Attribute Based Signatures with User Controlled Linkability without Random Oracles

Ali El Kaafarani[1] Essam Ghadafi[2]

[1]University of Oxford, UK

[2]University of the West of England, UK

IMACC 2017, Oxford, UK

Outline

UNIVERSITY OF OXFORD

Outline

UNIVERSITY OF OXFORD

Real world Examples

Who gets free prescriptions?

In England, certain groups of people don't have to pay for NHS prescriptions including those who:

- Are under 16
- Are 16, 17 or 18 and in full-time education
- Are 60 years old or over
- Are pregnant or have had a baby in the previous 12 months and have a valid maternity exemption certificate (MatEx)
- Have a specified medical condition and have a valid medical exemption certificate (MedEx)
- Have a valid prescription pre-payment certificate

Real world Examples

PART C: AUTHORITY TO BIND THE UNIVERSITY

13. The Seal of the University shall not be affixed to any document except by the Vice-Chancellor, the Registrar, or an officer or employee of the University deputed by the Registrar for this purpose (either generally or in relation to particular transactions).

Real world Examples

27. USE OF THE SEAL OF THE UNIVERSITY

In accordance with the provisions of Section 17.27 of the Statutes, power to affix the Seal of the University to a document may be exercised and witnessed either by two Members of the Council of the University or by one Member of the Council and the University Secretary (or, in the absence of the University Secretary, the Vice-Chancellor or Director of Finance).

Outline

UNIVERSITY OF OXFORD

Anonymous Signatures

- We don't want to reveal personal information that are not necessary to sign a certain document, aka minimum disclosure.
- For instance, think of proving eligibility to buy alcohol.
- These information might include the identity of the signer or any other attribute that he has.
- Anonymous signatures include:
  - Group Signatures
  - Ring Signatures
  - **Attribute-Based Signatures**
  - Anonymous Credentials

Outline

Attribute Based Signatures

- A policy is a relation between certain attributes, e.g.

$$\Psi = [(\text{Project manager}) \wedge (\text{Exp} > 7\text{years})] \vee \text{CEO}$$

- There are different types of policies: threshold, monotone, non-monotone, and circuits.
- Signers get their attributes from relevant Attribute Authorities.
- Any signer who can satisfy certain policy (predicate) Ψ is eligible to sign messages w.r.t. it.

Attribute Based Signatures

- A policy is a relation between certain attributes, e.g.

$$\Psi = [(\text{Project manager}) \wedge (\text{Exp} > 7\text{years})] \vee \text{CEO}$$

- There are different types of policies: threshold, monotone, non-monotone, and circuits.
- Signers get their attributes from relevant Attribute Authorities.
- Any signer who can satisfy certain policy (predicate) Ψ is eligible to sign messages w.r.t. it.

Attribute Based Signatures

- A policy is a relation between certain attributes, e.g.

$$\Psi = [(\text{Project manager}) \wedge (\text{Exp} > 7\text{years})] \vee \text{CEO}$$

- There are different types of policies: threshold, monotone, non-monotone, and circuits.
- Signers get their attributes from relevant Attribute Authorities.
- Any signer who can satisfy certain policy (predicate) Ψ is eligible to sign messages w.r.t. it.

Security Requirements

- **Anonymity:** Both the identity of the signer as well as the set of attributes that he uses to satisfy the predicate Ψ stay hidden from the verifier
- **Unforgeability:**
  - Only a signer with enough attributes to satisfy the predicate can sign w.r.t it.
  - Colluding users, cannot sign a message w.r.t. a certain predicate Ψ if none of them can satisfy on his own.

Security Requirements

- **Anonymity:** Both the identity of the signer as well as the set of attributes that he uses to satisfy the predicate Ψ stay hidden from the verifier
- **Unforgeability:**
  - Only a signer with enough attributes to satisfy the predicate can sign w.r.t it.
  - Colluding users, cannot sign a message w.r.t. a certain predicate Ψ if none of them can satisfy on his own.

