TRUST NEGOTIATIONS
CONCEPTS AND SYSTEMS

ANNA CINZIA SQUICCIARINI
The Pennsylvania State University
May 18th, 2009

Outline

♦ Foundations
♦ State of the Art
♦ Break ❼
♦ The Trust-X Project
♦ Open issues
  / Research challenges
  / Future Trends
In *open systems*, resources are shared across organizational boundaries

When users come from *outside a known domain or organization*, their identity is insufficient to determine their access rights.

The dynamic nature of open systems makes preregistration impractical and unscalable.

### Foundations - The idea

**Attribute based access control in open systems**

1. Project report request
2. Show your membership to the project, your citizenship and your phone number.

Attributes –not the identity!– of the prospective user are requested until the protected resources can be released.
Access control in open systems: issues

- How to model properties of the parties?
- How to establish mutual trust in a gradual fashion?
- Can a party apply alternative policies based on the interacting subject?
- Could we have alternative and feasible approaches to trust establishment?

On the notion of trust

- The Pudding of Trust: several notions of trust each emphasizing different set of issues[Bhargava,’04].
- Conceptual core, that provides a general construct to model a variety of senses of trust in different applications domains[Jones,01]. (A trusts B on doing Z)
- Literature of information systems
  - Human-system trust HCI (building long term relationship between human and computers)[Bickmore,00]
  - System-mediated trust Trust Management. MultiCentric Access control [Blaze,’96].
The solution... trust negotiation protocols

- **The goal**: establish mutual trust between possibly unknown parties.

- **The approach**: establish trust by verifying properties (*credentials*) of the other party.

- Protect sensitive credentials and services with ad hoc policies, namely disclosure policies.

- Instead of a one-shot authorization and authentication, trust is established incrementally through a sequence of bilateral credential disclosures.

Trust negotiation approaches to authorization have a number of advantages

- **Primary advantage**: intension vs. extension

Secondary benefits:
- *Bilateral* establishment of trust
- *Iterative* protocols
- *Automated* construction of machine-verifiable proofs
Application domains

- Healthcare applications
- Online transactions
- Collaborative Systems
- Grid Computing systems
- E-commerce
  - B2B and B2C transactions

Foundations - Language

- Credentials
  - Statements issued by trusted authorities stating properties of the credential holder
  - Could be typed
    - Type Name
    - Attribute Set (pair name/value)

IdCard(Issuer, First Name, Last Name, DateOfBirth,...)
Passport(Issuer, First Name, Last Name, Citizenship, Expiration Date)
Foundations - basic notions

- Disclosure Policies
  - Rule controlling a sensitive resource:
    - service, object, data, resource
  - Could be logic formula
  - Various notations employed

R↓IdCard V Passport
R↓IdCard(Issuer=USA, Name=Xet)

Conditions against credential’s attributes

Implementing the Trust Negotiation Language

- Credentials
  - X.509
  - SAML assertions
  - XML certificates

- Policies
  - XML
  - WS-Policy
  - XACML

Two typical X.509 Certification paths
The Protocol

- Trust evaluation depends on credentials’ exchange
  - Credentials possession
  - Attributes stated in credentials
  - Trustworthiness of the credential issuers

- Disclosure governed by credential exchange policy
- Rely on iterative disclosure of credentials and associated policies to ensure safe negotiation
The compliance checker modules

Service Provider’s Policy

(1)

Alice’s Disclosed Credentials

♦ Traditional Trust Management compliance checker
♦ Determines whether a set of credentials satisfies a policy


Compliance Checker

Bob’s Disclosed Policy

(2)

Alice’s Local Credentials

♦ Determines whether a policy is satisfied and how a policy is satisfied, but only produces a single satisfying set.
♦ IBM Trust Establishment (TE) and REFEREE produce a set of local credentials that satisfy the received policy.

Compliance Checker

Determines whether a policy is satisfied and how a policy is satisfied and produces all the possible satisfying credentials’ set.

Supported by PSPL [Bonatti-Samarati] and RT [Li et al.].


Outline

- Foundations
- State of the Art
- Break ♦
- The Trust-X Project
- Open issues
  - Research challenges
  - Future Trends
State of the Art – Trust Negotiation

- First introduced by Winsborough et al. in 2000
  “automated trust establishment between strangers through credential exchange when credentials are themselves potentially sensitive”.

- Two negotiation strategies:
  - an eager strategy: negotiators disclose each credential as soon as its access control policy is satisfied,
  - parsimonious strategy: negotiators disclose credentials only after exchanging sufficient policy content to ensure that a successful outcome is ensured.


