# Blank Digital Signatures: Optimization and Practical Experiences

David Derler, Christian Hanser, and Daniel Slamanig
{david.derler, christian.hanser, daniel.slamanig}@iaik.tugraz.at

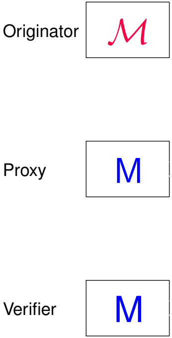
Institute for Applied Information Processing and Communications, Graz University of Technology

September 8, 2014

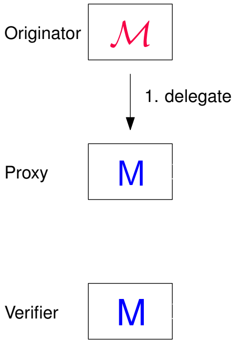
# Outline

- Proxy-Type Signatures
  - Blank Digital Signatures (BDS)
  - Motivation
- The BDS Scheme
  - Overview
  - Optimizations
  - Implementation
  - Performance
- Conclusion

# Proxy-Type Signatures

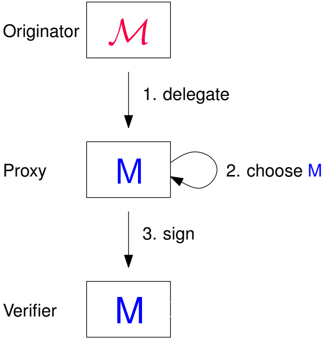


# Proxy-Type Signatures



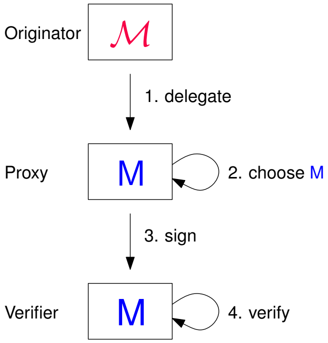

- Delegate signing rights for
  - Message space $\mathcal{M}$

# Proxy-Type Signatures




- Delegate signing rights for
  - Message space $\mathcal{M}$
- Choose message $M$ and sign

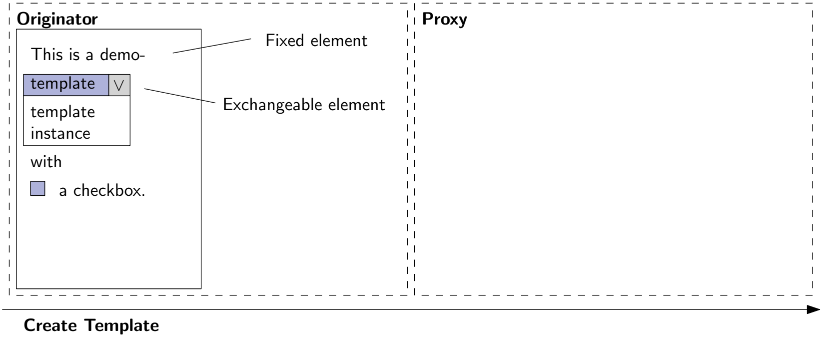
# Proxy-Type Signatures




- Delegate signing rights for
  - Message space $\mathcal{M}$
- Choose message $M$ and sign
- Verify
  - Integrity
  - Authenticity
  - $M \stackrel{?}{\in} \mathcal{M}$

