A Comprehensive Framework Enabling Data-Minimizing Authentication
Facebook* and you

If you’re not paying for it, you’re not the customer – you’re the product.
Facebook* and you

If you’re not paying for it, you’re not the customer – you’re the product.
What is the Problem?

Personal data is everywhere and data breaches happen.

Example (April 2011 - now):

- 100 Million user accounts of Sony services compromised
- Attributes including name, date of birth, or credit card
- Costs: 170 Mio. $ due to network outage, stock price drop, lawsuit (US), reputation loss
- Enables “spear phishing” and identity theft

What if Sony had never collected the data they don’t need?
What is the Problem?

Personal data is everywhere and data breaches happen.

Example (April 2011 - now):

- 100 Million user accounts of Sony services compromised
- Attributes including name, date of birth, or credit card
- Costs: 170 Mio. $ due to network outage, stock price drop, lawsuit (US), reputation loss
- Enables “spear phishing” and identity theft

What if Sony had never collected the data they don’t need?
What is the Problem?

Personal data is everywhere and data breaches happen.

Example (April 2011 - now):

- 100 Million user accounts of Sony services compromised
- Attributes including name, date of birth, or credit card
- Costs: 170 Mio. $ due to network outage, stock price drop, lawsuit (US), reputation loss
- Enables “spear phishing” and identity theft

What if Sony had never collected the data they don’t need?
What is the Problem?

Personal data is everywhere and data breaches happen.

Example (April 2011 - now):

- 100 Million user accounts of Sony services compromised
- Attributes including name, date of birth, or credit card
- Costs: 170 Mio. $ due to network outage, stock price drop, lawsuit (US), reputation loss
- Enables “spear phishing” and identity theft

What if Sony had never collected the data they don’t need?
Outline

1 Motivation & Goals
2 Data-Minimization Concepts
   - Credentials
   - Anonymous Credentials
3 Credential-based Authentication Framework
   - System Model
   - CARL Policy Language
   - Claim Language
   - Open Source Software
Outline

1 Motivation & Goals

2 Data-Minimization Concepts
   - Credentials
   - Anonymous Credentials

3 Credential-based Authentication Framework
   - System Model
   - CARL Policy Language
   - Claim Language
   - Open Source Software
Data-Minimization

Minimize possibility to collect personal data.

Minimize collection of personal data in remaining possibilities.

Minimize time collected data is stored.

[PfitzmannHansen 2010]
Data-Minimization

Minimize *possibility* to collect personal data.

Minimize *collection* of personal data in remaining possibilities.

Minimize *time* collected data is stored.

[PfitzmannHansen 2010]
Data-Minimization

Minimize **possibility** to collect personal data.

Minimize **collection** of personal data in remaining possibilities.

Minimize **time** collected data is stored.

[PfitzmannHansen 2010]
Data-Minimization

- **Safety** – due to protection against data loss
  - “What you don’t have, you cannot lose.”
  - Cost savings for data protection
  - Remove unused data immediately

- **Privacy**
  - Less profiling, identity theft, etc.
  - Policy specifies what is minimal

→ Data-Minimizing Authentication using attribute-based credentials
Data-Minimization

- Safety – due to protection against data loss
  - “What you don’t have, you cannot loose.”
  - Cost savings for data protection
  - Remove unused data immediately

- Privacy
  - Less profiling, identity theft, etc.
  - Policy specifies what is minimal

→ Data-Minimizing Authentication using attribute-based credentials
Data-Minimization

- **Safety** – due to protection against data loss
  - “What you don’t have, you cannot loose.”
  - Cost savings for data protection
  - Remove unused data immediately

- **Privacy**
  - Less profiling, identity theft, etc.
  - Policy specifies what is minimal

→ Data-Minimizing Authentication using attribute-based credentials
Attribute-Based Credentials

*Technically:*

- Authentic bundle of attribute-value pairs
- Issuer (e.g., Swiss Government)
- Type (e.g., identity card, university diploma, movie ticket)

*Conceptually: Technology-independent, long-lived, (digital)*

*Implementations: Anonymous Credentials (IBM Identity Mixer and MS U-Prove), X.509, Trusted LDAP, OpenID, etc.*
Attribute-Based Credentials

Technically:

- Authentic bundle of attribute-value pairs
- Issuer (e.g., Swiss Government)
- Type (e.g., identity card, university diploma, movie ticket)

Conceptually: Technology-independent, long-lived, (digital)

Implementations: Anonymous Credentials (IBM Identity Mixer and MS U-Prove), X.509, Trusted LDAP, OpenID, etc.
Attribute-Based Credentials

Technically:

- Authentic bundle of attribute-value pairs
- Issuer (e.g., Swiss Government)
- Type (e.g., identity card, university diploma, movie ticket)

Conceptually: Technology-independent, long-lived, (digital)

