Privacy-Enhancing Proxy Signatures from Non-Interactive Anonymous Credentials

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

July 14, 2014

Outline

- Privacy-enhancing proxy signatures
  - Blank Digital Signatures [HSa]
  - Warrant-Hiding Proxy Signatures [HSb]
  - Applications

Outline

- Privacy-enhancing proxy signatures
  - Blank Digital Signatures [HSa]
  - Warrant-Hiding Proxy Signatures [HSb]
  - Applications
- Building blocks
  - Anonymous credentials
    - Brands' credentials [Bra00]
    - CL credentials [CLa]
  - Non-interactive anonymous credentials

Outline

- Privacy-enhancing proxy signatures
  - Blank Digital Signatures [HSa]
  - Warrant-Hiding Proxy Signatures [HSb]
  - Applications
- Building blocks
  - Anonymous credentials
    - Brands' credentials [Bra00]
    - CL credentials [CLa]
  - Non-interactive anonymous credentials
- Our BDS/WHPS constructions

Outline

- Privacy-enhancing proxy signatures
  - Blank Digital Signatures [HSa]
  - Warrant-Hiding Proxy Signatures [HSb]
  - Applications
- Building blocks
  - Anonymous credentials
    - Brands' credentials [Bra00]
    - CL credentials [CLa]
  - Non-interactive anonymous credentials
- Our BDS/WHPS constructions
- Conclusion

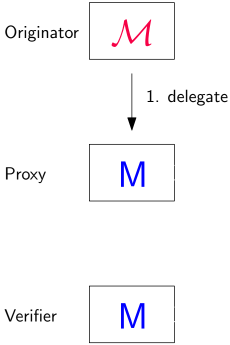
Privacy-Enhancing Proxy Signatures

Originator $\mathcal{M}$

Proxy M

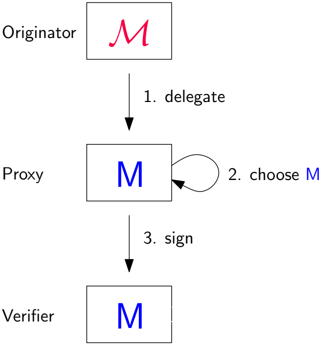
Verifier M

Privacy-Enhancing Proxy Signatures



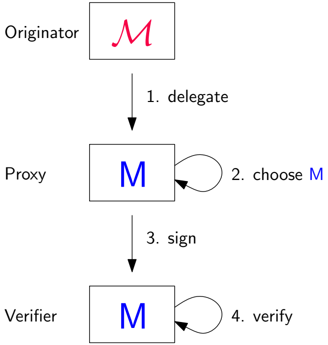
- Delegate signing rights for
  - Message space $\mathcal{M}$

Privacy-Enhancing Proxy Signatures



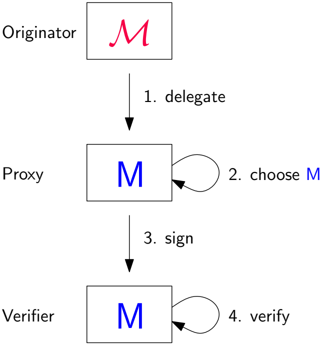
- Delegate signing rights for
  - Message space $\mathcal{M}$
- Choose message M and sign

Privacy-Enhancing Proxy Signatures



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

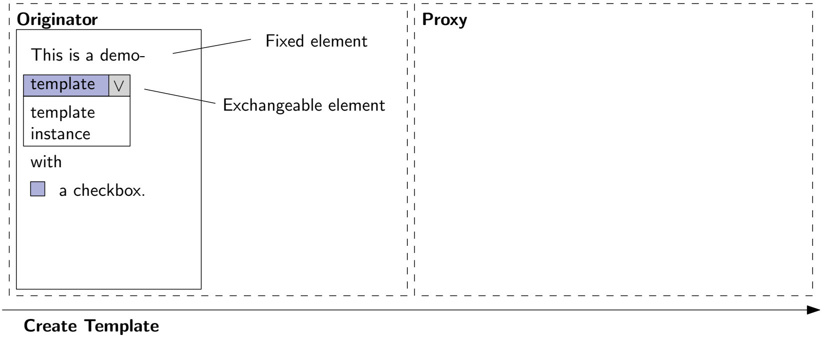
Privacy-Enhancing Proxy Signatures



- Delegate signing rights for
  - Message space $\mathcal{M}$
- Choose message M and sign
- Verify
  - Integrity
  - Authenticity
  - $\mathsf{M} \overset{?}{\in} \mathcal{M}$
- New: **Privacy property**
  - Hides $\mathcal{M} \setminus \mathsf{M}$

