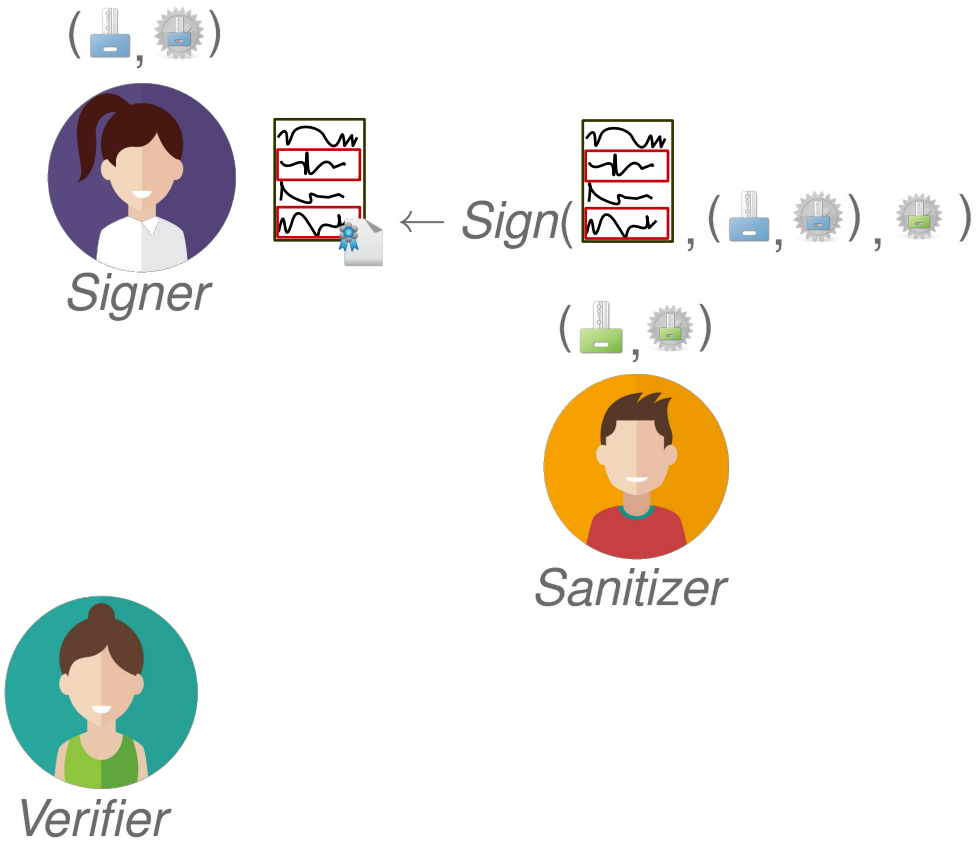
Signer



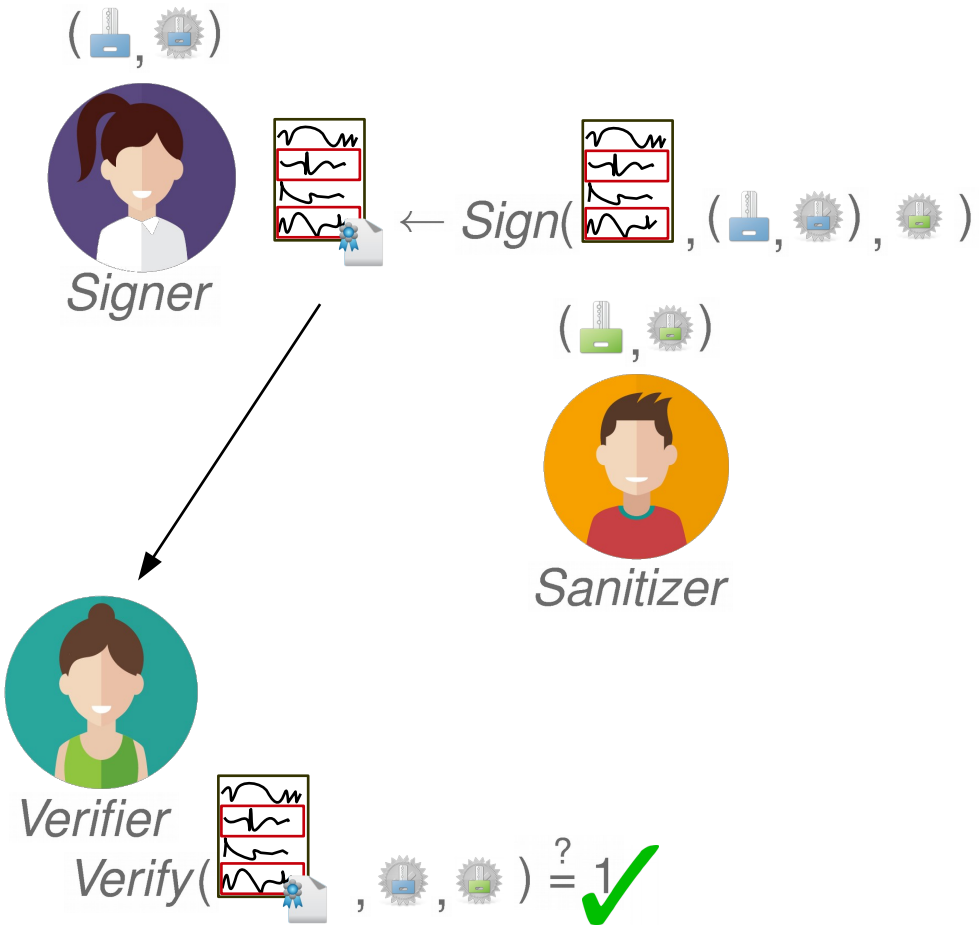
Sanitizer

