Anonymous Credentials

Anupam Datta
CMU
Fall 2015
Credentials: Motivation

• ID cards
  – Sometimes used for other uses
    • E.g. prove you’re over 21, or verify your address
  – Don’t necessarily need to reveal all of your information
  – Don’t necessarily want issuer of ID to track all of it’s uses
  – How can we get the functionality/verifiability of an physical id in electronic form without extra privacy loss
Credentials: Motivation

• The goal
  – Users should be able to
    • Obtain credentials
    • Show some properties
  – Without
    • Revealing additional information
    • Allowing tracking
Credentials: Motivation

• Other applications
  – Transit tokens/passes
  – Electronic currency
  – Online polling

• Implementations
  – Idemix (IBM), UProve (Microsoft)
Microsoft Research

U-Prove

U-Prove is an innovative cryptographic technology that allows users to minimally disclose certified information about themselves when interacting with online resource providers. U-Prove provides a superset of the security features of Public Key Infrastructure (PKI), and also provide strong privacy protections by offering superior user control and preventing unwanted user tracking.

Overview

A U-Prove token is a new type of credential similar to a PKI certificate that can encode attributes of any type, but with two important differences:

1) The issuance and presentation of a token is unlinkable due to the special type of public key and signature encoded in the token; the cryptographic "wrapping" of the attributes contain no correlation handles. This prevents unwanted tracking of users when they use their U-Prove tokens, even by colluding insiders.

2) Users can minimally disclose information about what attributes are encoded in a token in response to dynamic verifier policies. As an example, a user may choose to only disclose a subset of the encoded attributes, prove that her undisclosed name does not appear on a blacklist, or prove that she is of age without disclosing her actual birthdate.

These user-centric aspects make the U-Prove technology ideally suited to creating the digital
Identity governance

Project overview

All social and economic interactions among human beings in modern civilization require the exchange of personal data. In everyday situations, we decide intuitively which data to make available, for instance whether to state our name when shaking hands.

In the online world, each individual has to handle numerous accounts and data sets. These so-called partial identities will increasingly play a key role in future electronic services as well as in public security (such as at border checks). A partial identity may very well convey sensitive personal data, such as patient health data, employee data, or credit card data.

We envision user-controlled identity management systems within which the players concerned act together, mediated by technology, to enforce the rules established by law and by the contracting partners. In these systems, the user has control over his or her personal information and negotiates its disclosure in return for access to a service. The result of such a negotiation is an agreement between the user and the service provider, whereby the provider collects personal data for a stated, legitimate purpose (which may include the transfer of these data to other entities), and — in the case of certain providers — issues certified data to individuals.

All agents act within the strict bounds of the law, under anonymity, pseudonymity, or on the basis of terms explicitly agreed upon by the parties involved. In all cases, technology supports accountability and recourse.

Identity mixer

We are working on a suite of cryptographic protocols that allow privacy in identity management to be enhanced. In particular, we strive for:

- Theoretical results, i.e., cryptographic algorithms and protocols to realize an efficient anonymous credential system. This work is partially funded by PrimoLife.
- An open-source implementation of cryptographic protocols, some basic applications logic on the server and the client side, a wallet for users to manage tokens, share, and obtain their...
Today

Focus on one kind of anonymous credentials: electronic cash
Security without Identification
David Chaum 1985
Building Blocks

• Commitment schemes
• Blind signatures
Commitments

- Like locked box or safe
- Hiding – hard to tell which message is committed to
- Binding – there is a unique message corresponding to each commitment
Blind signatures

Alice learns only signature on her message.
Signer learns nothing.
Background on RSA Signatures

• Key Generation
  – Generate primes $p$, $q$; $N = pq$
  – Public key = $e$; private key = $d$ s.t.
    $ed = 1 \mod (p-1)(q-1)$

• Sign
  – $C = M^d \mod N$

• Verify
  – Check $M \mod N = C^e \mod N$
  – Note $C^e \mod N = M^{ed} \mod N = M \mod N$
Chaum’s scheme (1)

\[ N = pq \]
\[ e = 3 \]
\[ d \text{ is private} \]

\[ B = r^e f(x) \pmod{N} \]

- B is a blinded message: does not reveal information about f(x) to bank
- f(x) is a commitment to x
Chaum’s scheme (2)

N = pq
e = 3
d = multiplicative inverse of 3

\[ BC = r f(x)^{1/3} \pmod{n} \]

\[ C = f(x)^{1/3} \pmod{n} \]

- BC = B^d \pmod{n} is a blind signature on B
- Bank issues blinded coin and takes $1 from Alice’s account
- Alice extracts coin
Bob verifies bank’s signature on f(x) using bank’s public key

- Bob calls bank immediately to verify that the electronic coin has not been already spent
- Bank checks coin and, if OK, transfers $1 to Bob’s account
Can we do better?