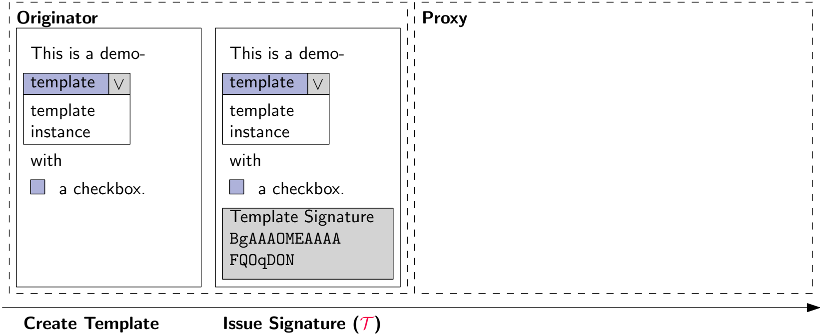
Blank Digital Signatures

- Message space defined by Template



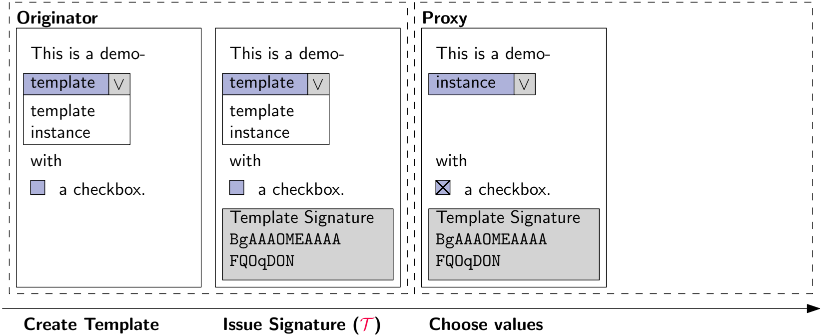
Blank Digital Signatures

- Message space defined by Template



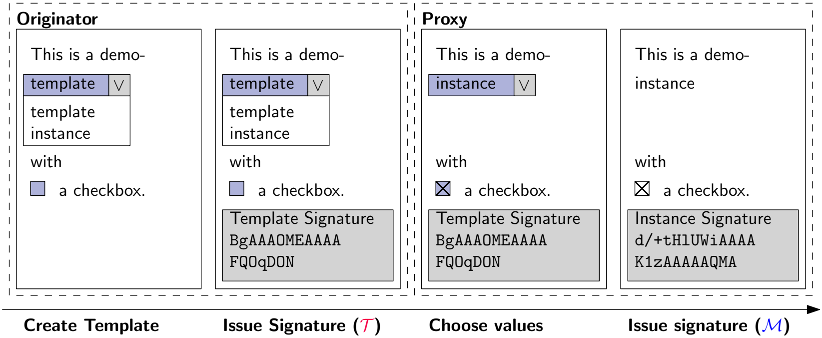
Blank Digital Signatures

- Message space defined by Template



Blank Digital Signatures

- Message space defined by Template



BDS Template/Message Representation

- Template $\mathcal{T} = (T_1, T_2, \ldots, T_n)$ with $T_i = \{M_{i_1}, M_{i_2}, \ldots, M_{i_k}\}$

BDS Template/Message Representation

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

BDS Template/Message Representation

- 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 $\mathcal{M} = (M_i)_{i=1}^{n}$

BDS Security

- Correctness

BDS Security

- Correctness
- Unforgeability
  - Without the knowledge of the respective secret keys it is intractable to (existentially) forge template or message signatures