Implementations: Anonymous Credentials (IBM Identity Mixer and MS U-Prove), X.509, Trusted LDAP, OpenID, etc.
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
- Attribute Properties (age $\leq 19$)
- Selective Disclosure to Third Parties
- Statement Signatures
- Consumption Control
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
  - Attribute Properties (age \leq 19)
  - Selective Disclosure to Third Parties
  - Statement Signatures
  - Consumption Control
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
- Attribute Properties (age ≤ 19)
- Selective Disclosure to Third Parties
- Statement Signatures
- Consumption Control
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
- Attribute Properties (age \( \leq 19 \))
- Selective Disclosure to Third Parties
- Statement Signatures
- Consumption Control
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
- Attribute Properties (age \( \leq 19 \))
- Selective Disclosure to Third Parties
- Statement Signatures
- Consumption Control
Anonymous Credentials

- Non-interactive Proof Of Ownership (vs. revealing whole credential)
- Selective Attribute Disclosure (vs. revealing whole credential)
- Attribute Properties (age $\leq 19$)
- Selective Disclosure to Third Parties
- Statement Signatures
- Consumption Control
Outline

1 Motivation & Goals
2 Data-Minimization Concepts
   - Credentials
   - Anonymous Credentials
3 Credential-based Authentication Framework
   - System Model
   - CARL Policy Language
   - Claim Language
   - Open Source Software
System Model

User

Credentials

Claim generation
\( t_a \equiv t_b \) (2a)

Claim selection
(2b)

Evidence generation
\( t_a \equiv t_b \) (2c)

Authenticator

Policy discovery & pre-evaluation (1a)

Policies

Evidence verification
\( t_a \equiv t_c \) (3b)

Claim verification
(3a)

applicable policy

Claim, evidence

(3) Claim, evidence

(2) Applicable policy

(1) AuthN request

(4) Success/Failure

(4) Success/Failure
Credential-based Authentication Requirements Language (CARL)

Discounted car rental policy

\[
\text{own } mc::\text{MemberShipCard issued-by CarRentalCo} \\
\text{own } cc::\text{CreditCard issued-by Amex, Visa} \\
\text{own } dl::\text{DriversLicense issued-by DeptMotorVehicles} \\
\text{own } li::\text{LiabilityIns issued-by InsuranceCo, OtherInsurance} \\
\text{reveal } cc.\text{number to } cc.\text{issuer under ‘purpose=payment’} \\
\text{reveal } li.\text{policyNo to EscrowAgent under ‘in case of damage’} \\
\text{sign ‘I agree with the general terms and conditions.’} \\
\text{where } dl.\text{vehicleCategoryB }== \text{ true } \land \\
\quad \text{li.guaranteedUSDAmount } \geq 30.000 \land \\
\quad (mc.\text{status }== \text{ ‘gold’ } \lor mc.\text{status }== \text{ ‘silver’ }) \land \\
\quad mc.\text{name }== \text{ dl.\text{name}}
\]
Credential-based Authentication Requirements Language (CARL)

Discounted car rental policy

own mc::MemberShipCard issued-by CarRentalCo

own cc::CreditCard issued-by Amex, Visa

own dl::DriversLicense issued-by DeptMotorVehicles

own li::LiabilityIns issued-by InsuranceCo, OtherInsurance

reveal cc.number to cc.issuer under ‘purpose=payment’

reveal li.policyNo to EscrowAgent under ‘in case of damage’

sign ‘I agree with the general terms and conditions.’

where dl.vehicleCategoryB == true ∧

li.guaranteedUSDAmount ≥ 30.000 ∧

(mc.status = ‘gold’ ∨ mc.status = ‘silver’) ∧

mc.name == dl.name
Discounted car rental policy

own \textit{mc}::\textit{MembershipCard} issued-by \textit{CarRentalCo}

own \textit{cc}::\textit{CreditCard} issued-by \textit{Amex, Visa}

own \textit{dl}::\textit{DriversLicense} issued-by \textit{DeptMotorVehicles}

own \textit{li}::\textit{LiabilityIns} issued-by \textit{InsuranceCo, OtherInsurance}

reveal \textit{cc.number} to \textit{cc.issuer} under ‘purpose=payment’

reveal \textit{li.policyNo} to \textit{EscrowAgent} under ‘in case of damage’

sign ‘I agree with the general terms and conditions.’

where \textit{dl.vehicleCategoryB} \texttt{==} \texttt{true} \land

\hspace{1cm} \textit{li.guaranteedUSDAmount} \texttt{\geq} \texttt{30.000} \land

\hspace{2cm} (\textit{mc.status} \texttt{=} ‘gold’ \lor \textit{mc.status} \texttt{=} ‘silver’) \land

\hspace{3cm} \textit{mc.name} \texttt{==} \textit{dl.name}
Credential-based Authentication Requirements Language (CARL)

Discounted car rental policy

own \textit{mc}::\textit{MemberShipCard} issued-by \textit{CarRentalCo}

own \textit{cc}::\textit{CreditCard} issued-by \textit{Amex, Visa}

own \textit{dl}::\textit{DriversLicense} issued-by \textit{DeptMotorVehicles}

own \textit{li}::\textit{LiabilityIns} issued-by \textit{InsuranceCo, OtherInsurance}

reveal \textit{cc.number} to \textit{cc.issuer} under ‘purpose=payment’

reveal \textit{li.policyNo} to \textit{EscrowAgent} under ‘in case of damage’

sign ‘I agree with the general terms and conditions.’