Verifier

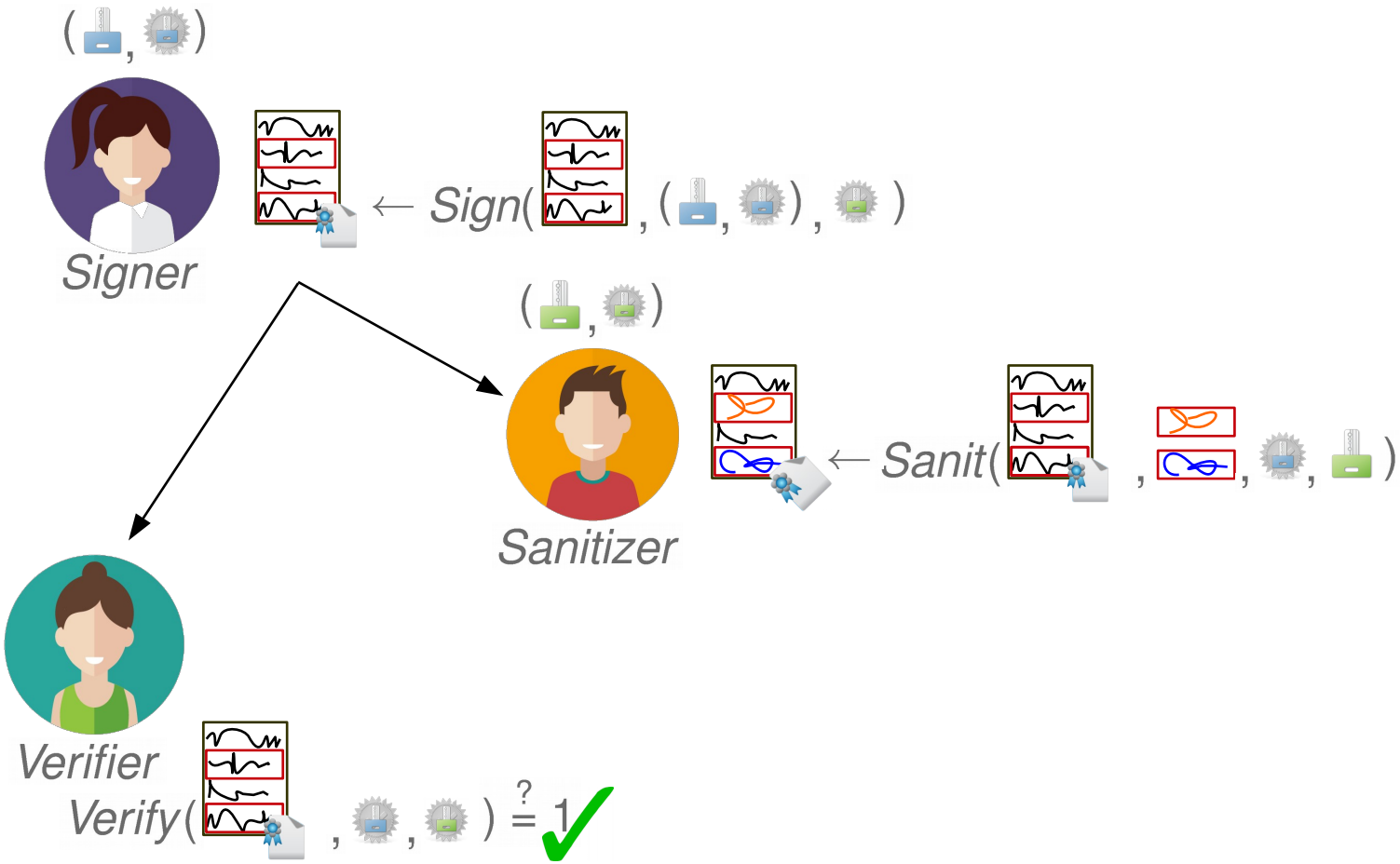
Sanitizable Signature Schemes (SSS) [ACdMT05]