BDS Security

- Correctness
- Unforgeability
  - Without the knowledge of the respective secret keys it is intractable to (existentially) forge template or message signatures
- Immutability
  - Similar to unforgeability
  - Additional access to proxy's keys and a template with corresponding signature

BDS Security

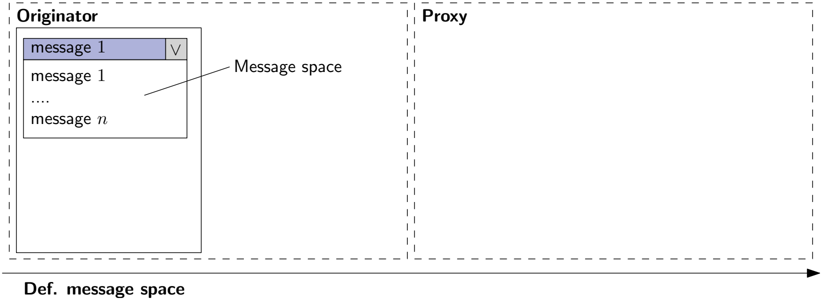
- Correctness
- Unforgeability
  - Without the knowledge of the respective secret keys it is intractable to (existentially) forge template or message signatures
- Immutability
  - Similar to unforgeability
  - Additional access to proxy's keys and a template with corresponding signature

Privacy

Verifier does not learn unused choices in the template

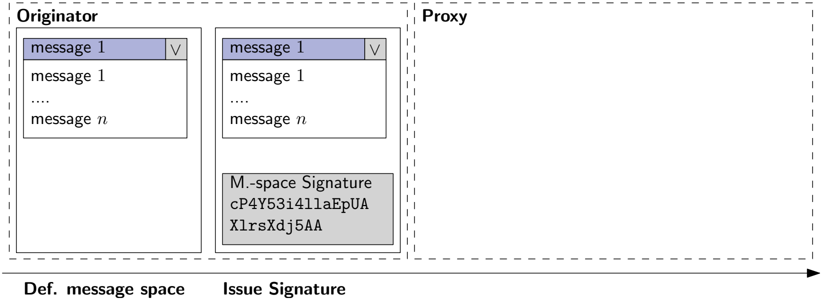
Warrant-Hiding Proxy Signatures

- Message space defined by set of messages



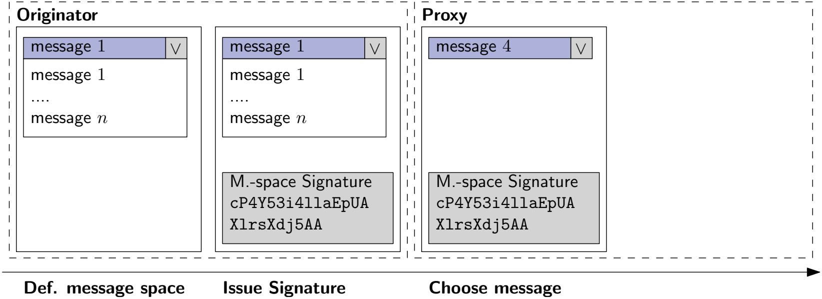
Warrant-Hiding Proxy Signatures

- Message space defined by set of messages



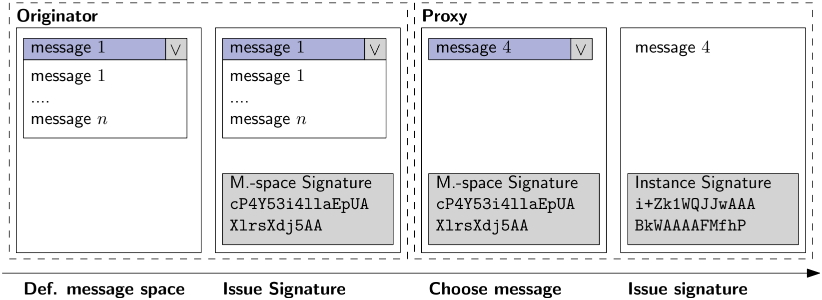
Warrant-Hiding Proxy Signatures

- Message space defined by set of messages



Warrant-Hiding Proxy Signatures

- Message space defined by set of messages



WHPS Message Space Representation

- Message Space $\mathcal{M} = \{M_i\}_{i=1}^{n}$

WHPS Message Space Representation

- Message Space $\mathcal{M} = \{M_i\}_{i=1}^{n}$
- Message $M = M_i, \quad 1 \leq i \leq n$