Security Requirements

- **Anonymity:** Both the identity of the signer as well as the set of attributes that he uses to satisfy the predicate Ψ stay hidden from the verifier
- **Unforgeability:**
  - Only a signer with enough attributes to satisfy the predicate can sign w.r.t it.
  - Colluding users, cannot sign a message w.r.t. a certain predicate Ψ if none of them can satisfy on his own.

Security Requirements

- **Anonymity:** Both the identity of the signer as well as the set of attributes that he uses to satisfy the predicate Ψ stay hidden from the verifier
- **Unforgeability:**
  - Only a signer with enough attributes to satisfy the predicate can sign w.r.t it.
  - Colluding users, cannot sign a message w.r.t. a certain predicate Ψ if none of them can satisfy on his own.

Security Requirements

- **Anonymity:** Both the identity of the signer as well as the set of attributes that he uses to satisfy the predicate Ψ stay hidden from the verifier
- **Unforgeability:**
  - Only a signer with enough attributes to satisfy the predicate can sign w.r.t it.
  - Colluding users, cannot sign a message w.r.t. a certain predicate Ψ if none of them can satisfy on his own.

Security Requirements

- **Decentralization:** No reliance on a central authority, multiple attribute authorities are involved (e.g. a doctor (Medical Council) registered with a hospital (hospital)).
- **Traceability:** A tracing authority can reveal the identity of the signer of any signature in case of abuse/misuse.
  - It is more for *troubleshooting*.
  - The correctness of the opening can be publicly verified to avoid framing scenarios! (also called judge of public opinion)

Security Requirements

- **Decentralization:** No reliance on a central authority, multiple attribute authorities are involved (e.g. a doctor (Medical Council) registered with a hospital (hospital)).
- **Traceability:** A tracing authority can reveal the identity of the signer of any signature in case of abuse/misuse.
  - It is more for *troubleshooting*.
  - The correctness of the opening can be publicly verified to avoid framing scenarios! (also called judge of public opinion)

Security Requirements

- **Decentralization:** No reliance on a central authority, multiple attribute authorities are involved (e.g. a doctor (Medical Council) registered with a hospital (hospital)).
- **Traceability:** A tracing authority can reveal the identity of the signer of any signature in case of abuse/misuse.
  - It is more for *troubleshooting*.
  - The correctness of the opening can be publicly verified to avoid framing scenarios! (also called judge of public opinion)

Security Requirements

- **Decentralization:** No reliance on a central authority, multiple attribute authorities are involved (e.g. a doctor (Medical Council) registered with a hospital (hospital)).
- **Traceability:** A tracing authority can reveal the identity of the signer of any signature in case of abuse/misuse.
  - It is more for *troubleshooting*.
  - The correctness of the opening can be publicly verified to avoid framing scenarios! (also called judge of public opinion)

Security Requirements

- **Decentralization:** No reliance on a central authority, multiple attribute authorities are involved (e.g. a doctor (Medical Council) registered with a hospital (hospital)).
- **Traceability:** A tracing authority can reveal the identity of the signer of any signature in case of abuse/misuse.
  - It is more for *troubleshooting*.
  - The correctness of the opening can be publicly verified to avoid framing scenarios! (also called judge of public opinion)

Outline

UNIVERSITY OF OXFORD

Related Work

- Attribute Based Signatures [MPR08]
- threshold ABS [SSN09]
- Decentralization[OT11,OT12]
- Decentralization and Traceability [EGK14]
- ABS for circuits [SAH16]

Related Work

- Attribute Based Signatures [MPR08]
- threshold ABS [SSN09]
- Decentralization[OT11,OT12]
- Decentralization and Traceability [EGK14]
- ABS for circuits [SAH16]

Related Work

- Attribute Based Signatures [MPR08]
- threshold ABS [SSN09]
- Decentralization[OT11,OT12]
- Decentralization and Traceability [EGK14]
- ABS for circuits [SAH16]

