Automated Trust Negotiation Using OACerts

Jiangtao Li

Purdue University
Outline

- Background and motivation
- Oblivious attribute certificates
  - [Li and Li, ACNS 2005]
- A new framework for trust negotiation
  - [Li, Li, and Winsborough, CCS 2005]
Decentralized Access Control

- Access control in decentralized, open, and distributed systems is different from traditional access control in operating systems.

- In open environments, access control decisions are often based on the attributes of the requester.

- Attributes are documented through digital credentials issued by trusted CAs.
  - E.g., citizenship, membership, date of birth, income, credit rating, security clearance.
Automated Trust Negotiation (ATN)

- Attribute information may be sensitive and needs to be protected
- In trust negotiation approach, each credential is protected by an access control policy
- ATN is a process in which two strangers establish trust via iterative exchange of digital certificates
An Example of Trust Negotiation

Student
Alice

Bookstore
Bob

I request a special bookstore discount
Only students younger than 21 can get the discount
I have a driver's license that documents my DoB
But prove to me that you are a member of BBB
Here's my driver's license and student card
Here is my BBB membership card
You are qualified to get the discount
Limitations on Existing ATN Approaches

Attribute information in a certificate is disclosed in an all-or-nothing fashion.

Alice.weight : false
Alice.DoB : true
Bob.discount : student \land age<21
Limitations on Existing ATN Approaches (cont.)

If the policy is sensitive, the only way to satisfy the policy is to reveal all related certificates unconditionally.

Alice.DoB : false
Alice.age : true

Bob.discount : age<21
Limitations on Existing ATN Approaches (cont.)

If there is a policy cycle, the negotiation will fail.

\[
\text{Alice.age : BBBmember}
\]
\[
\text{Bob.discount : student \land age<21}
\]
\[
\text{Bob.BBBmember : age}
\]
Summary of Our contribution

- We develop several techniques to address the previous limitations

- In particular, we develop
  - A new cryptographic certificate scheme
  - Several associated protocols
  - An ATN framework that supports various cryptographic certificates and protocols
Outline

- Background and motivation
- Oblivious attribute certificates
- A new framework for trust negotiation
Background Review: Cryptographic Commitment Scheme

- commit
- open
- prove the committed value satisfies some property without opening the commitment
Oblivious Attribute Certificates (OACerts)
Details of OCerts Scheme

- Issue OCerts
  - CA computes the commitments for each attribute and signs the certificate
  - CA gives the certificate and all the keys to Alice

- Direct usage of OCerts
  - Alice can show her OCerts to Bob without revealing any attribute values
  - Alice can open the commitments of some attributes
  - Alice can prove that her attributes satisfy some property using zero-knowledge proof techniques

- Additional features
  - Compatible with PKI and existing systems
  - Revocation can be handled using CRL
Oblivious Usage of An Attribute

- Bob’s policy is based on Alice’ attribute

- Alice can use her attribute to obtain Bob’s resource without leaking any information about it, not even whether she satisfies the policy

Motivation and application

- Break policy cycles
- Minimum information disclosure
Oblivious Commitment-Based Envelope (OCBE)

Case 1: \( \text{Pred}(\text{attr}) = \text{true} \)
- Message:
  - Policy: \( \text{Pred} \)
  - Attr: \( \text{attr} \)
  - Commitment: \( c \)

Case 2: \( \text{Pred}(\text{attr}) = \text{false} \)
- Security Property:
  - Sound
  - Oblivious
  - Secure against the receiver
Breaking Policy Cycles

I will show my birth-date to you only if you are older than 30

I will show my birth-date to you only if you are younger than 40

Alice can open the envelope only if she is younger than 40
Pedersen Commitment Scheme

- **Setup**
  - Outputs $\langle p, q, g, h \rangle$
  - $p, q$ are two large primes where $q | (p-1)$
  - $g, h$ are two random elements in $G_q$

- **Commit**
  - To commit $a$, chooses $r \leftarrow \mathbb{Z}_q$
  - Computes $c = \text{commit}(a, r) = g^a h^r \mod p$

- **Open**
  - Reveals $a$ and $r$.
  - The verifier verifies $c = g^a h^r \mod p$