WHPS Security

- Correctness

WHPS Security

- Correctness
- Unforgeability
  - Without delegator's secret key and the delegation key it is intractable to forge proxy signatures for messages inside/outside the warrant

WHPS Security

- Correctness
- Unforgeability
  - Without delegator's secret key and the delegation key it is intractable to forge proxy signatures for messages inside/outside the warrant

Privacy

Verifier does not learn unrevealed messages in the warrant.

Motivation

- Attorney makes business deal
  - ...on behalf of the client
  - Privacy property

$\mathcal{T} =$
$(\{"I, hereby, declare\ to\ pay\ "\},$
$\{"100\$", "120\$", "150\$"\},$
$\{"for\ this\ device."\})$

Motivation

- Attorney makes business deal
  - ... on behalf of the client
  - Privacy property

$\mathcal{T} =$
$(\{"I, hereby, declare\ to\ pay\ "\},$
$\{"100\$", "120\$", "150\$"\},$
$\{"for\ this\ device."\})$

- Governmental organizations publish forms
  - ... to be signed by any citizen

Motivation

- Attorney makes business deal
  - ... on behalf of the client
  - Privacy property

$$\mathcal{T} = \\ (\{"I, hereby, declare\ to\ pay\ "\}, \\ \{"100\$", "120\$", "150\$"\}, \\ \{"for\ this\ device."\})$$

- Governmental organizations publish forms
  - ... to be signed by any citizen
- Medical files
  - Doctor creates template containing all data
  - Patient can black-out critical parts

Motivation

- Attorney makes business deal
  - ...on behalf of the client
  - Privacy property

$\mathcal{T} =$
$(\{"I, hereby, declare\ to\ pay\ "\},$
$\{"100\$","120\$","150\$"\},$
$\{"for\ this\ device."\})$

- Governmental organizations publish forms
  - ...to be signed by any citizen
- Medical files
  - Doctor creates template containing all data
  - Patient can black-out critical parts
- Warrant-Hiding Proxy Signatures
  - Subset of BDS use cases

Anonymous Credentials

- Parties: Oranization o, Users $\mathsf{u_i}$

Anonymous Credentials

- Parties: Oranization o, Users u_i
- Organization issues credentials to users
  - w.r.t. set of attributes from a certain domain

Anonymous Credentials

- Parties: Oranization o, Users u_i
- Organization issues credentials to users
  - w.r.t. set of attributes from a certain domain
- Users can then anonymously demonstrate possession
  - and, thereby, selectively disclose a subset of attributes

Security of AC

- Correctness

Security of AC

- Correctness
- Unforgeability: The showing of a credential w.r.t. a set of attributes only succeeds when such a credential was issued for the user

Security of AC

- Correctness
- Unforgeability: The showing of a credential w.r.t. a set of attributes only succeeds when such a credential was issued for the user
- Anonymity: No one should be able to find anything about the user
  - Except for the fact that she owns a valid credential

Security of AC

- Correctness
- Unforgeability: The showing of a credential w.r.t. a set of attributes only succeeds when such a credential was issued for the user
- Anonymity: No one should be able to find anything about the user
  - Except for the fact that she owns a valid credential

Selective Disclosure

- Verifier learns nothing about non-shown attributes
- Informal requirement of all AC systems
- All known AC systems employ proofs of knowledge
  - Nothing beyond the shown attributes revealed by definition

Brands' Credentials

- Group $\mathbb{G}$ of prime order p (additive notation)

Brands' Credentials

- Group $\mathbb{G}$ of prime order p (additive notation)
- Generators $(P_1, \ldots, P_n) \in \mathbb{G}^n$
  - discrete logarithms between P_i unknown to users

