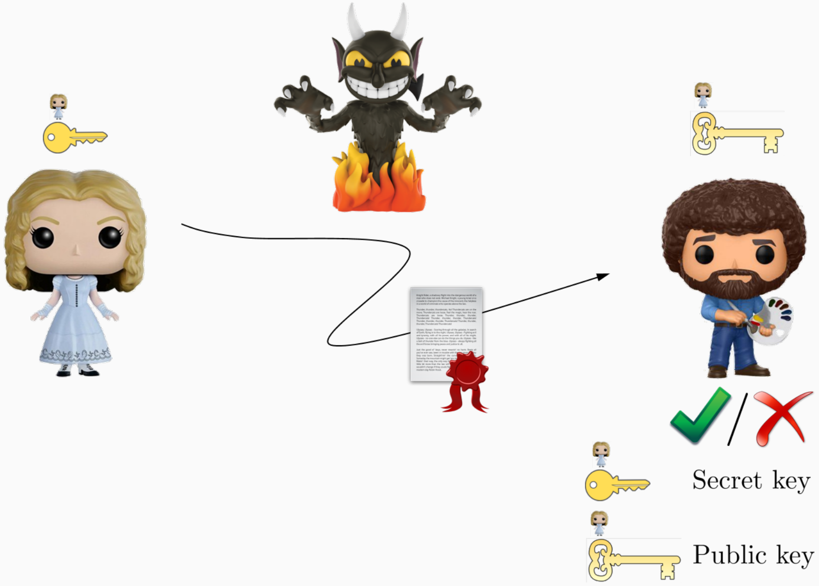
# Protean Signature Schemes

Stephan Krenn[‡], Henrich C. Pöhls[§], Kai Samelin[⋆], Daniel Slamanig[‡]

October 2, 2018—Cryptology And Network Security (CANS 2018), Naples, Italy

‡ AIT AUSTRIAN INSTITUTE OF TECHNOLOGY § UNIVERSITÄT PASSAU ⋆ TÜVRheinland®

## Digital Signatures



## Digital Signatures

- Establish the origin of a message (bind signer's identity to message)
- A valid signature guarantees
    - Message integrity (no modifications happened)
    - Identity of the signer

## Digital Signatures

- Establish the origin of a message (bind signer's identity to message)
- A valid signature guarantees
  - Message integrity (no modifications happened)
  - Identity of the signer

### Security (EUF-CMA)

- Obtain signatures on arbitrary messages
- Not able to produce valid signature for non-queried message

## Controlled Modification of Signed Messages

Modifications? That’s what we try to prevent?

## Controlled Modification of Signed Messages

Modifications? That's what we try to prevent?

Controlled modifications

- Signer determines *how* signed message can be altered
- Think of implicitly signing all possible messages

## Controlled Modification of Signed Messages

Modifications? That's what we try to prevent?

Controlled modifications

- Signer determines *how* signed message can be altered
- Think of implicitly signing all possible messages

Example: Medical documents

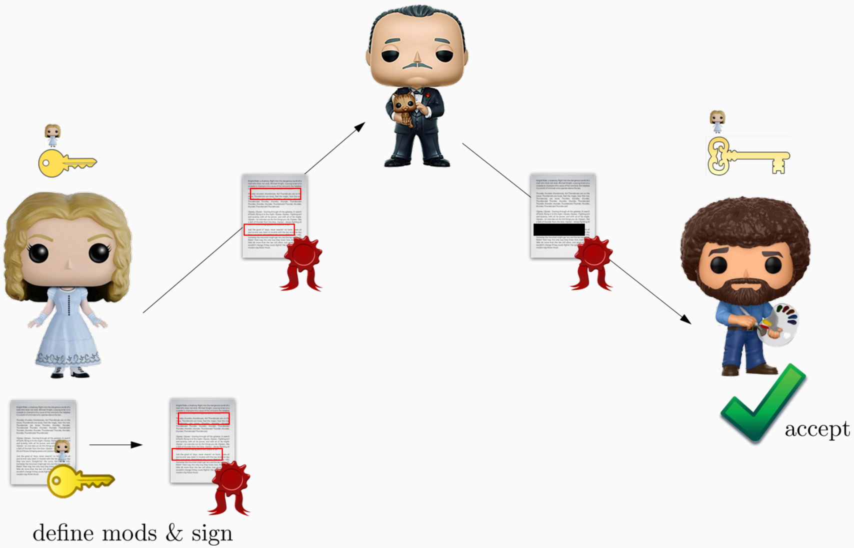
- Anonymization for research/accounting (still want authenticity guarantees)
- Removing exact diagnosis for sick leave