- **Security Property**
  - Unconditionally hiding and computationally binding
EQ-OCBE: an OCBE protocol for equality predicates

\[ \text{Pred} = EQ_{a_0}, \ c = \text{commit}(a, r) = g^a h^r \]

\[ \langle \eta = h^y, C = E_{H(\sigma)}[M] \rangle \]

**Receiver**

**Input:** \( EQ_{a_0}, c, a, r \)

**Steps:** if \( a = a_0 \)

1. Computes \( \sigma' = \eta^r \)
2. Decrypts \( C \) using \( H(\sigma') \)

**If** \( a = a_0 \), \( \sigma = (cg^{-a_0})^y = (g^{a-a_0} h^r)^y = (h^y)^r = \eta^r = \sigma' \)

**Sender**

**Input:** \( EQ_{a_0}, c, M \)

**Steps:**

1. Picks \( y \leftarrow Z_q^* \)
2. Computes \( \sigma = (cg^{-a_0})^y \)
Outline

- Background and motivation
- Oblivious attribute certificates
- A new framework for trust negotiation
Integrate OACerts into ATN

- Given OACerts and the associated protocols, how can we integrate them into ATN?
  - How do we model a credential?
  - How do we model an attribute?
  - How do we model delegation?
  - How do we model a private policy?
  - When to use these protocols?

These questions will be answered in the next few slides
A New Framework for ATN

- We propose an ATN framework that supports
  - diverse credentials
  - various cryptographic protocols
  - uncertified attribute information

- Our framework consists of
  - ATNL: a logic-based policy language
  - ETTG: a negotiation protocol
Language for Our Framework: Credentials

- Membership credentials
  \[\text{BBB.member} \leftrightarrow \text{Bob}\]

- Credential with attributes
  \[\text{CoS.student(prorgram='cs',level='soph')} \leftrightarrow \text{Alice}\]

- Credential with committed attributes
  \[\text{BMV.dLicense(name=commit(Alice), DoB=commit('03/07/86'))} \leftrightarrow \text{Alice}\]

- Delegation credentials
  \[\text{StateU.student} \leftrightarrow \text{CoS.student}\]
Language for Our Framework: 
Attribute Declarations

- **Certified attributes**
  - DoB = '03/07/86' ::
    - BMV.dLicense(DoB), Gov.passpt(DoB) :: sensitive

- **Uncertified attributes**
  - phoneNum = '(123)456-7890' :: :: sensitive

- **Non-sensitive attributes**
  - program = 'cs' :: CoS.student(program) :: non-sensitive
Language for Our Framework: Policies

- Policy
  \[ \text{Bob.discount} \leftarrow \text{Gov.employee} \]

- Policy with constraint
  \[ \text{Bob.discount} \leftarrow \text{BMV.dLicense(DoB=x)}; \]
  \[ x > '01/01/84' \]

- Private policy
  \[ \text{Bob.discount} \leftarrow \text{BMV.dLicense(DoB=x)}; \]
  \[ \text{false ! } x > '01/01/84' \]

- Policy that requires disclosure of attribute value
  \[ \text{Bob.discount} \leftarrow \text{Any.phoneNum(value } \Rightarrow x); \]
Language for Our Framework: More Policies

- **Ack policy**
  - Authorizes acknowledgement of possession of a credential
  
  \[
  \text{disclose}(\text{ack}, \text{StateU.student}) \leftrightarrow \text{BBB.member}
  \]

- **Access control policy**
  - Authorizes transmission of a credential
  
  \[
  \text{disclose}(\text{ac}, \text{StateU.student}) \leftrightarrow \text{BBB.member}
  \]
Language for Our Framework: More Policies

- **Full attribute policy**
  - Authorizes disclosure of the exact value of an attribute
  - `disclose(full, DoB) ← BBB.member`

- **Bit attribute policy**
  - Authorizes disclosure of whether the attribute satisfies a predicate chosen by the other party
  - `disclose(bit, DoB) ← BBB.member`

- **Range attribute policy**
  - Authorizes disclosure of the attribute value at a given level of precision
  - `disclose(range, DoB, year) ← BBB.member`
Questions?

Contact:

Jiangtao Li
jtli@cs.purdue.edu
http://www.cs.purdue.edu/homes/jtli