where \textit{dl.vehicleCategoryB} == true \land

\textit{li.guaranteedUSDAmount} \geq 30.000 \land

\left( \textit{mc.status} = \text{‘gold’} \lor \textit{mc.status} = \text{‘silver’} \right) \land

\textit{mc.name} == \textit{dl.name}
Discounted car rental policy

\[
\begin{align*}
\text{own } mc::\text{MemberShipCard} & \text{ issued-by CarRentalCo} \\
\text{own } cc::\text{CreditCard} & \text{ issued-by Amex, Visa} \\
\text{own } dl::\text{DriversLicense} & \text{ issued-by DeptMotorVehicles} \\
\text{own } li::\text{LiabilityIns} & \text{ issued-by InsuranceCo, OtherInsurance} \\
\text{reveal } cc\cdot\text{number} & \text{ to } cc\cdot\text{issuer} \text{ under ‘purpose=payment’} \\
\text{reveal } li\cdot\text{policyNo} & \text{ to EscrowAgent under ‘in case of damage’} \\
\text{sign ‘I agree with the general terms and conditions.’} \\
\text{where } dl\cdot\text{vehicleCategoryB} & \equiv \text{true} \land \\
\text{ } & \text{li.guaranteedUSDAmount} \geq 30.000 \land \\
\text{ } & \text{(mc.status = ‘gold’} \lor \text{mc.status = ‘silver’}) \land \\
\text{ } & \text{mc.name} \equiv \text{dl.name}
\end{align*}
\]
Credential-based Authentication Requirements Language (CARL)

Discounted car rental policy

own \texttt{mc::MemberShipCard} issued-by \texttt{CarRentalCo} 

own \texttt{cc::CreditCard} issued-by \texttt{Amex, Visa} 

own \texttt{dl::DriversLicense} issued-by \texttt{DeptMotorVehicles} 

own \texttt{li::LiabilityIns} issued-by \texttt{InsuranceCo, OtherInsurance} 

reveal \texttt{cc.number} to \texttt{cc.issuer} under ‘purpose=payment’ 

reveal \texttt{li.policyNo} to \texttt{EscrowAgent} under ‘in case of damage’ 

sign ‘I agree with the general terms and conditions.’ 

where \texttt{dl.vehicleCategoryB == true} \land 

\texttt{li.guaranteedUSDAmount} \geq 30.000 \land 

(mc.status = ‘gold’ \lor mc.status = ‘silver’) \land 

mc.name == dl.name
Credential-based Authentication Requirements Language (CARL)

Discounted car rental policy

own \texttt{mc::MembershipCard} issued-by \texttt{CarRentalCo}
own \texttt{cc::CreditCard} issued-by \texttt{Amex, Visa}
own \texttt{dl::DriversLicense} issued-by \texttt{DeptMotorVehicles}
own \texttt{li::LiabilityIns} issued-by \texttt{InsuranceCo, OtherInsurance}
reveal \texttt{cc.number} to \texttt{cc.issuer} under ‘purpose=payment’
reveal \texttt{li.policyNo} to \texttt{EscrowAgent} under ‘in case of damage’
sign ‘I agree with the general terms and conditions.’
where \texttt{dl.vehicleCategoryB} \texttt{== true} \land
  \texttt{li.guaranteedUSDAmout} \geq 30.000 \land
  (\texttt{mc.status} = \texttt{‘gold’} \lor \texttt{mc.status} = \texttt{‘silver’}) \land
  \texttt{mc.name} \texttt{== dl.name}
System Model

User

Claim generation

Claim selection

Evidence generation

Claim verification

Successful

Failure

Authenticator

Policy discovery & pre-evaluation

Policy

Evidence verification

Evidence verification
Pre-evaluate constant sub-expressions (e.g., \textit{today}())
System Model

User

Credentials

Claim generation

Claim selection

Evidence generation

Authenticator

Policy discovery & pre-evaluation

(1a)

Policy

Evidence verification

(3a)

Evidence

(3b)

(2c)

(2a)

(2b)

(3a)

(3b)

(4)

Success/Failure

(1) AuthN request

(2) Applicable policy

(3) Claim, evidence

13 / 25

System Model

- Determine all **claims** that fulfill the policy
Claim Language

Policy: Requirements expressed by authenticator.
Claim: Statement made by user – subset of policy language.

- Constraint 1: One Issuer per own-line
  own cc:: CreditCard issued-by AmericanExpress

- Constraint 2: No disclosure requests
  reveal if policyNo

- Obligation: Disclose requested attributes
  where if policyNo == "2011IBM/34780234"
Claim Language

Policy: Requirements expressed by authenticator.
Claim: Statement made by user – subset of policy language.

- Constraint 1: One Issuer per own-line
- Constraint 2: No disclosure requests
- Obligation: Disclose requested attributes where if policyNo = '2011IBM/23702311'
Claim Language

Policy: Requirements expressed by authenticator.
Claim: Statement made by user – subset of policy language.