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

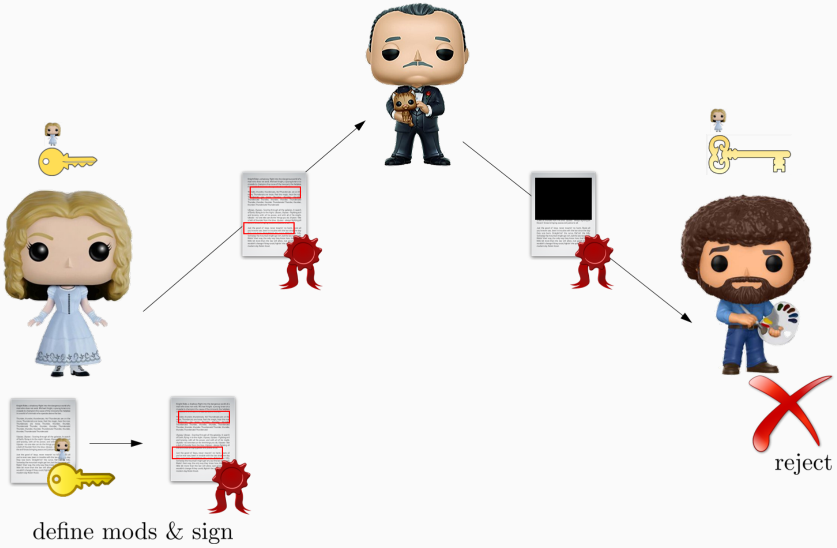
## Controlled Modification of Signed Messages

Will look at two common schemes: redactable and sanitizable signatures

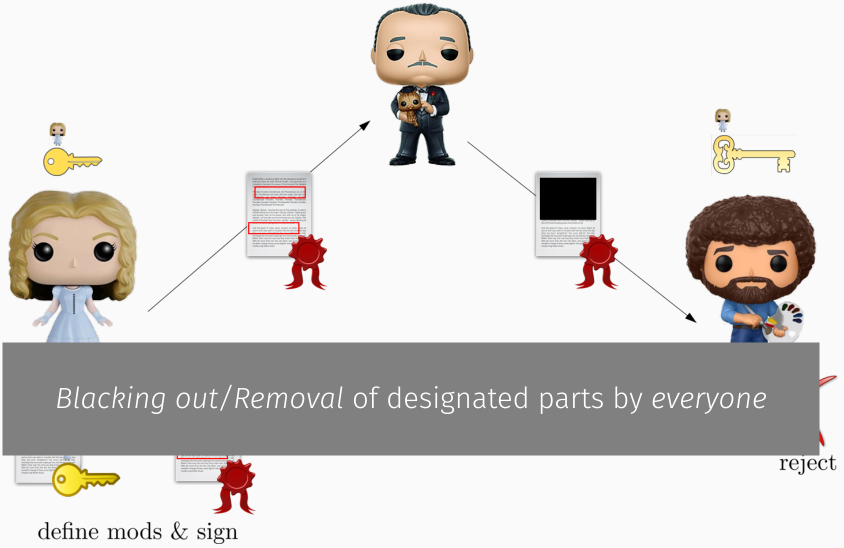
## Redactable Signatures (RSS)



## Redactable Signatures (RSS)



## Redactable Signatures (RSS)



*Blacking out*/*Removal* of designated parts by *everyone*

## Redactable Signatures (RSS)

### Security properties

- Unforgeability
    - In EUF-CMA sense: cannot come up with valid signature for a message not “derivable” from signed ones
- Privacy
    - Redacted signatures leaks no information about redacted parts
- Transparancy (optional)
    - Not visible if redaction happened or not
- Unlinkability

## Redactable Signatures (RSS)

Originally proposed in [SBZ, ICISC'01] and [JMSW, CT-RSA'02]