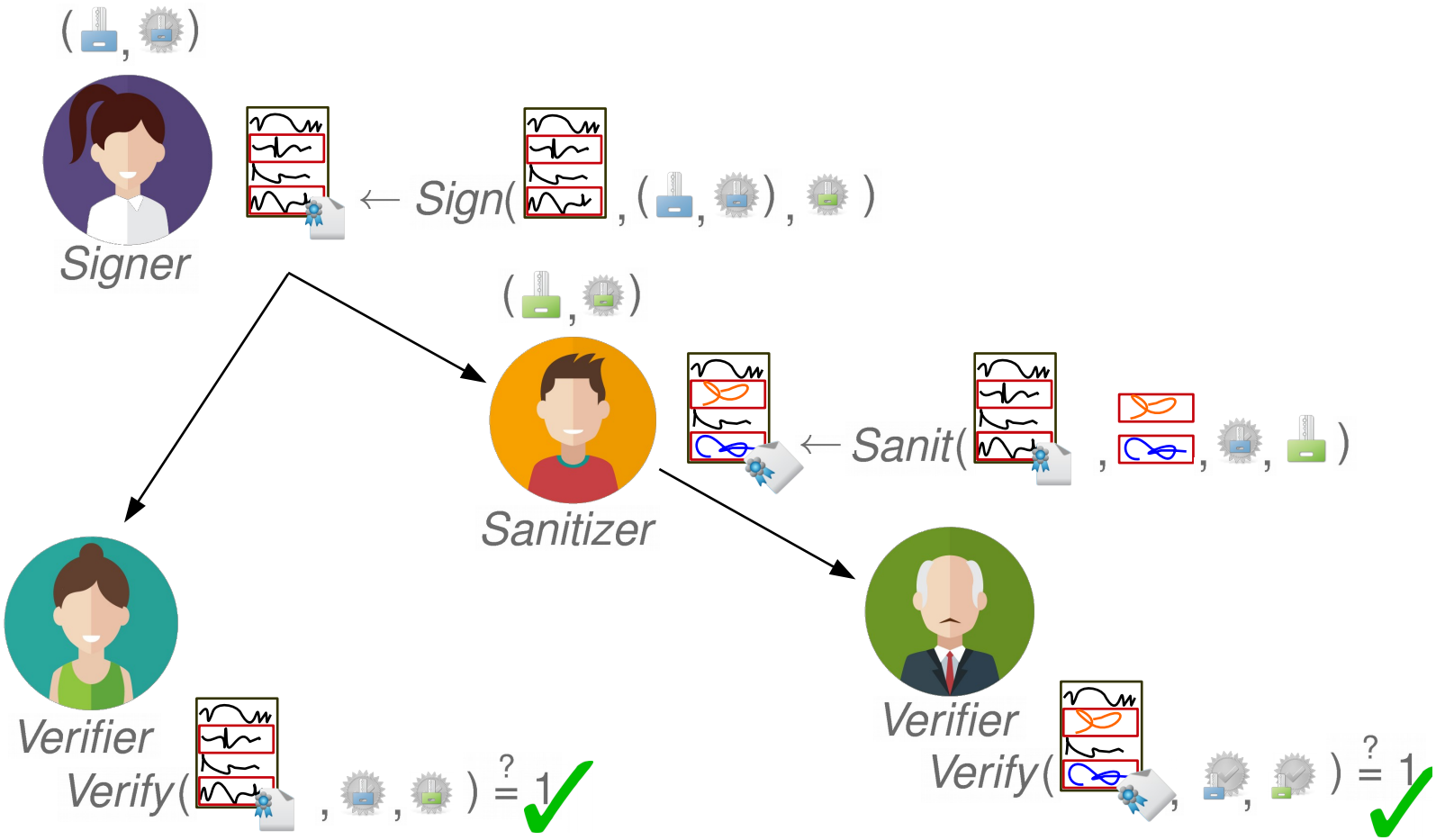
Sanitizable Signature Schemes (SSS) [ACdMT05]



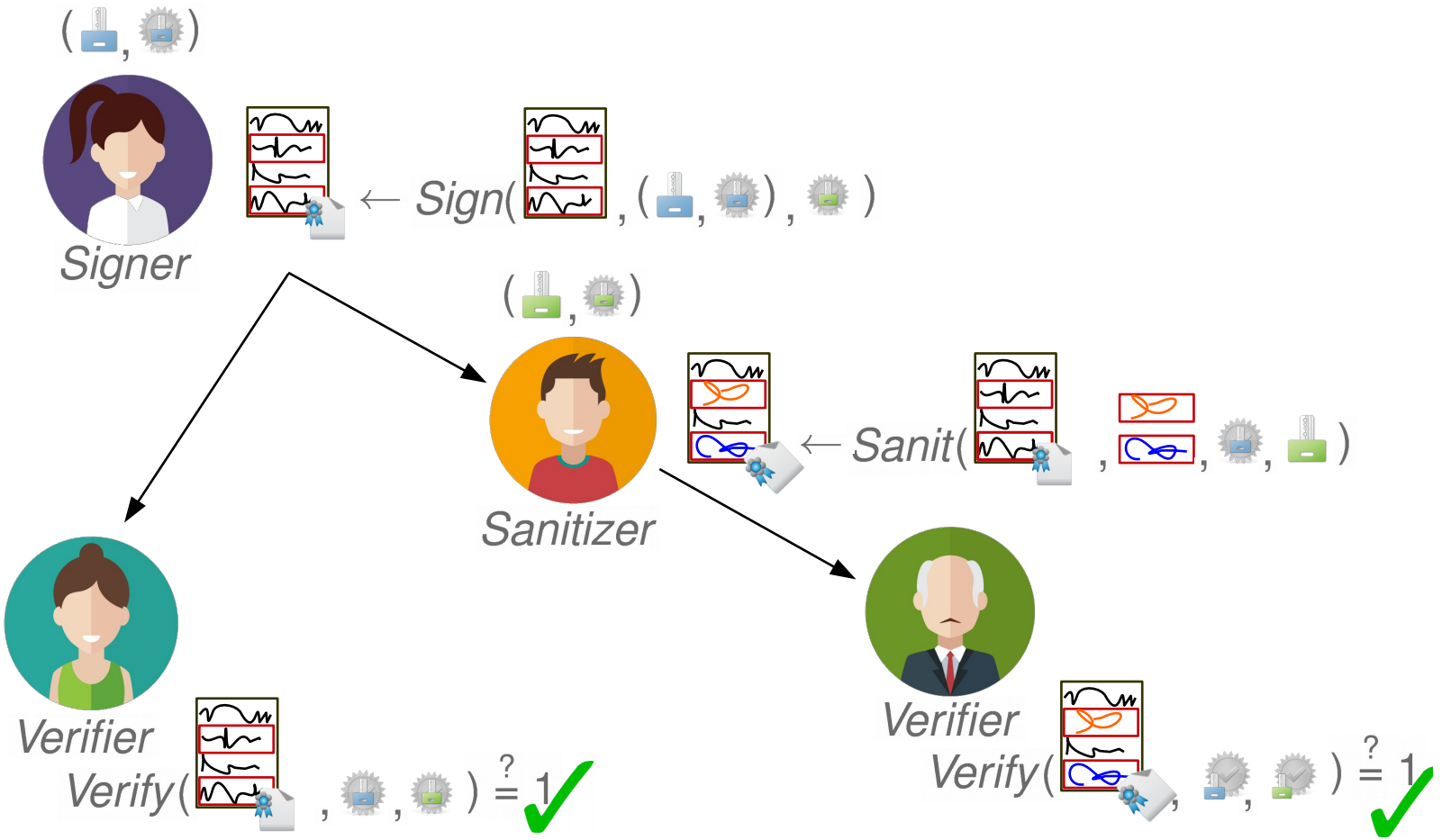
Sanitizable Signature Schemes (SSS) [ACdMT05]