Related Work

- Attribute Based Signatures [MPR08]
- threshold ABS [SSN09]
- Decentralization[OT11,OT12]
- Decentralization and Traceability [EGK14]
- ABS for circuits [SAH16]

Related Work

- Attribute Based Signatures [MPR08]
- threshold ABS [SSN09]
- Decentralization[OT11,OT12]
- Decentralization and Traceability [EGK14]
- ABS for circuits [SAH16]

Outline

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

User Controlled Linkability (UCL): idea

- Anonymity has its drawbacks: There are no sessions!
- With UCL: Verifiers can have a negotiation with the *same unknown* signer whose identity is still hidden. (analogous to cookies idea)
- The analogy with cookies is very strong: Signers are allowed to have different sessions.
- You can use it to link a current transaction to some of your earlier anonymous ones, e.g. to benefit from discounts.
- Or resume interrupted or lost authentication sessions between communicating parties.
- Traceability can't be helpful here- it's more for *troubleshooting*. UCL is intended to be built into *normal use!*

Outline

Features of ABS-UCL

- Decentralized: Multiple attribute authorities are involved with no reliance on a central authority.
- UCL: A signer can make any set of his signatures linkable, and bound to any recipient's (verifier) base name.

Features of ABS-UCL

- Decentralized: Multiple attribute authorities are involved with no reliance on a central authority.
- UCL: A signer can make any set of his signatures linkable, and bound to any recipient's (verifier) base name.

Security requirements–Correctness

- Correctness: if all parties are honest, valid signatures verify correctly, and valid *linkable* signatures verify and link correctly.

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathrm{id}_0, \mathrm{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathrm{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathrm{id}_0, \mathrm{id}_1$ should be honest, and $(\mathrm{id}_0, \mathrm{recip}), (\mathrm{id}_1, \mathrm{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathrm{id}_0, \mathrm{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathrm{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathrm{id}_0, \mathrm{id}_1$ should be honest, and $(\mathrm{id}_0, \mathrm{recip}), (\mathrm{id}_1, \mathrm{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathrm{id}_0, \mathrm{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathrm{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathrm{id}_0, \mathrm{id}_1$ should be honest, and $(\mathrm{id}_0, \mathrm{recip}), (\mathrm{id}_1, \mathrm{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathsf{id}_0, \mathsf{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathsf{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathsf{id}_0, \mathsf{id}_1$ should be honest, and $(\mathsf{id}_0, \mathsf{recip}), (\mathsf{id}_1, \mathsf{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathsf{id}_0, \mathsf{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathsf{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathsf{id}_0, \mathsf{id}_1$ should be honest, and $(\mathsf{id}_0, \mathsf{recip}), (\mathsf{id}_1, \mathsf{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathsf{id}_0, \mathsf{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathsf{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathsf{id}_0, \mathsf{id}_1$ should be honest, and $(\mathsf{id}_0, \mathsf{recip}), (\mathsf{id}_1, \mathsf{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Anonymity

- Anonymity: *The signer identity along with the set of signing attributes remain hidden from the verifier*
  - Adversary's power: Full control over all attribute authorities, i.e. signing, key generation, etc.
  - Challenge: The adversary chooses $(\mathsf{id}_0, \mathsf{id}_1, \mathcal{A}_0, \mathcal{A}_1, \Psi, m, \mathsf{recip})$ for which $\Psi(A_0) = \Psi(A_1) = 1$.
  - Both $\mathsf{id}_0, \mathsf{id}_1$ should be honest, and $(\mathsf{id}_0, \mathsf{recip}), (\mathsf{id}_1, \mathsf{recip})$ are never queried to the signing oracle. (So adversary cannot trivially win by exploiting the linkability feature).
  - The challenger signs m w.r.t the bit $b \leftarrow \{0, 1\}$.
  - Winning condition: if adversary's guessing bit $b^* = b$

