3PBCS: A Privacy-Preserving, Personhood-Based Credential System

Ksandros Apostoli, M.Sc. Cybersecurity
Supervised by: Simone Colombo (EPFL/DEDIS), Daniel Moser (CYD Campus)
Professor: Dr. Bryan Ford (EPFL/DEDIS)
Project Type: Master Thesis Project

Schweizerische Eidgenossenschaft
Confédération suisse
Confederazione Svizzera
Confederaziun svizra

armasuisse Science and Technology
Cyber-Defence Campus

EPFL

DEDIS
Decentralized and Distributed Systems

March 22, 2022

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).
. . . typically containing excessive amounts of **Personal Identifiable Information**.

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).
. . . typically containing excessive amounts of **Personal Identifiable Information**.

The average user:

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).
... typically containing excessive amounts of **Personal Identifiable Information**.

The average user:

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).
... typically containing excessive amounts of **Personal Identifiable Information**.

The average user:

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).
. . . typically containing excessive amounts of **Personal Identifiable Information**.

The average user:

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

State-of-the-Art misuse practices:

Motivation: Credential Misuse 101

Credential: A set of one or more claims made by the same entity (W3C, 2021).

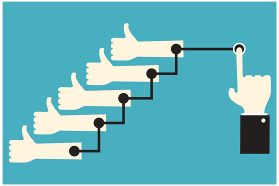
... typically containing excessive amounts of **Personal Identifiable Information**.

The average user:

- Holds **100** sets of *Digital Credentials* (Williams, 2020),
- Uses credentials to engage in **45** *Authentication Events* daily (Mare et al., 2016).

State-of-the-Art misuse practices:

Weak Privacy Guarantees

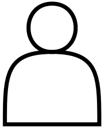


Lack of Sybil-Resistance

Lack of Accountability

What is on the Menu

Challenges in the Design of Credential Systems

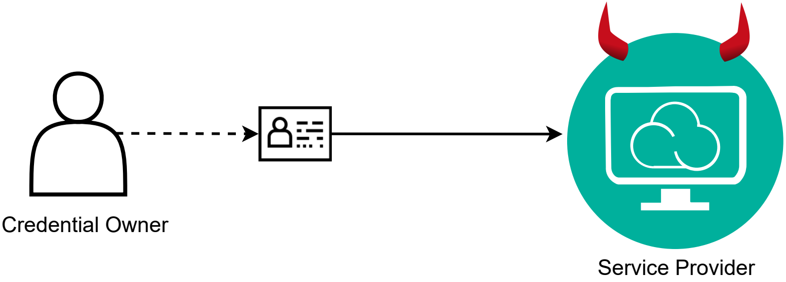


Credential Owner

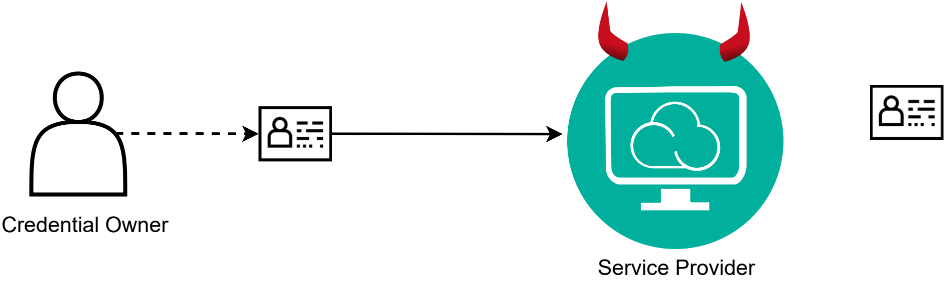
Service Provider

Challenges in the Design of Credential Systems

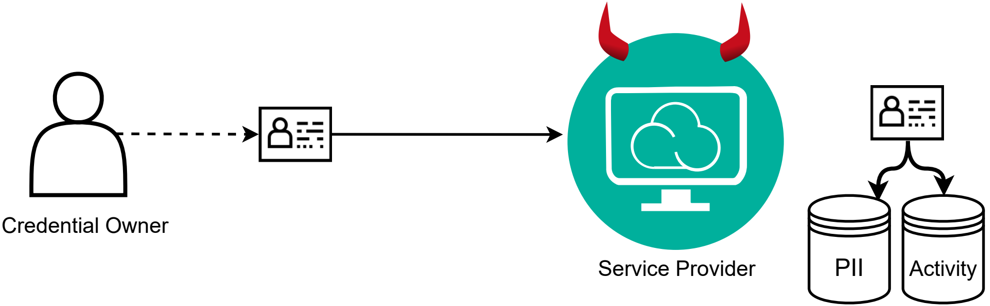
Challenges in the Design of Credential Systems



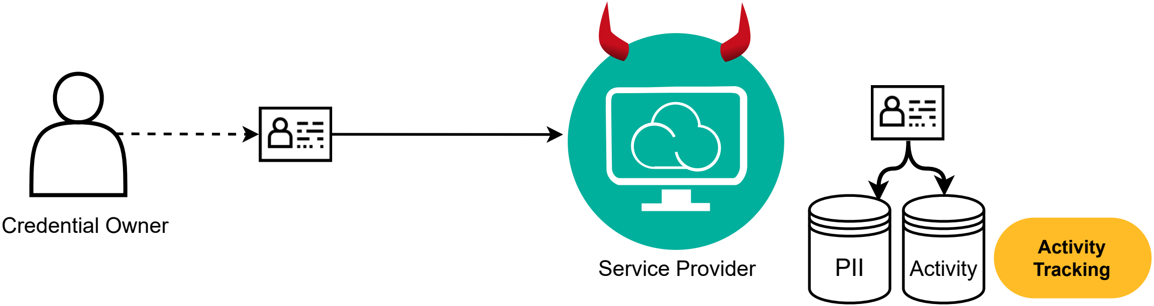
Challenges in the Design of Credential Systems



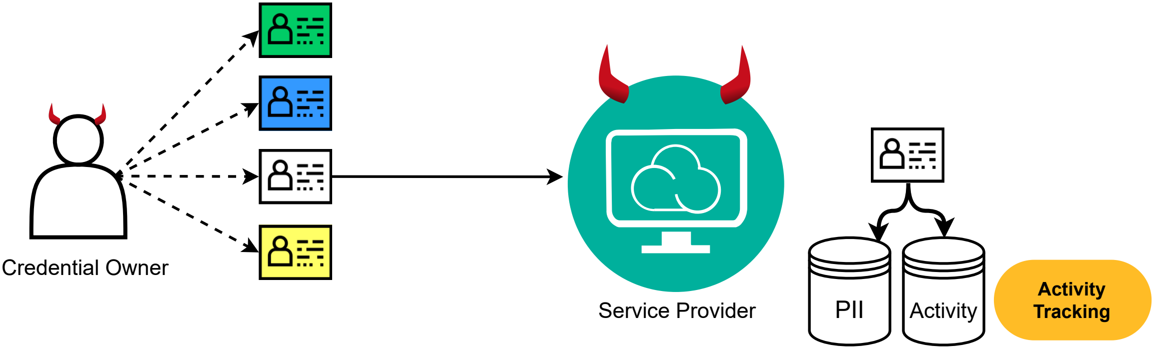
Challenges in the Design of Credential Systems



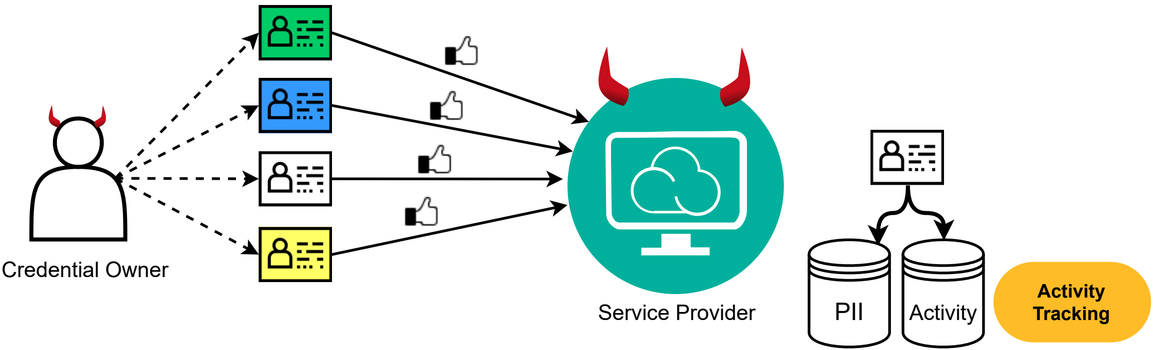
Challenges in the Design of Credential Systems



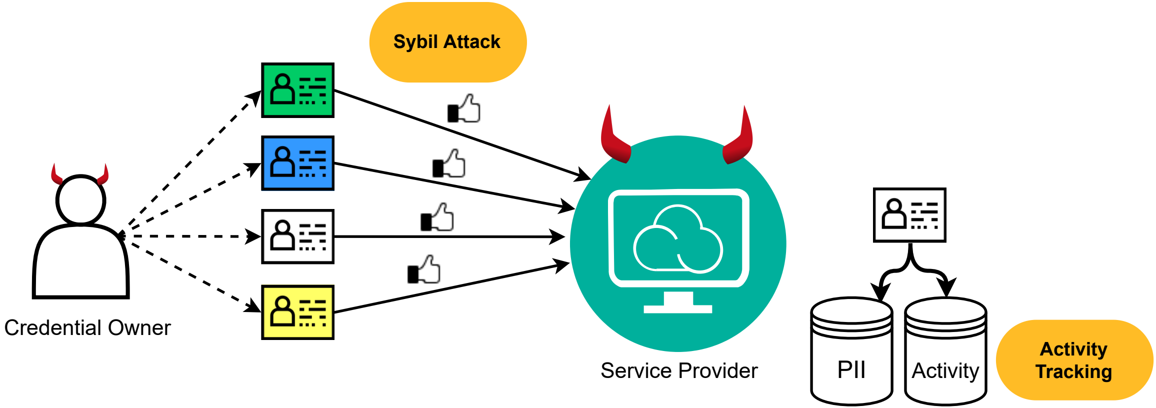
Challenges in the Design of Credential Systems



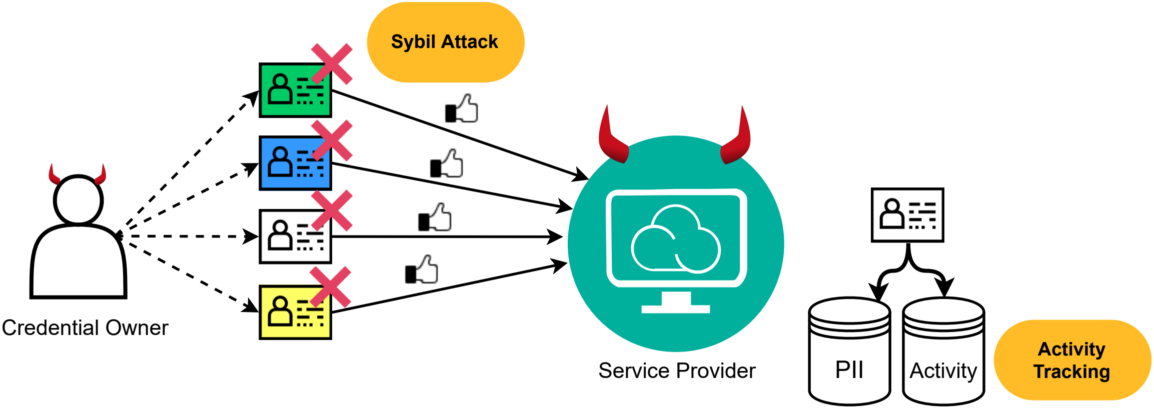
Challenges in the Design of Credential Systems



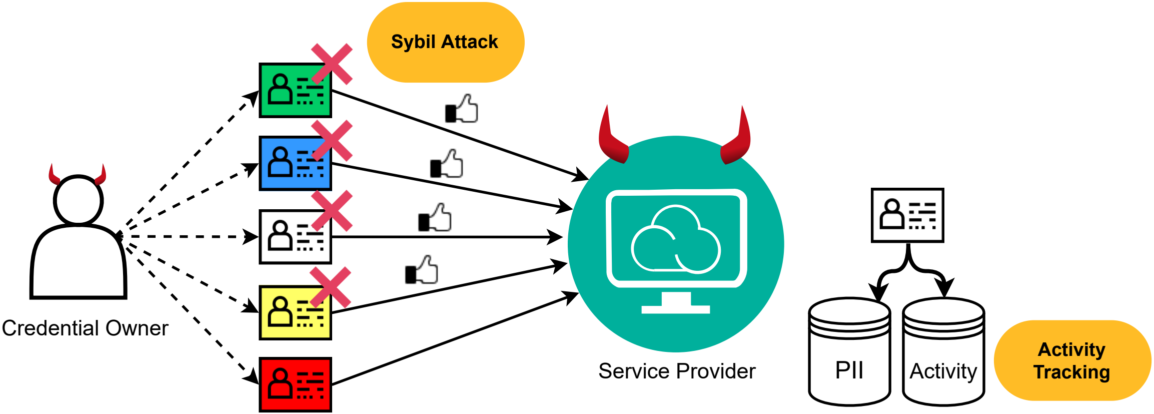
Challenges in the Design of Credential Systems



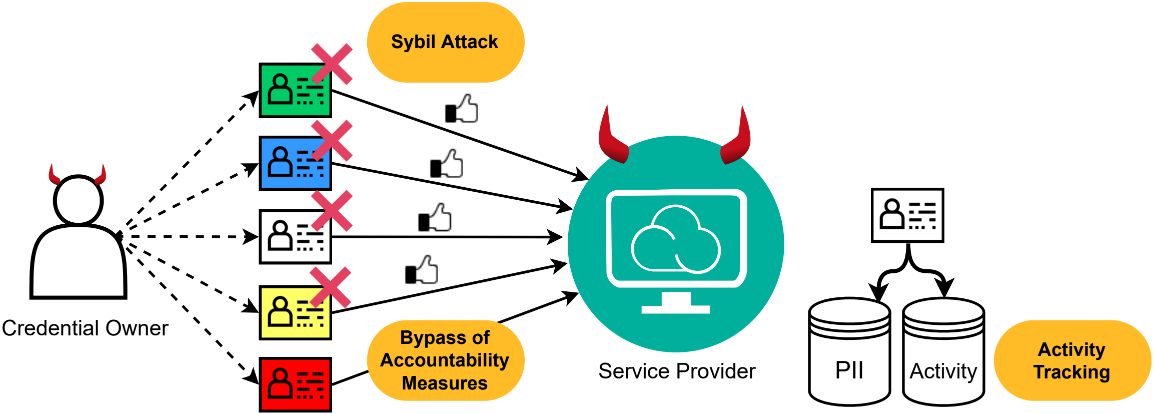
Challenges in the Design of Credential Systems



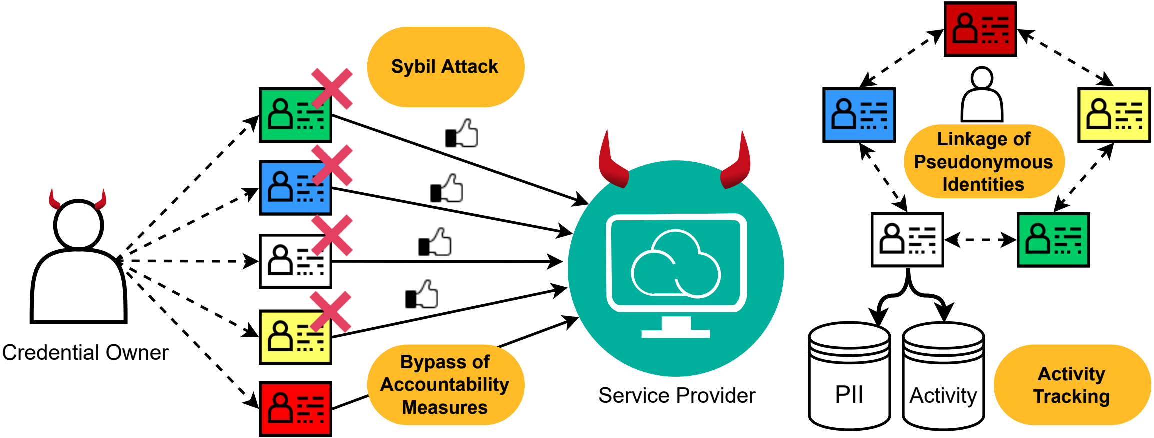
Challenges in the Design of Credential Systems



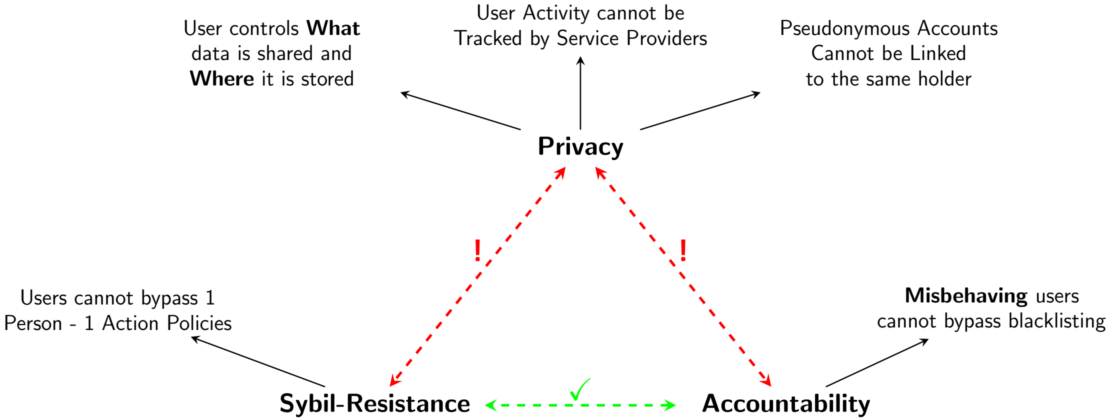
Challenges in the Design of Credential Systems



Challenges in the Design of Credential Systems



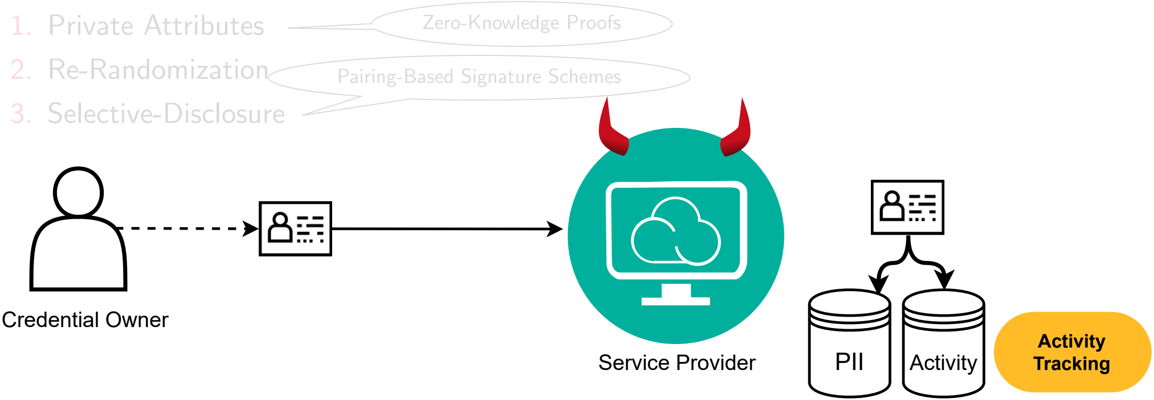
Security and Privacy Goals



Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

1. Private Attributes
2. Re-Randomization
3. Selective-Disclosure



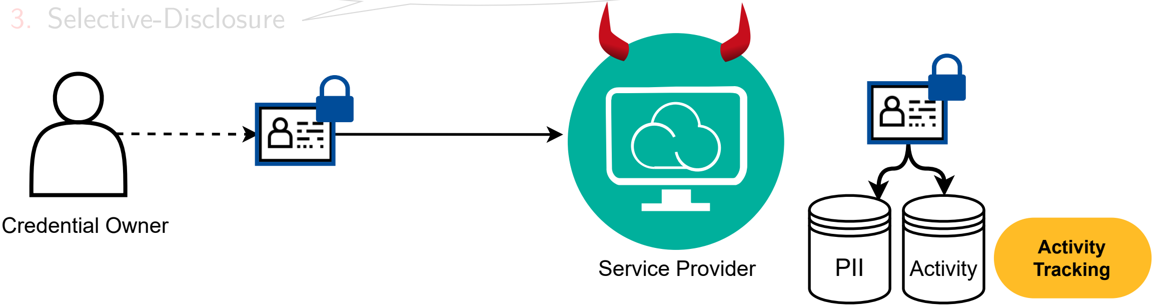
Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

1. Private Attributes
2. Re-Randomization
3. Selective-Disclosure

Zero-Knowledge Proofs

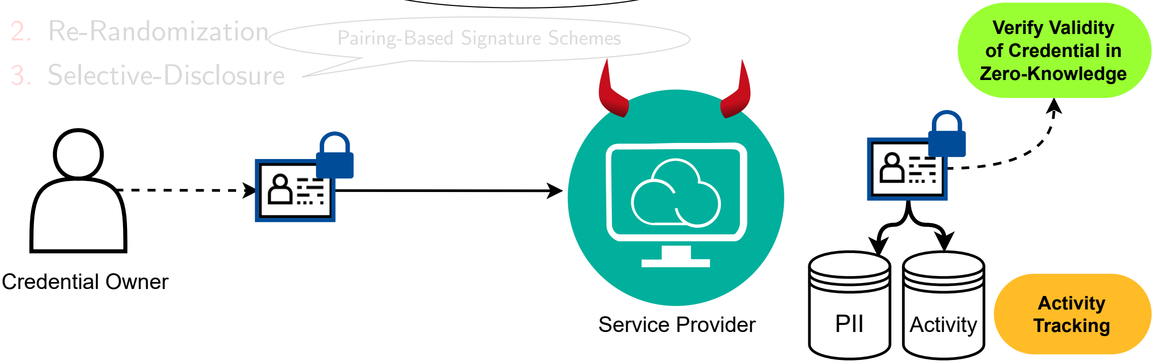
Pairing-Based Signature Schemes



Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

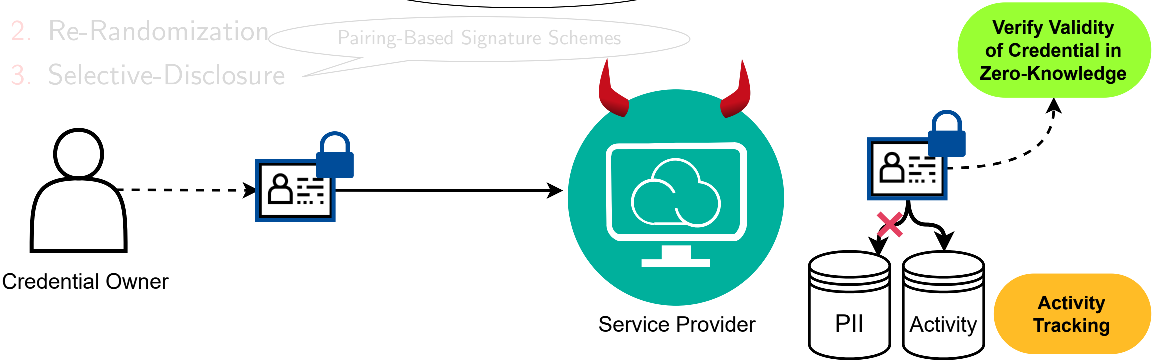
1. Private Attributes — Zero-Knowledge Proofs
2. Re-Randomization — Pairing-Based Signature Schemes
3. Selective-Disclosure



Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

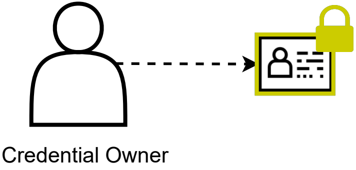
1. Private Attributes — Zero-Knowledge Proofs
2. Re-Randomization — Pairing-Based Signature Schemes
3. Selective-Disclosure

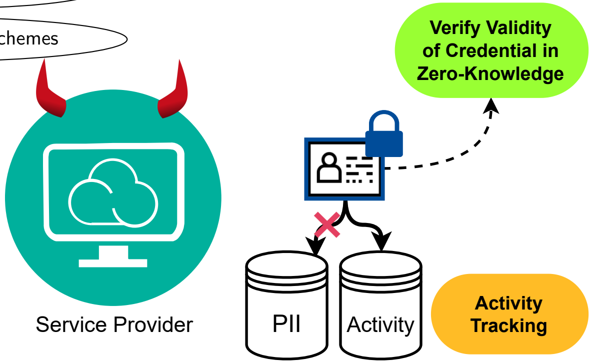


Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

1. Private Attributes — Zero-Knowledge Proofs
2. Re-Randomization — Pairing-Based Signature Schemes
3. Selective-Disclosure — Pairing-Based Signature Schemes

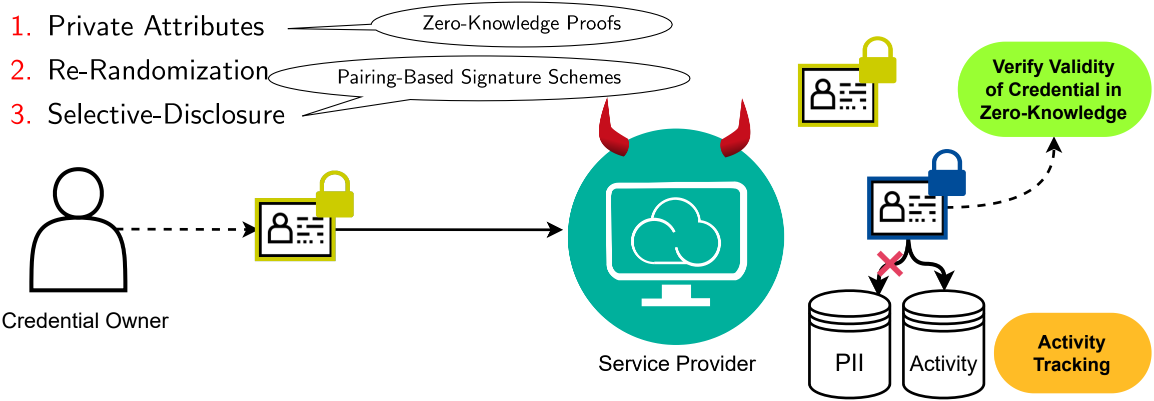




Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

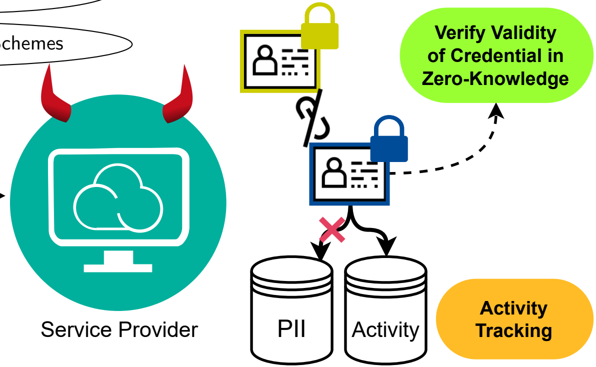
1. Private Attributes
2. Re-Randomization
3. Selective-Disclosure



Existing Solutions: Anonymous Credential Schemes

✓ Privacy-Enhancing Features:

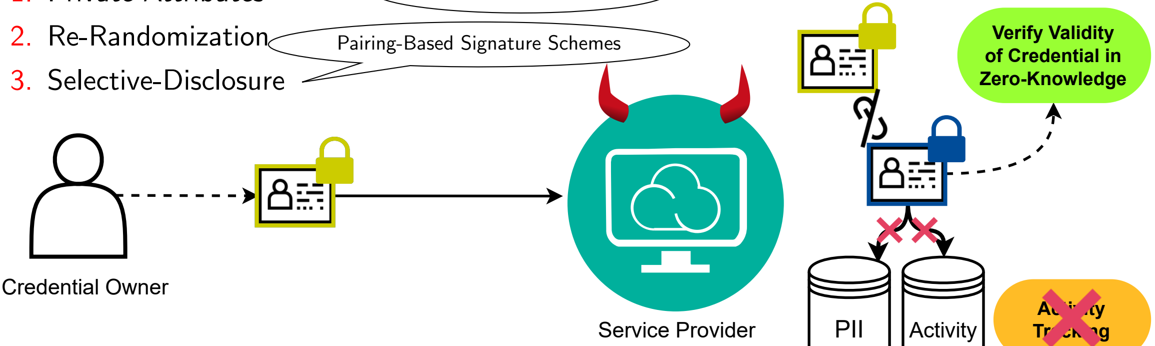
1. Private Attributes — Zero-Knowledge Proofs
2. Re-Randomization — Pairing-Based Signature Schemes
3. Selective-Disclosure



Existing Solutions: Anonymous Credential Schemes

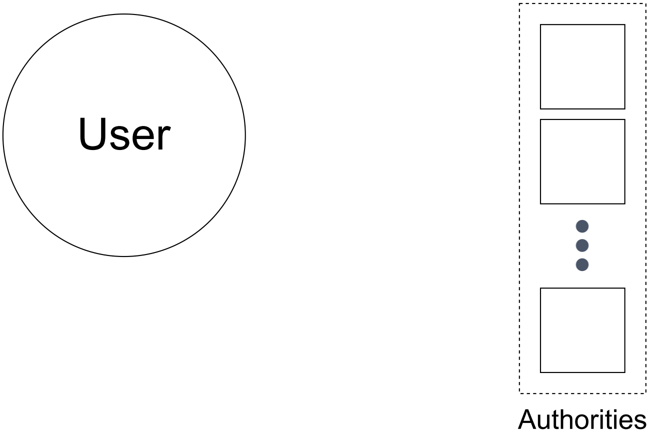
✓ Privacy-Enhancing Features:

1. Private Attributes — Zero-Knowledge Proofs
2. Re-Randomization — Pairing-Based Signature Schemes
3. Selective-Disclosure — Pairing-Based Signature Schemes



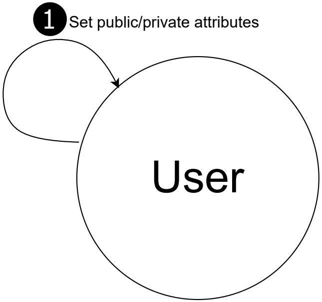
✗ Sybil-Resistance; ✗ Accountability

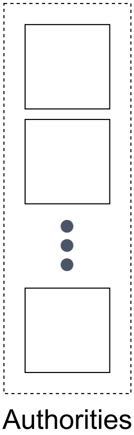
Existing Solutions: *Coconut* (Sonnino et al., 2018)



1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

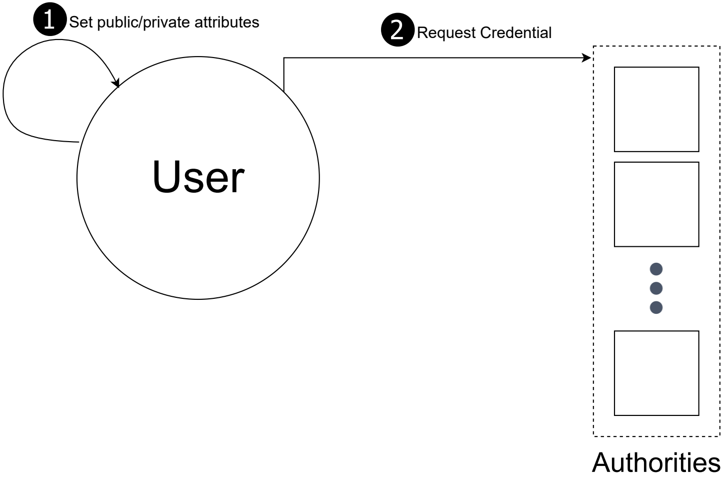
Existing Solutions: *Coconut* (Sonnino et al., 2018)





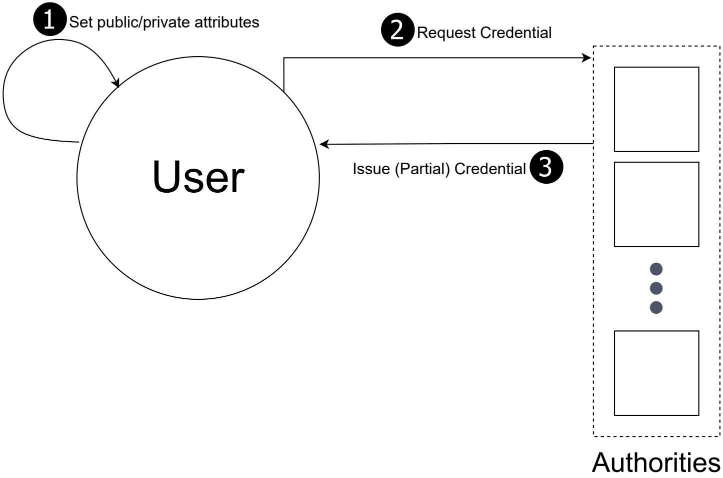
1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Coconut* (Sonnino et al., 2018)



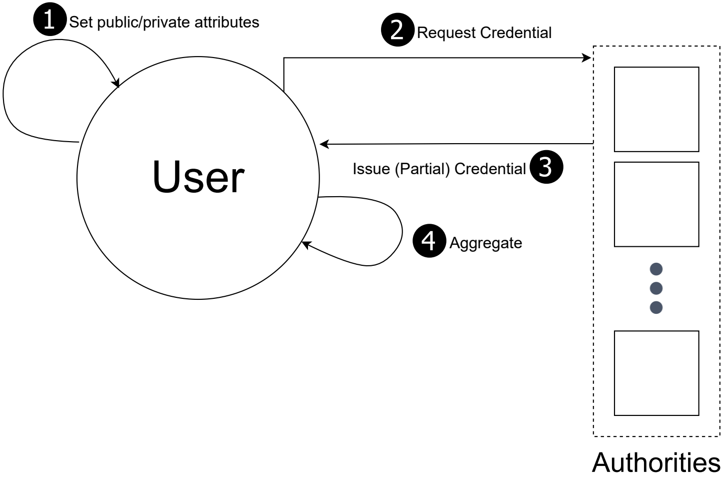
1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Coconut* (Sonnino et al., 2018)



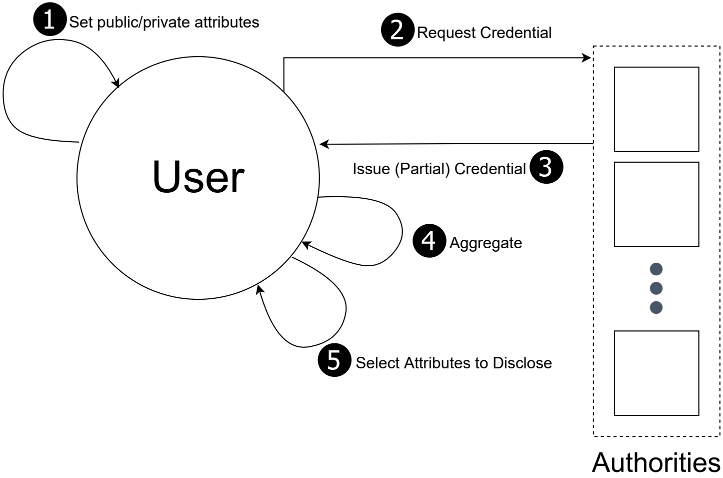
1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Coconut* (Sonnino et al., 2018)



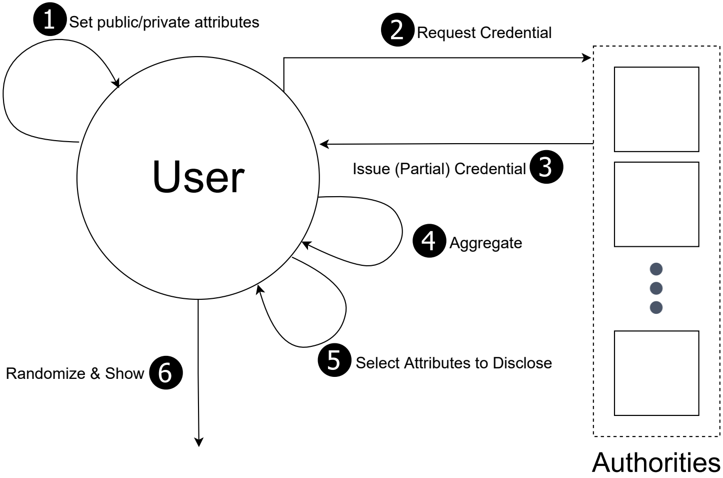
1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Coconut* (Sonnino et al., 2018)



1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Coconut* (Sonnino et al., 2018)



1. Choose attributes to show in clear.
2. For private attributes, provide commitments, together with ZKP on their validity.
3. Upon verification, sign (partial) credential.
4. Aggregate threshold of partial signatures.
5. Choose what to disclose.
6. Make credential presentation unlinkable from other presentations.

Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**

Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

$\rightarrow$ Create cryptographic artifacts that remain unique per person across different digital identities.

- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**

Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

→ Create cryptographic artifacts that remain unique per person across different digital identities.

How to Obtain Proof-of-Personhood? (Ford, 2020)

- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**

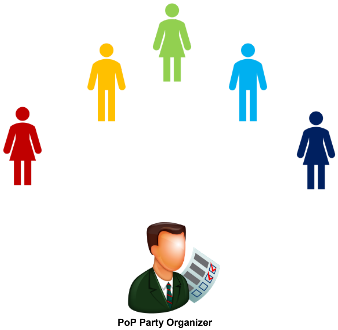
Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

$\rightarrow$ Create cryptographic artifacts that remain unique per person across different digital identities.

How to Obtain Proof-of-Personhood? (Ford, 2020)

- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**



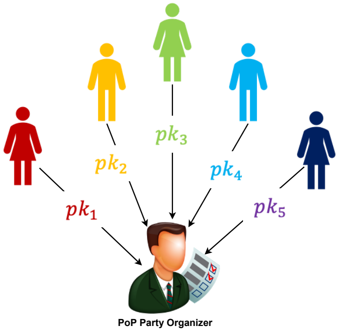
Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

→ Create cryptographic artifacts that remain unique per person across different digital identities.

How to Obtain Proof-of-Personhood? (Ford, 2020)

- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**



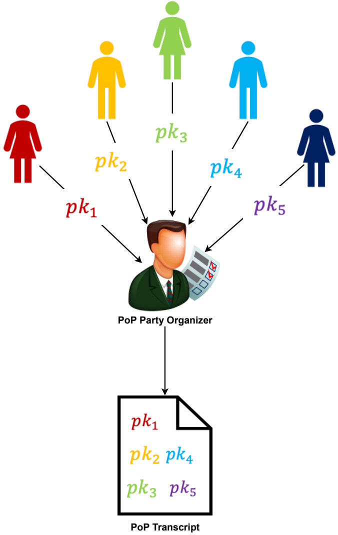
Existing Solutions: *Proof-of-Personhood* (Borge et al., 2017)

Idea: Bind every digital identity to a physical entity.

→ Create cryptographic artifacts that remain unique per person across different digital identities.

How to Obtain Proof-of-Personhood? (Ford, 2020)

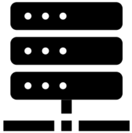
- User generates a (sk_u, pk_u) key-pair
- User presents pk_u to ta physical gathering known as a **PoP Party**
- **PoP Parties** conclude with organizers generating a list of all public keys, i.e. **PoP Transcript**
- (sk_u, pk_u) becomes the **PoP Token**



Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)

Alice

PoP Transcript

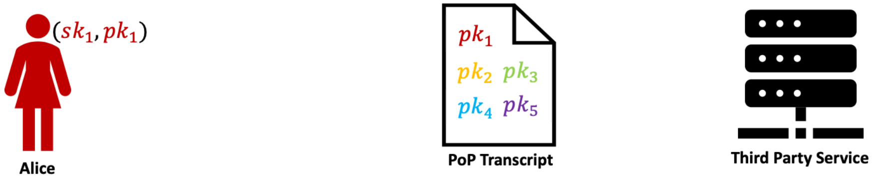


Third Party Service

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)

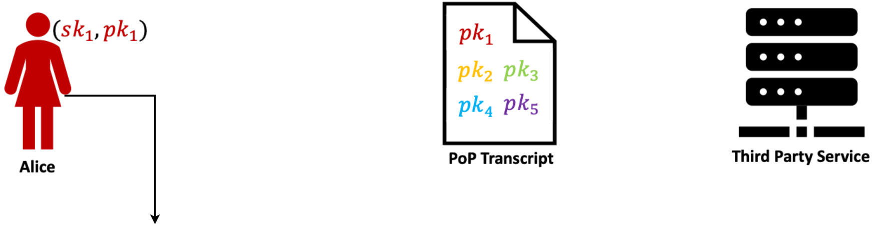


$LRS.\boldsymbol{Sign}(m, sk_1, pk_1, pk_2, pk_3, pk_4, pk_5)$

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)

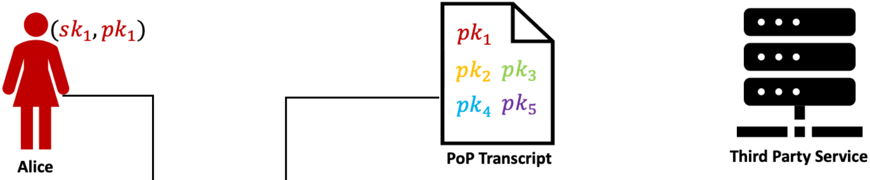


$LRS.\boldsymbol{Sign}(m, sk_1, pk_1, pk_2, pk_3, pk_4, pk_5)$

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)

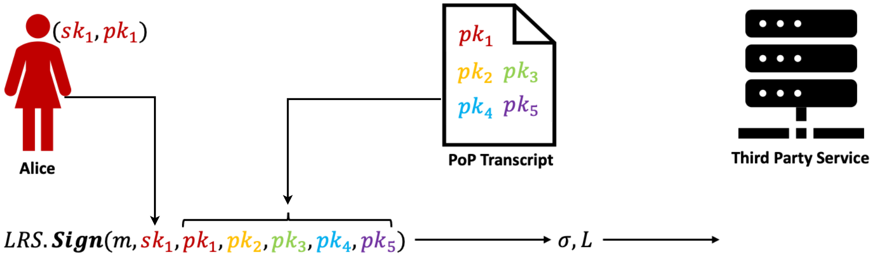


$$LRS.\mathbf{Sign}(m, sk_1, \overbrace{pk_1, pk_2, pk_3, pk_4, pk_5})$$

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

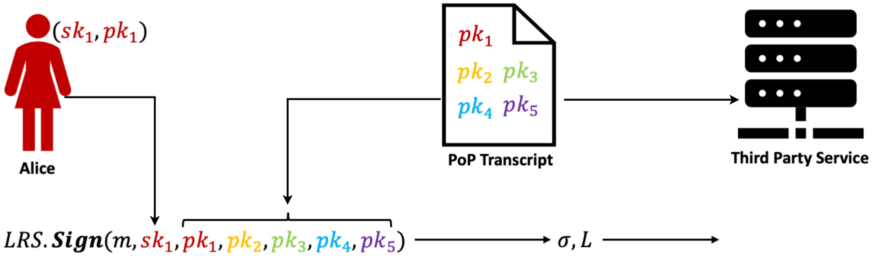
Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)



1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

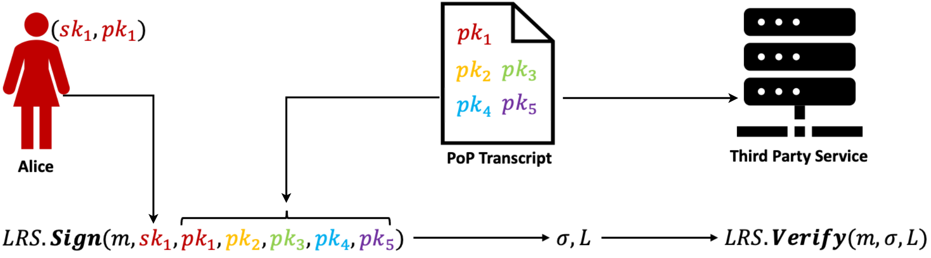
Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)



1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

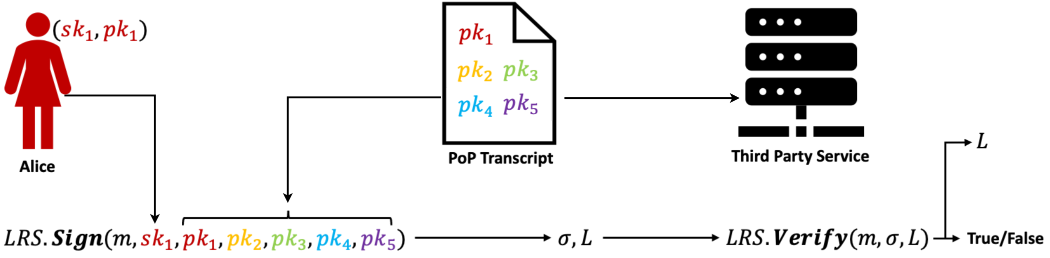
Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)



1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

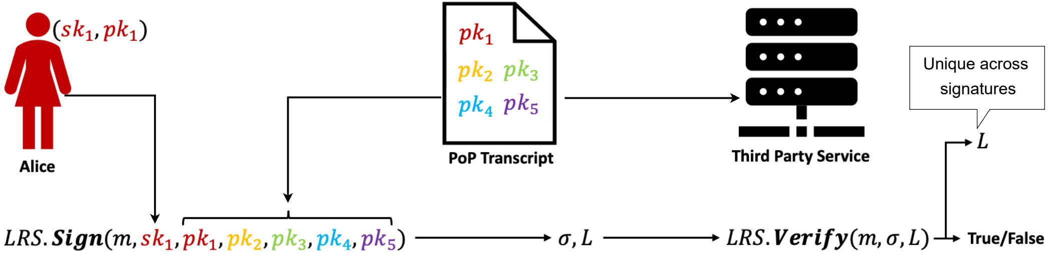
Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)



1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)

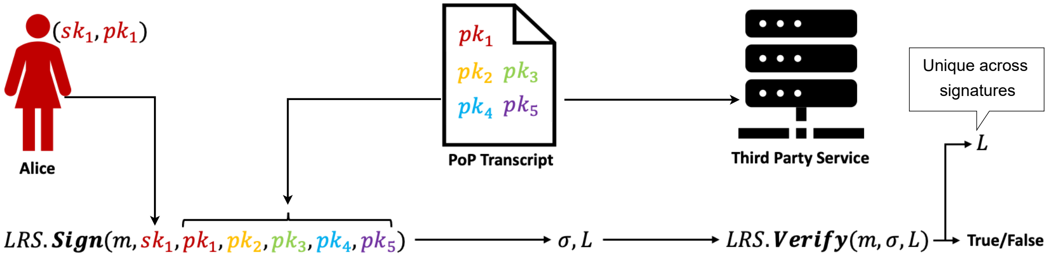


→ *Third-Party Service* can now:

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

Existing Solutions: *Linkable Ring Signatures* (Liu et al., 2005)



→ *Third-Party Service* can now:

1. Verify that *Alice* is a person
2. Use the **uniqueness** property of the tag L to enforce **Sybil-Resistance** and **Accountability**.

