Direct Anonymous Attestation: Revocation and Anonymity

Benjamin Benoy

Trusted Systems Research Group
National Security Agency

December 2, 2011
Joint Work With:

- Laura Fairfax  
  National Security Agency
- Angela Hennessy  
  Laboratory for Telecommunications Science
- Jonathan Katz  
  University of Maryland, College Park
- Laurie Law  
  National Security Agency
Direct Anonymous Attestation
Direct Anonymous Attestation (DAA)

Prove membership in group without revealing identity

Member and Verifier communicate directly, not through a third party

Demonstrations are anonymous and user-controlled unlinkable

Specialization of anonymous credentials
Anonymity and Unlinkability

Different aspects of general theme: “Demonstrations should be indistinguishable”
Anonymous: Given a demonstration and two members, can’t figure out which one made it
Unlinkable: Given two demonstrations, can’t tell whether they were made by one member or two
User Controlled Unlinkability a.k.a. Pseudonyms

Pseudonym is a persistent identity
Pseudonyms cannot be connected to each other
Members can recognize their own pseudonyms
Single-Use pseudonyms ↔ Anonymity
Players in the DAA world

**ISSUER** acts as central authority. Distributes credentials to members

**MEMBER** receives credentials and uses them to prove membership in group

**VERIFIER** is the “relying party”. **VERIFIER** confirms that credentials shown by **MEMBER** are valid and then accepts that **MEMBER** is actually in the group

**REVOKER** is in charge of maintaining revocation lists
Trust Model: Trust No One

Anonymity should be protected, even if Verifiers collude with each other...
and with the Issuer...
and with the Revoker(s)
What does a credential look like?

Membership credentials have two parts:

**Private Signing Key:**
- Used to create pseudonyms
- Known only to Member

**Digital Certificate:**
- Signed by Issuer
- May be known to Issuer, but never revealed
Making a Demonstration (Without Revocation)

1. **Member** creates a pseudonym $\sigma$ using Private Signing Key
2. **Member** creates a zero-knowledge proof $\Pi$ that she has a Certificate corresponding to the pseudonym $\sigma$
3. **Member** sends $(\sigma, \Pi)$ to **Verifier**
Verifying a Demonstration (Without Revocation)

**Verifier** checks that $\Pi$ is a valid proof of knowledge

**Verifier** needs public key of **Issuer**, but no direct contact
Revocation and Anonymity
Why do we need revocation?

Group membership is not a fixed property

- Members can leave the group
- Members can be forced from the group
- Credentials can be compromised
Revocation and Anonymity

Revocation is inherently in tension with Anonymity: Revocation requires some connection between demonstrations. Anonymity makes revocation decisions difficult

- Because demonstrations are anonymous, difficult to revoke a particular member
- Because demonstrations are unlinkable, difficult to revoke based on aggregated behavior
Signature Revocation List

Could be called “Pseudonym Revocation List”

List of “bad” pseudonyms

Maintained by Revocation Authority Revoker

When making a demonstration, Member proves she didn’t create any of the revoked pseudonyms

May not scale well
Demonstrations leak information: “I didn’t make any of those pseudonyms”

By manipulating Signature Revocation List may be able to link members to their demonstrations
Because demonstrations are anonymous it is difficult to make informed revocation decisions.

Difficult to implement “three strikes and you’re out”

This makes it easier to manipulate Signature Revocation List
Verifier-Local Blacklists

**Verifier** supplies common seed for pseudonyms → persistent identity with **Verifier**
All demonstrations with a given verifier are linkable
Each verifier maintains a list of “locally revoked” pseudonyms
Blacklists cannot be shared between verifiers
Because members have a history, revocation decisions are easier to make
Verifier-Local Blacklists (Cont.)

Signature Revocation List can enable cross-domain contamination:

\textbf{VERIFIER} submits local pseudonym to Signature Revocation List
\textbf{VERIFIER} can transform pseudonym before submitting
Key-Based Revocation List

List of compromised Private Signing Keys
Not possible to make demonstration using just Private Signing Key, also need corresponding Certificate
Given a signing key, can recognize pseudonyms created with that key
All demonstrations made with keys on Key-Based Revocation List are linkable
Key Compromise and Repudiation

Assume Enrolling Member can choose Private Signing Key

1. Alice loses control of her Private Signing Key $sk_{Alice}$
2. Eve gets Compromised Private Signing Key $sk_{Alice}$ before universal distribution on a revocation list
3. Eve enrolls as a new member, using $sk_{Alice}$
4. Eve makes demonstrations using $sk_{Alice}$
5. Eve’s demonstrations are linked to Alice’s demonstrations
Because the preceding scenario is possible, Alice can repudiate all signatures made after key is on Key-Based Revocation List. Alice: “I didn’t make those demonstrations. Someone must have taken my key from the revocation list and reenrolled using it!”
Conclusions
Conclusions

Interactions between revocation methods can be subtle and unpredictable.
Anonymity is not unconditional in the presence of revocation.
Revocation and Anonymity need to be balanced against each other.
Questions?
Thank you.