Various ad-hoc constructions for different message representations (linear, sets, trees)

Generic construction from EUF-CMA secure signatures and *indistinguishable* accumulators [DPSS, ICISC'15]

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \dots, x_n\}$ succinctly represented by $\mathsf{acc}_{\mathcal{X}}$

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \dots, x_n\}$ succinctly represented by $\mathsf{acc}_{\mathcal{X}}$
Witnesses $\mathsf{wit}_{x_i}$ certifying membership of $x_i$ in $\mathsf{acc}_{\mathcal{X}}$

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

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \dots, x_n\}$ succinctly represented by $\mathsf{acc}_{\mathcal{X}}$
Witnesses $\mathsf{wit}_{x_i}$ certifying membership of $x_i$ in $\mathsf{acc}_{\mathcal{X}}$

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

Indistinguishability [DS, CT-RSA'15]

- Neither **acc** nor **wit** leak information about $\mathcal{X}$

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \dots, x_n\}$ succinctly represented by $\mathbf{acc}_{\mathcal{X}}$
Witnesses $\mathbf{wit}_{x_i}$ certifying membership of $x_i$ in $\mathbf{acc}_{\mathcal{X}}$

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

Indistinguishability [DS, CT-RSA'15]

- Neither **acc** nor **wit** leak information about $\mathcal{X}$

**RSS:** To sign $m = \{m_1, \dots, m_n\}$

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \dots, x_n\}$ succinctly represented by $\mathbf{acc}_{\mathcal{X}}$
Witnesses $\mathbf{wit}_{x_i}$ certifying membership of $\boldsymbol{x_i}$ in $\mathbf{acc}_{\mathcal{X}}$

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

Indistinguishability [DS, CT-RSA'15]

- Neither **acc** nor **wit** leak information about $\mathcal{X}$

**RSS:** To sign $m = \{\boldsymbol{m_1}, \dots, \boldsymbol{m_n}\}$

- Compute $\mathbf{acc}_m$ and use $\Sigma$ to sign $\mathbf{acc}_m$

## Redactable Signatures (RSS)

Sketch of RSS for sets (accumulator + EUF-CMA signatures $\Sigma$):

**Accumulator:** $\mathcal{X} = \{x_1, \ldots, x_n\}$ succinctly represented by $\mathsf{acc}_{\mathcal{X}}$
Witnesses $\mathsf{wit}_{x_i}$ certifying membership of $x_i$ in $\mathsf{acc}_{\mathcal{X}}$

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

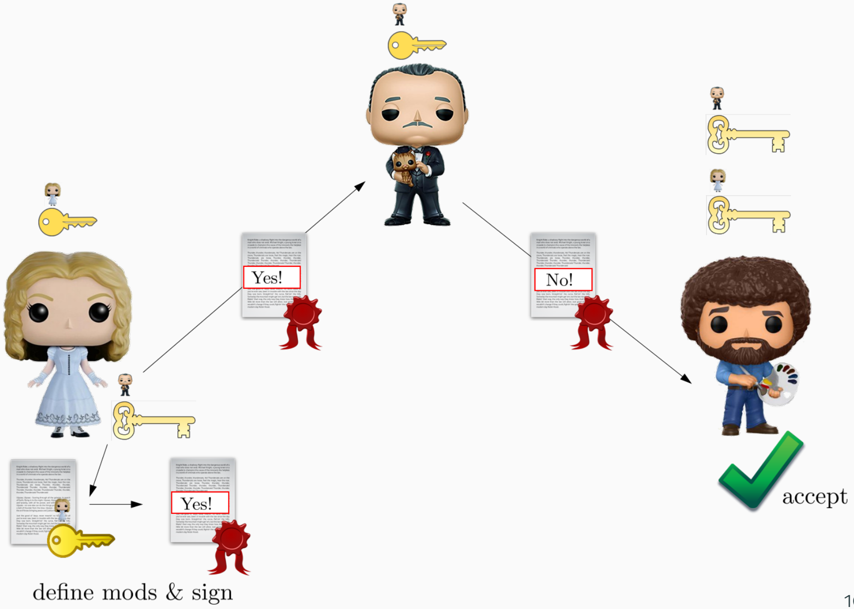
Indistinguishability [DS, CT-RSA'15]

