Policy-Based Sanitizable Signatures

Kai Samelin* and Daniel Slamanig** (presented by David Pointcheval)

*TÜV Rheinland i-sec GmbH
**AIT Austrian Institute of Technology
Outline

- Digital Signatures
- Sanitizable Signatures
- Policy-Based Sanitizable Signatures
- Conclusions & Take Home
Digital Signatures
Digital Signatures

Secret key

Signer

Public key

Modifications invalidate signature
Controlled Modifications of Signed Messages

• Modify signed messages without invalidating the signature?
  – But that’s what we try to prevent?!
  – Can be useful if controlled!

• Controlled modifications
  – Signer determines how signed message can be altered
  – Think of implicitly signing all possible messages

• Control who is allowed to modify
  – Signer specifies entity allowed to perform modifications
Example: Medical Documents

- Tokenization for research/accounting
- Removing exact diagnosis for sick leave
- Etc.

Re-signing after the fact might not be possible (availability, etc.)
Different Types of Schemes

- **Redactable Signatures**
  - *Blacking out/Removal* of designated parts by *everyone*

- **Sanitizable Signatures**
  - *Replacement* of designated parts by designated entity
  - Designated entity (=sanitizer) has its own key pair
Sanitizable Signatures
Sanitizable Signatures

Specify modifications and sanitizer

Sanitizer

Accept allowed modifications

Signer
Sanitizable Signatures

Signer

Sanitizer

Specify modifications and sanitizer

Reject disallowed modifications
Sanitizable Signatures

Generate proof that signature from signer or sanitizer
Security Properties

- **Unforgeability**
  - Non-sanitizers cannot come up with valid signature for non-signed message

- **Immutability**
  - Sanitizer cannot come up with valid signature for a message not “derivable” from signed ones

- **Privacy**
  - No information about sanitized parts can be learned

- **Signer/Sanitizer accountability**
  - Signer/sanitizer cannot blame the other party for having produced a signature

- **Transparancy**
  - Freshly signed and sanitized signatures are indistinguishable
Construction Idea I/II

- Originally proposed in [ACMT, ESORICS’05] and rigorous security model in [BFFLP+, PKC’09]
- Generic construction from secure signatures and chameleon-hash functions in [BFFLP+, PKC’09]

- Chameleon Hash [KT, NDSS’00]: Collision-resistant hash keyed with (sk, pk)
  - Hashing: $h \leftarrow \text{CHash}(pk, m; r)$
  - Collision: sk allows for any $h, m'$ to compute $r'$ s.t.

  \[
  \text{CHash}(pk, m; r) = \text{CHash}(pk, m'; r')
  \]
Construction Idea II/II

- Simplified construction idea:
  - To sign $m = (m_1, \ldots, m_n)$
  - Use signature scheme to sign $h = (h_1, \ldots, h_n)$, where

    $$h_i = \begin{cases} 
    CHash(pk, m_i; r_i) & \text{if sanitizable} \\
    m_i & \text{else}
    \end{cases}$$

  - As sanitizable signature provide signature on $h$; additionally include the randomness $r_i$ of sanitizable parts
Policy-Based Sanitizable Signatures
Drawbacks of Existing Sanitizable Signatures

- Restricted in flexibility of specifying potential sanitizers
  - Conventional model just considers a single sanitizer specified at signing time

- Some existing works allow multiple sanitizers, but then lose accountability
  - Accountability is a central feature: sanitizers must be traceable

Achieve fine-grained sanitization control with full accountability?
Specify a policy (e.g., over attributes) when signing: $A \text{ AND } (B \text{ OR } C)$

Concrete set of sanitizers not specified during signing!

Sanitizers have attributes: if attributes satisfy policy, then sanitizing works
Policy-Based Sanitizable Signatures (PB-SS)

- At signing time signer specifies an access-policy for sanitization and a sanitization group
- Sanitizers with their sanitizer key pair can join sanitization groups dynamically
  - Get then issued sanitization keys for attributes in the group
  - Sanitization works if sanitizer is in the group and attributes satisfy the policy
- Accountability: exact sanitizer can be identified

Achieves fine-grained sanitization control with full accountability!
Security of PB-SS

- Require same properties (but extended) as for sanitizable signatures **plus**
- Pseudonymity
  - Signature does not leak which party is accountable
- Proof-Soundness
  - Impossible to generate a proof for an adverserially chosen signature/message pair that points to different entities
- Traceability
  - Impossible to generate a verifying signature such that an honest signer cannot identify the accountable party

We also further strengthen (existing) notions whenever possible
Construction Idea: PB-SS

- Follows basically the idea outlined for sanitizable signatures
- Instead of using a chameleon hash we use a strengthened version of a policy-based chameleon hash [DSSS, NDSS’19]
  - Hash is computed with respect to a policy
  - Collisions can be found when policy is satisfied
  - Can be constructed from chameleon hashes with ephemeral trapdoors [CDKPSS, PKC’17] and CCA secure ciphertext-policy attribute-based encryption (CP-ABE) [BSW, Oakland’07]
- Achieving accountability requires some additional technicalities
  - Use of a non-interactive zero-knowledge proof (NIZK) for an OR-language
Potential Application of PB-SS

- Applicable to all existing applications
- A new application can be *redactable blockchains*
  - Introduced by [AMVA, EuroS&P’17]
  - [DSSS, NDSS’19] introduce the use of PBCH to hash transactions in blockchains
    - Update/rewrite transactions by computing collisions in the PBCH
    - Fine-grained approach by using policies
  - Use of PB-SS instead of PBCH to get additional properties such as accountability
Conclusions and Take Home
Conclusions & Take Home

- Sanitizable signatures are a tool to realize controlled modifications of signed messages
- Existing schemes are very limited in their expressiveness
- We introduce the notion of policy-based sanitizable signatures
  - Fine-grained sanitization control via policies
- We present a strong security model and a provable secure practical construction