Brands' Credentials

- Group $\mathbb{G}$ of prime order p (additive notation)
- Generators $(P_1, \ldots, P_n) \in \mathbb{G}^n$
  - discrete logarithms between P_i unknown to users
- Commit to attributes $(a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ using
  - DLREP: $H \leftarrow \sum_{i=1}^{n} a_i P_i$
  - Generalized Pedersen commitment with additional blinding

Brands' Credentials

- Group $\mathbb{G}$ of prime order p (additive notation)
- Generators $(P_1, \ldots, P_n) \in \mathbb{G}^n$
  - discrete logarithms between P_i unknown to users
- Commit to attributes $(a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ using
  - DLREP: $H \leftarrow \sum_{i=1}^{n} a_i P_i$
  - Generalized Pedersen commitment with additional blinding
- Issue a variant of a blind signature on H
  - Interpreted as credential

Brands' Credentials

- Group $\mathbb{G}$ of prime order p (additive notation)
- Generators $(P_1, \ldots, P_n) \in \mathbb{G}^n$
  - discrete logarithms between P_i unknown to users
- Commit to attributes $(a_1, \ldots, a_n) \in \mathbb{Z}_p^n$ using
  - DLREP: $H \leftarrow \sum_{i=1}^{n} a_i P_i$
  - Generalized Pedersen commitment with additional blinding
- Issue a variant of a blind signature on H
  - Interpreted as credential
- Showing
  - Verify blind signature
  - Prove knowledge of DLREP
  - Multiple showings are linkable

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
- A signature $\sigma = (R, A_i, B, B_i, C)$ is interpreted as credential:

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
- A signature $\sigma = (R, A_i, B, B_i, C)$ is interpreted as credential:
  - σ for a sequence of $n + 1$ attributes $(a_0, \ldots, a_n) \in \mathbb{Z}_p^{n+1}$,

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.
- A signature $\sigma = (R, A_i, B, B_i, C)$ is interpreted as credential:
  - σ for a sequence of $n+1$ attributes $(a_0, \ldots, a_n) \in \mathbb{Z}_p^{n+1}$,
  - w.r.t. the secret key $(x, y, z_1, \ldots z_n) \in \mathbb{Z}_p^{n+2}$:

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$.
- A signature $\sigma = (R, A_i, B, B_i, C)$ is interpreted as credential:
  - σ for a sequence of $n+1$ attributes $(a_0, \ldots, a_n) \in \mathbb{Z}_p^{n+1}$,
  - w.r.t. the secret key $(x, y, z_1, \ldots z_n) \in \mathbb{Z}_p^{n+2}$:
  - $R \xleftarrow{R} \mathbb{G}, \quad A_i \leftarrow z_i R, \quad B \leftarrow yR, \quad B_i \leftarrow yA_i,$
    $C \leftarrow (x + xya_0)R + \sum_{i=1}^{n} xya_i A_i$

CL Credentials

- Based on the CL Signature Scheme
  - Signatures are re-randomizable
- Instantiations in the known- and hidden-order group setting
- Group $\mathbb{G}$ of prime order p with a bilinear map $e : \mathbb{G} \times \mathbb{G} \to \mathbb{G}_T$.
- A signature $\sigma = (R, A_i, B, B_i, C)$ is interpreted as credential:
  - σ for a sequence of $n+1$ attributes $(a_0, \ldots, a_n) \in \mathbb{Z}_p^{n+1}$,
  - w.r.t. the secret key $(x, y, z_1, \ldots z_n) \in \mathbb{Z}_p^{n+2}$:
  - $R \xleftarrow{R} \mathbb{G}, \quad A_i \leftarrow z_i R, \quad B \leftarrow yR, \quad B_i \leftarrow yA_i,$
    $C \leftarrow (x + xya_0)R + \sum_{i=1}^{n} xya_i A_i$
- Showing
  - Verify re-randomized signature
  - Prove knowledge of attributes in C
  - Multiple showings unlinkable
    - Not needed in our context

Obtaining Non-interactive AC

- Honest-verifier zero-knowledge proofs used upon show
  - e.g., demonstrate knowledge of $x = \log_P Y$ to base P
  - ...only reveal that the prover knows x