Sanitizable Signature Schemes (SSS) [ACdMT05]



Sanitizable Signature Schemes (SSS) [ACdMT05]



Proof / Judge : original signature of signer or sanitized

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Immutability

- Malicious sanitizer cannot modify fixed message parts

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Immutability

- Malicious sanitizer cannot modify fixed message parts

Privacy

- **Sanitized information not recoverable**

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Immutability

- Malicious sanitizer cannot modify fixed message parts

Privacy

- **Sanitized information not recoverable**

Transparency

- Signatures of signer and sanitizer are indistinguishable

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Immutability

- Malicious sanitizer cannot modify fixed message parts

Privacy

- **Sanitized information not recoverable**

Transparency

- Signatures of signer and sanitizer are indistinguishable

Accountability

- Malicious signers/sanitizers unable to deny authorship

Security of SSS [ACdMT05]

Correctness, Unforgeability

- Straight forward

Immutability

- Malicious sanitizer cannot modify fixed message parts

Privacy

- **Sanitized information not recoverable**

Transparency

- Signatures of signer and sanitizer are indistinguishable

Accountability

- Malicious signers/sanitizers unable to deny authorship

Formalized in [BFF+09]

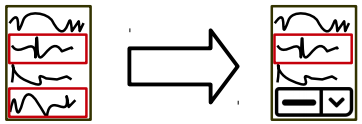
Arbitrary replacements?

- Sanitizer often **too powerful**
- **Limited expressiveness** of signed messages
- **Limited applicability** in several scenarios

Extensions

Several extensions proposed in [KL06]

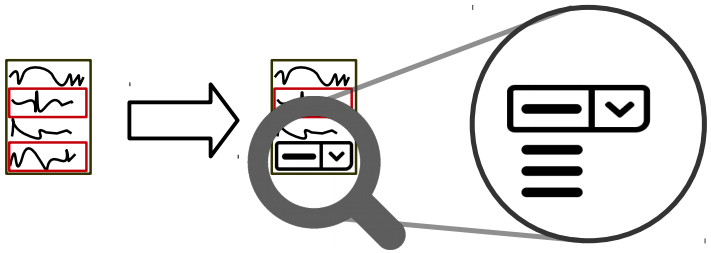
$\Rightarrow$ `LimitSet` feature



Extensions

Several extensions proposed in [KL06]

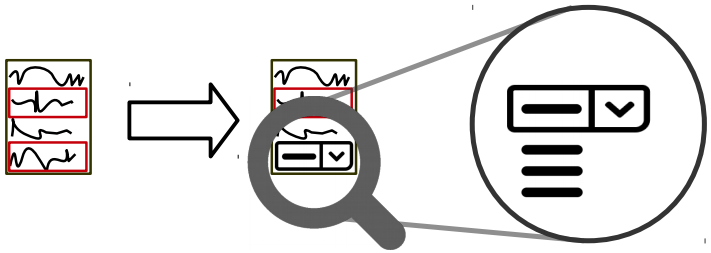
$\Rightarrow$ `LimitSet` feature



Extensions

Several extensions proposed in [KL06]

$\Rightarrow$ `LimitSet` feature

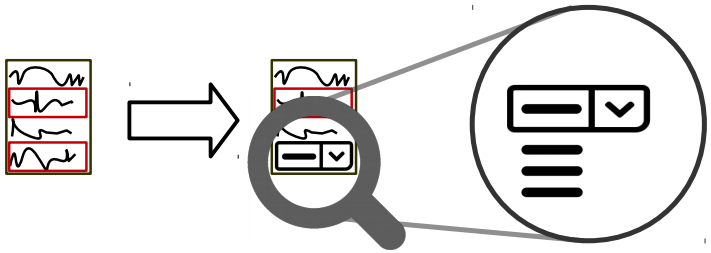


- Extended Sanitizable Signature Schemes (ESSS)

Extensions

Several extensions proposed in [KL06]

$\Rightarrow$ `LimitSet` feature



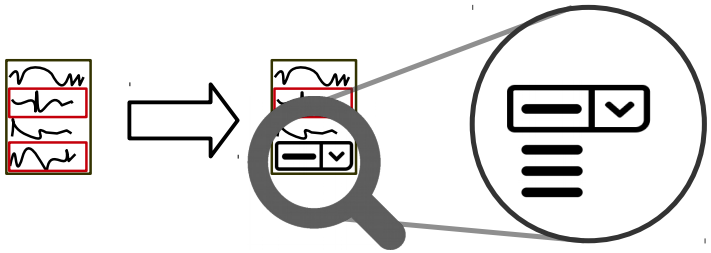
- Extended Sanitizable Signature Schemes (ESSS)

- Non-privacy-related features
  - Same changes in linked blocks
  - Modify at most *k* out of *n* blocks
  - Restrict number “versions” of a message

Extensions

Several extensions proposed in [KL06]

$\Rightarrow$ `LimitSet` feature



- Extended Sanitizable Signature Schemes (ESSS)

- Non-privacy-related features
  - Same changes in linked blocks
  - Modify at most k out of n blocks
  - Restrict number “versions” of a message

Unfortunately, no formal definitions.

Motivation

`LimitSet` useful tool in many applications

- Restrict power of sanitizer
- Automated processing: improve data quality

Motivation

`LimitSet` useful tool in many applications

- Restrict power of sanitizer
- Automated processing: improve data quality

Later formalized in [CJ10]

- Privacy not defined in original sense
- Recovery of admissible sets possible
  - Private scheme can leak all admissible sets

Motivation

`LimitSet` useful tool in many applications

- Restrict power of sanitizer
- Automated processing: improve data quality

Later formalized in [CJ10]

- Privacy not defined in original sense
- Recovery of admissible sets possible
  - Private scheme can leak all admissible sets

Proper formalization **important**!