! But it can also exploit the **uniqueness** of L to:

1. Track *Alice*'s activity.
2. Collapse pseudonymity of *Alice*'s identities.

→ **Weak Privacy**

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

\+ Privacy

– Sybil-Resistance

– Accountability

Proof-of-Personhood + LRS

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Bind digital identities with real persons:

- PoP Token: (sk_u, pk_u)
- Anonymous signatures verify personhood
- Unique, yet anonymous linkage tags, computed using sk_u

+ Sybil-Resistance
+ Accountability
– Privacy

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

\+ Privacy
− Sybil-Resistance
− Accountability

Proof-of-Personhood + LRS

Bind digital identities with real persons:

- PoP Token: (sk_u, pk_u)
- Anonymous signatures verify personhood
- Unique, yet anonymous linkage tags, computed using sk_u

\+ Sybil-Resistance
\+ Accountability
− Privacy

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Bind digital identities with real persons:

- PoP Token: (sk_u, pk_u)
- Anonymous signatures verify personhood
- Unique, yet anonymous linkage tags, computed using sk_u

+ Sybil-Resistance
+ Accountability
– Privacy

Recap of Existing Solutions

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

Bind digital identities with real persons:

- PoP Token: (sk_u, pk_u)
- Anonymous signatures verify personhood
- Unique, yet anonymous linkage tags, computed using sk_u

+ Sybil-Resistance
+ Accountability
– Privacy

Recap of Existing Solutions

EPFL

Anonymous Credential Schemes (Coconut)

Rely on ZKP and Pairing-Based Signatures to provide:

- Private Attributes
- Selective-Disclosure
- Re-randomisation (Un-linkability)

+ Privacy
– Sybil-Resistance
– Accountability

Proof-of-Personhood + LRS

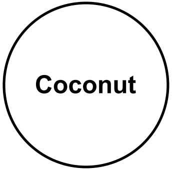
Bind digital identities with real persons:

- PoP Token: (sk_u, pk_u)
- Anonymous signatures verify personhood
- Unique, yet anonymous linkage tags, computed using sk_u

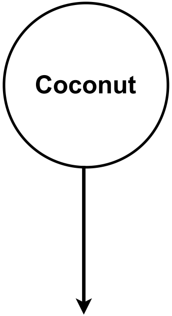
+ Sybil-Resistance
+ Accountability
– Privacy

3PBCS

3PBCS in a Nutshell

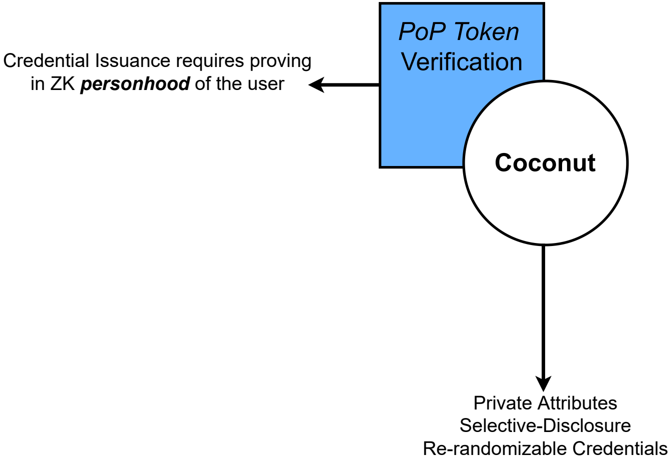


3PBCS in a Nutshell

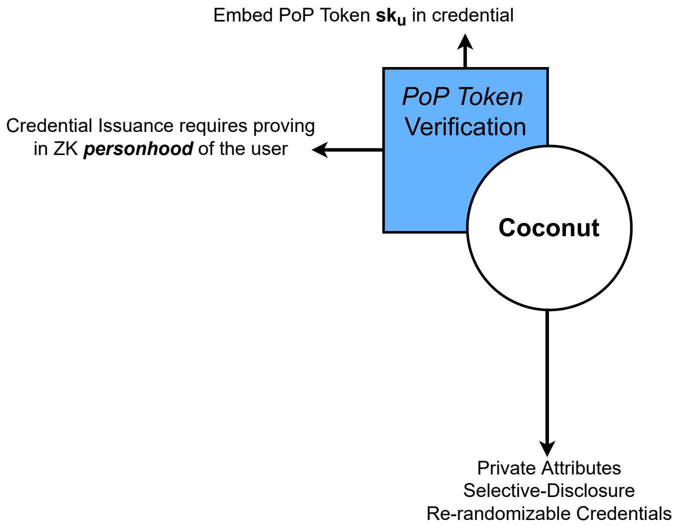


Private Attributes
Selective-Disclosure
Re-randomizable Credentials

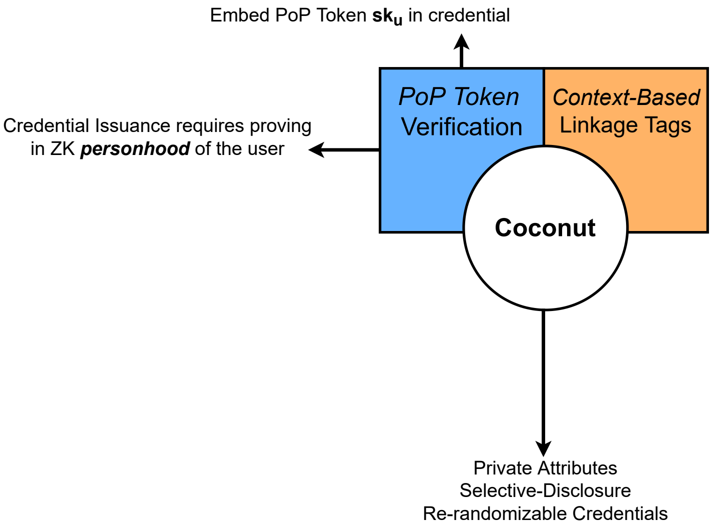
3PBCS in a Nutshell



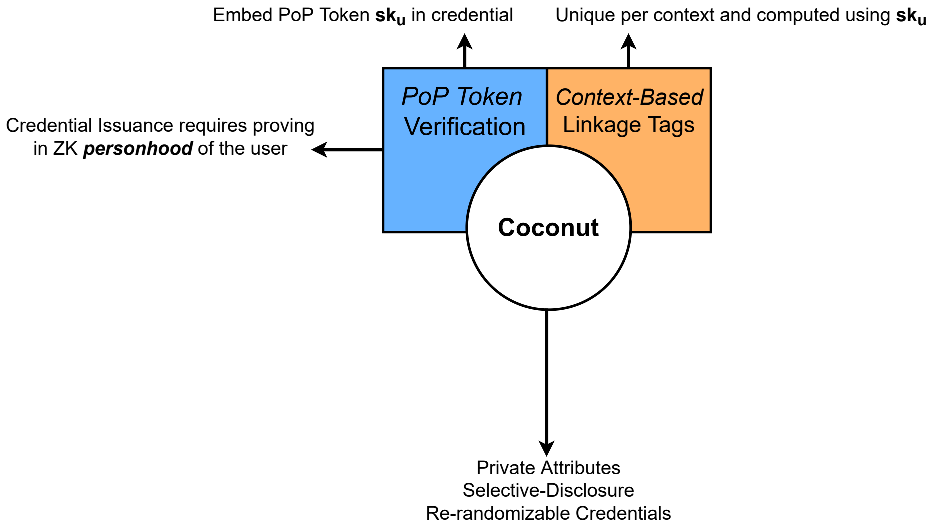
3PBCS in a Nutshell



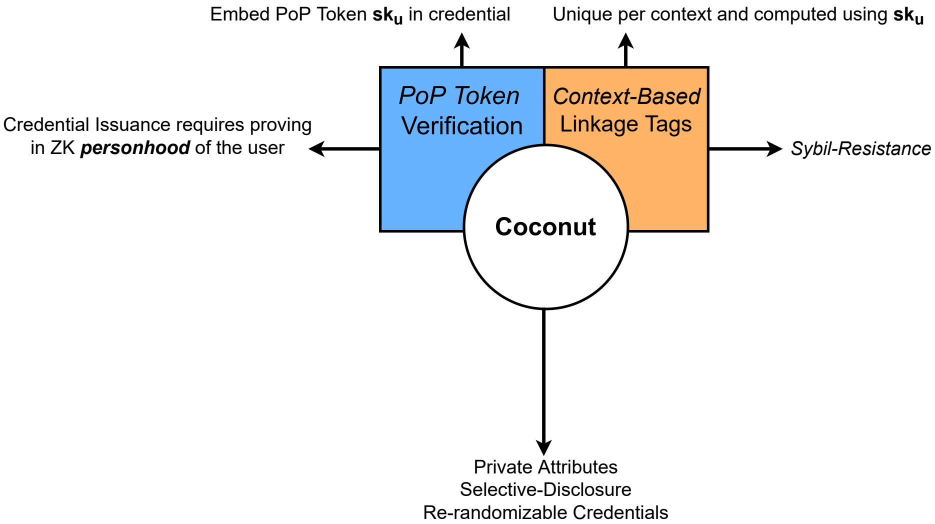
3PBCS in a Nutshell



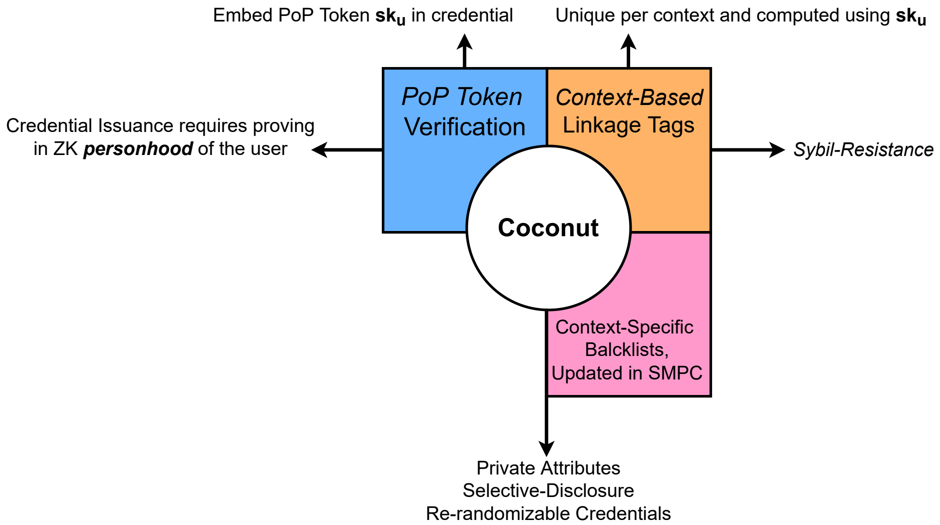
3PBCS in a Nutshell



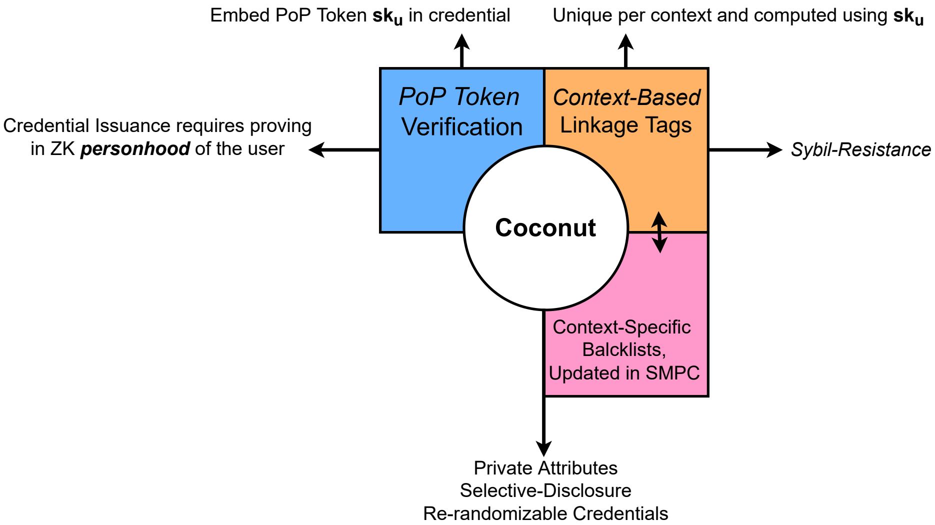
3PBCS in a Nutshell



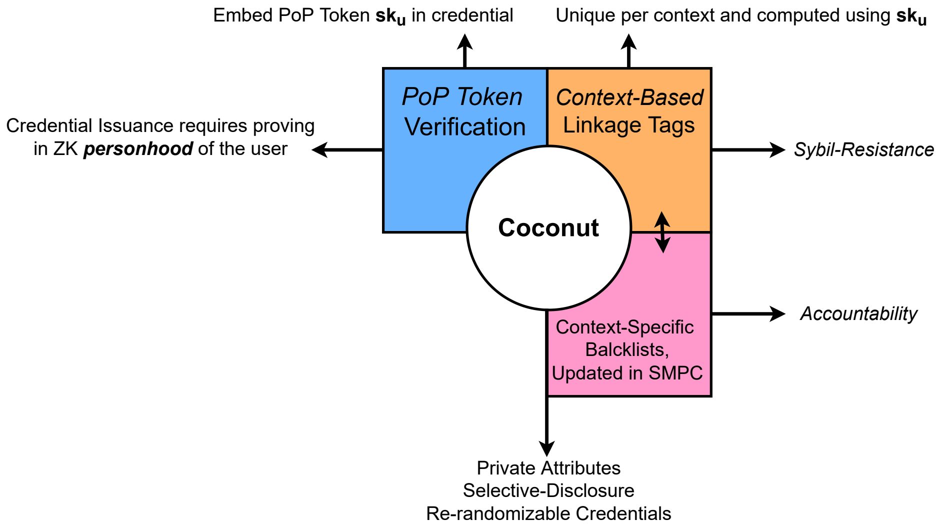
3PBCS in a Nutshell



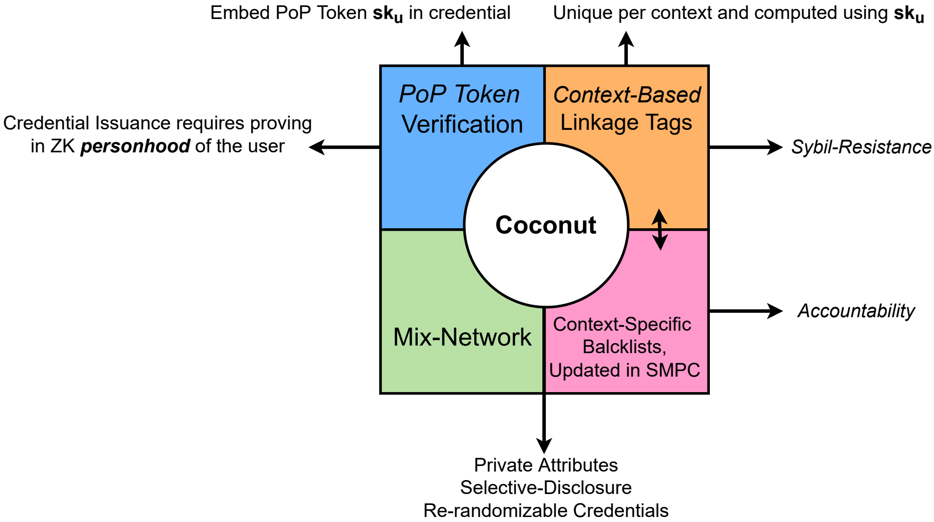
3PBCS in a Nutshell



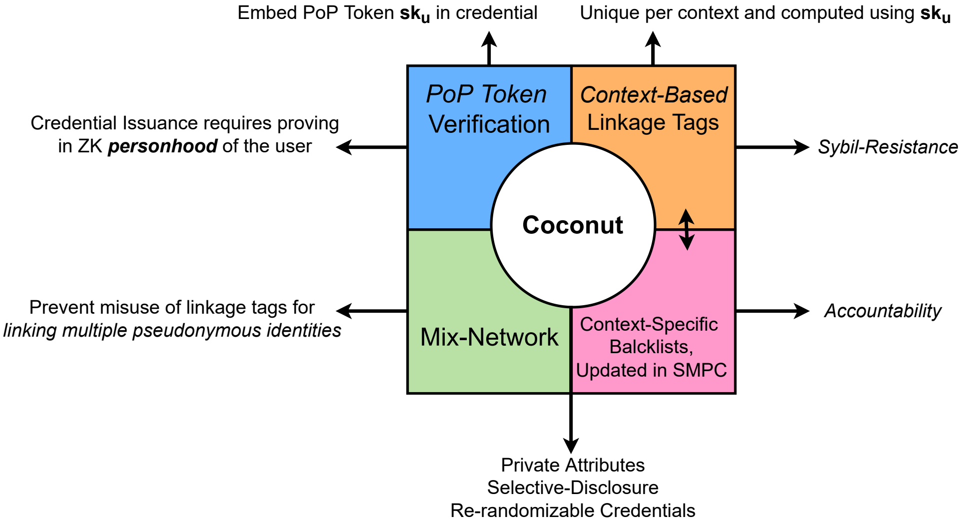
3PBCS in a Nutshell



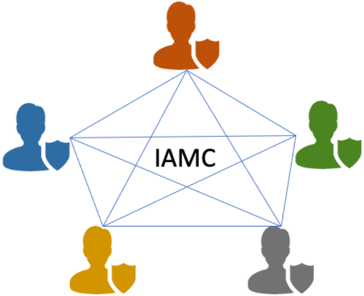
3PBCS in a Nutshell



3PBCS in a Nutshell



3PBCS: System Actors



User | *Identity and Accountability Management Cothority* | *Service Provider*

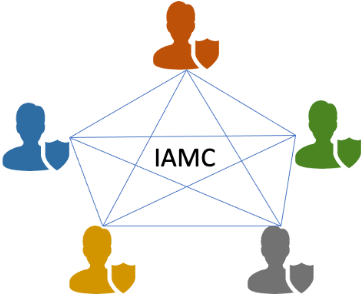
- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors

User



Identity and Accountability Management Cothority

Service Provider

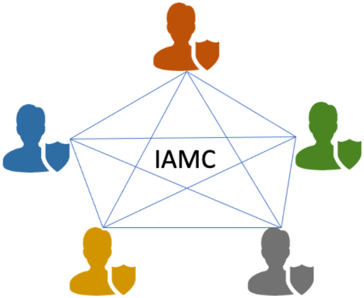
- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors

User



Identity and Accountability Management Cothority

Service Provider

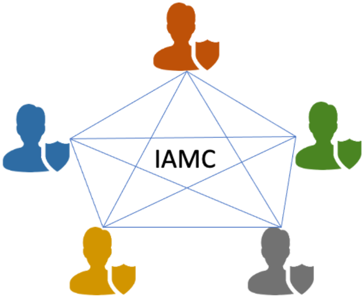
- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors

User



Identity and Accountability Management Cothority

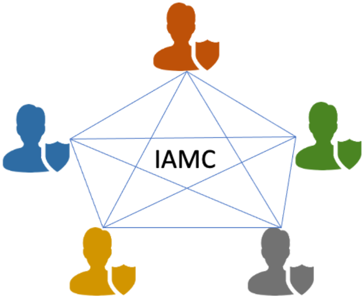
Service Provider

- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors



User *Identity and Accountability Management Cothority* *Service Provider*

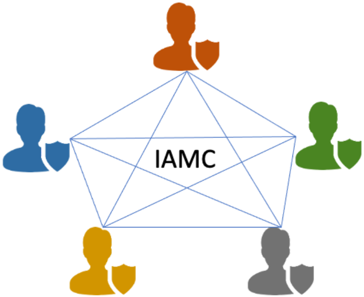
- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors

User



Identity and Accountability Management Cothority

Service Provider

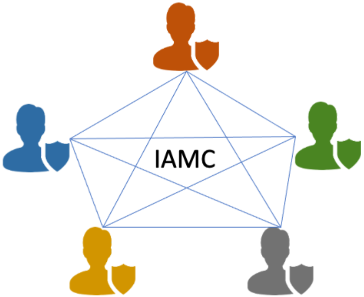
- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors

User



Identity and Accountability Management Cothority

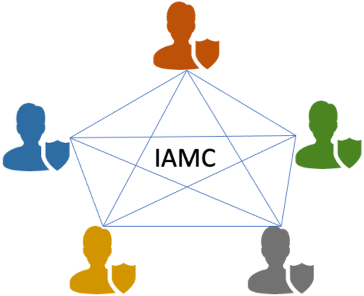
Service Provider

- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors



User

- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

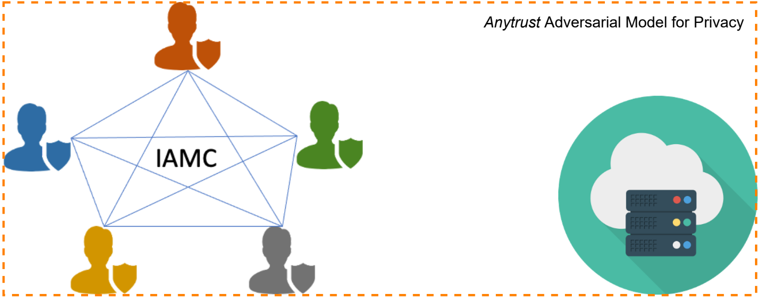
Identity and Accountability Management Cothority

- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

Service Provider

- Credential Verifier
- Linkage Tag Verifier

3PBCS: System Actors



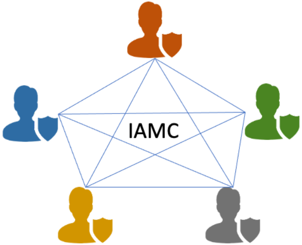
User | *Identity and Accountability Management Cothority* | *Service Provider*

- Credential Owner
- Holder of a valid PoP Token (sk_u, pk_u)

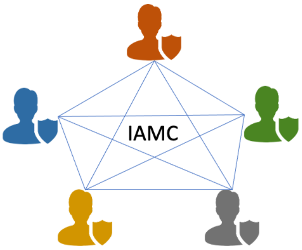
- Coconut Issuing Authority
- Mixnet Functionality
- SMPC Maintenance of Blacklists

- Credential Verifier
- Linkage Tag Verifier

3PBCS: Core System Overview



3PBCS: Core System Overview

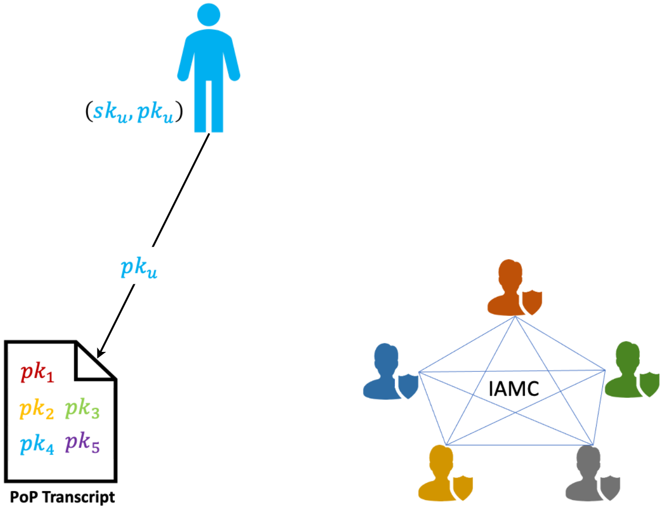


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

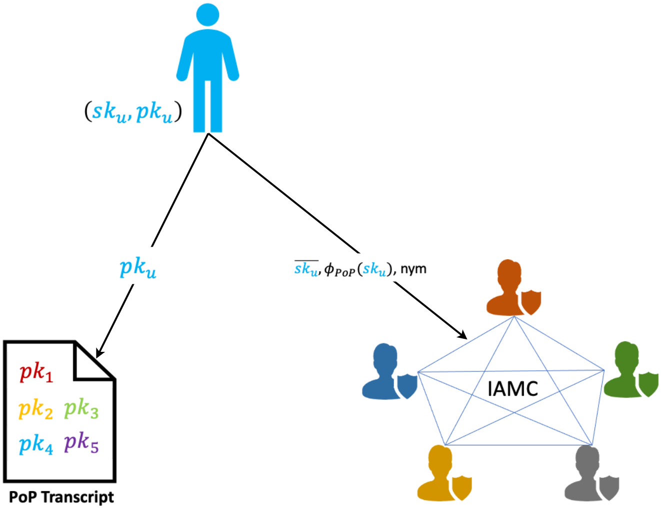


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

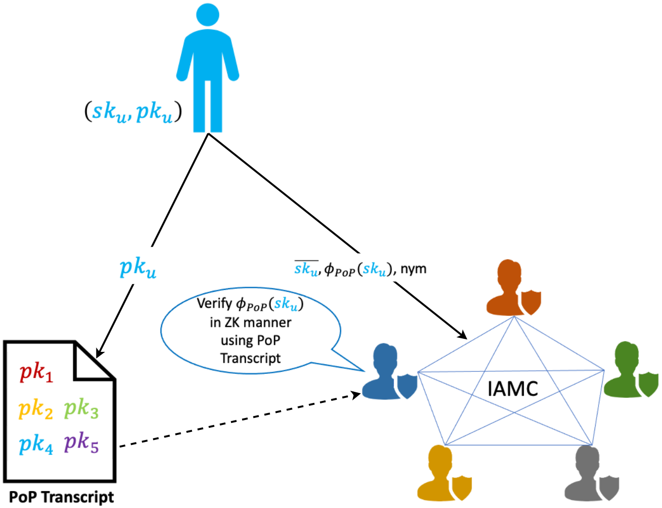


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

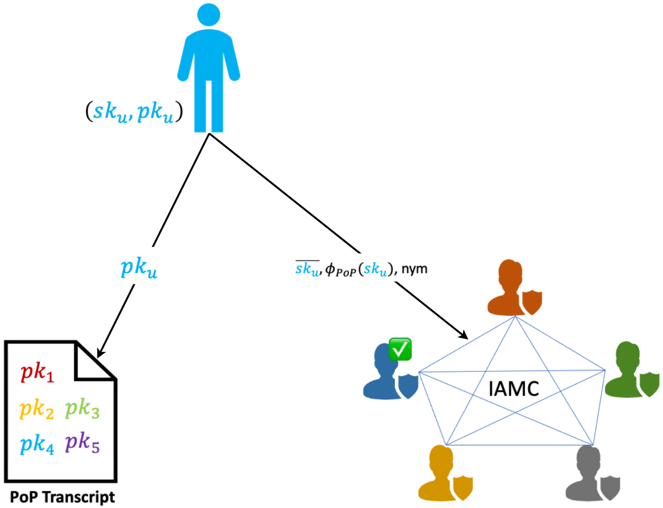


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

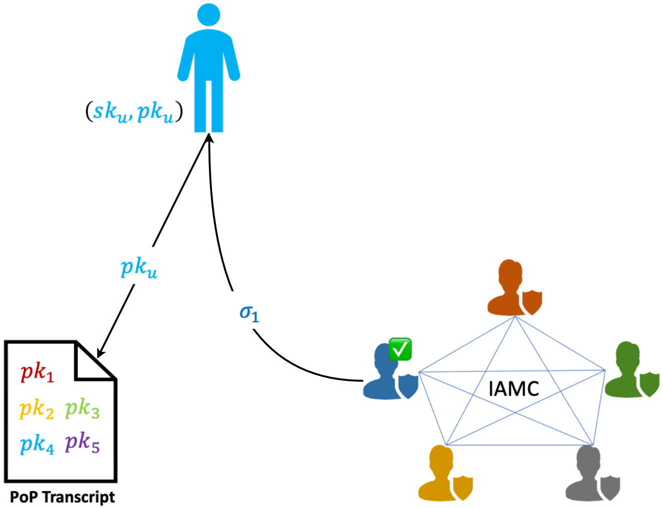


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

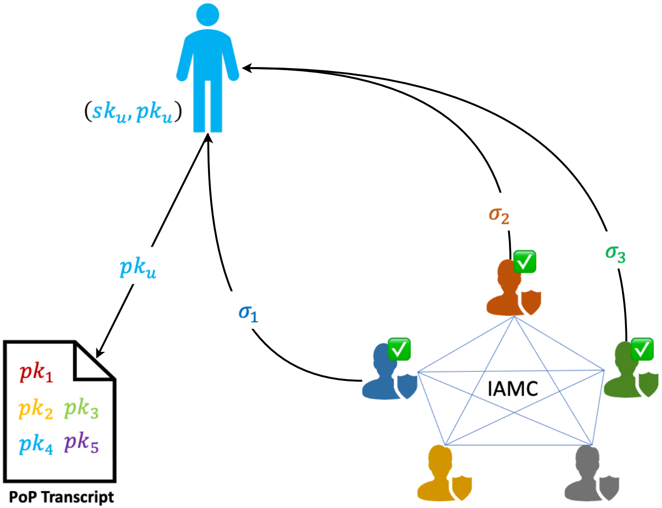


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

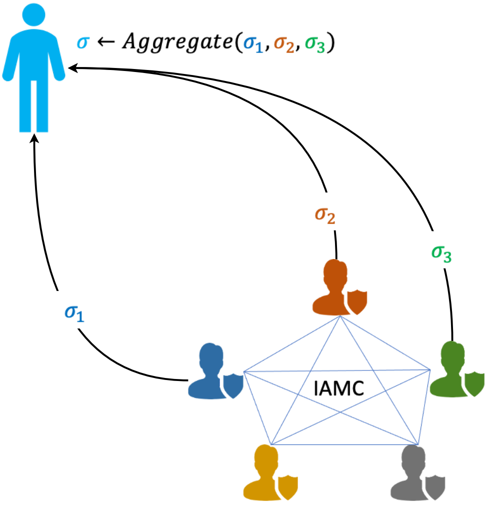


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

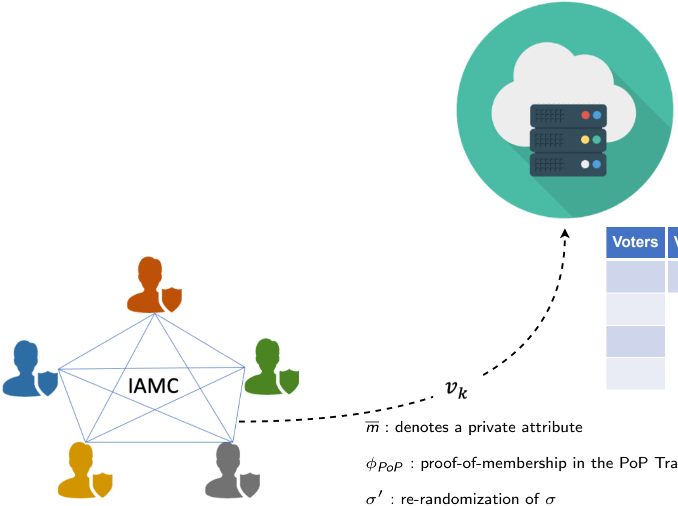


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



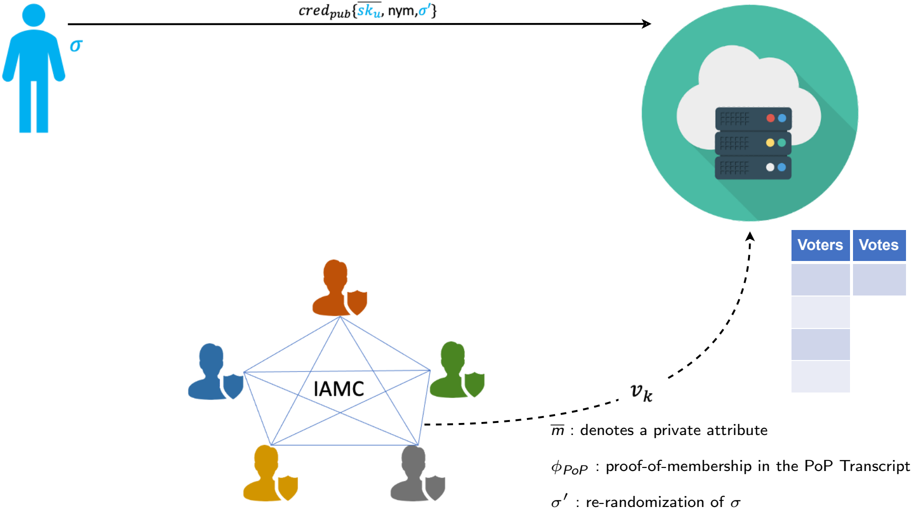
Voters	Votes

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

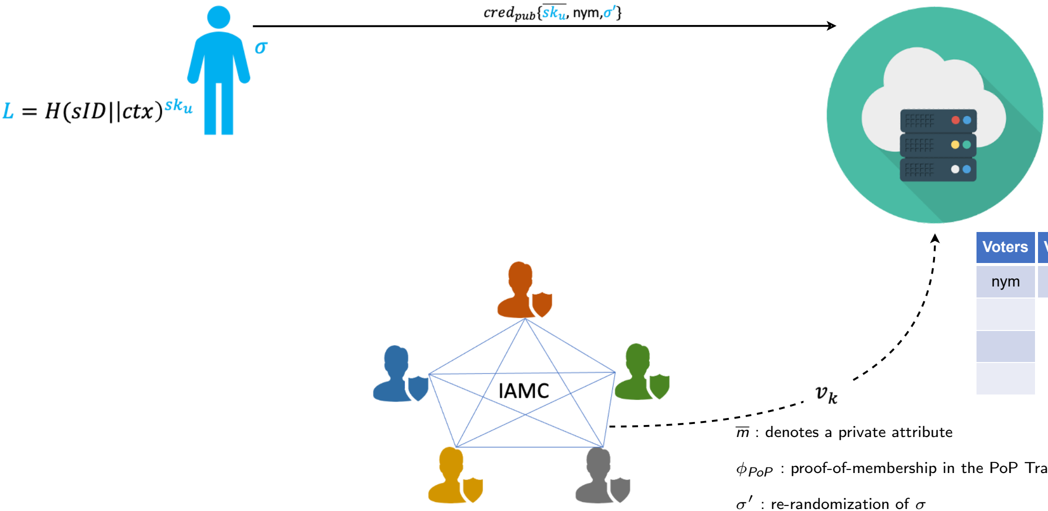


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



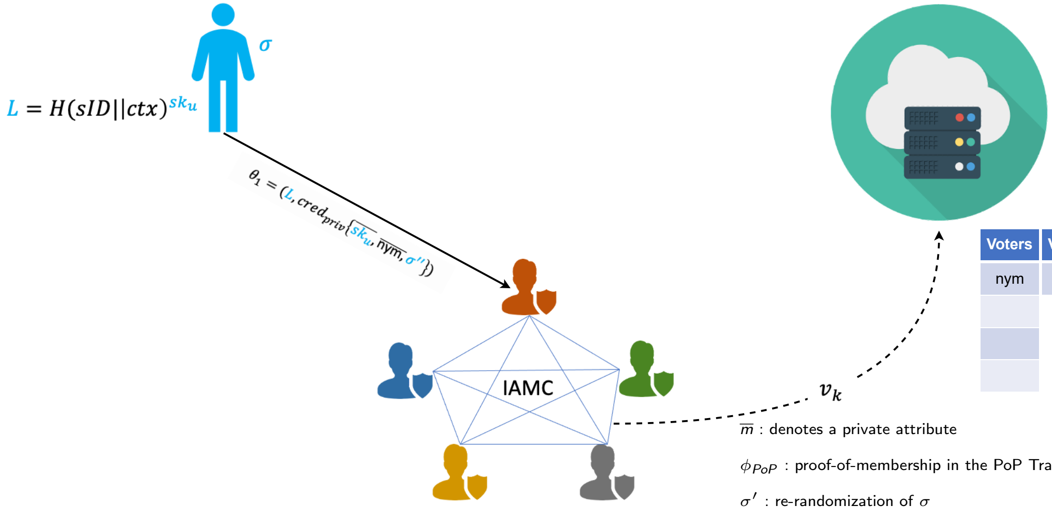
Voters	Votes
nym	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



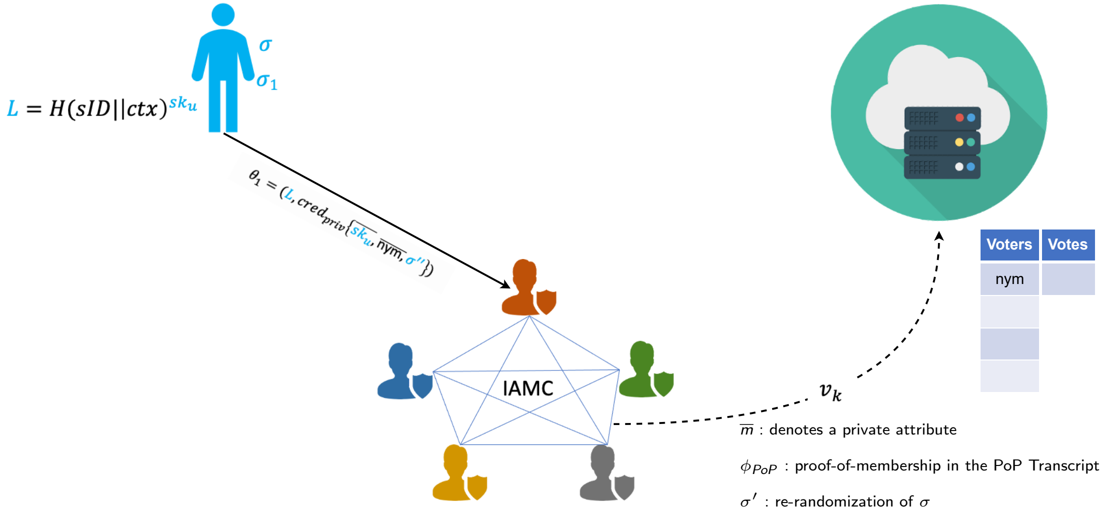
Voters	Votes
nym	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview

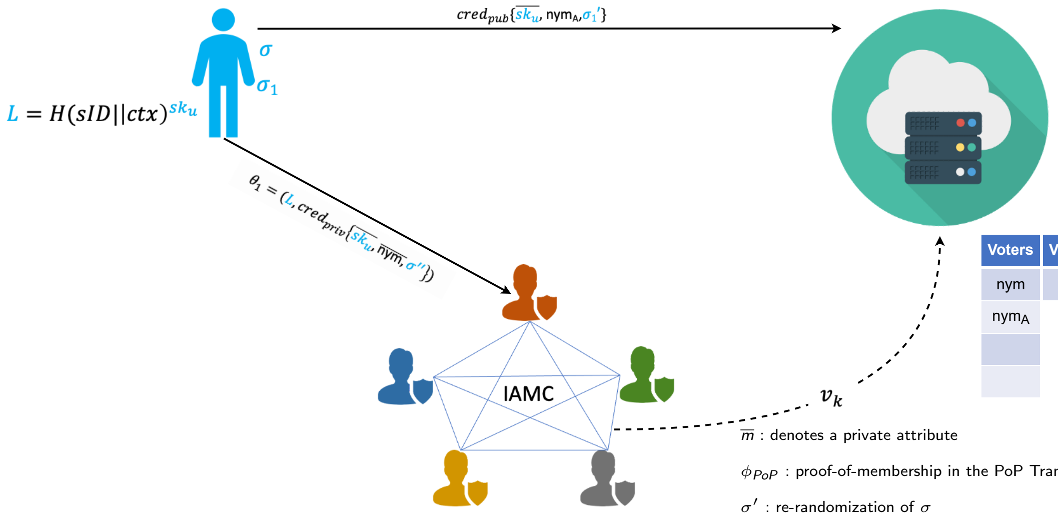


$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



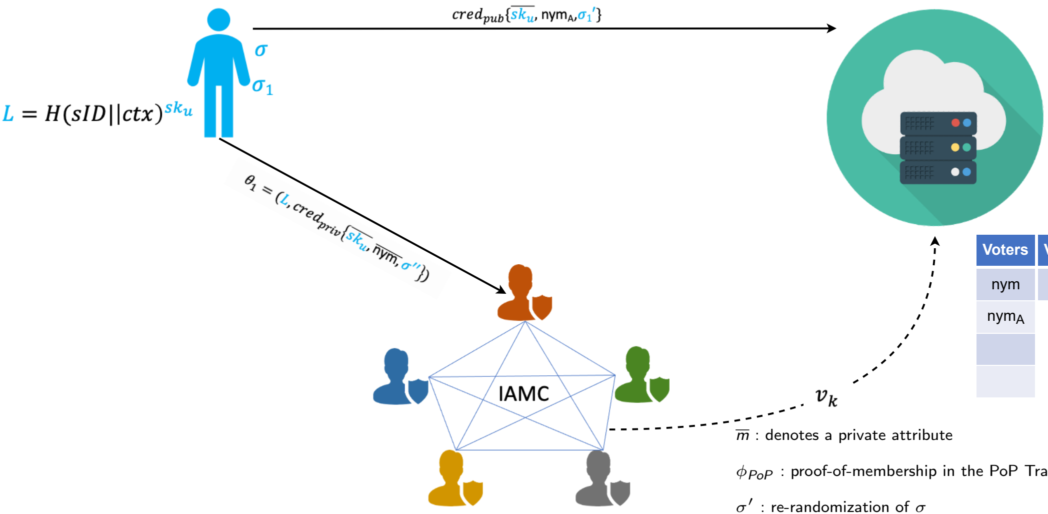
Voters	Votes
nym	
nym_A	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