- Constraint 1: One **Issuer** per own-line

  own cc::CreditCard issued-by Amex, Visa

- Constraint 2: No disclosure requests

  reveal li.policyNo

- Obligation: Disclose requested attributes

  where li.policyNo == ‘2011/3M/9570234’
Claim Language

Policy: Requirements expressed by authenticator.
Claim: Statement made by user – subset of policy language.

- Constraint 1: One **Issuer** per own-line
  
  own **cc::CreditCard** issued-by **Amex, Visa**

- Constraint 2: No disclosure requests
  
  reveal **li.policyNo**

- Obligation: Disclose requested attributes
  
  where **li.policyNo == '2011/3M/9570234'**
Claim Language

Policy: Requirements expressed by authenticator.
Claim: Statement made by user – subset of policy language.

- Constraint 1: One **ISSUE**R per own-line
  own cc::CreditCard issued-by **A**MEX, **V**ISA

- Constraint 2: No disclosure requests
  reveal li.p**o**licyNo

- Obligation: Disclose requested attributes
  where li.p**o**licyNo == '2011/3M/9570234'
Claim Generation

Procedure:

1. Determine all valid assignments w.r.t. policy & credentials

2. Create claim(s) for each assignment with tech.-spec. algorithm

Credential Assignment

A mapping from all credential variables occurring in a policy to credentials that a user owns.

\( m \) credentials, \( n \) credential variables \( c_i \): \( m^n \) possible assignments
Claim Generation

Procedure:

1. Determine all valid assignments w.r.t. policy & credentials
2. Create claim(s) for each assignment with tech.-spec. algorithm

Credential Assignment

A mapping from all credential variables occurring in a policy to credentials that a user owns.

\( m \) credentials, \( n \) credential variables \( c_i \): \( m^n \) possible assignments
Claim Generation Example

Discounted car rental policy

own mc::MemberShipCard issued-by CarRentalCo
own cc::CreditCard issued-by Amex, Visa
own dl::DriversLicense issued-by DeptMotorVehicles
reveal cc.number to cc.issuer under ‘purpose=payment’
where dl.vehicleCategoryB == true ∧
  (mc.status = ‘gold’ ∨ mc.status = ‘silver’) ∧
  mc.name == dl.name

Assignment 1

Assignment 2

Assignment 3
Claim Generation Example

Discounted car rental policy

own \textit{mc::MemberShipCard} issued-by \textit{CarRentalCo}

own \textit{cc::CreditCard} issued-by \textit{Amex, Visa}

own \textit{dl::DriversLicense} issued-by \textit{DeptMotorVehicles}

reveal \textit{cc.number} to \textit{cc.issuer} under ‘purpose=payment’

where \textit{dl.vehicleCategoryB} == true \land

(\textit{mc.status} = ‘gold’ \lor \textit{mc.status} = ‘silver’) \land

\textit{mc.name} == \textit{dl.name}
Discounted car rental policy

own \texttt{mc::MembershipCard} issued-by \texttt{CarRentalCo}

own \texttt{cc::CreditCard} issued-by \texttt{Amex, Visa}

own \texttt{dl::DriversLicense} issued-by \texttt{DeptMotorVehicles}

reveal \texttt{cc.number} to \texttt{cc.issuer} under ‘purpose=payment’

where \texttt{dl.vehicleCategoryB} == true \land

(\texttt{mc.status} = ‘gold’ \lor \texttt{mc.status} = ‘silver’) \land

\texttt{mc.name} == \texttt{dl.name}
Discounted car rental policy

own \texttt{mc::MembershipCard} issued-by \texttt{CarRentalCo}
own \texttt{cc::CreditCard} issued-by \texttt{Amex, Visa}
own \texttt{dl::DriversLicense} issued-by \texttt{DeptMotorVehicles}
reveal \texttt{cc.number} to \texttt{cc.issuer} under \'purpose=payment\'
where \texttt{dl.vehicleCategoryB == true \land}
\texttt{(mc.status = \textquoteleft \textquoteleft gold\textquoteright\ \lor mc.status = \textquoteleft \textquoteleft silver\textquoteright\ ) \land}
\texttt{mc.name == dl.name}
Claim Generation Example

**Assignment 1**

Discounted car rental policy

own mc::MemberShipCard issued-by CarRentalCo

own cc::CreditCard issued-by Amex, Visa

own dl::DriversLicense issued-by DeptMotorVehicles

reveal cc.number to cc.issuer under ‘purpose=payment’

where dl.vehicleCategoryB == true ∧

(mc.status = ‘gold’ ∨ mc.status = ‘silver’) ∧

mc.name == dl.name

**Assignment 2**

**Assignment 3**
Discounted car rental policy

\[
\text{own } mc::\text{MemberShipCard} \text{ issued-by CarRentalCo}
\]
\[
\text{own } cc::\text{CreditCard} \text{ issued-by Amex, Visa}
\]
\[
\text{own } dl::\text{DriversLicense} \text{ issued-by DeptMotorVehicles}
\]
\[
\text{reveal } cc\text{.number to } cc\text{.issuer under ‘purpose=payment’}
\]
\[
\text{where } dl\text{.vehicleCategoryB} == \text{true} \land
\]
\[
(mc\text{.status} = ‘\text{gold}’ \lor mc\text{.status} = ‘\text{silver}’) \land
\]
\[
mc\text{.name} == dl\text{.name}
\]
Claim Generation Example

