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]

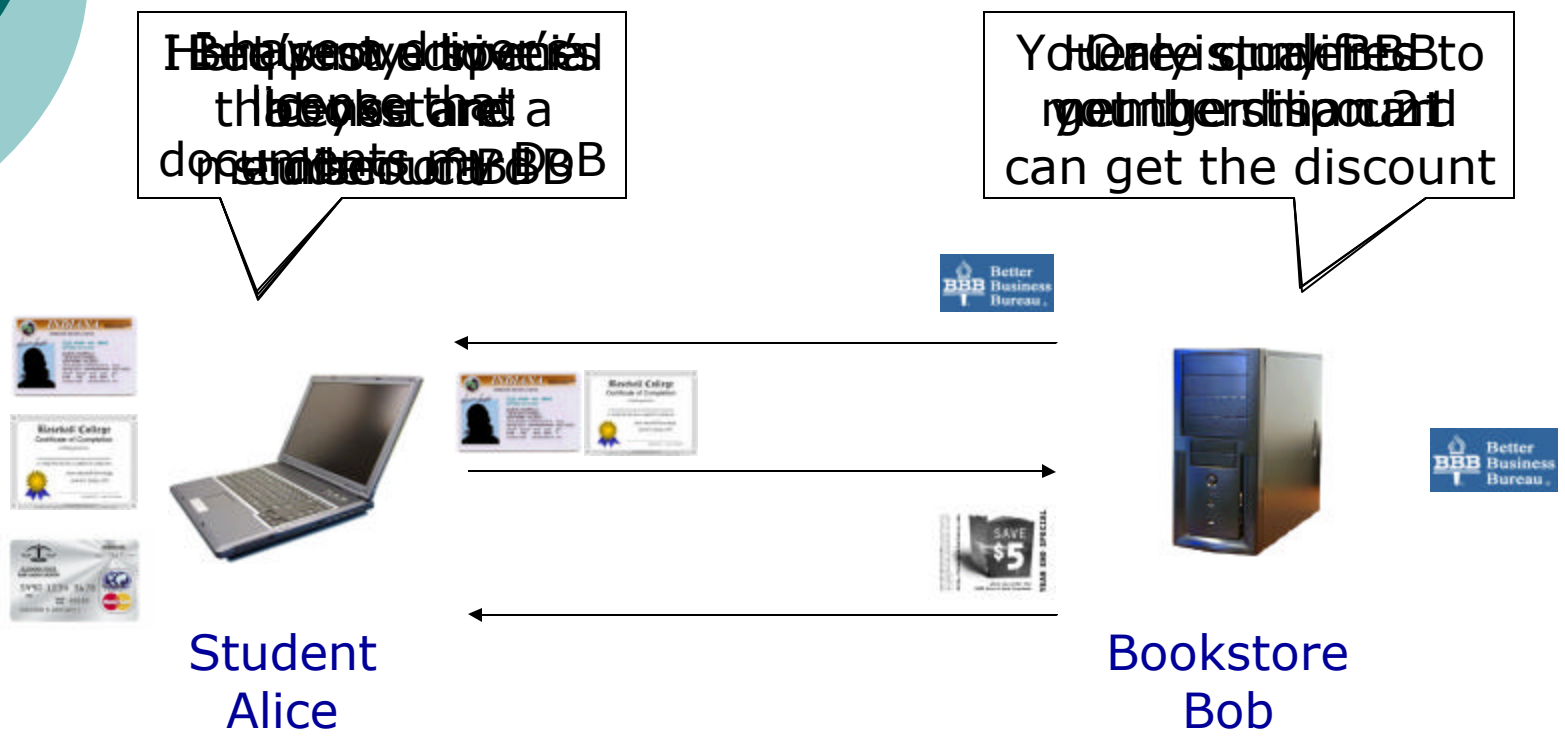
Decentralized Access Control

- 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)