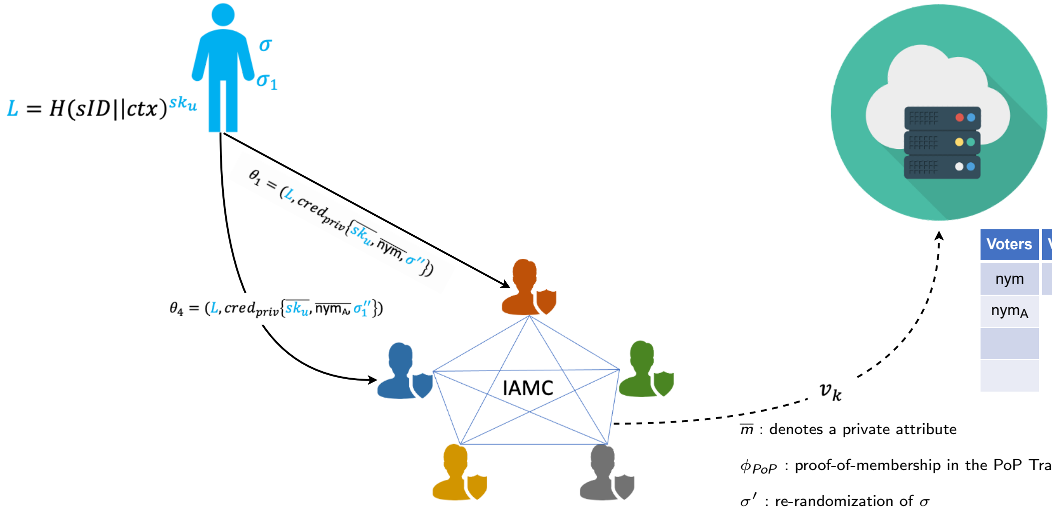
Voters	Votes
nym	
nym_A	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



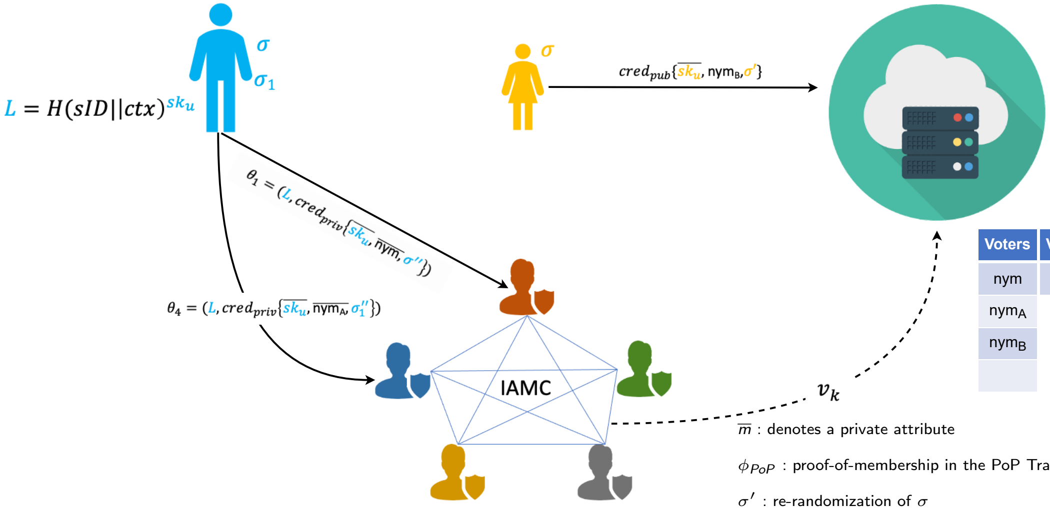
Voters	Votes
nym	
nym_A	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



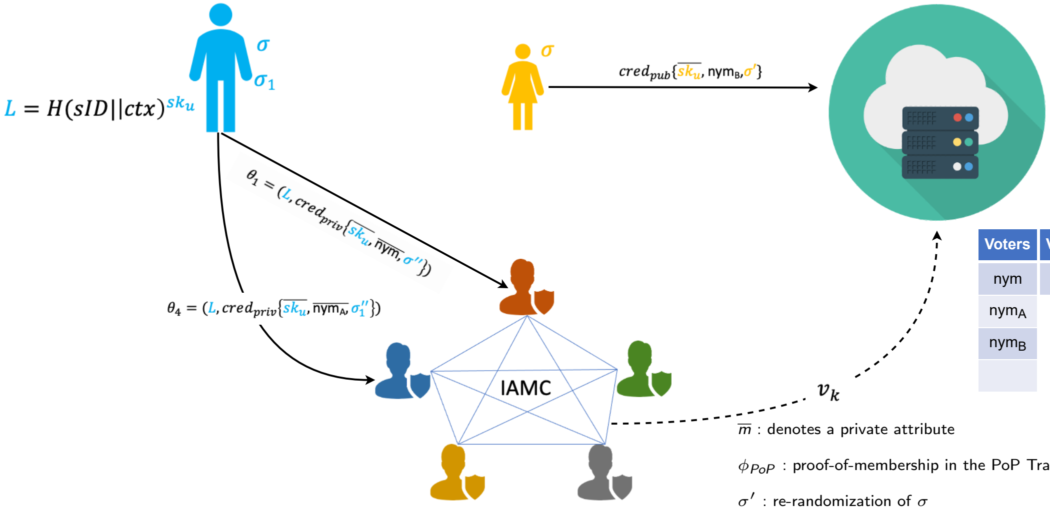
Voters	Votes
nym	
nym_A	
nym_B	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



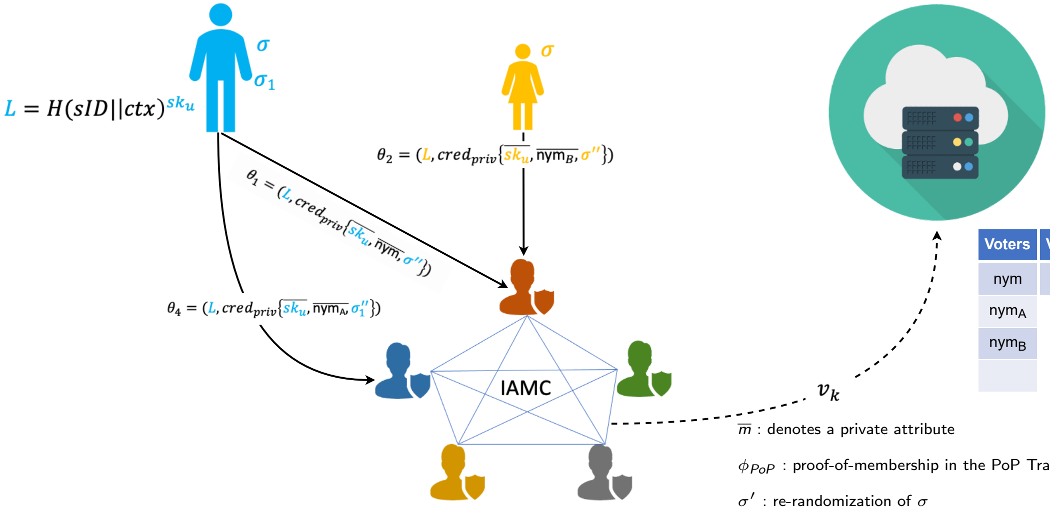
Voters	Votes
nym	
nym$_A$	
nym$_B$	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



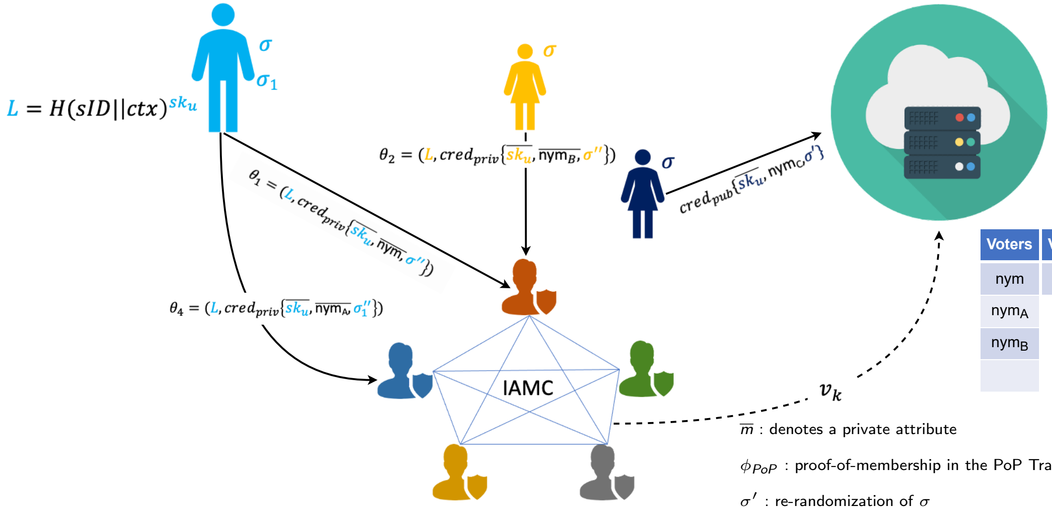
Voters	Votes
nym	
nym$_A$	
nym$_B$	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



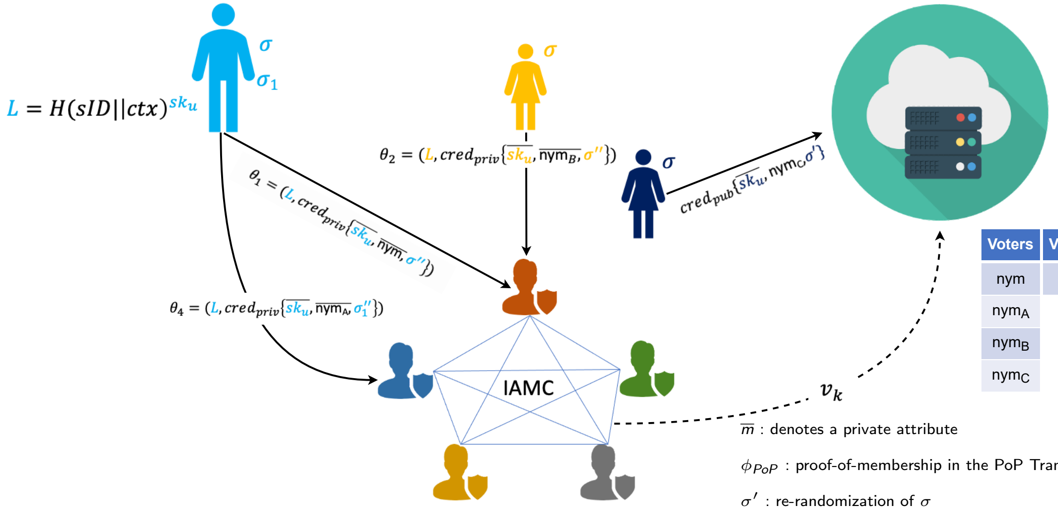
Voters	Votes
nym	
nym$_A$	
nym$_B$	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



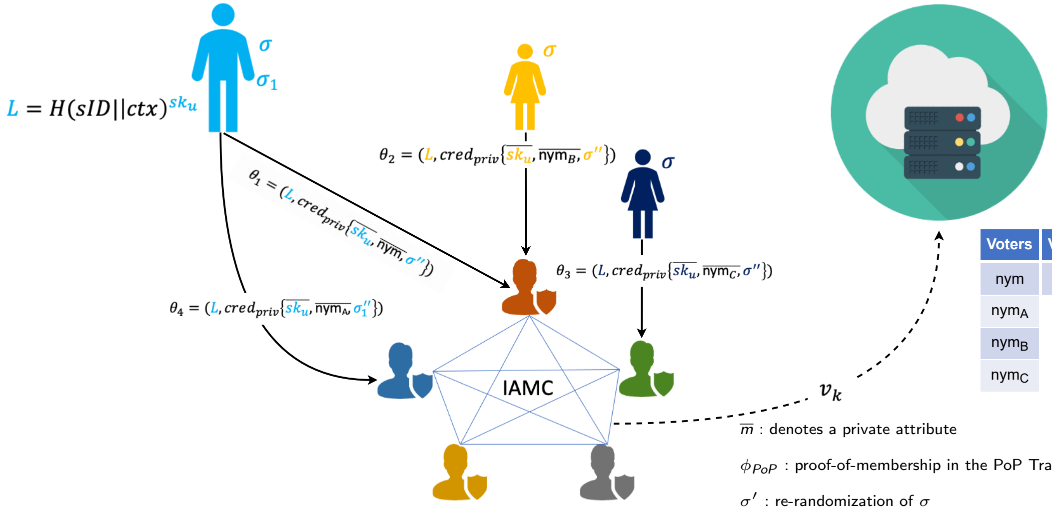
Voters	Votes
nym	
nym_A	
nym_B	
nym_C	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

σ' : re-randomization of σ

3PBCS: Core System Overview



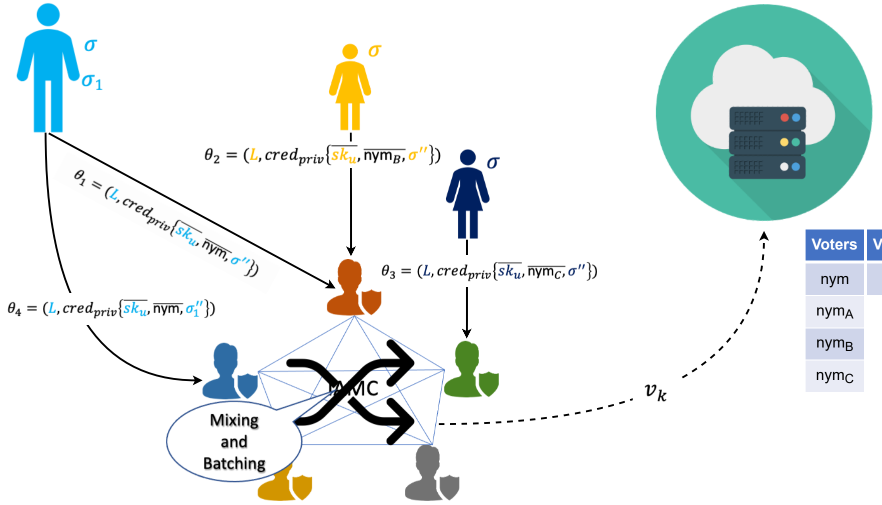
Voters	Votes
nym	
nym_A	
nym_B	
nym_C	

$\overline{m}$: denotes a private attribute

ϕ_{PoP} : proof-of-membership in the PoP Transcript

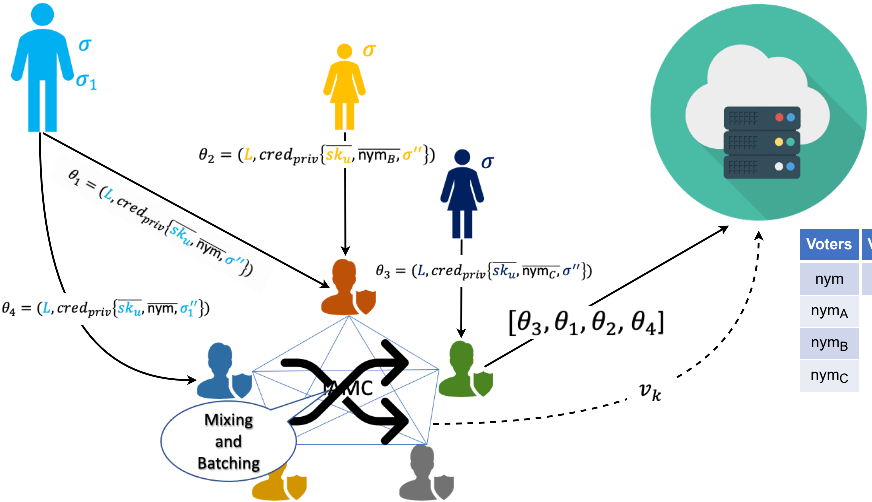
σ' : re-randomization of σ

3PBCS: Core System Overview



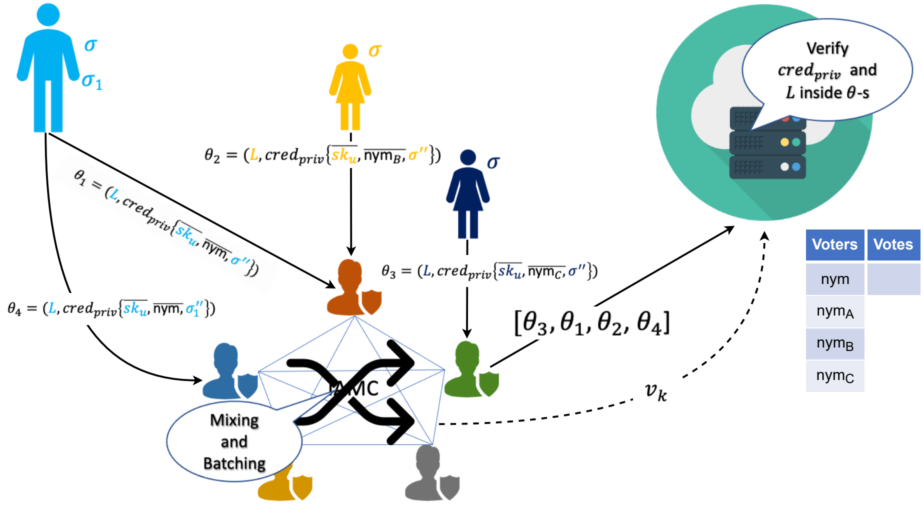
Voters	Votes
nym	
nym_A	
nym_B	
nym_C	

3PBCS: Core System Overview



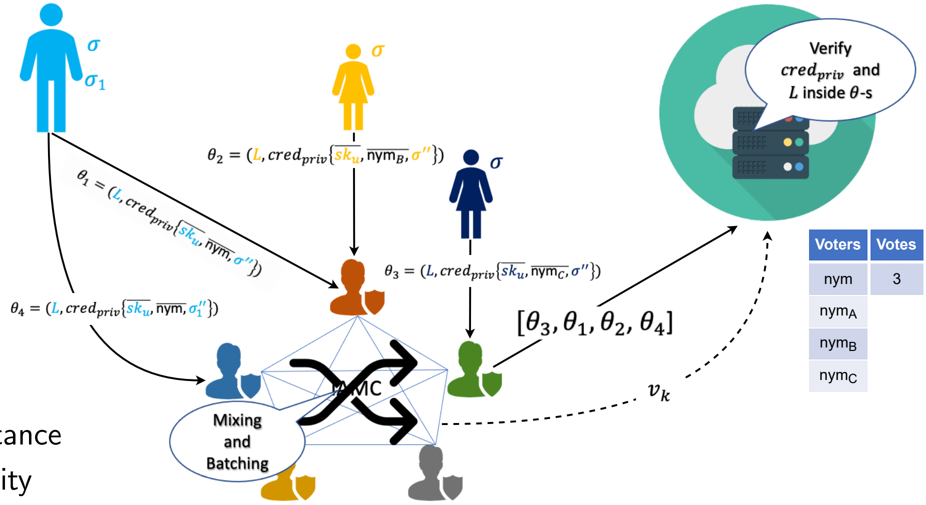
Voters	Votes
nym	
nym_A	
nym_B	
nym_C	

3PBCS: Core System Overview



Voters	Votes
nym	
nym_A	
nym_B	
nym_C	

3PBCS: Core System Overview



Voters	Votes
nym	3
nym_A	
nym_B	
nym_C	

- ✓ Privacy
- ✓ Sybil-Resistance
- ✗ Accountability

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

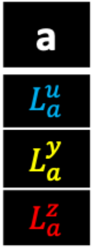
Solution: Make blacklists *context-specific* too - and *dynamically update them*.

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

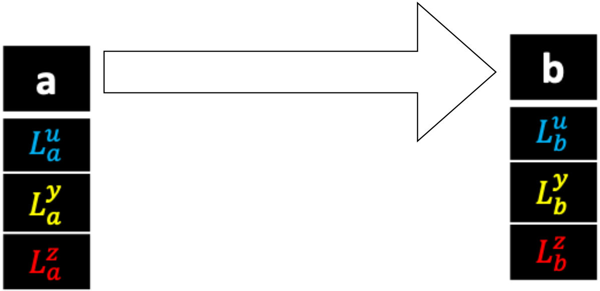


3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

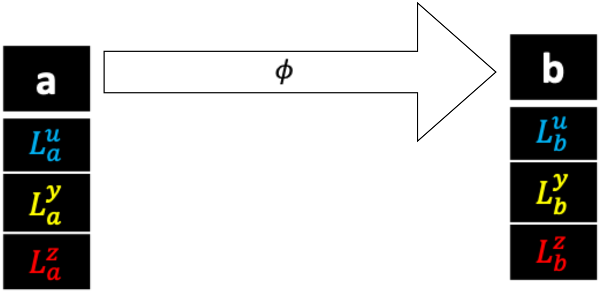


3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

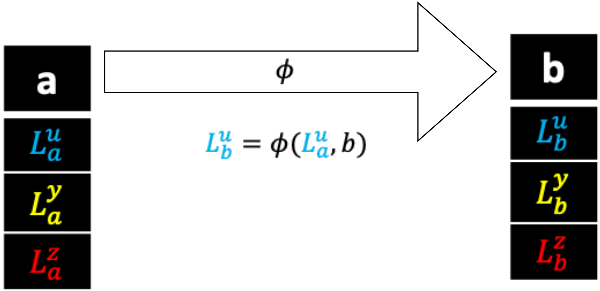


3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

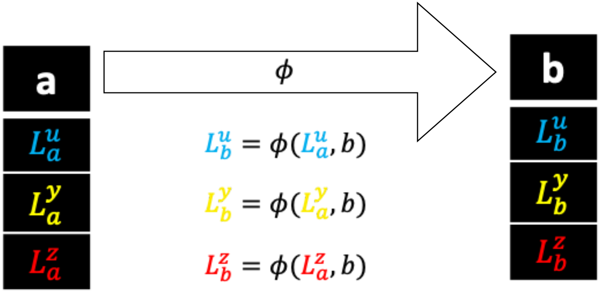


3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.

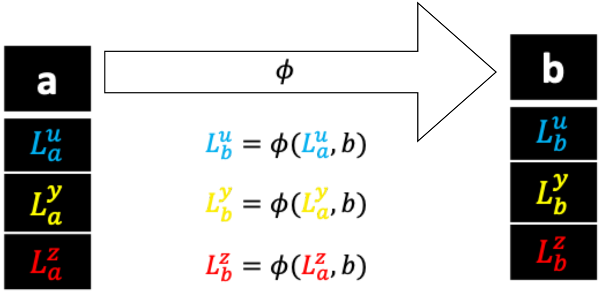


3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID||ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.