Assignment 1

Identity Mixer Credentials

Discounted car rental claim

\[ i \text{ own } mc::\text{MemberShipCard} \text{ issued-by CarRentalCo} \]
\[ i \text{ own } cc::\text{CreditCard} \text{ issued-by Amex, Visa} \]
\[ i \text{ own } dl::\text{DriversLicense} \text{ issued-by DeptMotorVehicles} \]
\[ i \text{ reveal } cc\text{.number} \text{ to Amex under 'purpose=payment'} \]
\[ \text{where } dl\text{.vehicleCategoryB} == \text{true} \wedge \]
\[ (mc\text{.status} == \text{'gold'} \vee mc\text{.status} == \text{'silver'}) \wedge \]
\[ mc\text{.name} == dl\text{.name} \]

X.509 Certificates

Discounted car rental claim

\[ […] \]
\[ \text{where } cc\text{.number} == \text{'4711 0221 4121 2112'} \wedge \]
\[ cc\text{.lastName} == \text{'Bichsel'} \wedge \]
\[ cc\text{.firstName} == \text{'}…’ \]
\[ mc\text{.name} == \text{'Bichsel'} \]
\[ mc\text{.firstName} == \text{'Patrik'} \]
\[ mc\text{.status} == \text{’…’} \]
\[ […] \]
Claim Generation Example

Assignment 1

discounted car rental policy

\[
\begin{align*}
\text{own } & mc::\text{MemberShipCard} \text{ issued-by } \text{CarRentalCo} \\
\text{own } & cc::\text{CreditCard} \text{ issued-by Amex, Visa} \\
\text{own } & dl::\text{DriversLicense} \text{ issued-by DeptMotorVehicles} \\
\text{reveal } & cc.number \text{ to } cc.issuer \text{ under ‘purpose=payment’} \\
\text{where } & dl\_vehicleCategoryB == \text{true} \land \\
& (mc.status == \text{‘gold’} \lor mc.status == \text{‘silver’}) \land \\
& mc.name == dl.name
\end{align*}
\]

Identity Mixer Credentials

discounted car rental claim

\[
\begin{align*}
i \text{ own } & mc::\text{MemberShipCard} \text{ issued-by } \text{CarRentalCo} \\
i \text{ own } & cc::\text{CreditCard} \text{ issued-by Amex} \\
i \text{ own } & dl::\text{DriversLicense} \text{ issued-by DeptMotorVehicles} \\
i \text{ reveal } & cc.number \text{ to Amex under ‘purpose=payment’} \\
\text{where } & dl\_vehicleCategoryB == \text{true} \land \\
& mc.status == \text{‘silver’} \land \\
& mc.name == dl.name
\end{align*}
\]

X.509 Certificates

discounted car rental claim

\[
\begin{align*}
\text{[...]} \\
\text{where } & cc.number == \text{‘4711 0221 4121 2112’} \land \\
& cc.lastName == \text{‘Bichsel’} \land \\
& cc.firstName == \text{‘...’} \\
& mc.name == \text{‘Bichsel’} \\
& mc.firstName == \text{‘Patrik’} \\
& mc.status == \text{‘...’} \\
\text{[...]} 
\end{align*}
\]
Claim Generation Example

Assignment 1

Identity Mixer Credentials

Discounted car rental claim

i own mc::MemberShipCard issued-by CarRentalCo
i own cc::CreditCard issued-by Amex, Visa
i own dl::DriversLicense issued-by DeptMotorVehicles
i reveal cc.number to cc.issuer under ‘purpose=payment’
where dl.vehicleCategoryB == true ∧
(mc.status = ‘gold’ ∨ mc.status = ‘silver’) ∧
mc.name == dl.name

X.509 Certificates

Discounted car rental claim

[...
where cc.number == ‘4711 0221 4121 2112’ ∧
cc.lastName == ‘Bichsel’ ∧
cc.firstName == ‘…’
mc.name == ‘Bichsel’
mc.firstName == ‘Patrik’
mc.status == ‘…’
[...]
System Model

User

Credentials

Claim selection (2b)

Evidence generation (2c)

Claim generation (2a)

(1) AuthN request

(2) Applicable policy

(3) Claim, evidence

(4) Success/Failure

Authenticator

Policy discovery & pre-evaluation (1a)

Evidence verification (3b)

Claim verification (3a)

Policies

 vestibule
System Model
Claim Selection User Interface

Adapted Card

**Revealed ownership:**
- MembershipCard from CarRentalCo.com.
- Driver's License from DeptMotorVehicles.com.

**Revealed attribute values:**
- vehicleCategoryB of Swiss Driver's License: true.
- cardNumber of Corporate American Express Card: 6473 7797 0122 2199.
- status of Hertz Membership Card: gold.