- Do not require Bob to call Bank immediately
- Catch Alice if she tries to spend the same coin twice
Untraceable Electronic Cash
Chaum, Fiat, Naor 1990
CFN90 scheme (1)

\[ N = pq \]
\[ e = 3 \]
\[ d \text{ is private} \]
\[ k \text{ is a security parameter} \]

- \( f, g \) are collision-resistant functions
- \( f(.,.,.) \) is a random oracle
- \( g(x,.) \) is a one-to-one function
Obtaining an Electronic Coin
CFN90 scheme (2)

\[ B_i = r_i e^{f(x_i, y_i) \pmod{n}} \quad 1 \leq i \leq k \]
where
\[ x_i = g(a_i, c_i) \]
\[ y_i = g(a_i \oplus (u \| (v+i)),d_i) \]

- \( B_i \) is a blinded message: does not reveal information about \( f(x,y) \) to bank
- \( f(x,y) \) is a commitment to \((x, y)\)
- \( x, y \) are constructed to reveal \( u \) in case Alice tries to spend the same coin twice
Check blinded candidates in R

R = random subset of k/2 indices

Reveal $a_i, c_i, d_i, r_i$ for $i$ in R

• Ensure Alice following protocol
• Assume R = \{k/2+1,\ldots,k\} to simplify notation
CFN90 scheme (4)

- Bank issues blinded coin and takes $1 from Alice’s account
- Bank and Alice increments Alice’s counter $v$ by $k$
- Alice extracts coin
Paying with an Electronic Coin
To pay Bob one dollar, Alice and Bob proceed as follows:

1. Alice sends $C$ to Bob.
2. Bob chooses a random binary string $z_1, z_2, \ldots, z_{k/2}$.
3. Alice responds as follows, for all $1 \leq i \leq k/2$:
   a. If $z_i = 1$, then Alice sends Bob $a_i$, $c_i$ and $y_i$.
   b. If $z_i = 0$, then Alice sends Bob $x_i$, $a_i \oplus (u \|(v + i))$ and $d_i$.
4. Bob verifies that $C$ is of the proper form and that Alice’s responses fit $C$.
5. Bob later sends $C$ and Alice’s responses to the bank, which verifies their correctness and credits his account.

- Steps 2, 3: Alice reveals her commitment
- Step 4: Bob check’s Alice’s commitment and Bank’s signature on coin $C$
- Step 5: Note Bob does not have to call Bank immediately
CFN90 scheme (6)

• What if Alice double-spends (gives the same coin to both Bob and Charlie)?

• Bank stores coin C, random strings $z_1, z_2, ..., z_{k/2}$ and $a_i$ (if $z_i = 1$) and $a_i \oplus (u \| (v+i))$ (if $z_i = 0$)

• If Alice double spends, then $\wp \frac{1}{2}$ Bank obtains $a_i$ and $a_i \oplus (u \| (v+i))$ for the same $i$ and thus obtains Alice’s identity and transaction counter $u \| (v+i)$
CFN90 scheme (7)

• What if Alice colludes with merchant Charlie and sends the same coin C and the same z to him as she did with Bob?

• Bank knows that one of Bob and Charlie are lying but not who; cannot trace back to Alice

• Solution: Every merchant has a fixed query string different from every other merchant + a random query string
Summary

• Electronic Cash
  – Untraceable if issued coins are used only once
  – Traceable if coin is double spent
  – (Some) collusion resistance

• Instance of Anonymous Credentials
Questions
Commitment

• Temporarily hide a value, but ensure that it cannot be changed later
  – Example: sealed bid at an auction

• 1\textsuperscript{st} stage: commit
  – Sender electronically “locks” a message in a box and sends the box to the Receiver

• 2\textsuperscript{nd} stage: reveal
  – Sender proves to the Receiver that a certain message is contained in the box
Properties of Commitment Schemes