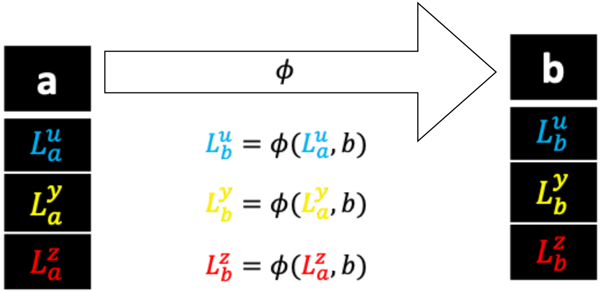
ϕ must be **only** possible to compute in **SMPC** manner by IAMC nodes!

3PBCS: Enforcing Accountability

Q: Why can't we use linkage tags $L = H(sID || ctx_a)^{sk_u}$ as blacklist entries?

A: Context-specific nature of L prevents from blacklisting across different contexts.

Solution: Make blacklists *context-specific* too - and *dynamically update them*.



ϕ must be **only** possible to compute in **SMPC** manner by IAMC nodes!

$\rightarrow$ Otherwise, our activity tracking guarantees collapse!

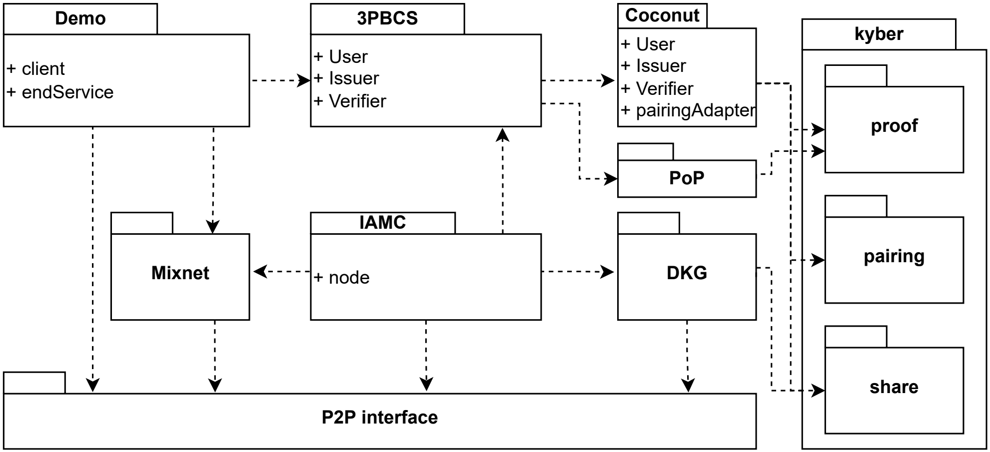
Summary

3PBCS is the first Credential System to our knowledge providing :

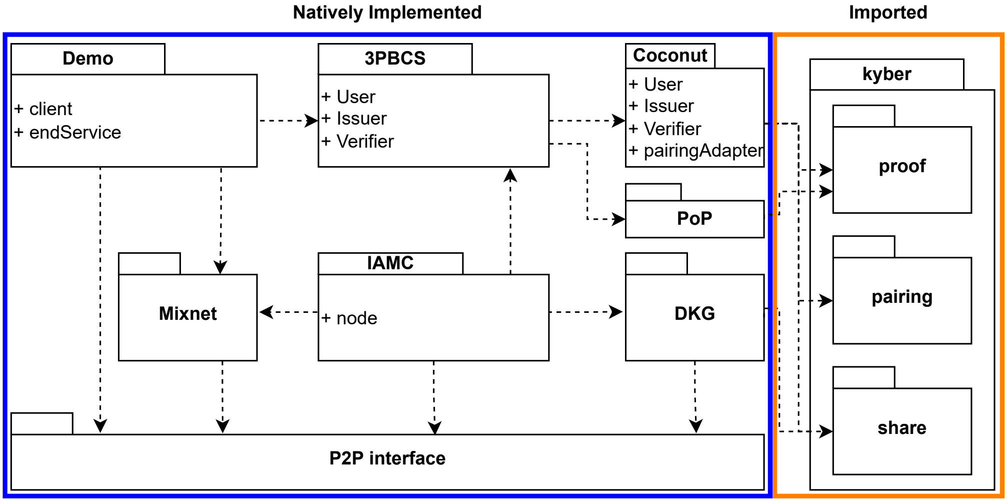
- Anonymous Credentials
- Sybil-Resistance
- Accountability
- Unlimited credential generation for a single user
- Enhanced Privacy guarantees (without risking any of the above)

$\longrightarrow$ These make 3PBCS a strong candidate for a variety of applications such as social platforms, whistleblowing apps, e-voting etc.

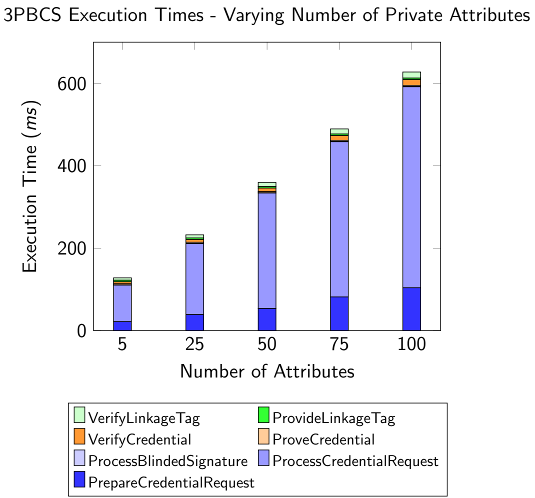
Proof-of-Concept Implementation & Challenges

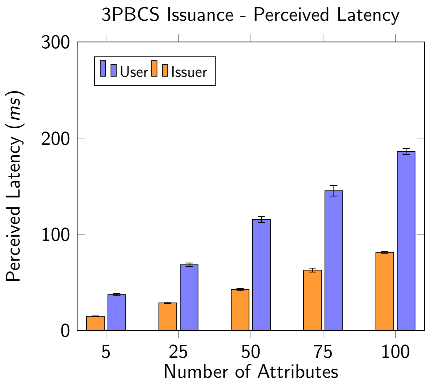


Proof-of-Concept Implementation & Challenges



Performance Evaluation





IAMC Nodes: 6; PoPTranscriptSize: 30;

Measurements generated over a sample of 5000 executions.

Demonstration

DEMO TIME!

Limitations and Future Directions

Limitations of 3PBCS:

- $\mathcal{O}(n)$ Computational and Space Complexity to the size of the PoP Transcript.
- Blacklisting is restricted to sequential actions only.
- Restricted Credential management at current state of advancement (e.g. Crendetial Recovery missing).
- PoC Implementation at present does not include our blacklist design.

Future Directions:

- Thorough Security Analysis of the scheme.
- Research towards alternative blacklisting methods.

Thank You for Your Attention!

Questions and Discussion...

References I

- Borge, Maria et al. (2017). "Proof-of-Personhood: Redemocratizing Permissionless Cryptocurrencies". In: *2017 IEEE European Symposium on Security and Privacy Workshops (EuroS PW)*, pp. 23–26. DOI: 10.1109/EuroSPW.2017.46.
- Camenisch, Jan and Markus Stadler (1997). "Efficient group signature schemes for large groups". In: *Annual International Cryptology Conference*. Springer, pp. 410–424.
- Ford, Bryan (2020). "Identity and Personhood in Digital Democracy: Evaluating Inclusion, Equality, Security, and Privacy in Pseudonym Parties and Other Proofs of Personhood". In: arXiv: 2011.02412 [cs.CY].
- Liu, Joseph K and Duncan S Wong (2005). "Linkable ring signatures: Security models and new schemes". In: *International Conference on Computational Science and Its Applications*. Springer, pp. 614–623.

References II

- Mare, Shrirang, Mary Baker, and Jeremy Gummeson (2016). "A study of authentication in daily life". In: *Twelfth symposium on usable privacy and security (SOUPS 2016)*, pp. 189–206.
- Sonnino, Alberto et al. (2018). "Coconut: Threshold Issuance Selective Disclosure Credentials with Applications to Distributed Ledgers". In: *CoRR* abs/1802.07344. arXiv: 1802.07344. URL: http://arxiv.org/abs/1802.07344.
- W3C (2021). *Verifiable credentials data model V1.1: Expressing verifiable information on the Web*. URL: https://www.w3.org/TR/vc-data-model/.
- Williams, Shannon (2020). *Average person has 100 passwords - study*. URL: https://securitybrief.co.nz/story/average-person-has-100-passwords-study.

EPFL

Definition (Linkable Ring Signature)

Let $\mathcal{U}$ be the set of r users, each associated with a public key pk_u of a standard signature scheme, where $(pk_u, sk_u) \in \mathcal{R}$, such that $\mathcal{R} \subseteq \mathcal{X} \times \mathcal{Y}$ denotes a secret-public key relation. We call $\mathcal{U}$ the *ring*. Let $\mathcal{L} = \{pk_1, \dots, pk_r\}$. Then, let the s-th member be the signer and denote their public key as $pk_s \in \mathcal{L}$ and the corresponding secret key sk_s. The generic Linkable Ring Signature Scheme is then described by the following:

- **LinkableRing.Sign**$(m, \mathcal{L}, sk_s) \to \sigma, L$:
  Output
  $$L = H(\mathcal{L})^{sk_s}$$
  and
  $$\sigma = SPK\left\{ sk_s : \vee_{i=1}^{r} ((sk_s, pk_i) \in \mathcal{R}) \wedge L = H(\mathcal{L})^{sk_s} \right\}(m)$$
  where *SPK* denotes a *Signature based on Proof-of-Knowledge* (Camenisch et al., 1997).
- **LinkableRing.Verify**$(m, \sigma, \mathcal{L}) \to True/False$:
  Output *True* if the corresponding Proof-of-Knowledge included in σ is verified to be correct. Else, output *False*.
- **LinkableRing.Link**$(L_1, L_2) \longrightarrow True/False$:
  Output *True* if $L_1 = L_2$, *False* otherwise.

Verifiable Credentials

Definition (Credential)

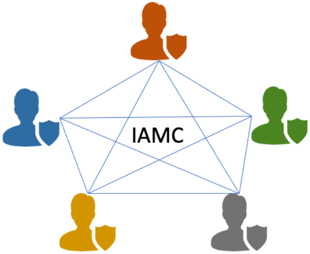
A credential is a 3-tuple

$$\texttt{cred} = \{\texttt{metadata}, \mathcal{C}, \sigma\}$$

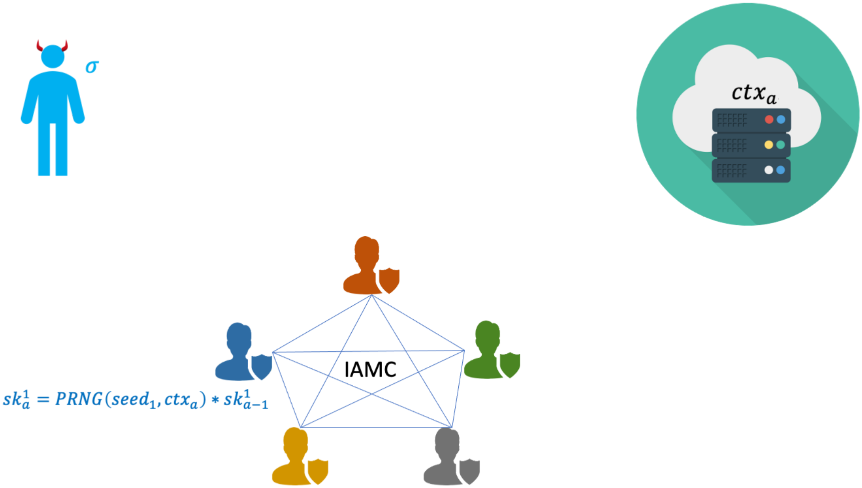
where:

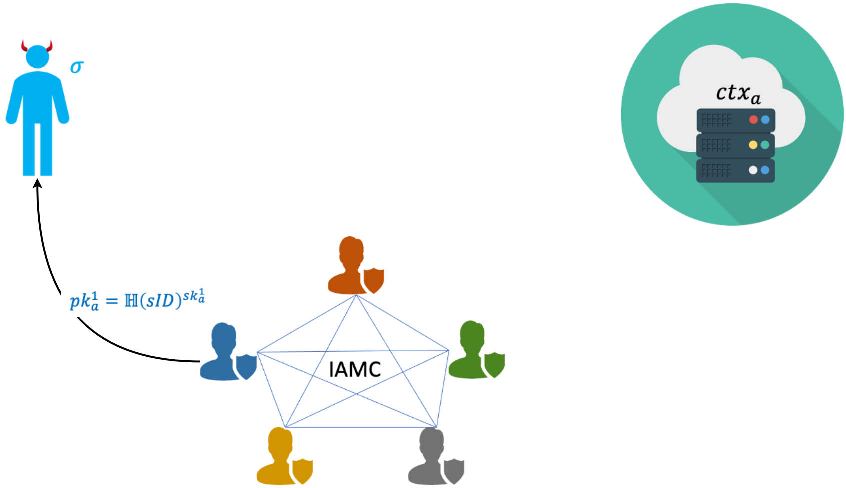
1. $\texttt{metadata}$ describes the metadata of the credential, i.e. a set of details regarding the use-case and context of usage of the credential, described by any data-type.
2. $\mathcal{C}$ denotes the set of claims embedded in the credential. Moreover, $\mathcal{C} = \mathcal{C}_{\texttt{pub}} \cup \mathcal{C}_{\texttt{priv}}$, where
   - if $\texttt{claim}_i \in \mathcal{C}_{\texttt{pub}}$, then
   $$\texttt{claim}_i = \{\texttt{attr}_i, \texttt{val}_i, \texttt{provider}_i\}$$
   - while if $\overline{\texttt{claim}}_i \in \mathcal{C}_{\texttt{priv}}$, then
   $$\overline{\texttt{claim}}_i = \{\texttt{attr}_i, \phi_i, \pi_{\phi_i}, \texttt{provider}_i\}$$
3. σ: the signature issued by the issuer over the metadata and claims embedded in the credential.

3PBCS: Dynamic, Context-Specific Blacklisting



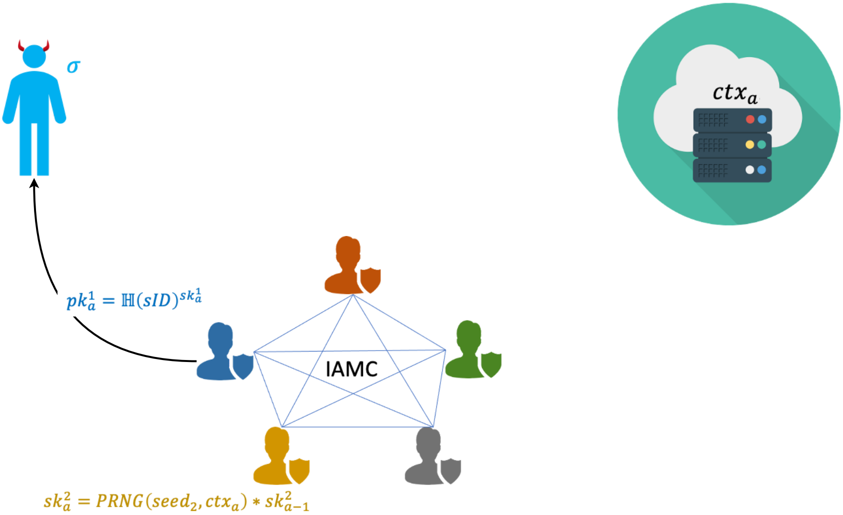
3PBCS: Dynamic, Context-Specific Blacklisting

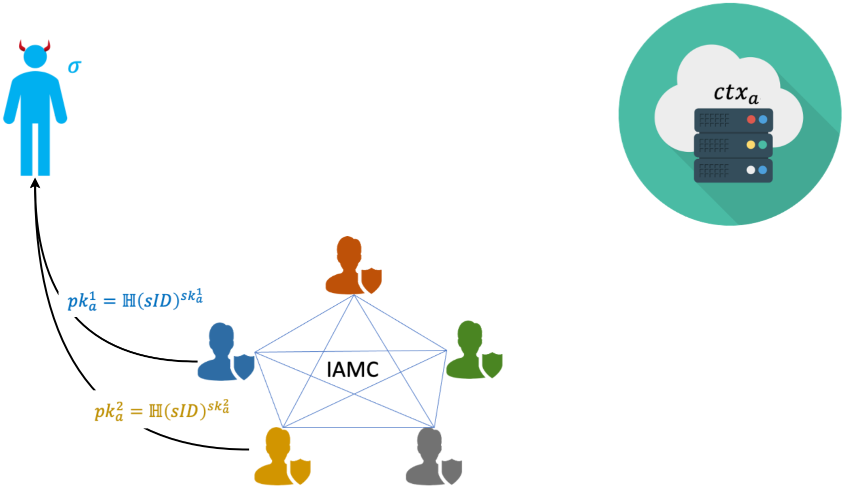




σ
pk_a^1 = ℍ(sID)^{sk_a^1}
IAMC
ctx_a

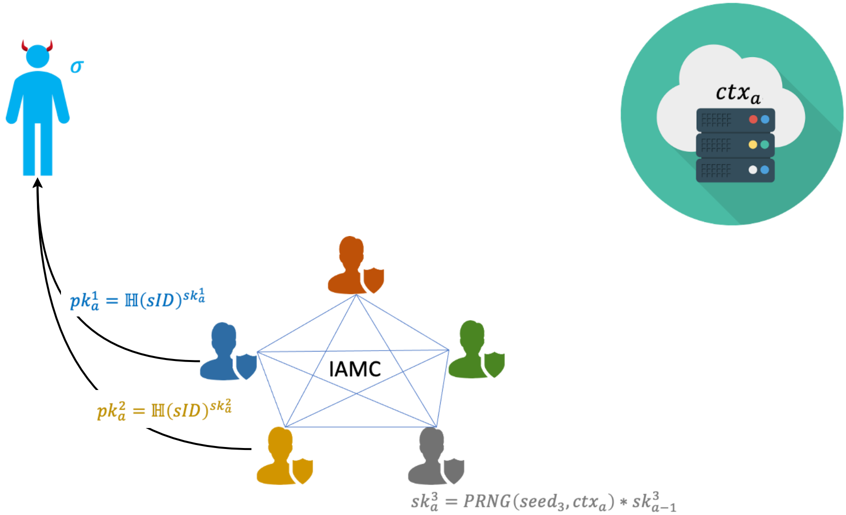
3PBCS: Dynamic, Context-Specific Blacklisting



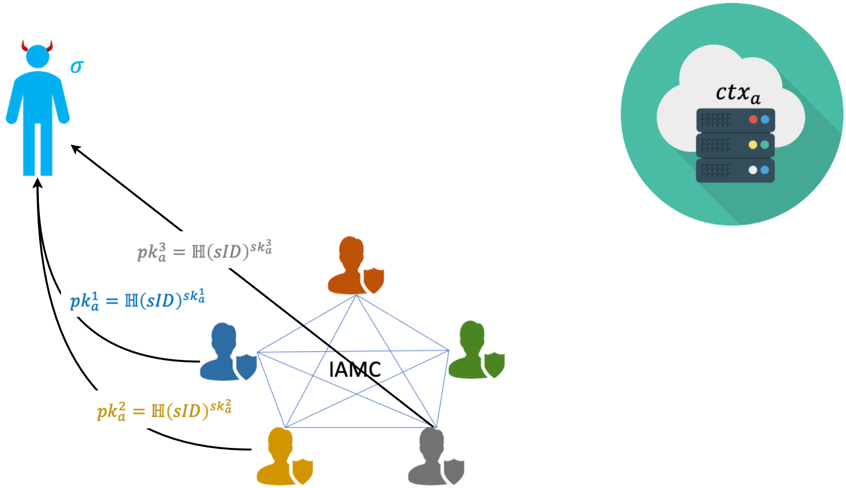


σ
$pk_a^1 = \mathbb{H}(sID)^{sk_a^1}$
$pk_a^2 = \mathbb{H}(sID)^{sk_a^2}$
IAMC
ctx_a

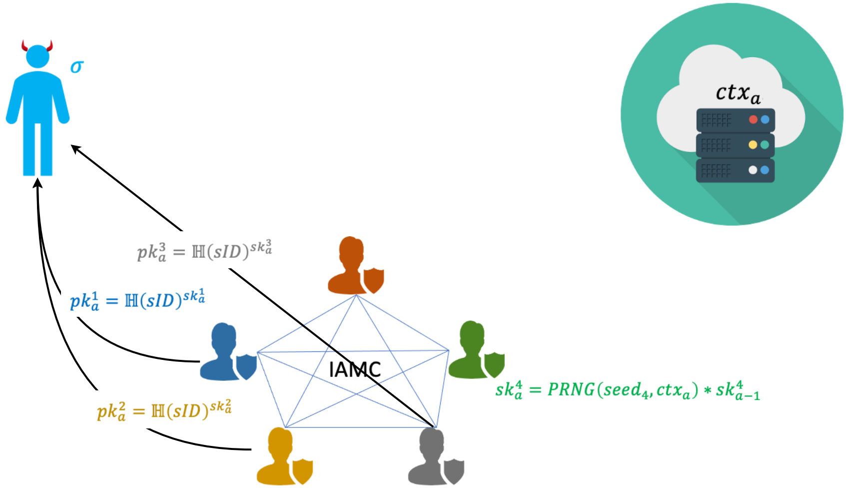
3PBCS: Dynamic, Context-Specific Blacklisting



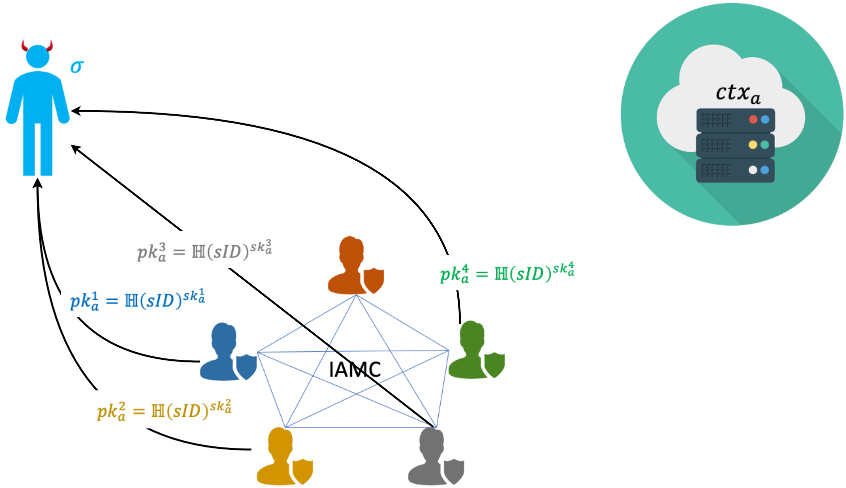
3PBCS: Dynamic, Context-Specific Blacklisting



