Decentralized Anonymous Credentials

Christina Garman, Matthew Green, Ian Miers
Johns Hopkins University
Privacy and Identity on the Internet

- Cannot make statements of identity privately
- But what about identity attributes?
Privacy and Identity on the Internet
Privacy and Identity on the Internet
Privacy and Identity on the Internet

Keith
Age = 62

Signature

Justin Bieber Fan Club
Privacy and Identity on the Internet

Keith
Age = 62

Signature

JUSTIN BIEBER FAN CLUB
Privacy and Identity on the Internet

“Welcome to the club!”
Privacy and Identity on the Internet

“Welcome to the club!”
Privacy and Identity on the Internet

Keith Alexander
earlier today....
Keith liked the Justin Bieber Fan Club.
Anonymous Credentials

- Introduced by Chaum [Chaum85] and extended in [Brands00, CL01, CL02, CL03, BCKL08,…]
- Prove that you have a credential issued by some organization without revealing anything other than that you have the credential
- Standard techniques use a specialized digital signature
Example of Anonymous Credentials
Example of Anonymous Credentials
Example of Anonymous Credentials

Keith
Age = 62
Example of Anonymous Credentials

Age > 13

Justin Bieber Fan Club
Example of Anonymous Credentials

Keith
Age = 62

Age > 13
Problems?

Keith
Age = 62

Age > 13
Solution?

Keith
Age = 62

Age > 13

Justin Bieber Fan Club
Our Contribution: Decentralized Anonymous Credentials

- Related to our electronic cash proposal [MGGR13]
  - Zerocoin (decentralized e-cash)

- **Decentralized** anonymous credentials
  - Decentralized credential issuance
  - Decentralized identity certification
  - Requires:
    - Public append-only ledger
    - Publicly verifiable identity claims
Public Append-Only Ledger

- Central ledger (audited by users)
- Broadcast networks
- Distributed consensus network
  - Bitcoin block chain
Publicly Verifiable Identity Claims

- Identity assertions are frequently publicly verifiable
- So why bother with (decentralized) anonymous credentials?
- Just because an identity assertion is publicly verifiable does not mean we want to link all of the information to every interaction!
Overview
Overview

cred
Overview

cred

[Diagram of networked computers and a notebook]
Overview
Overview

“A credential on the ledger says age > 13.”
Overview

“A credential on the ledger says age > 13.”
Cryptographic Building Blocks

- Commitments
- Zero-knowledge proofs
- Accumulators
Commitments

- Allow you to commit to and later reveal a value
- Binding: value cannot be tampered with
- Hiding: value cannot be read until revealed
- We use Pedersen commitments

\[ C = g^x h^r \mod q \]
Zero-knowledge Proofs

- Zero-knowledge [Goldwasser, Micali 1980s, and beyond]
- Prove a statement without revealing any other information
- Specific variant: non-interactive proof of knowledge
- Here we prove we know:
  1. The opening for a credential
  2. That the credential is in the ledger
An inefficient approach...

- Inefficient proof
  - Identify all valid credentials in the ledger (call them $C_{\downarrow 1}, \ldots, C_{\downarrow N}$)
  - Prove that you know the opening of a credential $C$ and $C = C_{\downarrow 1} \lor C = C_{\downarrow 2} \lor \ldots \lor C = C_{\downarrow N}$
  - This “OR” proof is O(N)
Cryptographic Accumulators

- Allow constant size set membership proofs
- Strong RSA accumulator originally due to Benaloh and de Mare
- Efficient proof for accumulation of primes proposed by Camenisch and Lysyanskaya ‘01

\[ N = p \cdot q, u \in \mathbb{QR}_N(u \neq 1) \]

\[ A = u^{C_1 \cdot C_2 \cdot \ldots \cdot C_n} \mod N \]

\[ \omega_i = u^{C_1 \cdot C_2 \cdot \ldots \cdot C_{i-1} \cdot C_{i+1} \cdot \ldots \cdot C_n} \mod N \]
Basic Decentralized Anonymous Credentials
Basic Decentralized Anonymous Credentials

\[ c = g_0^r g_1^s k \quad \text{age}=62 \]

\[ \pi_c \]
Basic Decentralized Anonymous Credentials
Basic Decentralized Anonymous Credentials

$C_1, C_2, \ldots, C_n$
Basic Decentralized Anonymous Credentials

\[ A = u^{c_1 \cdot c_2 \cdots \cdot c_n} \mod N \]
Basic Decentralized Anonymous Credentials

\[ A = u^{c_1 \cdot c_2 \cdot \cdots \cdot c_n} \mod N \]
Basic Decentralized Anonymous Credentials

\[ A = u^{c_1 \cdot c_2 \cdots c_n} \mod N \]
Applications

- Anonymous resource management in ad hoc networks
- Decentralized Direct Anonymous Attestation
- Auditable credentials
- Mitigating Sybil attacks in ad hoc networks
Protecting Against Sybil Attacks
Protecting Against Sybil Attacks

“I am John Smith”, $
Protecting Against Sybil Attacks

John Smith

veriSign®
Protecting Against Sybil Attacks

John Smith

veriSign®
Protecting Against Sybil Attacks

“I am John Smith”, $\rightarrow$ John Smith

[Diagram showing a computer connecting to a certificate of John Smith and a network of computers connected with a VeriSign logo]
Protecting Against Sybil Attacks
Protecting Against Sybil Attacks

cred

“I have paid 1 BTC”
Protecting Against Sybil Attacks
"I am not a Sybil"
Protecting Against Sybil Attacks

“I am not a Sybil”
Performance

- Basic scheme implemented as stand-alone library
  - Proofs 50 KB
Future Work

- Better, smaller “proofs” of knowledge:
  - **Succinct Non-Interactive ARguments of Knowledge (zkSNARKs)** [PHGR13, BCGTV13]
    - 288 byte proof for arbitrary-sized arithmetic circuits
    - 8 ms verification time

- Additional applications?
Questions?
Potential Alternatives

- Threshold cryptography
  - High setup cost for large number of parties
  - Difficult for parties to come and go
- Ring signatures [RST01]
  - Grow linearly with the number of participating signers
  - Expensive to generate
Non Publicly Verifiable Credentials

- Credential transform service
- Allows user to transform a credential to an anonymous credential without additional trust assumption
- Works for any statement that an authority can certify
Proof of Work for Sybil Attacks

- Proof of resource expenditure instead of payment
- Cannot reuse proof of work with different peers
  - Not anonymous
  - Clonable
- Do not want to have to do a proof of work with each peer in the system
- Instead do one proof of work per k interactions
Resource Management

- Publicly verifiable proofs of resources
- File storage, bandwidth, etc.
- Do not want to link resources provided to resources consumed
  - Files uploaded vs. files downloaded