Obtaining Non-interactive AC

- Honest-verifier zero-knowledge proofs used upon show
  - e.g., demonstrate knowledge of $x = \log_P Y$ to base P
  - …only reveal that the prover knows x
- Non-interactive AC Versions
  - Apply Fiat-Shamir transform [FS] to proofs

Obtaining Non-interactive AC

- Honest-verifier zero-knowledge proofs used upon show
  - e.g., demonstrate knowledge of $x = \log_P Y$ to base P
  - …only reveal that the prover knows x
- Non-interactive AC Versions
  - Apply Fiat-Shamir transform [FS] to proofs
- Non-interactive Proof
  - …together with proving knowledge of a secret key
  - Secure digital signature in the random oracle model [CLb]
  - Interpreted as the proxy's signature

Bringing it Together

- Credentials encode a finite set of attributes

Bringing it Together

- Credentials encode a finite set of attributes
- . . . and allow to disclose a subset of the attributes upon showing

Bringing it Together

- Credentials encode a finite set of attributes
- ... and allow to disclose a subset of the attributes upon showing
- Why not use this for BDS/WHPS?
  - Encode template elements/message space within attributes

Bringing it Together

- Credentials encode a finite set of attributes
- . . . and allow to disclose a subset of the attributes upon showing
- Why not use this for BDS/WHPS?
  - Encode template elements/message space within attributes
  - Provide non-interactive showings
    - Reveal subset of the attributes
    - Prove knowledge of secret key and remaining attributes

BDS Encoding

- Template uniquely defined by its elements
  - Fixed elements
    - Position i in the template
    - Corresponding message m_i
  - Exchangeable elements
    - Position i in the template
    - j messages m_{i_j}

BDS Encoding

- Template uniquely defined by its elements
  - Fixed elements
    - Position i in the template
    - Corresponding message m_i
  - Exchangeable elements
    - Position i in the template
    - j messages m_{i_j}
- Hashing them together
  - Collision resistant hash function
  - Mapping to the attribute domain

BDS Encoding

- Template uniquely defined by its elements
  - Fixed elements
    - Position i in the template
    - Corresponding message m_i
  - Exchangeable elements
    - Position i in the template
    - j messages m_{i_j}
- Hashing them together
  - Collision resistant hash function
  - Mapping to the attribute domain
- Template element $\mapsto$ AC attribute

$$\mathcal{T} = (\{m_{1_1}\}, \{m_{2_1}, m_{2_2}, m_{2_3}\})$$
$$\downarrow$$
$$\mathcal{T}^{\text{enc}} = (H(m_{1_1}||1), H(m_{2_1}||2), H(m_{2_2}||2), H(m_{2_3}||2))$$

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$
- Most credential systems implicitly assign order to attributes

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$
- Most credential systems implicitly assign order to attributes
- Template structure may leak
  - Last two attributes are not shown
  - $\implies$ exchangeable element has cardinality 3

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$
- Most credential systems implicitly assign order to attributes
- Template structure may leak
  - Last two attributes are not shown
  - $\implies$ exchangeable element has cardinality 3
- Thus apply a secret random permutation ϕ to $\mathcal{T}^{\mathsf{enc}}$
  - $(H(m_{2_2}||2), H(m_{2_1}||2), H(m_{1_1}||1), H(m_{2_3}||2))$

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$
- Most credential systems implicitly assign order to attributes
- Template structure may leak
  - Last two attributes are not shown
  - $\implies$ exchangeable element has cardinality 3
- Thus apply a secret random permutation ϕ to $\mathcal{T}^{\mathsf{enc}}$
  - $(H(m_{2_2}||2), H(m_{2_1}||2), H(m_{1_1}||1), H(m_{2_3}||2))$
- ... and the same permutation ϕ to $\mathcal{M}^{\mathsf{enc}}$
  - $(\blacksquare, H(m_{2_1}||2), H(m_{1_1}||1), \blacksquare)$