Security requirements–Linkability

- Linkability: *Only valid signatures directed at the same recipient and which were produced by the same user link.*
- Adversary's power: full control over attribute authorities
- Wining conditions: Adversary outputs two valid signatures σ_1 on $(\mathsf{sk}_1, m_1, \mathsf{recip}_1, \mathcal{A}_1, \Psi_1)$ and σ_2 on $(\mathsf{sk}_2, m_2, \mathsf{recip}_2, \mathcal{A}_2, \Psi_2)$. He wins in the following cases:
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 1$, although σ_1, σ_2 are not supposed to link.
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 0$, although σ_1, σ_2 are supposed to link.

Security requirements–Linkability

- Linkability: *Only valid signatures directed at the same recipient and which were produced by the same user link.*
- Adversary's power: full control over attribute authorities
- Wining conditions: Adversary outputs two valid signatures σ_1 on $(\mathsf{sk}_1, m_1, \mathsf{recip}_1, \mathcal{A}_1, \Psi_1)$ and σ_2 on $(\mathsf{sk}_2, m_2, \mathsf{recip}_2, \mathcal{A}_2, \Psi_2)$. He wins in the following cases:
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 1$, although σ_1, σ_2 are not supposed to link.
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 0$, although σ_1, σ_2 are supposed to link.

Security requirements–Linkability

- Linkability: *Only valid signatures directed at the same recipient and which were produced by the same user link.*
- Adversary's power: full control over attribute authorities
- Wining conditions: Adversary outputs two valid signatures σ_1 on $(\mathsf{sk}_1, m_1, \mathsf{recip}_1, \mathcal{A}_1, \Psi_1)$ and σ_2 on $(\mathsf{sk}_2, m_2, \mathsf{recip}_2, \mathcal{A}_2, \Psi_2)$. He wins in the following cases:
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 1$, although σ_1, σ_2 are not supposed to link.
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 0$, although σ_1, σ_2 are supposed to link.

Security requirements–Linkability

- Linkability: *Only valid signatures directed at the same recipient and which were produced by the same user link.*
- Adversary's power: full control over attribute authorities
- Wining conditions: Adversary outputs two valid signatures σ_1 on $(\mathsf{sk}_1, m_1, \mathsf{recip}_1, \mathcal{A}_1, \Psi_1)$ and σ_2 on $(\mathsf{sk}_2, m_2, \mathsf{recip}_2, \mathcal{A}_2, \Psi_2)$. He wins in the following cases:
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 1$, although σ_1, σ_2 are not supposed to link.
  - if $\mathsf{link}(\sigma_1, \sigma_2) = 0$, although σ_1, σ_2 are supposed to link.

Security Requirements–Unforgeability

- Unforgeability: *users cannot output signatures on (message, recipient tag) pairs w.r.t. a signing policy not satisfied by their set of attributes, even if they pool their attributes together, which ensures collusion-resistance. They also cannot produce signatures which link to other signatures by an honest user.*

Generic construction

- Building blocks:
  - A NIZK system $\mathcal{NIZK}$
  - An existentially unforgeable Digital Signature scheme $\mathcal{DS}$
  - A Linkable Indistinguishable Tag scheme $\mathcal{LIT}$ (A special deterministic digital signature with two additional security requirement, i.e. Linkability and f-Indistinguishability)
  - Span Program $\mathcal{SP}$

Generic construction

- Building blocks:
  - A NIZK system $\mathcal{NIZK}$
  - An existentially unforgeable Digital Signature scheme $\mathcal{DS}$
  - A Linkable Indistinguishable Tag scheme $\mathcal{LIT}$ (A special deterministic digital signature with two additional security requirement, i.e. Linkability and f-Indistinguishability)
  - Span Program $\mathcal{SP}$