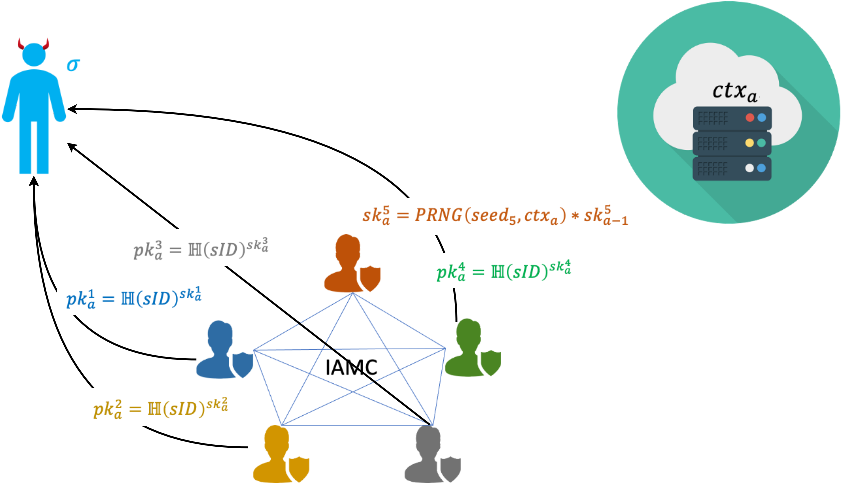
3PBCS: Dynamic, Context-Specific Blacklisting



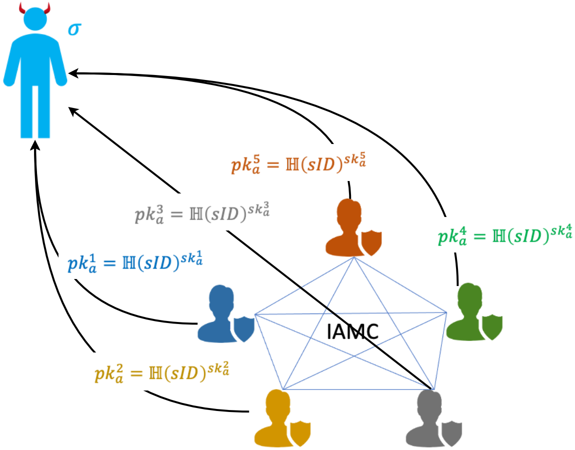
3PBCS: Dynamic, Context-Specific Blacklisting



3PBCS: Dynamic, Context-Specific Blacklisting

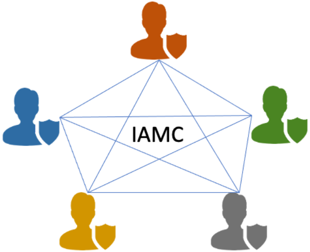


3PBCS: Dynamic, Context-Specific Blacklisting



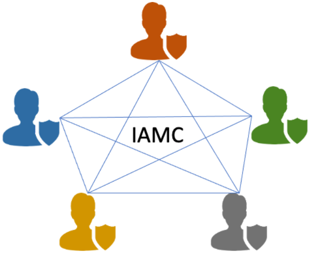
3PBCS: Dynamic, Context-Specific Blacklisting

$$pk_a = pk_a^1 * pk_a^2 * pk_a^3 * pk_a^4 * pk_a^5$$

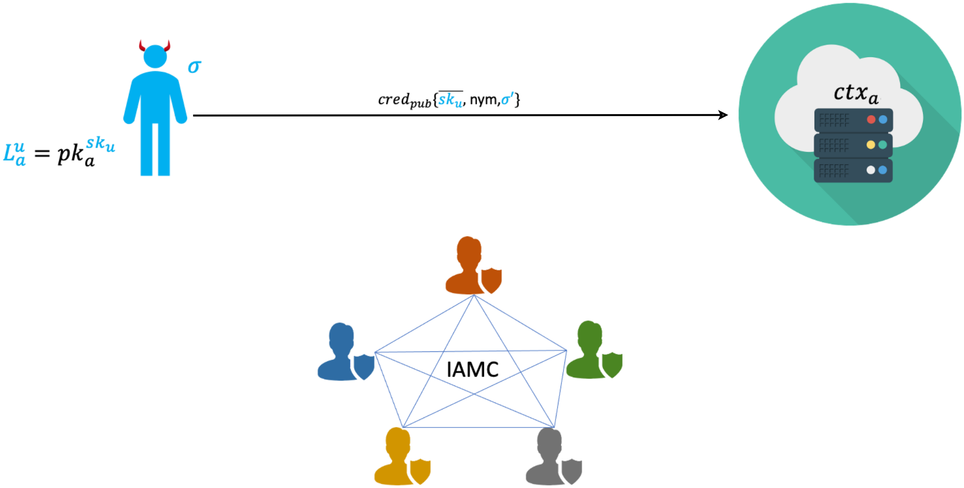


$$L_a^u = pk_a^{sk_u}$$

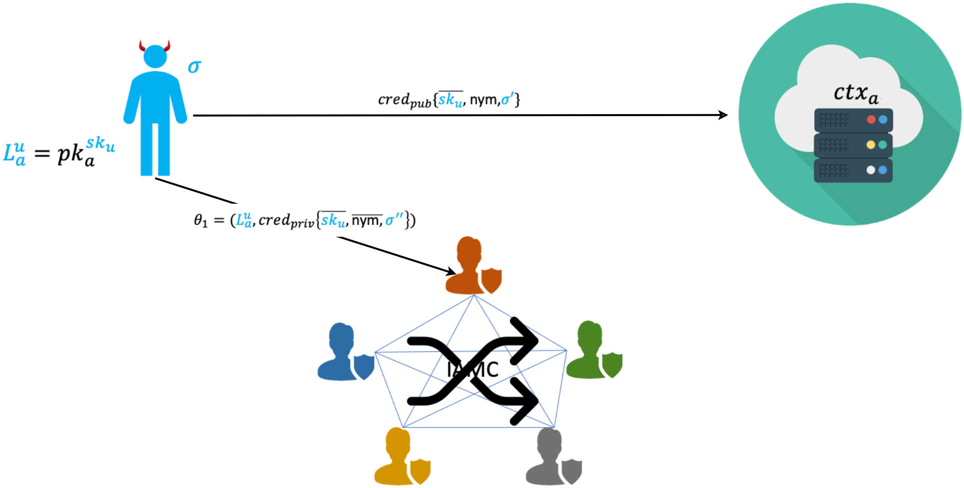
$$pk_a = pk_a^1 * pk_a^2 * pk_a^3 * pk_a^4 * pk_a^5$$



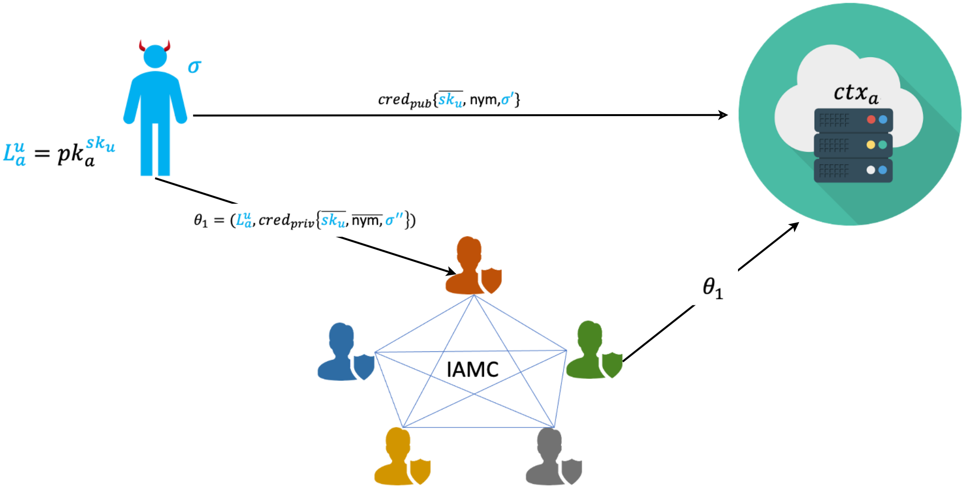
3PBCS: Dynamic, Context-Specific Blacklisting



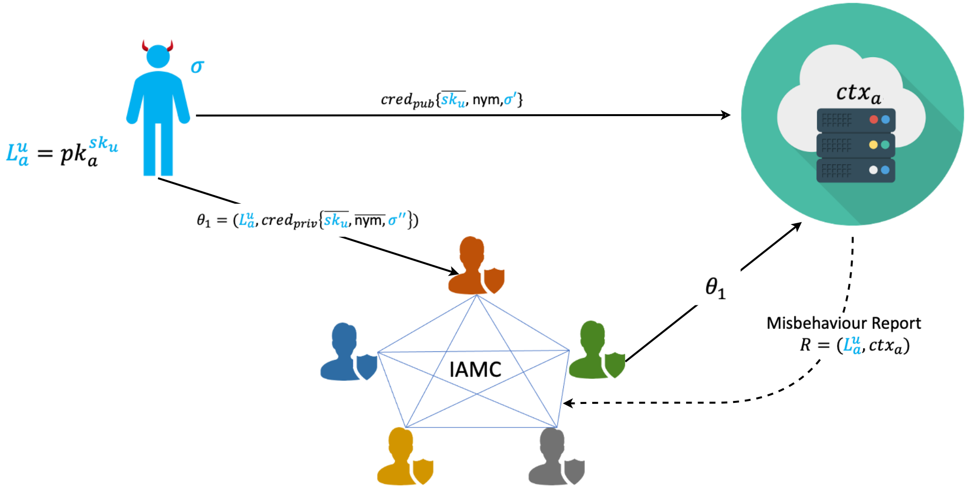
3PBCS: Dynamic, Context-Specific Blacklisting

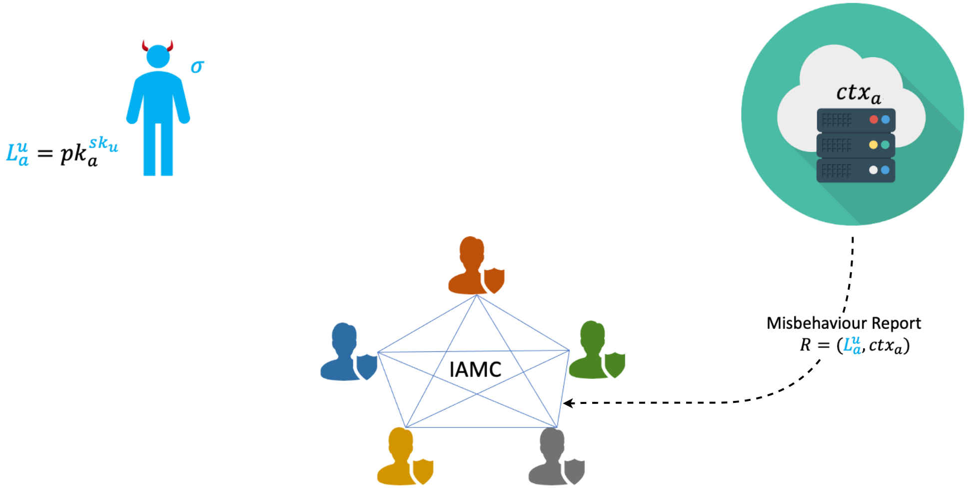


3PBCS: Dynamic, Context-Specific Blacklisting

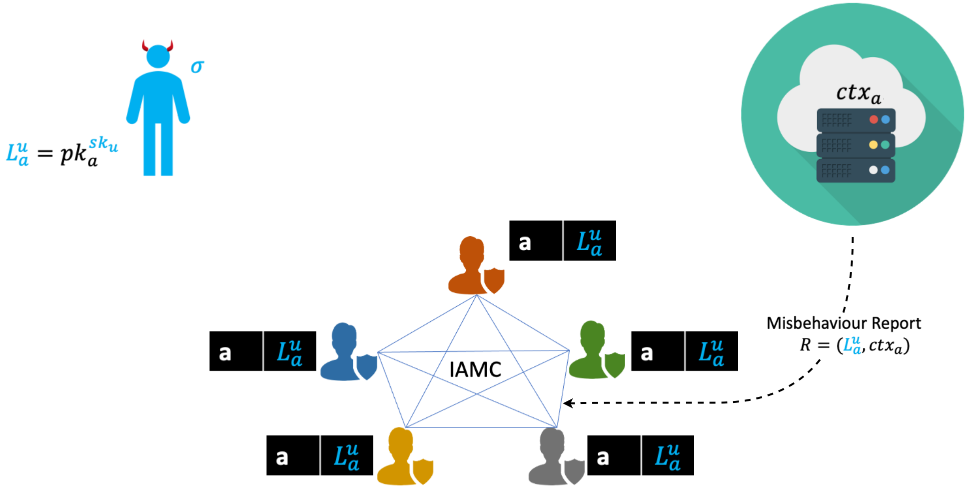


3PBCS: Dynamic, Context-Specific Blacklisting

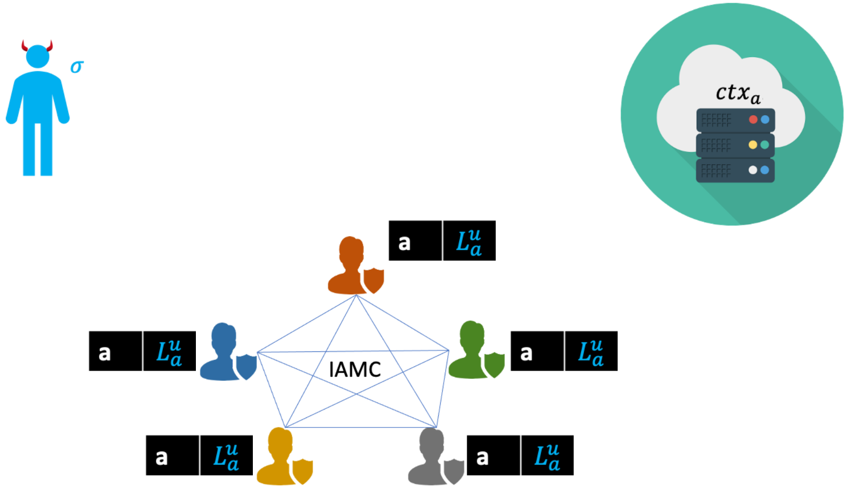



σ
$L_a^u = pk_a^{sk_u}$
ctx_a
IAMC
Misbehaviour Report
$R = (L_a^u, ctx_a)$

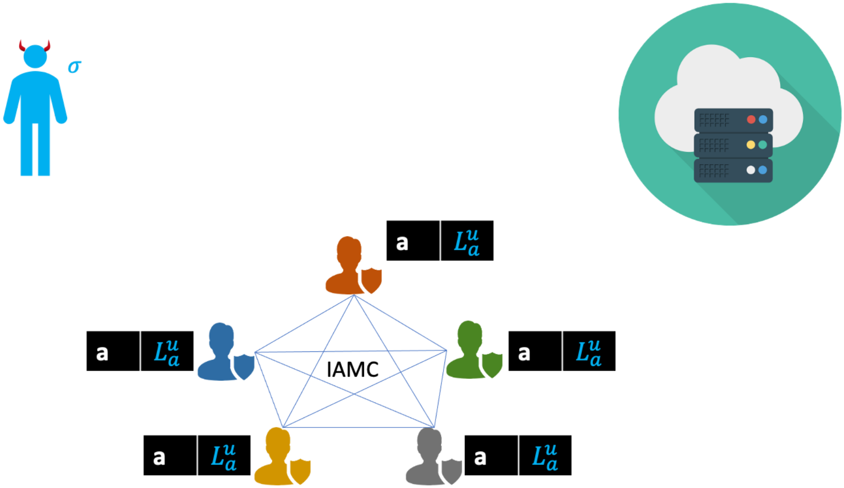
3PBCS: Dynamic, Context-Specific Blacklisting



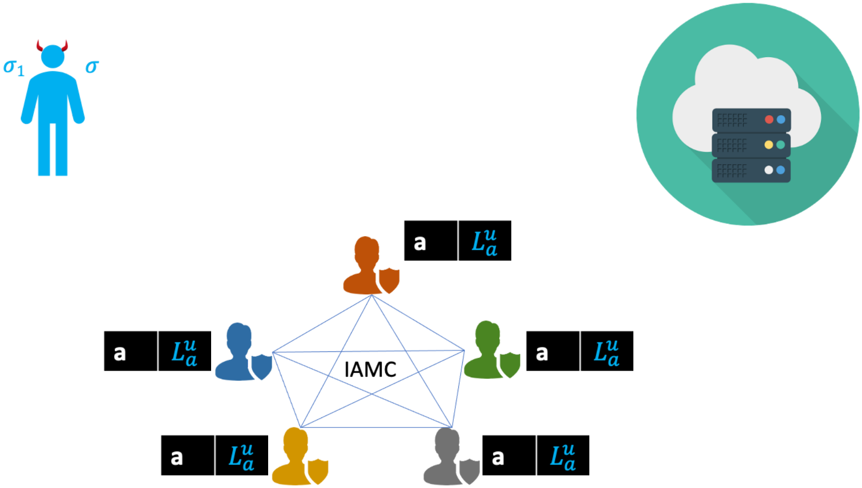
3PBCS: Dynamic, Context-Specific Blacklisting



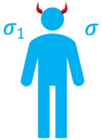
3PBCS: Dynamic, Context-Specific Blacklisting

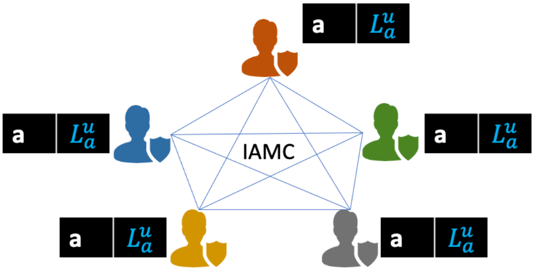


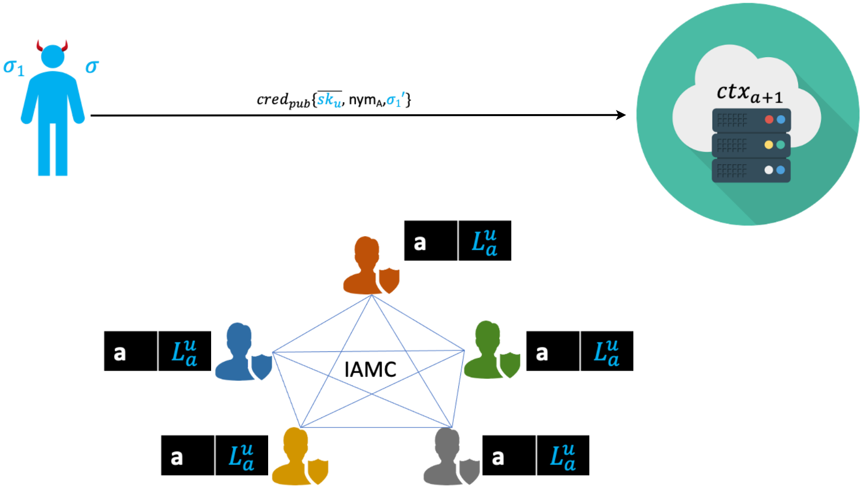
3PBCS: Dynamic, Context-Specific Blacklisting



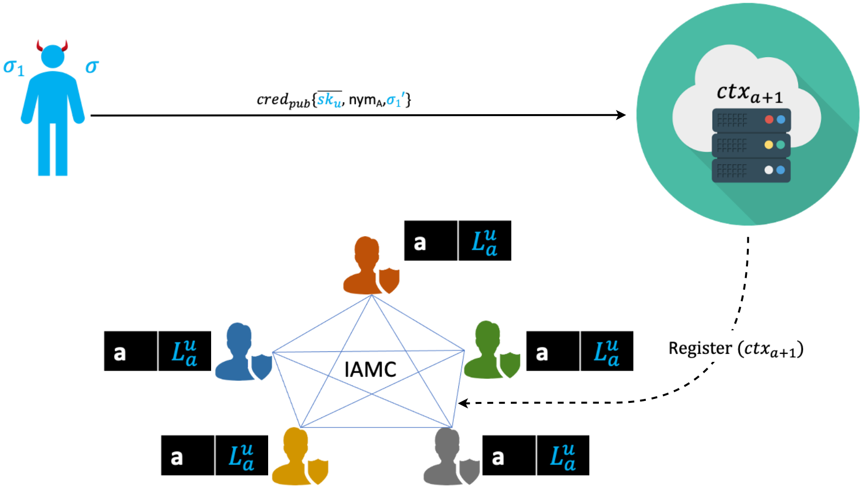
3PBCS: Dynamic, Context-Specific Blacklisting



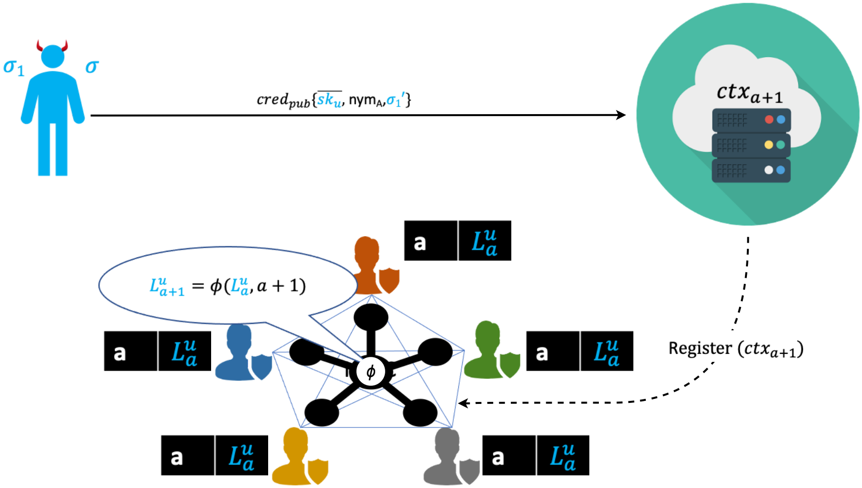



σ_1
σ
$cred_{pub}\{\overline{sk_u}$, nym$_A$,$\sigma_1{}'\}$
ctx_{a+1}
a L_a^u
a L_a^u
a L_a^u
IAMC
a L_a^u
a L_a^u