Is a weak privacy notion a problem?

Doctor signs medical records

- Patient replaces sensitive information
- with less sensitive information

Is a weak privacy notion a problem?

Doctor signs medical records

- Patient replaces sensitive information
- with less sensitive information

Bank signs authorized financial transactions

- Enterprise reveals subset
- Sensitive transactions replaced by $\perp$

Is a weak privacy notion a problem?

Doctor signs medical records

- Patient replaces sensitive information
- with less sensitive information

Bank signs authorized financial transactions

- Enterprise reveals subset
- Sensitive transactions replaced by $\perp$

Sanitized documents published

$\Rightarrow$ **Verifiers may learn sensitive information!**

Contribution

Stronger privacy notion for ESSS: **strong privacy**

- Captures privacy in original sense
- Allows practically efficient instantiations

Contribution

Stronger privacy notion for ESSS: **strong privacy**

- Captures privacy in original sense
- Allows practically efficient instantiations
- **Relations** to existing privacy notions

Contribution

Stronger privacy notion for ESSS: **strong privacy**

- Captures privacy in original sense
- Allows practically efficient instantiations
- **Relations** to existing privacy notions

Practically efficient **generic** construction of ESSS

- From any SSS
  - Secure in the established model of SSS [BFF+09]
- ... and cryptographic accumulators

Contribution

Stronger privacy notion for ESSS: **strong privacy**

- Captures privacy in original sense
- Allows practically efficient instantiations
- **Relations** to existing privacy notions

Practically efficient **generic** construction of ESSS

- From any SSS
  - Secure in the established model of SSS [BFF+09]
- ... and cryptographic accumulators
- Strongly private if accu is indistinguishable [DHS15]

Outline

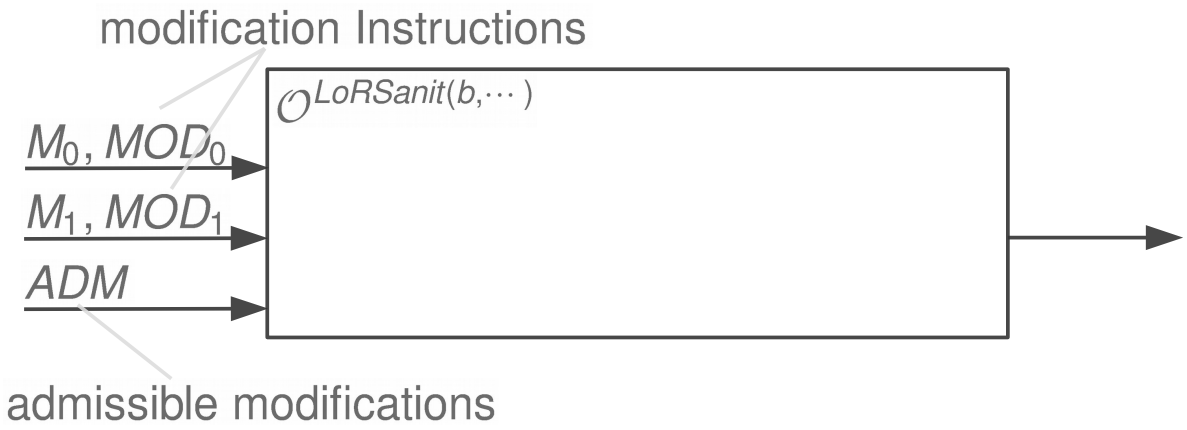
1. Introduction
2. Revisiting Privacy
3. Generic Construction
4. Conclusions

Original Privacy Definition [BFF+09]

Indistinguishability-based notion (left-or-right oracle)

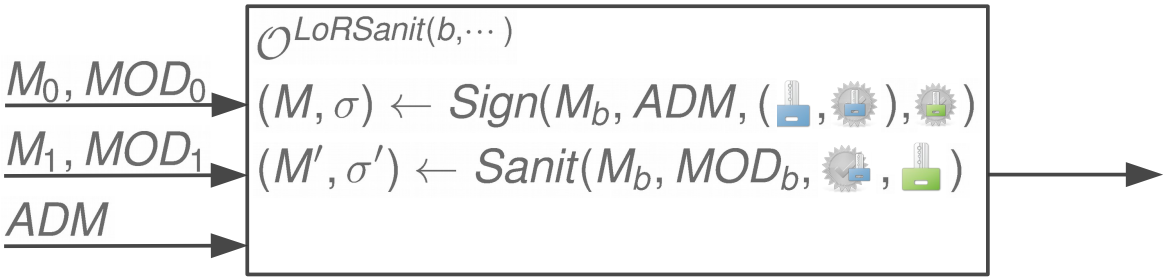
Original Privacy Definition [BFF+09]

Indistinguishability-based notion (left-or-right oracle)



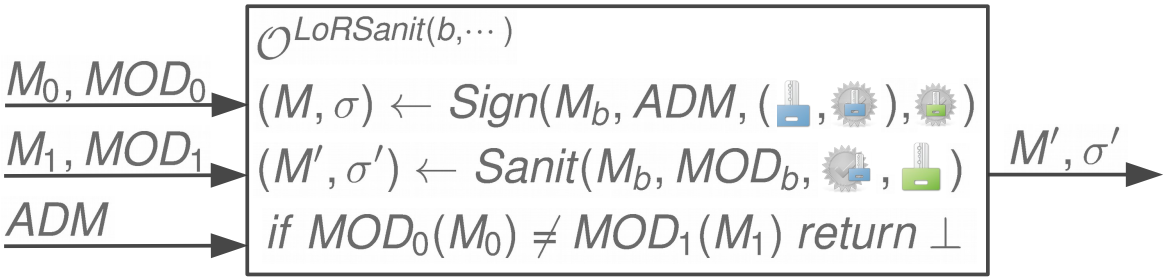
Original Privacy Definition [BFF+09]