**Revealed facts:**
- expirationDate of your Corporate American Express Card is after 2011-09-01.
- givenName of your Hertz Membership Card equals firstName of your Swiss Driver's License.
- surname of your Hertz Membership Card equals lastName of your Swiss Driver's License.
System Model

User

- Credentials
  - Claim generation: $t_a$, $t_b$ (2a)
  - Claim selection: $t_a$ (2b)
  - Evidence generation: $t_a$, $t_b$ (2c)

Authenticator

- Policy discovery & pre-evaluation (1a)
- Applicable policy
- Claim verification (3a)
- Claim, evidence
- Evidence verification: $t_a$, $t_c$ (3b)
- bool

Success/Failure

Evidence verification

Policies
System Model

User

Credentials

Claim generation

Claim selection

Evidence generation

(1) AuthN request

(2) Applicable policy

(3) Claim, evidence

(4) Success/Failure

Authenticator

Policy discovery & pre-evaluation (1a)

Evidence verification (3a)

Evidence verification (3b)

bool

Claims

$C_1$, $C_2$, $C_k$

$t_a$, $t_b$

$t_a$, $t_b$, $t_c$
### Evidence Generation

**Identity Mixer Proof Specification**

```xml
<ProofSpecification>
  <Declaration>
    <AttributeId name='id1' proofMode='revealed'/>
    <AttributeId name='id2 ' proofMode='unrevealed'/>
    <AttributeId name='id3 ' proofMode='unrevealed'/>
    ...
  </Declaration>
  <Specification>
    <Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>
      <Attribute name='lastName'>id5</Attribute>
      <Attribute name='number'>id2</Attribute>
    </Credential>
    ...
    <VerifiableEncryptions>
      <VerifiableEncryption name='jd2e0a' pk='Amex' label='payment'>id2</VerifiableEncryption>
    </VerifiableEncryptions>
  </Specification>
</ProofSpecification>
```

**Discounted car rental claim**

- *i own mc::*
  - MembershipCard issued-by CarRentalCo
- *i own cc::*
  - CreditCard issued-by Amex
- *i own dl::*
  - DriversLicense issued-by DeptMotorVehicles

**I reveal cc.number to Amex under ‘purpose=payment’ where dl.vehicleCategoryB == true ^**

- mc.status == ‘silver’ ^
- mc.name == dl.name
Evidence Generation

Discounted car rental claim

\[
i \text{own mc::MembershipCard issued-by CarRentalCo} \\
i \text{own cc::CreditCard issued-by Amex} \\
i \text{own dl::DriversLicense issued-by DeptMotorVehic} \\
i \text{reveal cc.number to Amex under ‘purpose=payment’} \\
\text{where dl.vehicleCategoryB == true} \land \\
\text{mc.status = ‘silver’} \land \\
\text{mc.name == dl.name}
\]

Identity Mixer Proof Specification

\[
<\text{ProofSpecification}> \\
<\text{Declaration}> \\
<\text{AttributeId name='id1' proofMode='revealed'/>} \\
<\text{AttributeId name='id2' proofMode='unrevealed'/>} \\
<\text{AttributeId name='id3' proofMode='unrevealed'/>} \\
\ldots \\
</\text{Declaration}> \\
<\text{Specification}> \\
<\text{Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>} \\
<\text{Attribute name='lastName'>id5</Attribute> \\
<\text{Attribute name='number'>id2</Attribute> \\
</\text{Credential}> \\
\ldots \\
<\text{VerifiableEncryptions}> \\
<\text{VerifiableEncryption name='jd2e0a' pk='Amex' label='payment'>id2</VerifiableEncryption> \\
</\text{VerifiableEncryptions}> \\
</\text{Specification}> \\
</\text{ProofSpecification}>
Evidence Generation

**Discounted car rental claim**

- I own `mc::MembershipCard` issued-by `CarRentalCo`
- I own `cc::CreditCard` issued-by `Amex`
- I own `dl::DriversLicense` issued-by `DeptMotorVehicles`

I reveal `cc.number` to `Amex` under ‘purpose=payment’ where `dl.vehicleCategoryB == true ∧ mc.status = ‘silver’ ∧ mc.name == dl.name`

**Identity Mixer Proof Specification**

```xml
<ProofSpecification>
  <Declaration>
    <AttributeId name='id1' proofMode='revealed'/>
    <AttributeId name='id2' proofMode='unrevealed'/>
    <AttributeId name='id3' proofMode='unrevealed'/>
    ...
  </Declaration>
  <Specification>
    <Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>
      <Attribute name='lastName'>id5</Attribute>
      <Attribute name='number'>id2</Attribute>
    </Credential>
    ...
    <VerifiableEncryptions>
      <VerifiableEncryption name='jd2e0a' pk='Amex'
        label='payment'>id2</VerifiableEncryption>
    </VerifiableEncryptions>
  </Specification>
</ProofSpecification>
```
Evidence Generation

Discounted car rental claim

\[
\begin{align*}
i & \text{own } mc::\text{MemberShipCard} \text{ issued-by CarRentalCo} \\
i & \text{own } cc::\text{CreditCard} \text{ issued-by Amex} \\
i & \text{own } dl::\text{DriversLicense} \text{ issued-by DeptMotorVehic} \\
i & \text{reveal } cc.\text{number} \text{ to Amex under ‘purpose=payment’} \\
& \text{where } dl.\text{vehicleCategoryB} = \text{true} \land \\
& \quad mc.\text{status} = ‘\text{silver’} \land \\
& \quad mc.\text{name} = dl.\text{name}
\end{align*}
\]