- Neither **acc** nor **wit** leak information about $\mathcal{X}$

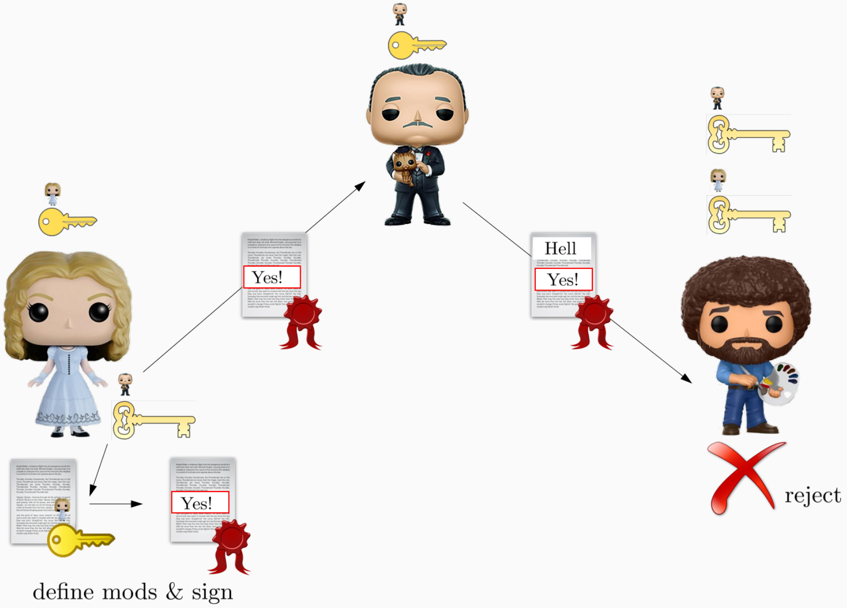
**RSS:** To sign $m = \{m_1, \ldots, m_n\}$

- Compute $\mathsf{acc}_m$ and use $\Sigma$ to sign $\mathsf{acc}_m$
- As redactable signature provide signature of $\Sigma$ and $\{\mathsf{wit}_{m_i}\}$

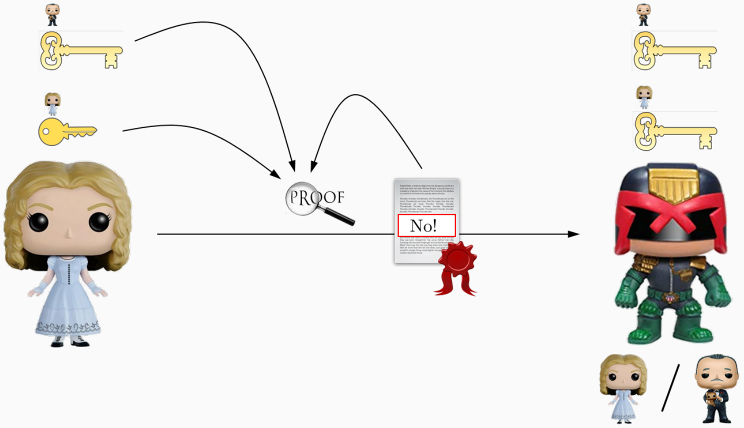
## Sanitizable Signatures (SSS)



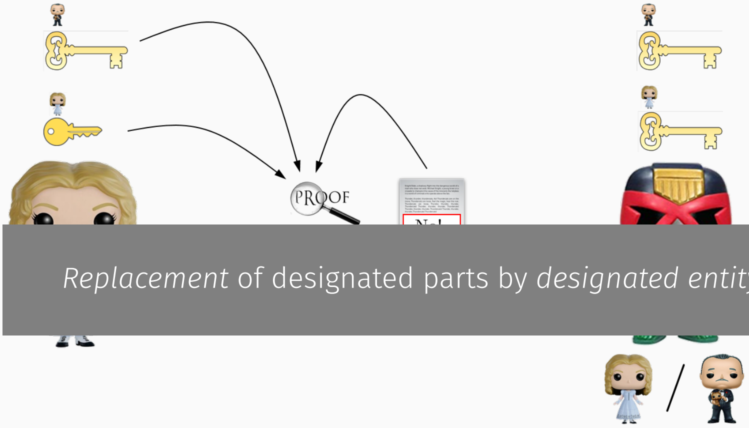
## Sanitizable Signatures (SSS)