Generic construction

- Building blocks:
  - A NIZK system $\mathcal{NIZK}$
  - An existentially unforgeable Digital Signature scheme $\mathcal{DS}$
  - A Linkable Indistinguishable Tag scheme $\mathcal{LIT}$ (A special deterministic digital signature with two additional security requirement, i.e. Linkability and f-Indistinguishability)
  - Span Program $\mathcal{SP}$

Generic construction

- Building blocks:
  - A NIZK system $\mathcal{NIZK}$
  - An existentially unforgeable Digital Signature scheme $\mathcal{DS}$
  - A Linkable Indistinguishable Tag scheme $\mathcal{LIT}$ (A special deterministic digital signature with two additional security requirement, i.e. Linkability and f-Indistinguishability)
  - Span Program $\mathcal{SP}$

Generic construction

- Building blocks:
  - A NIZK system $\mathcal{NIZK}$
  - An existentially unforgeable Digital Signature scheme $\mathcal{DS}$
  - A Linkable Indistinguishable Tag scheme $\mathcal{LIT}$ (A special deterministic digital signature with two additional security requirement, i.e. Linkability and f-Indistinguishability)
  - Span Program $\mathcal{SP}$

Generic construction

- Setup(1^λ): On input a security parameter, it returns public parameters $\mathcal{P}$.
- AASetup(aid, $\mathcal{P}$): Is run locally by attribute authority AA_{aid} to generate its public/secret key pair (vk_{AA}, sk_{AA}). The authority publishes vk_{AA} and keeps sk_{AA} secret.
- UKeyGen(id, $\mathcal{P}$): Is run by user id to generate his personal secret key sk_{id}.
- AttKeyGen(id, $f(sk_{id})$, a, sk_{AA}): Is run by attribute authority AA that is responsible for the attribute a, where f is an injective one-way function, it generates the user id's secret key $sk_{id,a}$, bound to his identity id and $f(sk_{id})$.

$$sk_{id,a} \leftarrow \mathcal{DS}.\text{Sign}((id, f(sk_{id}), a), sk_{AA})$$

Generic construction

- Setup(1^λ): On input a security parameter, it returns public parameters $\mathcal{P}$.
- AASetup(aid, $\mathcal{P}$): Is run locally by attribute authority $\mathsf{AA}_{\mathsf{aid}}$ to generate its public/secret key pair $(\mathsf{vk}_{\mathsf{AA}}, \mathsf{sk}_{\mathsf{AA}})$. The authority publishes $\mathsf{vk}_{\mathsf{AA}}$ and keeps $\mathsf{sk}_{\mathsf{AA}}$ secret.
- UKeyGen(id, $\mathcal{P}$): Is run by user id to generate his personal secret key $\mathsf{sk}_{\mathsf{id}}$.
- AttKeyGen(id, $f(\mathsf{sk}_{\mathsf{id}}), a, \mathsf{sk}_{\mathsf{AA}}$): Is run by attribute authority AA that is responsible for the attribute a, where f is an injective one-way function, it generates the user id's secret key $\mathsf{sk}_{\mathsf{id},a}$, bound to his identity id and $f(\mathsf{sk}_{\mathsf{id}})$.

$$\mathsf{sk}_{\mathsf{id},a} \leftarrow \mathcal{DS}.\mathsf{Sign}((\mathsf{id}, f(\mathsf{sk}_{\mathsf{id}}), a), \mathsf{sk}_{\mathsf{AA}})$$