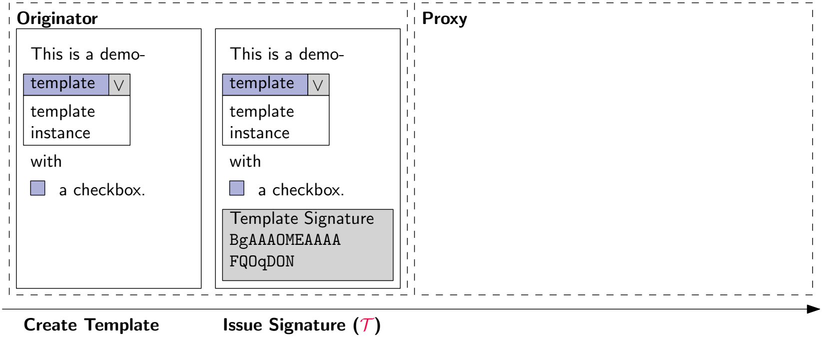
# Blank Digital Signatures

- Message space defined by Template



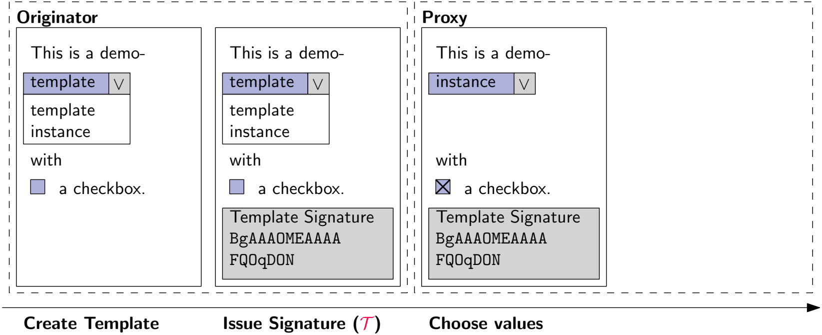
# Blank Digital Signatures

- Message space defined by Template



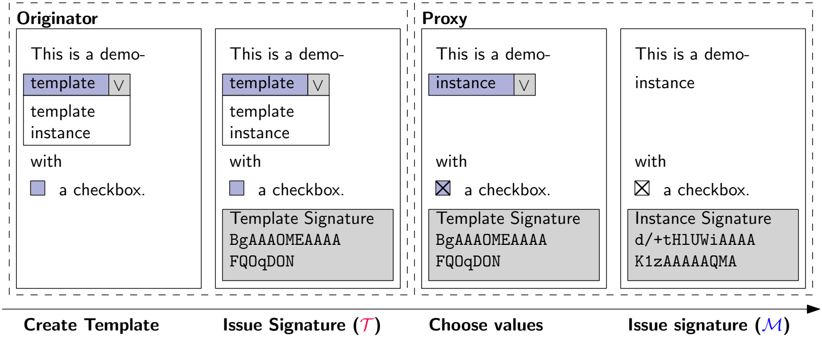
# Blank Digital Signatures

- Message space defined by Template



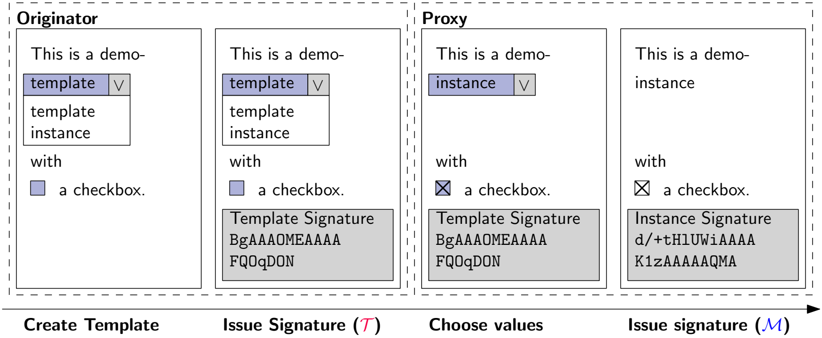
# Blank Digital Signatures

- Message space defined by Template



# Blank Digital Signatures

- Message space defined by Template




- New: **Privacy property**
  - Hides $\mathcal{T} \setminus \mathcal{M}$

# Motivation

- Attorney makes business deal
  - ...on behalf of the client
  - Privacy property
- Medical files
  - Doctor creates template containing all data
  - Patient can black-out critical parts
- Governmental organizations publish forms
  - to be signed by any citizen

$\mathcal{T} =$
$(\{\text{''}\textit{I, hereby, declare to pay }\text{''}\},$
$\{\text{''}100\$\text{''}, \text{''}120\$\text{''}, \text{''}150\$\text{''}\},$
$\{\text{''}\textit{for this device.}\text{''}\})$