# Sanitizable Signatures (SSS)



## Sanitizable Signatures (SSS)



*Replacement* of designated parts by *designated entity*

## Sanitizable Signatures (SSS)

### Security properties

- Unforgeability
- Immutability
  - In EUF-CMA sense: Sanitizer cannot come up with valid signature for a message not “derivable” from signed ones
- Privacy
- Signer/Sanitizer accountability
  - Signer/sanitizer cannot blame the other party for having produced a signature
- Transparancy (optional)
- Invisibility (optional)
  - Signature does not leak which parts are sanitizable
- Unlinkability

## Sanitizable Signatures (SSS)

Originally proposed in [ACMT, ESORICS'05] and rigorous security model [BFFLP+, PKC'09]

Various constructions with different properties and sanitizing restrictions, e.g., limit sanitizing to defined set

Generic construction from EUF-CMA secure signatures and *chameleon hash functions* [BFFLP+, PKC'09]

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

Chameleon Hash: Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

Chameleon Hash: Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

- Hashing: $h \leftarrow \mathsf{CHash}(\mathsf{pk}, m; r)$

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

Chameleon Hash: Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

- Hashing: $h \leftarrow \mathsf{CHash}(\mathsf{pk}, m; r)$
- Collision: $\mathsf{sk}$ allows for any $h$, $\hat{m}$ to compute $\hat{r}$ s.t. $\mathsf{CHash}(\mathsf{pk}, m; r) = \mathsf{CHash}(\mathsf{pk}, \hat{m}; \hat{r})$

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

Chameleon Hash: Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

- Hashing: $h \leftarrow \mathsf{CHash}(\mathsf{pk}, m; r)$
- Collision: $\mathsf{sk}$ allows for any $h$, $\hat{m}$ to compute $\hat{r}$ s.t. $\mathsf{CHash}(\mathsf{pk}, m; r) = \mathsf{CHash}(\mathsf{pk}, \hat{m}; \hat{r})$

SSS: To sign $m = (m_1, \ldots, m_n)$

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

**Chameleon Hash:** Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

- Hashing: $h \leftarrow \mathsf{CHash}(\mathsf{pk}, m; r)$
- Collision: $\mathsf{sk}$ allows for any $h$, $\hat{m}$ to compute $\hat{r}$ s.t. $\mathsf{CHash}(\mathsf{pk}, m; r) = \mathsf{CHash}(\mathsf{pk}, \hat{m}; \hat{r})$

**SSS:** To sign $m = (m_1, \dots, m_n)$
Use $\Sigma$ to sign $h = (h_1, \dots, h_n)$ where

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

## Sanitizable Signatures (SSS)

Sketch of SSS (chameleon hashes + EUF-CMA signatures $\Sigma$):

**Chameleon Hash:** Collision resistant hash keyed with $(\mathsf{sk}, \mathsf{pk})$

- Hashing: $h \leftarrow \mathsf{CHash}(\mathsf{pk}, m; r)$
- Collision: $\mathsf{sk}$ allows for any $h$, $\hat{m}$ to compute $\hat{r}$ s.t. $\mathsf{CHash}(\mathsf{pk}, m; r) = \mathsf{CHash}(\mathsf{pk}, \hat{m}; \hat{r})$

**SSS:** To sign $m = (m_1, \ldots, m_n)$
Use $\Sigma$ to sign $h = (h_1, \ldots, h_n)$ where

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

As sanitizable signature provide signature of $\Sigma$ and $r$

## Merging the Functionalities

Provide a primitive that supports removal and editing at the same time

Generalize RSS and SSS into a single primitive having all desired properties of RSS and SSS

Motivating example (k-anonymization):

- Removal of attributes
- Generalization of attributes

## Using Existing Approaches?

We could use SSS to mimic the functionality of RSS

- Use limited set replacement with a special symbol denoting redaction

## Using Existing Approaches?

We could use SSS to mimic the functionality of RSS

- Use limited set replacement with a special symbol denoting redaction

Destroys transparency!