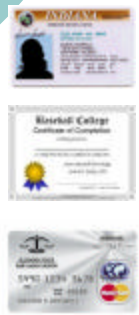
- Attribute information may be sensitive and needs to be protected
- In trust negotiation approach, each credential is protected by an access control policy
- ATN is a process in which two strangers establish trust via iterative exchange of digital certificates

An Example of Trust Negotiation



Limitations on Existing ATN Approaches

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



Negotiation Failed

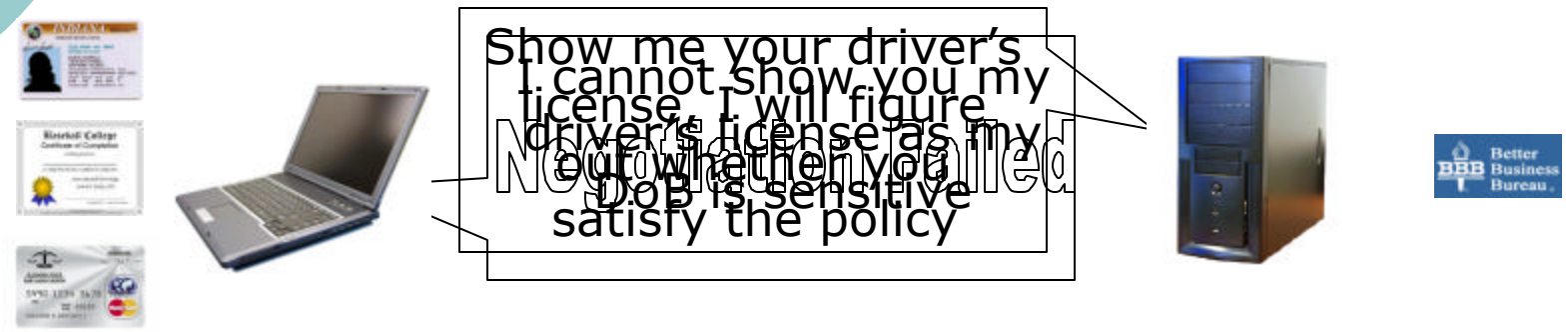
Alice.weight : false

Alice.DoB : true

Bob.discount : student $\wedge$ 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.




Negotiation Failed

Alice.age : BBBmember

Bob.discount : student $\wedge$ age<21

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

value

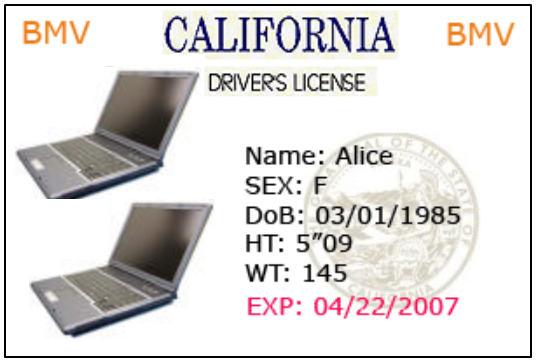
Prover
Alice

value

Verifier
Bob

- commit
- open
- prove the committed value satisfies some property without opening the commitment

Oblivious Attribute Certificates (OACerts)



California Driver License

Expired: 04-22-07

Name: Alice
DoB: 03/01/1985
SEX: F

HT: 5"09
WT: 145

Signed by BMV

X.509 Certificate

California Driver License

Expired: 04-22-07

Name:
DoB:
SEX:

HT:
WT:

Signed by BMV

OACerts

Details of OCerts Scheme

- Issue OACerts
  - CA computes the commitments for each attribute and signs the certificate
  - CA gives the certificate and all the keys to Alice
- Direct usage of OACerts
  - Alice can show her OACerts 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)