# Blank Digital Signature Scheme

- Proposed in [HS13]
- Combination of
  - Conventional Digital Signature Scheme
    - Providing a warrant for the delegation
  - Polynomial Commitments
    - Templates and messages bound to commitment
    - Optimized version of [KZG10]
    - Based on pairing friendly elliptic curve groups
    - Hiding Commitments $\rightarrow$ *privacy property*

# Encoding

- Template $\mathcal{T} = (T_1, T_2, \ldots, T_n)$ with $T_i = \{M_{i_1}, M_{i_2}, \ldots, M_{i_k}\}$
- $|T_i| = \begin{cases} > 1 & \text{for exchangeable elements} \\ = 1 & \text{for fixed elements} \end{cases}$
- Message $\mathcal{M} = (M_i)_{i=1}^{n}$

# Encoding

- Template $\mathcal{T} = (T_1, T_2, \ldots, T_n)$ with $T_i = \{M_{i_1}, M_{i_2}, \ldots, M_{i_k}\}$
- $|T_i| = \begin{cases} > 1 & \text{for exchangeable elements} \\ = 1 & \text{for fixed elements} \end{cases}$
- Message $\mathcal{M} = (M_i)_{i=1}^{n}$
- $\mathcal{T} = (\{\text{"}I, hereby, declare\ to\ pay\text{"}\}, \{\text{"}100\$\text{"}, \text{"}120\$\text{"}, \text{"}150\$\text{"}\}, \{\text{"}for\ this\ device.\text{"}\})$, $\mathcal{M} = (\text{"}I, hereby, declare\ to\ pay\text{"}, \text{"}120\$\text{"}, \text{"}for\ this\ device.\text{"})$

# Encoding (2) - Template Encoding Polynomial

Fixed element

Exchangeable element

$$t(X) = (X - H(M_1||id_{\mathcal{T}}||1)) \cdot \frac{\frac{(X - H(M_{2_1}||id_{\mathcal{T}}||2))}{(X - H(M_{2_2}||id_{\mathcal{T}}||2))}}{(X - H(M_{2_3}||id_{\mathcal{T}}||2))} \cdots (X - H(M_n||id_{\mathcal{T}}||n))$$

Fixed element

- $H$ ... collision resistant hash function

# Encoding (3) - Message Encoding Polynomial

$$m(X) =$$

$$(X - H(M_1||id_{\mathcal{T}}||1)) \cdot \frac{\frac{(X - H(M_{2_1}||id_{\mathcal{T}}||2))}{(X - H(M_{2_2}||id_{\mathcal{T}}||2))}}{(X - H(M_{2_3}||id_{\mathcal{T}}||2))} \cdots (X - H(M_n||id_{\mathcal{T}}||n))$$

- $H \ldots$ collision resistant hash function

# Encoding (4) - Complementary Message Polynomial

$$\overline{m}(X) =$$

$$(X - H(M_1||id_{\mathcal{T}}||1)) \cdot \frac{\dfrac{(X - H(M_{2_1}||id_{\mathcal{T}}||2))}{(X - H(M_{2_2}||id_{\mathcal{T}}||2))}}{(X - H(M_{2_3}||id_{\mathcal{T}}||2))} \cdots (X - H(M_n||id_{\mathcal{T}}||n))$$

- $H$ . . . collision resistant hash function

# Scheme

- **Sign:**
  - Commit to template encoding polynomial $t(X) \rightarrow \mathcal{C}_t$
  - Designation
    - Sign $\mathcal{C}_t$ and identity of proxy $\rightarrow$ DSS

# Scheme

- **Sign:**
  - Commit to template encoding polynomial $t(X) \rightarrow \mathcal{C}_t$
  - Designation
    - Sign $\mathcal{C}_t$ and identity of proxy $\rightarrow$ DSS
- **Verify$_{\mathcal{T}}$ (only for proxy):**
  - Recompute commitment and compare
  - Verify designation $\rightarrow$ DSS

# Scheme (2)

- **Inst:**
  - Choose final values
    - for exchangeable elements

# Scheme (2)

- **Inst:**
  - Choose final values
    - for exchangeable elements
  - Compute message encoding polynomial $m(X)$