## Using Existing Approaches?

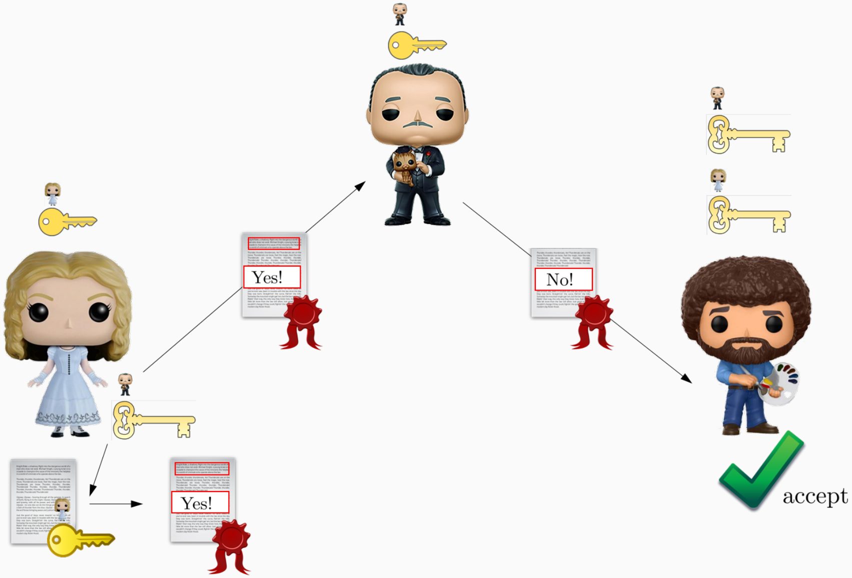
We could use SSS to mimic the functionality of RSS

- Use limited set replacement with a special symbol denoting redaction

Destroys transparency!

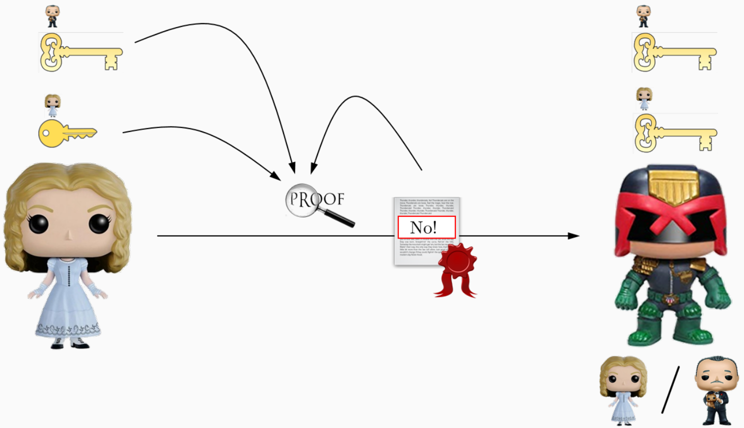
Ideally have efficient construction providing all properties

## Protean Signature Schemes (PSS)

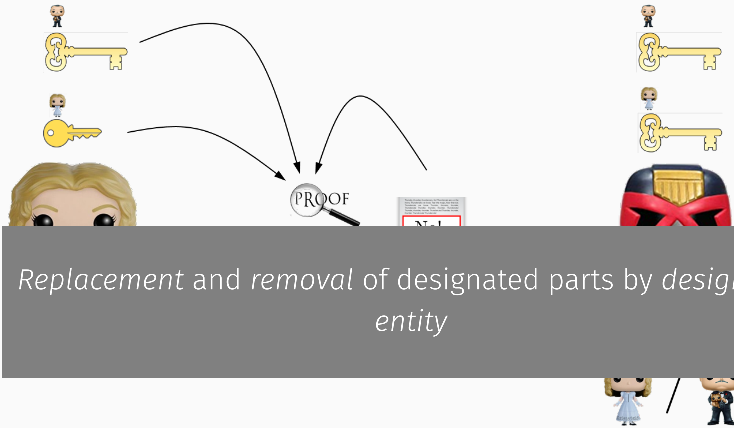


define mods & sign

## Protean Signature Schemes (PSS)



## Protean Signature Schemes (PSS)



*Replacement* and *removal* of designated parts by *designated entity*

## Protean Signature Schemes (PSS)

Security properties

- Unforgeability
- Immutability
- Privacy
- Signer/Sanitizer accountability
- Transparancy (optional)
- Invisibility (optional)