Identity Mixer Proof Specification

\[
<\text{ProofSpecification}>
<\text{Declaration}>
<\text{AttributeId name='id1' proofMode='revealed'/>}
<\text{AttributeId name='id2' proofMode='unrevealed'/>}
<\text{AttributeId name='id3' proofMode='unrevealed'/>}
\text{...}
</\text{Declaration}>
<\text{Specification}>
<\text{Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>}
<\text{Attribute name='lastName'>id5</Attribute>
<\text{Attribute name='number'>id2</Attribute>
</\text{Credential}>
\text{...}
<\text{VerifiableEncryptions}>
<\text{VerifiableEncryption name='jd2e0a' pk='Amex' label='payment'>id2</VerifiableEncryption>
</\text{VerifiableEncryptions}>
</\text{Specification}>
</\text{ProofSpecification}>
\]
Evidence Generation

Discounted car rental claim

\[i \text{ own } mc::\text{MembershipCard} \text{ issued-by CarRentalCo}\]
\[i \text{ own } cc::\text{CreditCard} \text{ issued-by Amex}\]
\[i \text{ own } dl::\text{DriversLicense} \text{ issued-by DeptMotorVehicle}\]
\[i \text{ reveal } cc\text{.number} \text{ to Amex under ‘purpose=payment’}\]
where \[dl\text{.vehicleCategoryB} == \text{true} \land \]
\[mc\text{.status} == \text{‘silver’} \land \]
\[mc\text{.name} == dl\text{.name}\]

Identity Mixer Proof Specification

\[<\text{ProofSpecification}>\]
\[<\text{Declaration}>\]
\[<\text{AttributeId} \text{name='id1' proofMode='revealed'/>}\]
\[<\text{AttributeId} \text{name='id2' proofMode='unrevealed'/>}\]
\[<\text{AttributeId} \text{name='id3' proofMode='unrevealed'/>}\]
\[...\]
\[</\text{Declaration}>\]
\[<\text{Specification}>\]
\[<\text{Credential} \text{ipk='Amex' credStruct='CreditCard' name='kdsf23'>}\]
\[<\text{Attribute} \text{name='lastName'>id5</Attribute}>\]
\[<\text{Attribute} \text{name='number'>id2</Attribute}>\]
\[</\text{Credential}>\]
\[...\]
\[<\text{VerifiableEncryptions}>\]
\[<\text{VerifiableEncryption} \text{name='jd2e0a' pk='Amex'>}\]
\[<\text{Attribute} \text{name='payment'>id2</Attribute}>\]
\[</\text{VerifiableEncryption}>\]
\[</\text{VerifiableEncryptions}>\]
\[</\text{Specification}>\]
\[</\text{ProofSpecification}>\]
Evidence Generation

Discounted car rental claim

\[ i \text{ own } mc \text{::MembershipCard issued-by CarRentalCo} \]
\[ i \text{ own } cc \text{::CreditCard issued-by Amex} \]
\[ i \text{ own } dl \text{::DriversLicense issued-by DeptMotorVehicles} \]
\[ i \text{ reveal } cc \text{.number to Amex under 'purpose=payment'} \]
where \( dl \text{.vehicleCategoryB} \equiv \text{true} \land \)
\[ mc \text{.status} = \text{'silver'} \land \]
\[ mc \text{.name} \equiv dl \text{.name} \]

Identity Mixer Proof Specification

\[
<\text{ProofSpecification}>
<\text{Declaration}>
  <\text{AttributeId name='id1' proofMode='revealed'/}>
  <\text{AttributeId name='id2 ' proofMode='unrevealed'/}>
  <\text{AttributeId name='id3 ' proofMode='unrevealed'/}>
  ...
</\text{Declaration}>
<\text{Specification}>
  <\text{Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>}
    <\text{Attribute name='lastName'}>id5</\text{Attribute}>
    <\text{Attribute name='number'}>id2</\text{Attribute}>
  </\text{Credential}>
  ...
  <\text{VerifiableEncryptions}>
    <\text{VerifiableEncryption name='jd2e0a' pk='Amex' label='payment'}>id2</\text{VerifiableEncryption}>
  </\text{VerifiableEncryptions}>
</\text{Specification}>
</\text{ProofSpecification}>


Evidence Generation

Discounted car rental claim

\[
i \text{own mc::MembershipCard issued-by CarRentalCo}
\]
\[
i \text{own cc::CreditCard issued-by Amex}
\]
\[
i \text{own dl::DriversLicense issued-by DeptMotorVehic}
\]
\[
i \text{reveal cc.number to Amex under 'purpose=payment'}
\]

where \( dl.\text{vehicleCategoryB} = \text{true} \land \)
\[
mc.\text{status} = \text{‘silver’} \land
\]
\[
mc.\text{name} = dl.\text{name}
\]