Indistinguishability-based notion (left-or-right oracle)



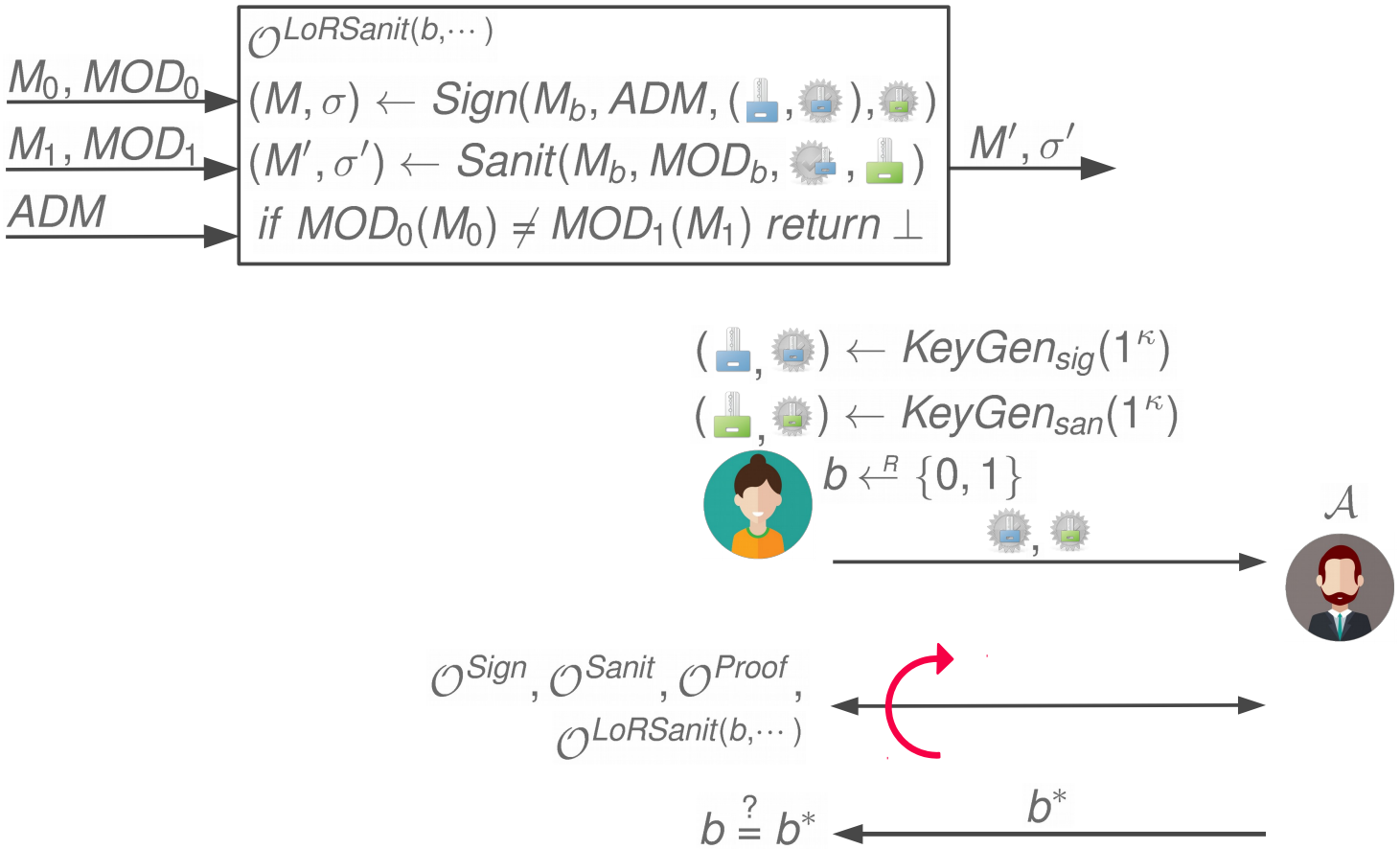
Original Privacy Definition [BFF+09]

Indistinguishability-based notion (left-or-right oracle)



Original Privacy Definition [BFF+09]

Indistinguishability-based notion (left-or-right oracle)



Privacy for ESSS

`LimitSet` $\Rightarrow$ additionally need to specify set limitations

- for `LimitSet` blocks

Privacy for ESSS

`LimitSet` $\Rightarrow$ additionally need to specify set limitations

- for `LimitSet` blocks

[CJ10] modified $\mathcal{O}^{LoRSanit}$

- Random set limitations are internally chosen
- Compatible with initially submitted messages,
- and with both sanitized messages.

Privacy for ESSS

`LimitSet` $\Rightarrow$ additionally need to specify set limitations

- for `LimitSet` blocks

[CJ10] modified $\mathcal{O}^{LoRSanit}$

- Random set limitations are internally chosen
- Compatible with initially submitted messages,
- and with both sanitized messages.

$\Rightarrow$ Set limitations independent of challenge bit

Privacy for ESSS

`LimitSet` $\Rightarrow$ additionally need to specify set limitations

- for `LimitSet` blocks

[CJ10] modified $\mathcal{O}^{LoRSanit}$

- Random set limitations are internally chosen
- Compatible with initially submitted messages,
- and with both sanitized messages.

$\Rightarrow$ Set limitations independent of challenge bit

Possible motivation for weak formalization?

- Preserves implication of privacy by transparency [BFF+09]

Adapt Unlinkability to ESSS?

Stronger privacy notion

- Modify $\mathcal{O}^{LoR}$:
  - Submit two message-signature pairs
  - ... together with modification instructions,
  - such that modified messages are equivalent.

Adapt Unlinkability to ESSS?