Generic construction

- Setup(1^λ): On input a security parameter, it returns public parameters $\mathcal{P}$.
- AASetup(aid, $\mathcal{P}$): Is run locally by attribute authority $\mathsf{AA}_{\mathsf{aid}}$ to generate its public/secret key pair $(\mathsf{vk}_{\mathsf{AA}}, \mathsf{sk}_{\mathsf{AA}})$. The authority publishes $\mathsf{vk}_{\mathsf{AA}}$ and keeps $\mathsf{sk}_{\mathsf{AA}}$ secret.
- UKeyGen(id, $\mathcal{P}$): Is run by user id to generate his personal secret key $\mathsf{sk}_{\mathsf{id}}$.
- AttKeyGen(id, $f(\mathsf{sk}_{\mathsf{id}})$, a, $\mathsf{sk}_{\mathsf{AA}}$): Is run by attribute authority AA that is responsible for the attribute a, where f is an injective one-way function, it generates the user id's secret key $\mathsf{sk}_{\mathsf{id},a}$, bound to his identity id and $f(\mathsf{sk}_{\mathsf{id}})$.

$$\mathsf{sk}_{\mathsf{id},a} \leftarrow \mathcal{DS}.\mathsf{Sign}((\mathsf{id}, f(\mathsf{sk}_{\mathsf{id}}), a), \mathsf{sk}_{\mathsf{AA}})$$

Generic construction

- Setup(1^λ): On input a security parameter, it returns public parameters $\mathcal{P}$.
- AASetup(aid, $\mathcal{P}$): Is run locally by attribute authority $\mathsf{AA}_{\mathsf{aid}}$ to generate its public/secret key pair $(\mathsf{vk}_{\mathsf{AA}}, \mathsf{sk}_{\mathsf{AA}})$. The authority publishes $\mathsf{vk}_{\mathsf{AA}}$ and keeps $\mathsf{sk}_{\mathsf{AA}}$ secret.
- UKeyGen(id, $\mathcal{P}$): Is run by user id to generate his personal secret key $\mathsf{sk}_{\mathsf{id}}$.
- AttKeyGen(id, $f(\mathsf{sk}_{\mathsf{id}}), a, \mathsf{sk}_{\mathsf{AA}}$): Is run by attribute authority AA that is responsible for the attribute a, where f is an injective one-way function, it generates the user id's secret key $\mathsf{sk}_{\mathsf{id},a}$, bound to his identity id and $f(\mathsf{sk}_{\mathsf{id}})$.

$$\mathsf{sk}_{\mathsf{id},a} \leftarrow \mathcal{DS}.\mathsf{Sign}((\mathsf{id}, f(\mathsf{sk}_{\mathsf{id}}), a), \mathsf{sk}_{\mathsf{AA}})$$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

UNIVERSITY OF OXFORD

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

UNIVERSITY OF OXFORD

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on (id, a, $f(\underline{\mathsf{sk}_{\mathsf{id}}})$) (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}$, recip):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}$.Tag($\underline{\mathsf{sk}_{\mathsf{id}}}$, recip) (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on (id, a, $f(\underline{\mathsf{sk}_{\mathsf{id}}})$) (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m,$ recip).
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on $(\mathsf{id}, a, f(\underline{\mathsf{sk}_{\mathsf{id}}}))$ (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on (id, a, $f(\underline{\mathsf{sk}_{\mathsf{id}}})$) (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Generic construction

Sign($m, \Psi, \mathsf{sk}_{\mathsf{id}}, \mathsf{sk}_{\mathsf{id},\mathcal{A}}, \mathsf{recip}$):

- Linkable signature: (recip $\neq \perp$)
  - Produce a NIZK proof π that :
    1. $\sigma_{link} \leftarrow \mathcal{LIT}.\mathsf{Tag}(\underline{\mathsf{sk}_{\mathsf{id}}}, \mathsf{recip})$ (and)
    2. $\forall a \in \mathcal{A}$, he owns a $\mathcal{DS}$ signature on (id, a, $f(\underline{\mathsf{sk}_{\mathsf{id}}})$) (or)
    3. He owns a signature on a special attribute, i.e.$\mathcal{H}(\Psi, m, \mathsf{recip})$.
    4. The span program $\mathcal{SP}$ allows the signer to hide the set of signing attributes.
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$
- non-Linkable signature:(recip $= \perp$)
  - $\sigma_{link} = \perp$
  - $\sigma_{ABS-UCL} = \{\sigma_{link}, \pi\}$