# Scheme (2)

- **Inst:**
  - Choose final values
    - for exchangeable elements
  - Compute message encoding polynomial $m(X)$
  - Commit to: $\overline{m}(X) = \frac{t(X)}{m(X)} \rightarrow \mathcal{C}_{\overline{m}}$
    - Thus, $m(X) \cdot \overline{m}(X) = t(X)$ and "$\mathcal{C}_m \otimes \mathcal{C}_{\overline{m}} = \mathcal{C}_t$"

# Scheme (2)

- **Inst:**
  - Choose final values
    - for exchangeable elements
  - Compute message encoding polynomial $m(X)$
  - Commit to: $\overline{m}(X) = \frac{t(X)}{m(X)} \to \mathcal{C}_{\overline{m}}$
    - Thus, $m(X) \cdot \overline{m}(X) = t(X)$ and "$\mathcal{C}_m \otimes \mathcal{C}_{\overline{m}} = \mathcal{C}_t$"
  - Sign $\mathcal{C}_{\overline{m}} \to$ DSS

# Scheme (3)

- **Verify$_{\mathcal{M}}$ (public):**
  - Compute commitment to message encoding polynomial $\mathcal{C}_m$

# Scheme (3)

- **Verify$_{\mathcal{M}}$ (public):**
  - Compute commitment to message encoding polynomial $\mathcal{C}_m$
  - Check $\mathcal{C}_m \otimes \mathcal{C}_{\overline{m}} \stackrel{?}{=} \mathcal{C}_t$

# Scheme (3)

- **Verify$_{\mathcal{M}}$ (public):**
  - Compute commitment to message encoding polynomial $\mathcal{C}_m$
  - Check $\mathcal{C}_m \otimes \mathcal{C}_{\overline{m}} \stackrel{?}{=} \mathcal{C}_t$
  - Verify designation, signature over $\mathcal{C}_t, \mathcal{C}_{\overline{m}} \rightarrow$ DSS

# Optimizations

- Original protocol uses inefficient symmetric pairings

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T, \text{ with } \mathbb{G}_1 = \mathbb{G}_2$$

- Asymmetric Type-3 pairings ($\mathbb{G}_1 \neq \mathbb{G}_2$)
  - Duplicating some points

# Optimizations

- Original protocol uses inefficient symmetric pairings

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \to \mathbb{G}_T, \text{ with } \mathbb{G}_1 = \mathbb{G}_2$$

- Asymmetric Type-3 pairings ($\mathbb{G}_1 \neq \mathbb{G}_2$)
  - Duplicating some points
- Computations in $\mathbb{G}_2$ more expensive
  - Moving $\mathbb{G}_2$ computations to instantiation step **Inst**
  - $\textbf{Verify}_{\mathcal{M}}$ fast

# Optimizations

- Original protocol uses inefficient symmetric pairings

$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T, \text{ with } \mathbb{G}_1 = \mathbb{G}_2$$

- Asymmetric Type-3 pairings ($\mathbb{G}_1 \neq \mathbb{G}_2$)
  - Duplicating some points
- Computations in $\mathbb{G}_2$ more expensive
  - Moving $\mathbb{G}_2$ computations to instantiation step **Inst**
  - **Verify**$_{\mathcal{M}}$ fast
- Aggregation of fixed elements
  - $X - H(id_{\mathcal{T}}||M_1||1||M_2||2||\ldots||M_n||n)$

# Optimizations

- Original protocol uses inefficient symmetric pairings

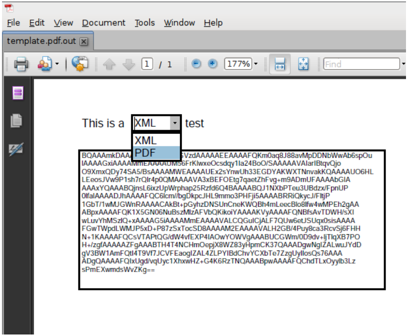
$$e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T, \text{ with } \mathbb{G}_1 = \mathbb{G}_2$$