Stronger privacy notion

- Modify $\mathcal{O}^{LoR}$:
  - Submit two message-signature pairs
  - ... together with modification instructions,
  - such that modified messages are equivalent.
- Return sanitized version of (M_b, σ_b)

$\Rightarrow$ Wins if it correctly guesses b

Adapt Unlinkability to ESSS?

Stronger privacy notion

- Modify $\mathcal{O}^{LoR}$:
  - Submit two message-signature pairs
  - ... together with modification instructions,
  - such that modified messages are equivalent.
- Return sanitized version of (M_b, σ_b)

$\Rightarrow$ Wins if it correctly guesses b

Set limitations required to remain hidden

- **No** practically efficient instantiations

Adapt Unlinkability to ESSS?

Stronger privacy notion

- Modify $\mathcal{O}^{LoR}$:
  - Submit two message-signature pairs
  - ... together with modification instructions,
  - such that modified messages are equivalent.
- Return sanitized version of (M_b, σ_b)

$\Rightarrow$ Wins if it correctly guesses b

Set limitations required to remain hidden

- **No** practically efficient instantiations

We look for notion **between** privacy and unlinkability!

Introducing Strong Privacy

Extension of privacy

- Additionally submit set limitations per message
- Final sanitized messages must be equivalent

Introducing Strong Privacy

Extension of privacy

- Additionally submit set limitations per message
- Final sanitized messages must be equivalent

Then, also set limitations depend on bit *b*

$\Rightarrow$ Signature is required to hide set limitations

Introducing Strong Privacy

Extension of privacy

- Additionally submit set limitations per message
- Final sanitized messages must be equivalent

Then, also set limitations depend on bit *b*

$\Rightarrow$ Signature is required to hide set limitations

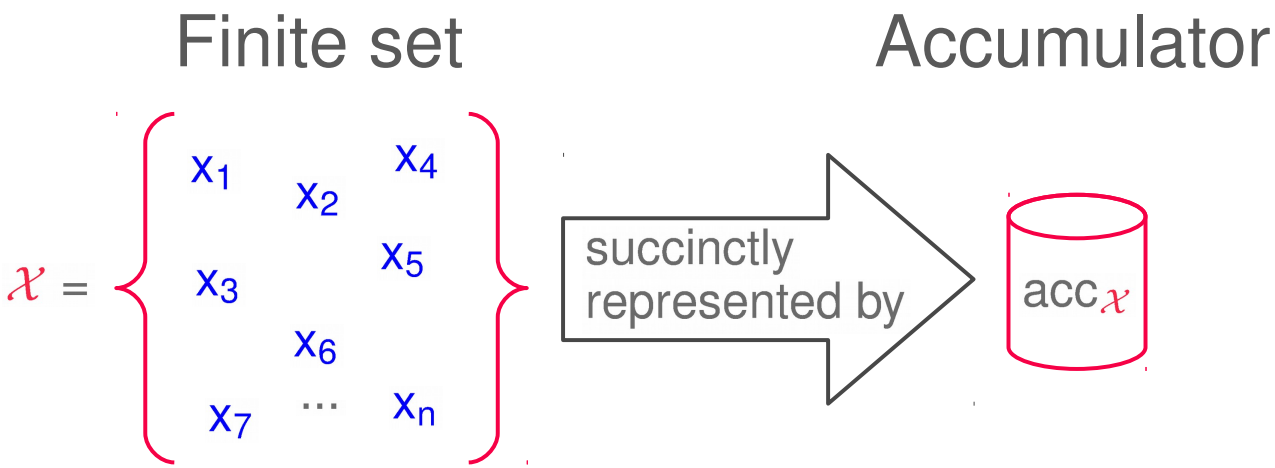
We show that

- Privacy **strictly weaker** than strong privacy
- (Strong) unlinkability **strictly stronger** than strong privacy