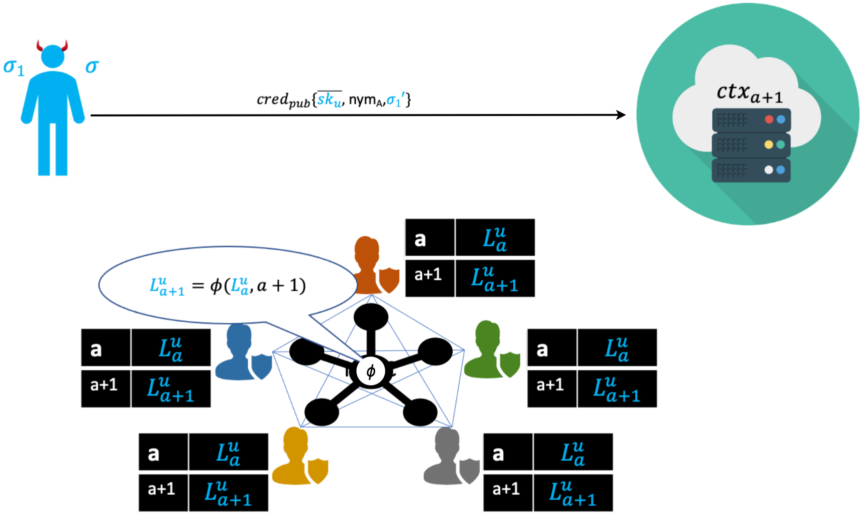
3PBCS: Dynamic, Context-Specific Blacklisting



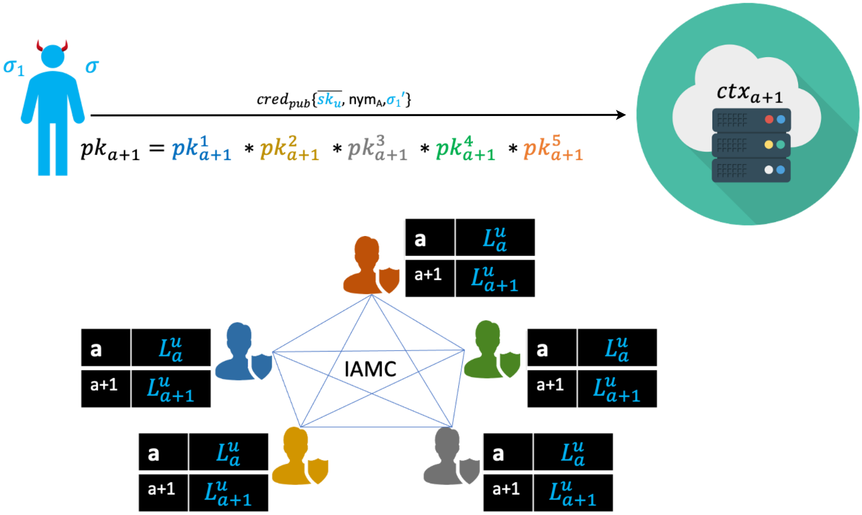
3PBCS: Dynamic, Context-Specific Blacklisting



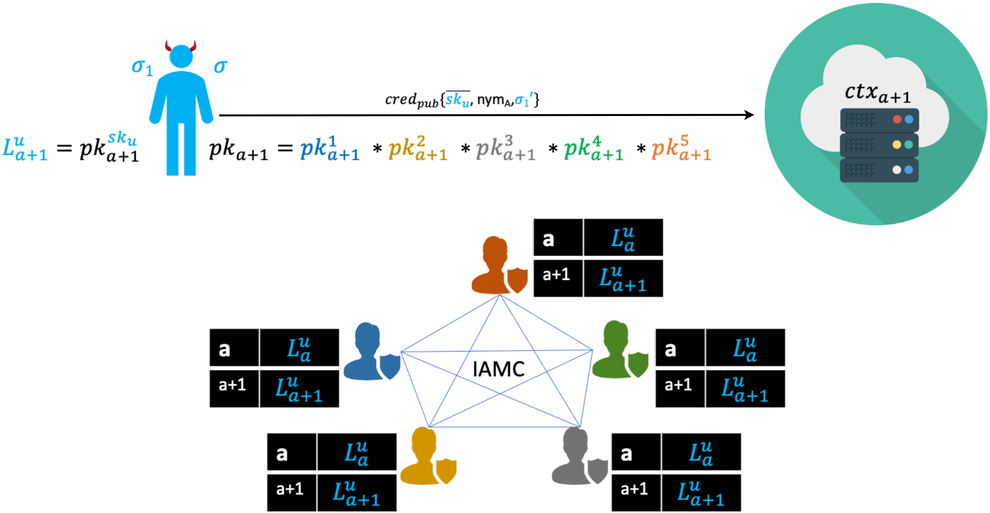
3PBCS: Dynamic, Context-Specific Blacklisting



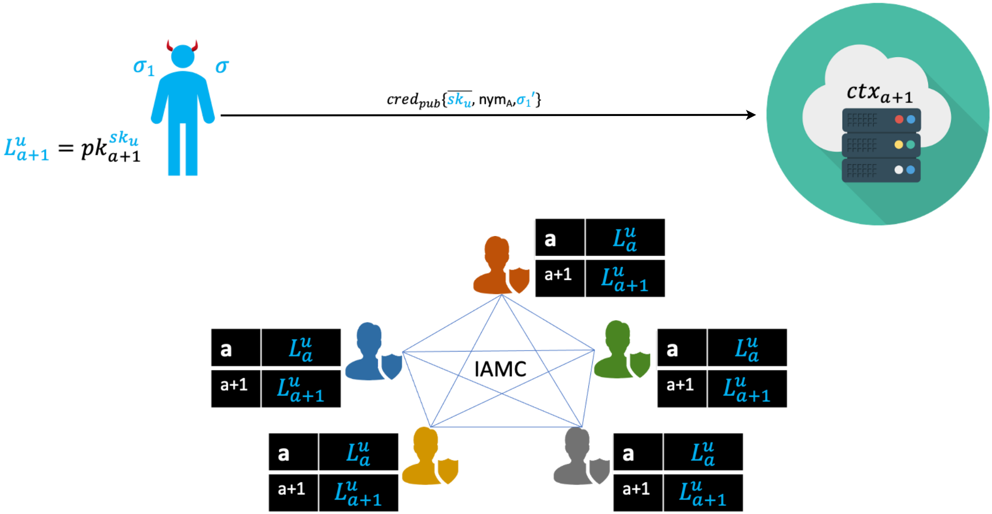
3PBCS: Dynamic, Context-Specific Blacklisting



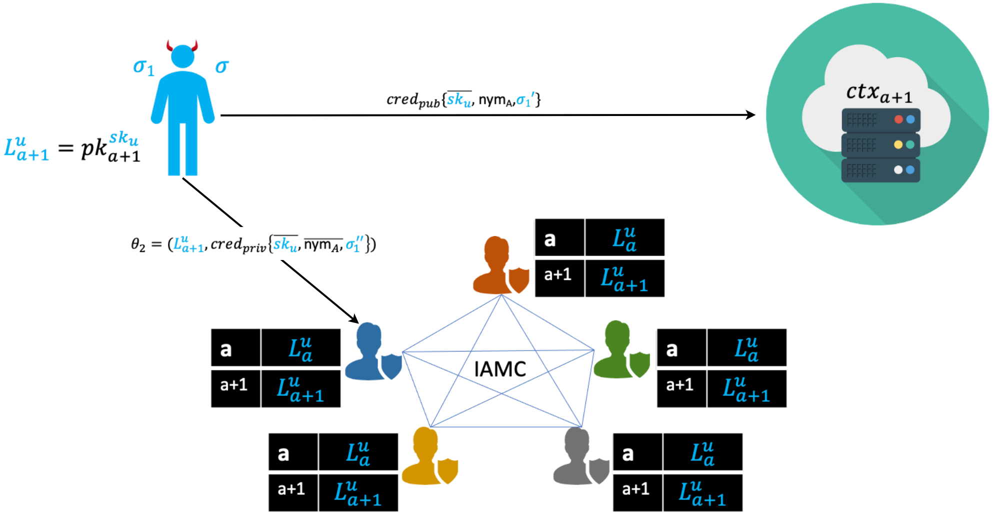
3PBCS: Dynamic, Context-Specific Blacklisting



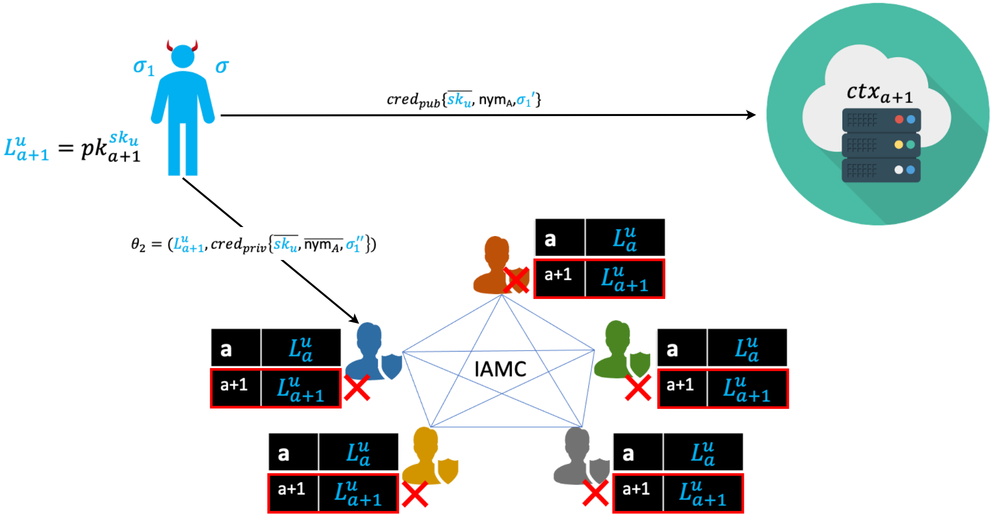
3PBCS: Dynamic, Context-Specific Blacklisting



3PBCS: Dynamic, Context-Specific Blacklisting



3PBCS: Dynamic, Context-Specific Blacklisting



Dynamic, Context-specific Blacklists: Context Registration I

Upon receiving registration request from the service provider, IAMC node i sets:

$$R_k^i = PRNG(seed_i, \mathtt{ctx}_k)$$

Then,

- *If* $k = 0$, i.e. it is the first registered context, set

$$sk_k^i = R_k^i$$

- *Else*, set

$$sk_k^i = R_k^i * sk_{k-1}, \text{ where } sk_k = \sum_{i \in [n]} sk_k^i$$

Then the public key share for node i for context $\mathtt{ctx}_k$ will be:

$$pk_k^i = h_{\mathtt{sID}}^{sk_k^i}$$

where $h_{\mathtt{sID}} = H_{\mathbb{G}_1}(\mathtt{sID})$ and $H_{\mathbb{G}_1}(\cdot)$ is a cryptographic hash function mapping to elements of the group $\mathbb{G}_1$. Lastly, the shared public key for context $\mathtt{ctx}_k$, is

$$pk_k = h_{\mathtt{sID}}^{sk_k} = \prod_{i \in [n]} pk_k^i$$

Note that $pk_k^i = pk_{k-1}^{R_k^i}$, and therefore no party needs to learn the common shared secret key of any context at any point in time.

The user after receiving the context identifier $\mathtt{ctx}_k$ from the third-party service, queries the IAMC nodes for their public key shares pk_k^i for this context. Upon receiving all such shares, the user can compute the shared public key $pk_k = \prod_{i \in [n]} pk_k^i$.

Providing Linkage Tags I

◇ **ProvideLinkageTag**$(\texttt{ctx}_k, pk_k, sk_u, \sigma)$: First the user computes a context-specific linkage tag, using the public key for this context derived from IAMC nodes, and the user's secret-key sk_u.

$$L_k^u = pk_k^{sk_u}$$

Next, the user, prepares a credential that contains a single private claim

$$\overline{\texttt{claim}}_{sk_u} : \{\overline{sk_u}, \phi'_L, \pi_{\phi'_L}, \sigma'_L\}$$

where ϕ'_L: "L_k^u `was properly computed using` sk_u" and

$$\pi_{\phi_L} = NIZK\left\{sk_u : L_k^u = pk_k^{sk_u}\right\}$$

whereas σ'_L is a re-randomization of the signature σ received upon issuance of the credential.

Providing Linkage Tags II

This can be done using the feature of *Selective-Disclosure* in the *Coconut*, setting all attributes as private, i.e. $M = M_{\mathrm{prv}}$, and using ϕ'_L as described above as a single predicate:

$$\begin{aligned}\mathtt{cred}_{\mathrm{anon}} =& \left\{ \textbf{ProveCred}(vk_0, M_{\mathrm{prv}}, \sigma, \phi'_L), M_{\mathrm{pub}} \right\} \\ =& \left\{ \{M_{\mathrm{prv}}, \Theta_L, \phi'_L\}, M_{\mathrm{pub}} = \emptyset, \sigma'_L \right\}\end{aligned}$$

Note that the credential above does not contain any information on the user, apart from the fact that they hold a legitimately signed credential, and that the secret-key sk_u embedded in this credential has been used to compute L^u_k.

Using the anonymous credential prepared and the linkage tag computed,the user composes the following message object:

$$Tag^u_k = \{\mathtt{ctx}_k, \mathtt{cred}_{\mathrm{anon}}, L^u_k\}$$

The MixNetwork

IAMC nodes form a layered mixnet architecture of three layers, with entry (IN), first layer (L1) and exit nodes (OUT) on each path.

Let the pair-wise disjoint sets $\mathcal{S}_{\texttt{in}}, \mathcal{S}_{\texttt{L1}}, \mathcal{S}_{\texttt{out}} \subset \mathcal{S}$ denote the nodes corresponding to each of these layers respectively. Paths are computed in a *source-routing* manner as follows then:

- **SetMixRoute**$(Tag_k^u) \longrightarrow (mix_{\texttt{in}}, mix_{\texttt{L1}}, mix_{\texttt{out}})$:

  Parse Tag_k^u as $\{\texttt{ctx}_k, \texttt{cred}_{\texttt{anon}}, L_k^u\}$
  1. $mix_{\texttt{in}} \leftarrow \mathcal{H}_{\texttt{mix}}(\texttt{ctx}_{\texttt{k}})$, where
     $\mathcal{H}_{\texttt{mix}} : \{0,1\}^{\lambda} \rightarrow \mathcal{S}_{\texttt{in}}$ is a cryptographic hash function public to all users.
  2. $mix_{\texttt{L1}} \xleftarrow{\$} \mathcal{S}_{\texttt{L1}}$.
  3. $mix_{\texttt{out}} \xleftarrow{\$} \mathcal{S}_{\texttt{out}}$.

  Return $(mix_{\texttt{in}}, mix_{\texttt{L1}}, mix_{\texttt{out}})$.

- The user performs layered encryption on tag message, using public keys of all nodes in the path.
- Having tags of the same context sent to unique entry nodes, enables threshold batching: the entry nodes will ensure that they have received a threshold $\tau \geq 2$ of tags for each context, before relaying them to the next node in layer L1.

Blacklist Updates I

Upon initialization, each $node_i \in \mathcal{S}$ from the IAMC, initializes its local blacklist hashtable `blacklist[]` $\leftarrow \emptyset$.

Then, upon receiving the misbehaviour report $\{\mathtt{ctx_j}, L^u_j\}$, for user u under context ctx_j each $node_i \in \mathcal{S}$ proceeds as follows:

◊ **BlacklistReport**$(L^u_j, \mathtt{ctx_j})$:

$\mathtt{for}\ \ n = 0;\ \ j + n \leq k;\ \ n\texttt{++}:$

$\qquad \mathtt{blacklist}_i\mathtt{[ctx_{j+n}]}\ \ += L^u_{j+n};$

$\qquad R^i_n = PRNG(seed_i, \mathtt{ctx}_{j+n+1});$

$\qquad \mathtt{broadcast}\ \ (L^u_{j+n})^{R^i_n};$

$\qquad \mathtt{while\ not}\ \ ((L^u_{j+n})^{R^s_n}\ \ \mathtt{received}\ \forall s \in \mathcal{S}):$

$\qquad\qquad \mathtt{wait();}$

$\qquad L^u_{j+n+1}\ \ = \prod_{s \in \mathcal{S}} (L^u_{j+n})^{R^s_n};$

Blacklist Updates II

Additionally, to maintain the blacklist across new contexts being created, the nodes run the following procedure every time a new context $\mathtt{ctx_k}$ is registered (by a third-party service) and **RegisterContext**($\mathtt{ctx_k}$) is executed:

- **UpdateBlacklist**($\mathtt{ctx_k}$) :

  $R_k^i = PRNG(seed_i, \mathtt{ctx}_k)$;

  for L in blacklist[$\mathtt{ctx_{k\text{-}1}}$]:

      broadcast $(L)^{R_k^i}$;

      while not $((L)^{R_k^s}$ received $\forall s \in \mathcal{S})$:

          wait();

      blacklist[$\mathtt{ctx}_k$] $+= \prod_{s\in\mathcal{S}} (L)^{R_n^s}$;

Correctness of Blacklist Entries

We recall that a linkage tag for context $\texttt{ctx}_k$ from user u is computed according to the procedures **RegisterContext** and **ProvideLinkageTag**, described above, where

$$L_k^u = pk_k^{sk_u} = h_{\texttt{sID}}^{sk_u * sk_k} = h_{\texttt{sID}}^{sk_u \sum_{s \in \mathcal{S}} sk_k^s}$$

Moreover, recall that $\forall s \in \mathcal{S}$ we have that $sk_{k+1}^s = R_{k+1}^s * sk_k$, where $R_{k+1}^s = PRNG(seed_i, \texttt{ctx}_{k+1})$, yielding

$$sk_{k+1} = \sum_{s \in \mathcal{S}} (sk_{k+1}^s) = \sum_{s \in \mathcal{S}} (R_{k+1}^s * sk_k) = sk_k \sum_{s \in \mathcal{S}} R_{k+1}^s$$

Note that in procedures **BlacklistReport** and **UpdateBlacklist** above we have

$$\begin{aligned} L_{k+1}^u &= \prod_{s \in \mathcal{S}} (L_k^u)^{R_{k+1}^s} = \prod_{s \in \mathcal{S}} h_{\texttt{sID}}^{sk_u * sk_k * R_{k+1}^s} \\ &= h_{\texttt{sID}}^{sk_u * sk_k * \sum_{s \in \mathcal{S}} R_{k+1}^s} = h_{\texttt{sID}}^{sk_u * sk_{k+1}} \\ &= pk_{k+1}^{sk_u} \end{aligned}$$

Hence, the linkage tag collectively computed by the IAMC nodes, corresponds to the tag that would be computed by the user themselves for context $\texttt{ctx}_{k+1}$.

Security of Blacklist Entries

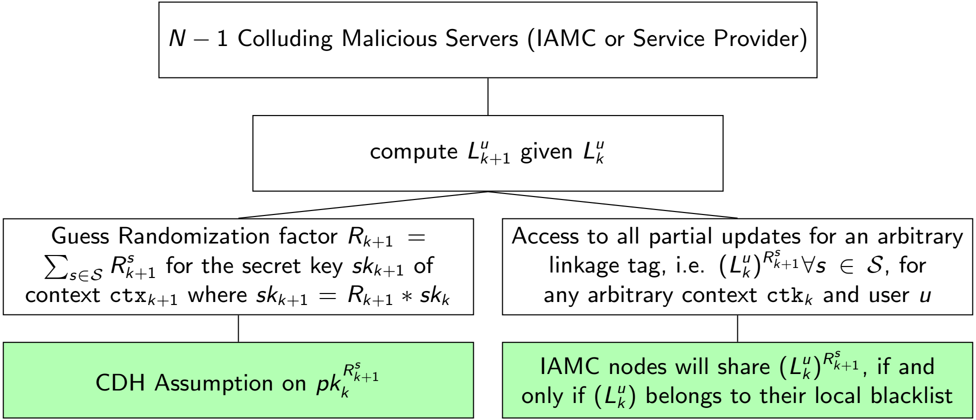


Figure: Attack Tree for Blacklist Entries

Complexity Analysis

Procedure	Communication	Size
RequestCredential	$\mathcal{O}(n)$	$\mathcal{O}(m+q)$
IssueCredential	$\mathcal{O}(n)$	$\mathcal{O}(m)$
ProveCredential	$\mathcal{O}(1)$	$\mathcal{O}(m)$
ProvideLinkageTag	$\mathcal{O}(n+r)$	$\mathcal{O}(m)$
VerifyCredential	$\mathcal{O}(1)$	$\mathcal{O}(1)$
VerifyLinkageTag	$\mathcal{O}(1)$	$\mathcal{O}(1)$
RegisterContext	$\mathcal{O}(n)$	$\mathcal{O}(1)$
UpdateBlacklist	$\mathcal{O}(n^2)$	$\mathcal{O}(1)$

Table: Communication and Size Complexity for 3PBCS procedures. n - number of IAMC nodes; m - number of private credential attributes; q - PoP Transcript Size; r - length of mix-route.

Evaluation: Effect of PoP Transcript Size on Performance

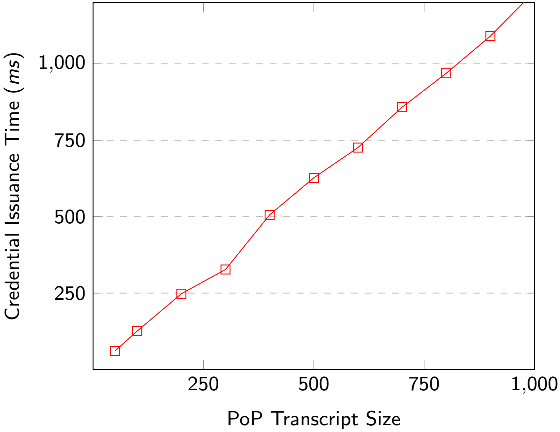


Figure: Effect of PoP Transcript size on credential issuance in 3PBCS.