- Asymmetric Type-3 pairings ($\mathbb{G}_1 \neq \mathbb{G}_2$)
  - Duplicating some points
- Computations in $\mathbb{G}_2$ more expensive
  - Moving $\mathbb{G}_2$ computations to instantiation step **Inst**
  - $\mathbf{Verify}_{\mathcal{M}}$ fast
- Aggregation of fixed elements
  - $X - H(id_{\mathcal{T}}||M_1||1||M_2||2||\ldots||M_n||n)$
- Optimizations preserve the security

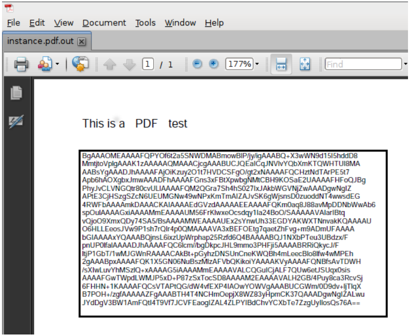
# Implementation Aspects

- Integrated within Java Cryptography Architecture
  - Key generation: `KeyPairGenerator` implementation
  - Sign, $\mathsf{Verify}_{\mathcal{T}}$: `Signature`
  - Inst, $\mathsf{Verify}_{\mathcal{M}}$: `Signature`
- Using PKIX
  - Integration of public keys in X.509 certificates
  - `KeyFactory` implementations for X.509 key extraction
  - Revocation mechanisms of PKIX can be employed
- Two example signature formats
  - XML
  - PDF

# PDF Signature Format




Template

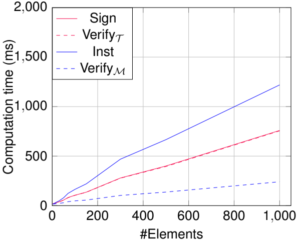



Message

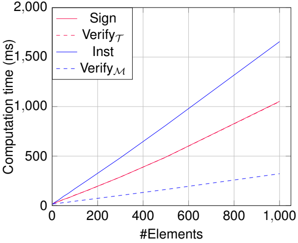
# Performance

- BNPairings library by Geovandro and Barreto [GB12]
  - Optimal Ate pairing on BN Curves
  - 256 bit group size
- Timings performed on a single core of a
  - Lenovo ThinkPad T420s
  - Intel Core i5 2540M with 2.6/3.3 GHz
  - 8GB RAM
  - Java 1.7.0_55 on top of Ubuntu 14.04/amd64
- Different template constellations
- ... and numbers of elements

# Performance (2)




(a) 50% fixed




(b) 33% fixed

Figure : Computation times in relation to #Elements

# Conclusion

- Optimized BDSS
- Integration into JCA and PKIX
- Two signature formats
  - PDF forms $\rightarrow$ practical applications
  - Integration of XML format into XMLDsig or XAdES
    - XAdES-A $\rightarrow$ long term validation
- Fully feasible for practical use
  - 100 elements $\rightarrow$ each step $\leq$ 180ms
- Future Work
  - Comparison to BDSS from anonymous credentials [DHS14]
  - Integration of BDSS into PDF reader plug-in

# Thank you.

**References**


David Derler, Christian Hanser, and Daniel Slamanig.
Privacy-Enhancing Proxy Signatures from Non-interactive Anonymous Credentials.
In *Data and Applications Security and Privacy XXVIII*, volume 8566 of *LNCS*, pages 49–65. Springer, 2014.

C. C. F. Pereira Geovandro and Paulo S. L. M. Barreto.
bnpairings - A Java implementation of efficient bilinear pairings and elliptic curve operations.
Public Google code project at: `https://code.google.com/p/bnpairings/`, 5 November 2012.

Christian Hanser and Daniel Slamanig.
Blank Digital Signatures.
Cryptology ePrint Archive, Report 2013/130, 2013.
`http://eprint.iacr.org/`.

Aniket Kate, Gregory M. Zaverucha, and Ian Goldberg.
Constant-Size Commitments to Polynomials and Their Applications.
In *ASIACRYPT*, volume 6477 of *LNCS*, pages 177–194. Springer, 2010.