## Sketch of Construction (Ingredients)

We provide a black-box construction of a protean signature scheme

Ingredients

- A secure sanitizable signature scheme (SSS)
- A secure redactable signature scheme (RSS)
- A CCA2 secure labeled public key encryption scheme
    - Only required if RSS provides auxiliary redaction information RED
    - RED typically makes redactions more efficient

## Sketch of Construction (Keys)

Signer keys (keys from SSS and RSS)

- $\mathsf{sk}_{\mathsf{sig}} \leftarrow (\mathsf{sk}_{\mathsf{sig}}^{\mathsf{SSS}}, \mathsf{sk}^{\mathsf{RSS}})$
- $\mathsf{pk}_{\mathsf{sig}} \leftarrow (\mathsf{pk}_{\mathsf{sig}}^{\mathsf{SSS}}, \mathsf{pk}^{\mathsf{RSS}})$

Sanitizer keys (keys from SSS)

- $\mathsf{sk}_{\mathsf{san}} \leftarrow \mathsf{sk}_{\mathsf{san}}^{\mathsf{SSS}}$
- $\mathsf{pk}_{\mathsf{san}} \leftarrow \mathsf{pk}_{\mathsf{san}}^{\mathsf{SSS}}$

## Sketch of Construction (Signing)

Let us consider a message $m = (m_1, m_2, m_3)$

## Sketch of Construction (Signing)

Let us consider a message $m = (m_1, m_2, m_3)$

Inner SSS

$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}}$ $\boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}}$ $\boxed{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}$

## Sketch of Construction (Signing)

Let us consider a message $m = (m_1, m_2, m_3)$

Inner SSS

$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}}$ $\boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}}$ $\boxed{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}$

RSS

$\boxed{\underbrace{(\tau_1, \tau_2, \tau_3, \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma^{\mathsf{RSS}}}}$

## Sketch of Construction (Signing)

Let us consider a message $m = (m_1, m_2, m_3)$

Inner SSS

$$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathrm{pk}_{\mathrm{sig}}, \mathrm{pk}_{\mathrm{san}})}_{\sigma_1^{\mathrm{SSS}}}} \quad \boxed{\underbrace{(m_2, \tau, \tau_2, \mathrm{pk}_{\mathrm{sig}}, \mathrm{pk}_{\mathrm{san}})}_{\sigma_2^{\mathrm{SSS}}}} \quad \boxed{\underbrace{(m_3, \tau, \tau_3, \mathrm{pk}_{\mathrm{sig}}, \mathrm{pk}_{\mathrm{san}})}_{\sigma_3^{\mathrm{SSS}}}}$$

RSS

$$\boxed{\underbrace{(\tau_1, \tau_2, \tau_3, \tau, \mathrm{pk}_{\mathrm{sig}}, \mathrm{pk}_{\mathrm{san}})}_{\sigma^{\mathrm{RSS}}}}$$

Outer SSS

$$\boxed{\underbrace{((m_1, m_2, m_3), \sigma^{\mathrm{RSS}}, (\tau_1, \tau_2, \tau_3, \sigma_1^{\mathrm{SSS}}, \sigma_2^{\mathrm{SSS}}, \sigma_3^{\mathrm{SSS}}), \tau, \mathrm{pk}_{\mathrm{sig}}, \mathrm{pk}_{\mathrm{san}})}_{\sigma_0^{\mathrm{SSS}}}}$$

## Sketch of Construction (Editing)

Edit the message $m = (m_1, m_2, m_3)$ to $\hat{m} = (m_1, \hat{m}_2)$

# Sketch of Construction (Editing)

Edit the message $m = (m_1, m_2, m_3)$ to $\hat{m} = (m_1, \hat{m}_2)$

Inner SSS

$$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}} \quad \boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}} \quad \boxed{\cancel{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}}$$

## Sketch of Construction (Editing)

Edit the message $m = (m_1, m_2, m_3)$ to $\hat{m} = (m_1, \hat{m}_2)$

Inner SSS

$$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}} \quad \boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}} \quad \boxed{\cancel{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}}$$