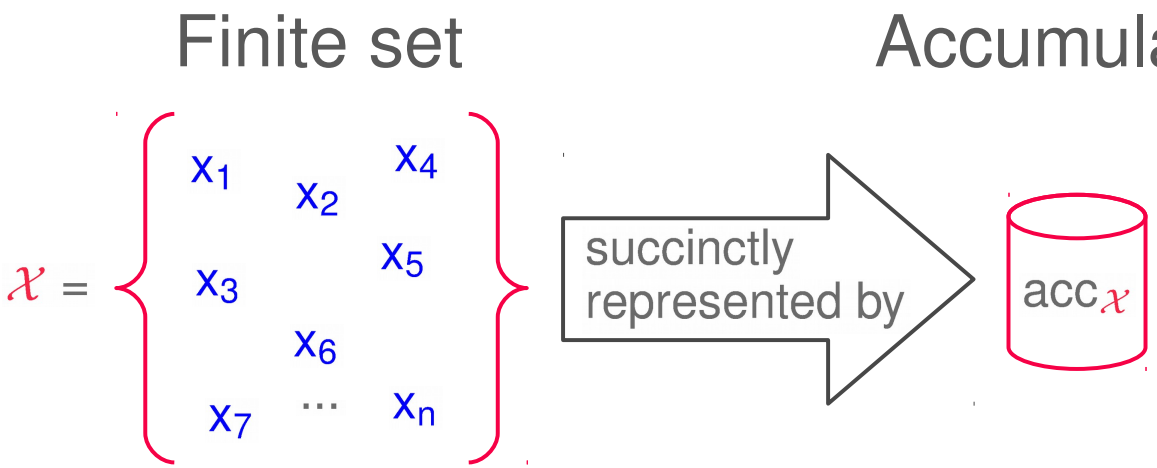
Outline

1. Introduction
2. Revisiting Privacy
3. Generic Construction
4. Conclusions

Indistinguishable Accumulators



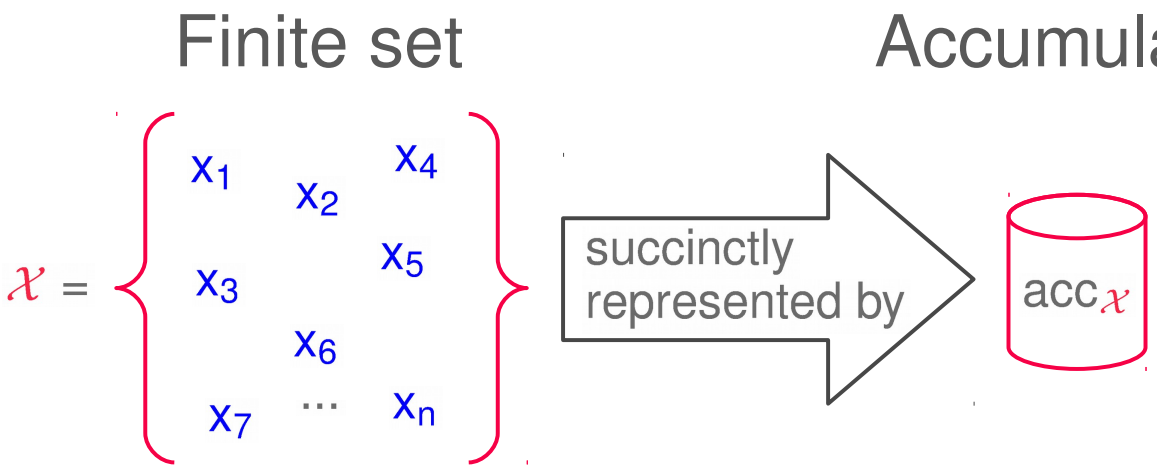
Indistinguishable Accumulators



Witnesses wit_x certifying membership of x in $acc_{\mathcal{X}}$

- Efficiently computable $\forall\ x \in \mathcal{X}$, intractable $\forall\ x \notin \mathcal{X}$

Indistinguishable Accumulators



Witnesses wit_x certifying membership of x in $acc_{\mathcal{X}}$

- Efficiently computable $\forall\ x \in \mathcal{X}$, intractable $\forall\ x \notin \mathcal{X}$

Indistinguishability

- Neither accu nor witnesses leak information about $\mathcal{X}$

Existing Constructions Supporting `LimitSet`

Problems of existing constructions

- Tailored to specific instantiations
- Meaningful indistinguishability notion very recent [DHS15]
  - Not considered in existing constructions
  - Plain accumulators not required to hide accumulated set

  $\Rightarrow$ **No** strong privacy!

Existing Constructions Supporting `LimitSet`

Problems of existing constructions

- Tailored to specific instantiations
- Meaningful indistinguishability notion very recent [DHS15]
  - Not considered in existing constructions
  - Plain accumulators not required to hide accumulated set

  $\Rightarrow$ **No** strong privacy!

Existing constructions follow paradigm

- We show that this paradigm is generally applicable

How does the Extension Work?

For each `LimitSet` block

- Include actually chosen message as variable element
- Accumulate sets of admissible changes
- Include accumulators as additional fixed elements
- Include witnesses in the signature

How does the Extension Work?

For each `LimitSet` block

- Include actually chosen message as variable element
- Accumulate sets of admissible changes
- Include accumulators as additional fixed elements
- Include witnesses in the signature

Verification

- Conventional verification
- \+ Accumulator membership for `LimitSet` blocks
- Unambiguous encoding required!

Security I

Correctness

- Correctness of underlying primitives

Unforgeability

- Unforgeability of underlying SSS

Immutability

- Immutability of underlying SSS
- Collision freeness of accumulator

Security II

Privacy, Transparency

- Privacy, Transparency of underlying SSS

Accountability

- Accountability of underlying SSS

Security II

Privacy, Transparency

- Privacy, Transparency of underlying SSS

Accountability

- Accountability of underlying SSS

Strong Privacy

Holds if

- `LimitSet` instantiated with indistinguishable accumulator
- Underlying SSS is private

Outline

1. Introduction
2. Revisiting Privacy
3. Generic Construction
4. Conclusions

Conclusions

Strengthened privacy notion

- Strong enough for many applications
- Allows particularly efficient instantiations

Conclusions

Strengthened privacy notion

- Strong enough for many applications
- Allows particularly efficient instantiations

Relation of **strong privacy** to other privacy notions

Conclusions

Strengthened privacy notion

- Strong enough for many applications
- Allows particularly efficient instantiations

Relation of **strong privacy** to other privacy notions

Generic construction of ESSS

- Providing strong privacy
- Obtain practically efficient implementations with low effort
  - ...by building upon existing schemes

Thank you.

david.derler@iaik.tugraz.at

Full version: http://eprint.iacr.org/2015/843

Supported by prisma cloud

References I

[ACdMT05] Giuseppe Ateniese, Daniel H. Chou, Breno de Medeiros, and Gene Tsudik. Sanitizable signatures. In *ESORICS'05*, LNCS, 2005.

[BFF+09] Christina Brzuska, Marc Fischlin, Tobias Freudenreich, Anja Lehmann, Marcus Page, Jakob Schelbert, Dominique Schröder, and Florian Volk. Security of sanitizable signatures revisited. In *PKC'09*, LNCS, 2009.

[CJ10] Sébastien Canard and Amandine Jambert. On extended sanitizable signature schemes. In *CT-RSA'10*, LNCS, 2010.

[DHS15] David Derler, Christian Hanser, and Daniel Slamanig. Revisiting cryptographic accumulators, additional properties and relations to other primitives. In *CT-RSA'15*, LNCS, 2015.

References II

[KL06] Marek Klonowski and Anna Lauks. Extended sanitizable signatures. In *ICISC'06*, LNCS, 2006.

Avatars designed by Freepic.