Security of the construction

- **Anonymity** because the NIZK system is zero-knowledge, $\mathcal{H}$ is collision resistant, and $\mathcal{LIT}$ is indistinguishable
- **Unforgeability** because the NIZK system is sound, $\mathcal{H}$ is collusion resistant, $\mathcal{LIT}$ is linkable, and $\mathcal{DS}$ is unforgeable.
- **Linkability** because the hash function $\mathcal{H}$ is collusion-resistant and $\mathcal{LIT}$ is linkable.

Security of the construction

- **Anonymity** because the NIZK system is zero-knowledge, $\mathcal{H}$ is collision resistant, and $\mathcal{LIT}$ is indistinguishable
- **Unforgeability** because the NIZK system is sound, $\mathcal{H}$ is collusion resistant, $\mathcal{LIT}$ is linkable, and $\mathcal{DS}$ is unforgeable.
- **Linkability** because the hash function $\mathcal{H}$ is collusion-resistant and $\mathcal{LIT}$ is linkable.

UNIVERSITY OF OXFORD

Security of the construction

- **Anonymity** because the NIZK system is zero-knowledge, $\mathcal{H}$ is collision resistant, and $\mathcal{LIT}$ is indistinguishable
- **Unforgeability** because the NIZK system is sound, $\mathcal{H}$ is collusion resistant, $\mathcal{LIT}$ is linkable, and $\mathcal{DS}$ is unforgeable.
- **Linkability** because the hash function $\mathcal{H}$ is collusion-resistant and $\mathcal{LIT}$ is linkable.

Outline

UNIVERSITY OF OXFORD

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

UNIVERSITY OF OXFORD

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

- Three Instantiations in the Standard Model that use the following building blocks;
  - Bilinear groups (type-3)
  - NIZK system: Groth-Sahai system.
  - $\mathcal{DS}$: Boneh-Boyen signature scheme [BB04], PSPS [G17], and a new PSPS scheme presented in the paper.
  - $\mathcal{LIT}$: weak-Boneh-Boyen (the one way function f will be the Dlog in this case)
- For ABS-UCL, the signature size is: $(15|\Psi| + 15) \cdot |\mathbb{G}| + (14|\Psi| + 22) \cdot |\mathbb{H}| + (\beta + 3) \cdot |p|$, where β is the number of columns in the span program matrix $\mathcal{SP}$.
- For tABS-UCL, the signature size is: $27|\mathbb{G}| + 28|\mathbb{H}|$.

Scheme	Anonymity	Traceability	Decentralized	UCL
[KGK13,G14]	✓	✓	✓	✘
[OT12]	✓	✘	✓	✘
[MPR08]	✓	✘	✘	✘
Ours	✓	✘	✓	✓

Table: Existing ABS schemes and their features

Summary

- Security model for Attribute based signatures with user-controlled linkability (ABS-UCL)
- A Generic construction of ABS-UCL
- Two Instantiations of ABS-UCL in the Standard Model
- One Instantiation of a tABS-UCL in the Standard Model

Summary

- Security model for Attribute based signatures with user-controlled linkability (ABS-UCL)
- A Generic construction of ABS-UCL
- Two Instantiations of ABS-UCL in the Standard Model
- One Instantiation of a tABS-UCL in the Standard Model

Summary

- Security model for Attribute based signatures with user-controlled linkability (ABS-UCL)
- A Generic construction of ABS-UCL
- Two Instantiations of ABS-UCL in the Standard Model
- One Instantiation of a tABS-UCL in the Standard Model

Summary

- Security model for Attribute based signatures with user-controlled linkability (ABS-UCL)
- A Generic construction of ABS-UCL
- Two Instantiations of ABS-UCL in the Standard Model
- One Instantiation of a tABS-UCL in the Standard Model

Thanks for your attention!