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\text{enc}}$
- Most credential systems implicitly assign order to attributes
- Template structure may leak
  - Last two attributes are not shown
  - $\implies$ exchangeable element has cardinality 3
- Thus apply a secret random permutation ϕ to $\mathcal{T}^{\text{enc}}$
  - $(H(m_{2_2}||2), H(m_{2_1}||2), H(m_{1_1}||1), H(m_{2_3}||2))$
- ... and the same permutation ϕ to $\mathcal{M}^{\text{enc}}$
  - $(\blacksquare, H(m_{2_1}||2), H(m_{1_1}||1), \blacksquare)$
- Encode number of elements l into first attribute
  - Always opened

BDS Encoding (2)

- Template instantiation
  - $\mathcal{M} = (m_{1_1}, m_{2_1}) \mapsto (H(m_{1_1}||1), H(m_{2_1}||2), \blacksquare, \blacksquare) = \mathcal{M}^{\mathsf{enc}}$
- Most credential systems implicitly assign order to attributes
- Template structure may leak
  - Last two attributes are not shown
  - $\implies$ exchangeable element has cardinality 3
- Thus apply a secret random permutation ϕ to $\mathcal{T}^{\mathsf{enc}}$
  - $(H(m_{2_2}||2), H(m_{2_1}||2), H(m_{1_1}||1), H(m_{2_3}||2))$
- ... and the same permutation ϕ to $\mathcal{M}^{\mathsf{enc}}$
  - $(\blacksquare, H(m_{2_1}||2), H(m_{1_1}||1), \blacksquare)$
- Encode number of elements l into first attribute
  - Always opened
- Ensure that one attribute m_{i_j} is shown for each $1 \leq i \leq l$

WHPS Encoding

- Message space $\mathcal{M}$ defined by contained messages m_i

WHPS Encoding

- Message space $\mathcal{M}$ defined by contained messages m_i
- Encoding a lot simpler
  - No order of messages in the message space
  - Random permutation not needed
  - . . . no useful information leaks

WHPS Encoding

- Message space $\mathcal{M}$ defined by contained messages m_i
- Encoding a lot simpler
  - No order of messages in the message space
  - Random permutation not needed
  - ... no useful information leaks
- $\mathcal{M} = \{m_1, \ldots, m_n\} \mapsto (H(m_1), \ldots, H(m_n))$

WHPS Encoding

- Message space $\mathcal{M}$ defined by contained messages m_i
- Encoding a lot simpler
  - No order of messages in the message space
  - Random permutation not needed
  - ... no useful information leaks
- $\mathcal{M} = \{m_1, \ldots, m_n\} \mapsto (H(m_1), \ldots, H(m_n))$
- Instantiation: $\{\blacksquare, \ldots, \blacksquare, \ldots, H(m_i), \ldots, \blacksquare\}$

Modeling the Delegation

- Keys compatible with system parameters of used ACs
  - Secret key $\mathsf{sk} \in \mathbb{Z}_p^*$
  - Public key $\mathsf{pk} = \mathsf{sk} \cdot P$ (P generates used group $\mathbb{G}$)

Modeling the Delegation

- Keys compatible with system parameters of used ACs
  - Secret key $\mathsf{sk} \in \mathbb{Z}_p^*$
  - Public key $\mathsf{pk} = \mathsf{sk} \cdot P$ (P generates used group $\mathbb{G}$)
- In addition to encoded attributes
  - Incorporate sk as attribute without disclosing it
  - ... by using pk as public commitment

Modeling the Delegation

- Keys compatible with system parameters of used ACs
  - Secret key $\mathsf{sk} \in \mathbb{Z}_p^*$
  - Public key $\mathsf{pk} = \mathsf{sk} \cdot P$ (P generates used group $\mathbb{G}$)
- In addition to encoded attributes
  - Incorporate sk as attribute without disclosing it
  - ... by using pk as public commitment
- Possible for Brands' and CL credentials

Modeling the Delegation

- Keys compatible with system parameters of used ACs
  - Secret key $\mathsf{sk} \in \mathbb{Z}_p^*$
  - Public key $\mathsf{pk} = \mathsf{sk} \cdot P$ (P generates used group $\mathbb{G}$)
- In addition to encoded attributes
  - Incorporate sk as attribute without disclosing it
  - ... by using pk as public commitment
- Possible for Brands' and CL credentials
- If not
  - Incorporate public key as attribute
  - Prove knowledge by providing a signature

