Identity Linking

Arne Bochem

University of Goettingen,
Institute for Computer Science,
Telematics Group

TELEMATICS
GROUP

GEORG-AUGUST-UNIVERSITÄT
GÖTTINGEN

Identity Linking

So we heard before:

- Everybody has an identity with public/private key pair
- An identity can have multiple pseudonyms
- Pseudonyms are signed using blind signatures, unlinking them from identities

Side note: No currency ⇒ no transaction graph analysis.

Identity Linking

At some point you will want to prove:

- I wrote that paper
- I wrote that review
- Somebody well known reviewed my paper

How do we link pseudonyms back to identities?

Public Linking

You want everybody to know and to be able to verify that this pseudonym corresponds to your identity.

Public Linking

You want everybody to know and to be able to verify that this pseudonym corresponds to your identity.

Pretty simple:

- Pseudonym signs message: I am that identity
- Identity signs the signed message message as well
- Anybody can verify using the pseudonym's and identity's public key

Private Linking

You want to prove that a pseudonym and identity are linked, but you only want to prove this to one specific party.

Private Linking

You want to prove that a pseudonym and identity are linked, but you only want to prove this to one specific party.

OTR-like approach without non-repudiation:

- ECC for shared secret: $A_{\text{pub}} * B_{\text{priv}} = B_{\text{pub}} * A_{\text{priv}}$
- Messages as for public linking, but no signature, instead:
- Message Authentication Code (e.g. HMAC-SHA256)
- Requires shared secret to generate and verify
- Can be generated by either party ⇒ Even if published, you can claim the other side generated it

API

Probably should be integrated in the identity API.

- ProveIdentity(identity, pseudonym) → double signed message
- PrivateProveIdentity(identity, pseudonym, provePeer) → double MACed message