• Commitment must be hiding
  – At the end of the 1\textsuperscript{st} stage, no adversarial receiver learns information about the committed value
  – If receiver is probabilistic polynomial-time, then \textit{computationally} hiding; if receiver has unlimited computational power, then \textit{perfectly} hiding

• Commitment must be binding
  – At the end of the 2\textsuperscript{nd} stage, there is only one value that an adversarial sender can successfully “reveal”
  – Perfectly binding vs. computationally binding

• Can a scheme be perfectly hiding and binding?
Discrete Logarithm Problem

• Intuitively: given $g^x \mod p$ where $p$ is a large prime, it is “difficult” to learn $x$
  – Difficult = there is no known polynomial-time algorithm

• $g$ is a generator of a multiplicative group $\mathbb{Z}_p^*$
  – Fermat’s Little Theorem
    • For any integer $a$ and any prime $p$, $a^{p-1}=1 \mod p$.
    – $g^0, g^1 \ldots g^{p-2} \mod p$ is a sequence of distinct numbers, in which every integer between 1 and $p-1$ occurs once
      • For any number $y \in [1..p-1]$, $\exists x$ s.t. $g^x = y \mod p$
    – If $g^q=1$ for some $q>0$, then $g$ is a generator of $\mathbb{Z}_q$, an order-$q$ subgroup of $\mathbb{Z}_p^*$
Pedersen Commitment Scheme

• **Setup:** receiver chooses...
  – Large primes $p$ and $q$ such that $q$ divides $p-1$
  – Generator $g$ of the order-$q$ subgroup of $\mathbb{Z}_p^*$
  – Random secret $a$ from $\mathbb{Z}_q$
  – $h = g^a \mod p$
    – Values $p, q, g, h$ are public, $a$ is secret

• **Commit:** to commit to some $x \in \mathbb{Z}_q$, sender chooses random $r \in \mathbb{Z}_q$ and sends $c = g^x h^r \mod p$ to receiver
  – This is simply $g^{x(a)} r = g^{x + ar} \mod p$

• **Reveal:** to open the commitment, sender reveals $x$ and $r$, receiver verifies that $c = g^x h^r \mod p$
Security of Pedersen Commitments

• Perfectly hiding
  – Given commitment c, every value x is equally likely to be the value committed in c
  – Given x, r and any x’, exists r’ such that $g^x h^r = g^{x'} h^{r'}$
    $r' = (x-x')a^{-1} + r \mod q$ (but must know a to compute r’)

• Computationally binding
  – If sender can find different x and x’ both of which open commitment $c = g^x h^r$, then he can solve discrete log
    • Suppose sender knows x,r,x’,r’ s.t. $g^x h^r = g^{x'} h^{r'} \mod p$
    • Because $h = g^a \mod p$, this means $x+ar = x'+ar' \mod q$
    • Sender can compute a as $(x'-x)(r-r')^{-1}$
    • But this means sender computed discrete logarithm of h!
RSA Blind Signatures

One of the simplest blind signature schemes is based on RSA signing. A traditional RSA signature is computed by raising the message $m$ to the secret exponent $d$ modulo the public modulus $N$. The blind version uses a random value $r$, such that $r$ is relatively prime to $N$ (i.e. $\gcd(r, N) = 1$). $r$ is raised to the public exponent $e$ modulo $N$, and the resulting value $r^e \mod N$ is used as a blinding factor. The author of the message computes the product of the message and blinding factor, i.e.

$$m' \equiv mr^e \pmod{N}$$

and sends the resulting value $m'$ to the signing authority. Because $r$ is a random value and the mapping $r \mapsto r^e \mod N$ is a permutation it follows that $r^e \mod N$ is random too. This implies that $m'$ does not leak any information about $m$. The signing authority then calculates the blinded signature $s'$ as:

$$s' \equiv (m')^d \pmod{N}.$$  

$s'$ is sent back to the author of the message, who can then remove the blinding factor to reveal $s$, the valid RSA signature of $m$:

$$s \equiv s' \cdot r^{-1} \pmod{N}.$$  

This works because RSA keys satisfy the equation $r^{ed} \equiv r \pmod{N}$ and thus

$$s \equiv s' \cdot r^{-1} \equiv (m')^d r^{-1} \equiv m^d r^{-1} \equiv m^d \pmod{N},$$

hence $s$ is indeed the signature of $m$. 

34