Receiver
Alice

c

Sender
Bob

Attr: attr
commitment: c

Message:
Policy: Pred

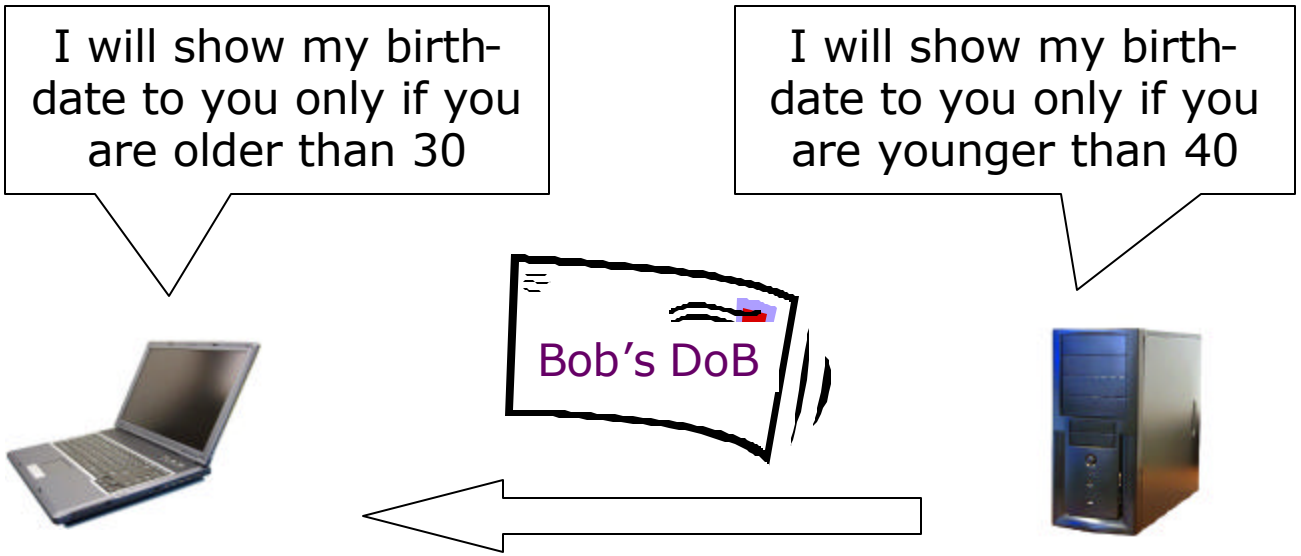
Case 1: Pred(attr) = true

Case 2: Pred(attr) = false

Security Property:

- Sound
- Oblivious
- Secure against the receiver

Breaking Policy Cycles



Pedersen Commitment Scheme

- Setup
  - Outputs ⟨p,q,g,h⟩
  - p, q are two large primes where q|p-1
  - g, h are two random elements in G_q
- Commit
  - To commit a, chooses $r \leftarrow Z_q$
  - Computes $c = \text{commit}(a, r) = g^a h^r \bmod p$
- Open
  - Reveals a and r,
  - The verifier verifies $c = g^a h^r \bmod p$

- Security Property
  - Unconditionally hiding and computationally binding

EQ-OCBE: an OCBE protocol for equality predicates

$\text{Pred} = \text{EQ}_{a_0},\ c = \text{commit}(a, r) = g^a h^r$

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

Receiver

Sender

Input: EQ_{a_0}, c, a, r

Steps: if $a = a_0$

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

Input: EQ_{a_0}, c, M

Steps:

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

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

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

  BBB.member ← Bob

- Credential with attributes

  CoS.student(prorgram='cs',level='soph') ← Alice

- Credential with committed attributes

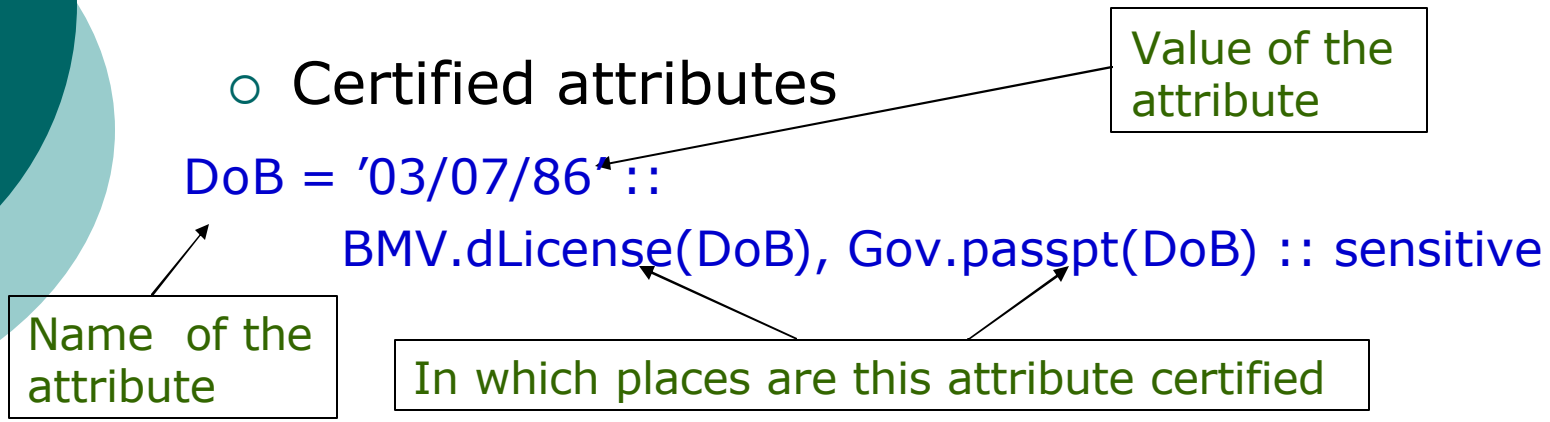
  BMV.dLicense(name=commit(Alice),
  DoB=commit('03/07/86')) ← Alice

- Delegation credentials

  StateU.student ← CoS.student

Language for Our Framework: Attribute Declarations

- Certified attributes



- Uncertified attributes

  phoneNum = '(123)456-7890' :: :: sensitive

- Non-sensitive attributes

program = 'cs' :: CoS.student(program) :: non-sensitive

Language for Our Framework: Policies

- Policy

  Bob.discount $\leftarrow$ Gov.employee

- Policy with constraint

  Bob.discount $\leftarrow$ BMV.dLicense(DoB=x);
  x > ’01/01/84’

- Private policy

  Bob.discount $\leftarrow$ BMV.dLicense(DoB=x);
  false ! x > ’01/01/84’

- Policy that requires disclosure of attribute value

  Bob.discount $\leftarrow$ Any.phoneNum(value $\Rightarrow$ x);

Language for Our Framework: More Policies

- Ack policy
  - Authorizes acknowledgement of possession of a credential

  disclose(ack, StateU.student) $\leftarrow$ BBB.member

- Access control policy
  - Authorizes transmission of a credential

  disclose(ac, StateU.student) $\leftarrow$ BBB.member

Language for Our Framework: More Policies

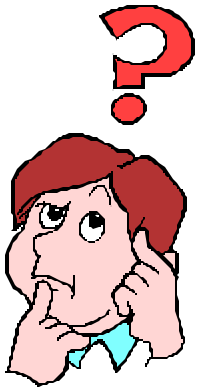
- Full attribute policy
  - Authorizes disclosure of the exact value of an attr

  disclose(full, DoB) $\leftarrow$ BBB.member
- Bit attribute policy
  - Authorizes disclosure of whether the attr satisfies a predicate chosen by the other party

  disclose(bit, DoB) $\leftarrow$ BBB.member
- Range attribute policy
  - Authorizes disclosure of the attr value at a given level of precision

  disclose(range, DoB, year) $\leftarrow$ BBB.member

Questions?



Contact:

Jiangtao Li

jtli@cs.purdue.edu

http://www.cs.purdue.edu/homes/jtli