RSS

$$\boxed{\underbrace{(\tau_1, \tau_2, \cancel{\tau_3}, \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\hat{\sigma}^{\mathsf{RSS}}}}$$

## Sketch of Construction (Editing)

Edit the message $m = (m_1, m_2, m_3)$ to $\hat{m} = (m_1, \hat{m}_2)$

Inner SSS

$$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}} \quad \boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}} \quad \boxed{\cancel{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}}$$

RSS

$$\boxed{\underbrace{(\tau_1, \tau_2, \cancel{\tau_3}, \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\hat{\sigma}^{\mathsf{RSS}}}}$$

Outer SSS

$$\boxed{\underbrace{((m_1, m_2, m_3), \sigma^{\mathsf{RSS}}, (\tau_1, \tau_2, \tau_3, \sigma_1^{\mathsf{SSS}}, \sigma_2^{\mathsf{SSS}}, \sigma_3^{\mathsf{SSS}}), \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_0^{\mathsf{SSS}}}}$$

## Sketch of Construction (Editing)

Edit the message $m = (m_1, m_2, m_3)$ to $\hat{m} = (m_1, \hat{m}_2)$

Inner SSS

$\boxed{\underbrace{(m_1, \tau, \tau_1, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_1^{\mathsf{SSS}}}}$ $\boxed{\underbrace{(m_2, \tau, \tau_2, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_2^{\mathsf{SSS}}}}$ $\boxed{\cancel{\underbrace{(m_3, \tau, \tau_3, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\sigma_3^{\mathsf{SSS}}}}}$

RSS

$\boxed{\underbrace{(\tau_1, \tau_2, \cancel{\tau_3}, \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\hat{\sigma}^{\mathsf{RSS}}}}$

$\boxed{\underbrace{((m_1, \hat{m}_2), \hat{\sigma}^{\mathsf{RSS}}, (\tau_1, \tau_2, \sigma_1^{\mathsf{SSS}}, \sigma_2^{\mathsf{SSS}}), \tau, \mathsf{pk}_{\mathsf{sig}}, \mathsf{pk}_{\mathsf{san}})}_{\hat{\sigma}_0^{\mathsf{SSS}}}}$

## Open Issues and Potential Directions

Limiting sets: Integrate features to limit sanitizing to signer-chosen sets of values

## Open Issues and Potential Directions

Limiting sets: Integrate features to limit sanitizing to signer-chosen sets of values

Block-level: Accountability notions on a block instead of message level

## Open Issues and Potential Directions

Limiting sets: Integrate features to limit sanitizing to signer-chosen sets of values

Block-level: Accountability notions on a block instead of message level

Dependencies: Enforce restrictions such as “if block $i$ is redacted, also block $j$ needs to be redacted”

## Open Issues and Potential Directions

Limiting sets: Integrate features to limit sanitizing to signer-chosen sets of values

Block-level: Accountability notions on a block instead of message level

Dependencies: Enforce restrictions such as “if block $i$ is redacted, also block $j$ needs to be redacted”

Unlinkability: Seems hard to achieve with our construction paradigm

## Conclusions and Take-Home

- RSS and SSS allow controlled modifications of signed messages

## Conclusions and Take-Home

- RSS and SSS allow controlled modifications of signed messages
- RSS and SSS provide different features (e.g., remove vs. replace)

## Conclusions and Take-Home

- RSS and SSS allow controlled modifications of signed messages
- RSS and SSS provide different features (e.g., remove vs. replace)
- We generalize RSS and SSS into protean signatures (PSS)

## Conclusions and Take-Home

- RSS and SSS allow controlled modifications of signed messages
- RSS and SSS provide different features (e.g., remove vs. replace)
- We generalize RSS and SSS into protean signatures (PSS)
- PSS provide all features and strong privacy guarantees

## Conclusions and Take-Home

- RSS and SSS allow controlled modifications of signed messages
- RSS and SSS provide different features (e.g., remove vs. replace)
- We generalize RSS and SSS into protean signatures (PSS)
- PSS provide all features and strong privacy guarantees
- We provide a generic construction based on RSS and SSS (and labeled PKE)

## Thank you! Questions?

@drl3c7er

Supported by EU H2020 and ECSEL