Security

- Although similar goals
  - BDS and WHPS rely on different security models

Security

- Although similar goals
  - BDS and WHPS rely on different security models
- Correctness notions are compatible

Security

- Although similar goals
  - BDS and WHPS rely on different security models
- Correctness notions are compatible
- BDS
  - $\mathit{AC.Unforgeability} \implies \mathit{BDS.Unforgeability}$
  - $\mathit{AC.Unforgeability} \implies \mathit{BDS.Immutability}$
  - $\mathit{AC.SelectiveDisclosure} \implies \mathit{BDS.Privacy}$

Security

- Although similar goals
  - BDS and WHPS rely on different security models
- Correctness notions are compatible
- BDS
  - $\mathit{AC.Unforgeability} \implies \mathit{BDS.Unforgeability}$
  - $\mathit{AC.Unforgeability} \implies \mathit{BDS.Immutability}$
  - $\mathit{AC.SelectiveDisclosure} \implies \mathit{BDS.Privacy}$
- WHPS
  - $\mathit{AC.Unforgeability} \implies \mathit{WHPS.Unforgeability}$
  - $\mathit{AC.SelectiveDisclosure} \implies \mathit{WHPS.Privacy}$

Conclusion

- Performance quite comparable
  - Linear signature sizes in our constructions
  - Templates quite small in most practical use cases

Conclusion

- Performance quite comparable
  - Linear signature sizes in our constructions
  - Templates quite small in most practical use cases
- Multiple implementations Brands' and CL Credentials
  - e.g. EU Project ABC4Trust
  - Basis for practical implementations

Conclusion

- Performance quite comparable
  - Linear signature sizes in our constructions
  - Templates quite small in most practical use cases
- Multiple implementations Brands' and CL Credentials
  - e.g. EU Project ABC4Trust
  - Basis for practical implementations
- Flexibility regarding underlying constructions

Conclusion

- Performance quite comparable
  - Linear signature sizes in our constructions
  - Templates quite small in most practical use cases
- Multiple implementations Brands' and CL Credentials
  - e.g. EU Project ABC4Trust
  - Basis for practical implementations
- Flexibility regarding underlying constructions
- First approach to build special signature schemes from AC
  - Inspiration for other constructions

Conclusion

- Performance quite comparable
  - Linear signature sizes in our constructions
  - Templates quite small in most practical use cases
- Multiple implementations Brands' and CL Credentials
  - e.g. EU Project ABC4Trust
  - Basis for practical implementations
- Flexibility regarding underlying constructions
- First approach to build special signature schemes from AC
  - Inspiration for other constructions
- Proposed encoding might also be useful for AC

Thank you.

david.derler@iaik.tugraz.at

Extended Version: http://eprint.iacr.org/2014/285

Stefan Brands.
Rethinking Public-Key Infrastructures and Digital Certificates: Building in Privacy.
MIT Press, 2000.

Jan Camenisch and Anna Lysyanskaya.
Signature Schemes and Anonymous Credentials from Bilinear Maps.
In *CRYPTO'04*, volume 3152 of *LNCS*, pages 56–72.

Melissa Chase and Anna Lysyanskaya.
On Signatures of Knowledge.
In *CRYPTO'06*, volume 4117 of *LNCS*, pages 78–96.

Amos Fiat and Adi Shamir.
How to Prove Yourself: Practical Solutions to Identification and Signature Problems.
In *CRYPTO'87*, volume 263 of *LNCS*, pages 186–194.

Christian Hanser and Daniel Slamanig.
Blank Digital Signatures.
In *ACM ASIACCS'13*, pages 95–106. ACM.
ext.: IACR ePrint 2013/130.

Christian Hanser and Daniel Slamanig.
Warrant-Hiding Delegation-by-Certificate Proxy Signature Schemes.
In *INDOCRYPT'13*, volume 8250 of *LNCS*.
ext.: IACR ePrint 2013/544.