Identity Mixer Proof Specification

\[
<\text{ProofSpecification}>
\]
\[
<\text{Declaration}>
\]
\[
<\text{AttributeId name='id1' proofMode='revealed'}/>
\]
\[
<\text{AttributeId name='id2 ' proofMode='unrevealed'}/>
\]
\[
<\text{AttributeId name='id3 ' proofMode='unrevealed'}/>
\]
\[
\ldots
\]
\[
</\text{Declaration}>
\]
\[
<\text{Specification}>
\]
\[
<\text{Credential ipk='Amex' credStruct='CreditCard' name='kdsf23'>}
\]
\[
<\text{Attribute name='lastName'>id5</Attribute>
\]
\[
<\text{Attribute name='number'>id2</Attribute>
\]
\[
</\text{Credential}>
\]
\[
\ldots
\]
\[
<\text{VerifiableEncryptions}>
\]
\[
<\text{VerifiableEncryption name='jd2e0a' pk='Amex' label='payment'>id2</VerifiableEncryption>
\]
\[
</\text{VerifiableEncryptions}>
\]
\[
</\text{Specification}>
\]
\[
</\text{ProofSpecification}>
\]
Evidence Generation

Discounted car rental claim

\[\begin{align*}
i & \text{own mc::MembershipCard issued-by CarRentalCo} \\
i & \text{own cc::CreditCard issued-by Amex} \\
i & \text{own dl::DriversLicense issued-by DeptMotorVehic} \\
i & \text{reveal cc.number to Amex under ‘purpose=payment’} \\
\text{where dl.vehicleCategoryB == true \land} \\
\text{mc.status == ‘silver’ \land} \\
\text{mc.name == dl.name} \\
\end{align*}\]
## Claim Transformation

<table>
<thead>
<tr>
<th>Type</th>
<th>Examples</th>
<th>Idemix</th>
<th>U-Prove</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proof of Ownership</td>
<td><em>i own c::CreditCard issued-by Visa</em></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Message Signatures</td>
<td><em>i sign ‘terms and conditions’</em></td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Disclosure To Third Parties</td>
<td><em>i reveal li.pNo to ECRAGT under ‘damage’</em></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Attribute Predicate</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logics</td>
<td>((a.b == c.d) \land (e.f &lt; 1984/01/01))</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>((a.b == c.d) \lor (e.f &lt; 1984/01/01))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Equality (incl. Selective Disclosure)</td>
<td>(a.b == c.d)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(a.b == ‘silver’)</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>((a.b + 2 \cdot c.d) == 7)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Not Equal To</td>
<td>(a.b \neq c.d)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(a.b \neq 7; a.b \neq ‘male’; a.b \neq 1984/01/01)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>((a.b + 2 \cdot c.d) \neq 7)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>InEquality</td>
<td>(a.b &lt; c.d; a.b \geq c.d)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(a.b &gt; 8, a.b &gt; 1984/01/01)</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td></td>
<td>((a.b - 2 \cdot c.d) &gt; 25)</td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>
System Model

Claim selection (2b)

Evidence generation (2c)

Policy discovery & pre-evaluation (1a)

Evidence verification (3b)

Claim generation (2a)

Claim verification (3a)

Credentials

User

Authenticator

(1) AuthN request

(2) Applicable policy

(3) Claim, evidence

(4) Success/Failure

Policies

Evidence

Claim

credentialed

credential

claim

claim

Evidence

credential

credential
System Model

User

- Claim selection
  - (2b)
  - Claims
  - (t_a, t_b)

- Evidence generation
  - (2c)
  - Evidence
  - (t_a, t_b)

- Policy discovery & pre-evaluation
  - (1a)
  - Policies
  - (A_1, A_2, A_3)

Authenticator

- Claim verification
  - (3a)
  - Claim
  - Evidence
  - (t_a, t_c)

- Evidence verification
  - (3b)
  - Evidence
  - (t_a, t_c)

- Success/Failure
  - bool

Credentials

- Credentials
  - (C_1, C_2, C_k)
System Model

1. **AuthN request** (1a)
2. **Applicable policy** (2a)
3. **Claim, evidence** (3a)
4. **Success/Failure** (4a)

- **User**
  - **Credentials**
  - **Claim generation**
    - $t_a$, $t_b$ (2a)
  - **Claim selection** (2b)
  - **Evidence generation**
    - $t_a$, $t_b$ (2c)

- **Authenticator**
  - **Policy discovery & pre-evaluation** (1a)
  - **Applicable policy**
  - **Claim verification** (3a)
  - **Evidence verification**
    - $t_a$, $t_b$, $t_c$ (3b)
  - **Success/Failure** (4a)
System Model
Open Source Implementation

User

(1) AuthN request

(2a) Claim generation

(2b) Claim selection

(2) Applicable policy

(3) Claim, evidence

(3a) Claim verification

(3b) Evidence verification

(4) Success/Failure

Authenticator

(1a) Policy discovery & pre-evaluation

Policies

Evidence generation

Evidence

Inbox

Credentials

IDMX
Thank you!

- Credential-based Authentication Framework
  www.primelife.eu/results/opensource/140-abcauth

- Identity Mixer
  idemix.wordpress.com
  prime.inf.tu-dresden.de/idemix/