Research streams in Trust Negotiations

- Theoretical Foundations
- Logics
  - Bonatti and Samarati CCS 2000
  - Winslett et al. CCS 2005
- Policy Languages
  - Herzog et al. Oakland 2000
  - Li and Mitchell DISCEX 2003
  - Becker and Sewell POLICY 2004
  - Bertino et al. TKDE 2004
- Strategies
  - Yu et al. TISSEC 2003
  - Bertino et al. TKDE 2004
  - Li et al. CCS 2005
  - Frikken et al. NDSS 2006
- Privacy, Anonymity
  - Bradshaw et al., CCS ’04
  - Frikken et al., WPES ’03
  - Squicciarini et al. TISSEC 2007
  - Li and Li 2006 CCS 06
- Consistency and Concurrency Control
  - Lee and Winslett CCS 2006
  - Lee and Winslett TISSEC
  - Lee et al. TISSEC (submitted)
- System Prototypes
- Deployment Considerations
  - Squicciarini et al. TISSEC 2007
  - Lee et al. SACMAT 2006
  - Lee and Winslett Trust 2004
  - Lee et al. TISSEC 2008
Requirements of a good trust negotiation system

- Credential ownership
- Unified formalism
- Privacy protection mechanisms
- Compliance checker modes
- Formal semantics
- Runtime system
- Support of complex policies


Trust Negotiation Logics

- Problem: how do we possibly represent trust negotiation protocols?
- Solutions: (Many logic frameworks proposed...some examples)
  - **RT**: role-based trust management framework used for negotiations [Li, Winsborough, 02]
  - **Trusted Graph Protocol**: Trust target graph (TTG), directed graph representing the state of negotiation [Winsborough Li ’05]
  - **PeerAccess**: Infrastructure and framework to reason about authorization in open system [Winslett et al. ’05]


Safety In Trust Negotiations

Problem: it is not trivial to define when a negotiation protocol is safe, and what is safety in Trust Negotiations.

Solutions:

- Safe disclosure sequences. [Yu, Winslett, ’04]
- Indistinguishability. Strong guarantees regarding the amount of private information leaked during the negotiation [Winsborough, Li, ’04]
- Safety of the information, based on the idea of information gain [Irwin, Yu, ’05]
- Safety of the decision as a result of the consistency of the underlying state information used during policy evaluation [Lee, Winslett, ’06]


Practice and Theory

Problem: There is a big gap between theory and practice in Trust Negotiation research

- Mismatch between theoretical models and asynchronous systems
- Lack of suitable frameworks and software components for deploying and evaluating systems
- Algorithmic complexity of theoretically “easy” problems

Solution:

- Safety and consistency: Theory doesn’t consider the effect of asynchronous systems. Lee’s work addresses the issue [Lee,Winslett ’06]
- Deployment concerns: Traust architecture: TN implementations talking to legacy services [Lee et. al. 06]; Recovery protocols for faulty negotiations [Squicciarini et. al. 2007]

Systems/Architectural considerations: TrustBuilder2 framework for prototyping and experimenting with TN protocols; Clouseau compliance checker is a high-performance policy compiler and compliance checker (Lee, Winslett 08)

The problem of sensitive credentials

1. Show your driver license, I will figure out whether you satisfy the policy.
2. I cannot show you my driver’s license as my date of Birth is sensitive.

Privacy and Cryptographic Protocols

Problem:
- Attribute information in a certificate is disclosed in an all-or-nothing fashion.
- If the policy is sensitive, the only way to satisfy the policy is to reveal all related certificates unconditionally.
- By analyzing a policy the negotiating party can infer a lot of information about the counterpart’s profile.
Cryptographic based Trust Negotiations

**Solutions:**
- secret handshakes [Balfanz et. al. ‘03]: Solution built from pairing based key agreements.
- hidden credentials [Bradshaw et al. ‘04]: By using Identity Based Encryption scheme, one can verify other’s properties without disclosing even the policies.
- oblivious signature based envelope, oblivious commitment based envelope [Frikken et. al. ’04]: Based on ZKP protocols.

---

**One interesting solution**

**Problem:**
- While these protocols are useful tools and building blocks for ATN, they are not general enough to solve arbitrary trust negotiation problems in a systematic way.

**Solution:**
- [Li, Li,’06] propose OaCerts, based on zero knowledge proof protocols.
- A framework that supports diverse credentials and various cryptographic protocol.

---


Outline

- Foundations
- State of the Art
- Break
- The Trust-X Project
- Open issues
  - Research challenges
  - Future Trends

Trust-X at a glance

- A comprehensive and innovative XML based framework for trust negotiations:
  - Trust negotiation language (X-TNL)
  - Algorithms and strategies to carry out the negotiation process
- System prototype
  - Web Service architecture
  - Support of various trust negotiation modes

Bertino, Ferrari Squicciarini, Trust-X a comprehensive Framework for Trust Negotiations IEEE TKDE 2004;
Privacy solutions for trust negotiation

Language Tools
- Minimal information disclosure
- Selective disclosure of credential
- Policy definition

Strategies
- Loss of data control
- Inference problem
- Minimal information disclosure
- Inference problem

Trust negotiation framework
- Integration with P3P Platform
- Loss of data control

Privacy enhanced credential

Credential Header (plain)
- <Credential......ID......>
- TYPE
- ISSUER

Credential Content (Blinded at first release)
- <NAME>
- <ADDRESS>
- <CITIZENSHIP>

Signature computed over the whole credential

Header used as credential proof

Attribute Names, Values, Random Numbers

Modeling negotiation: Logic formalism

Disclosure policies are expressed in terms of logical expressions which can specify either simple or composite conditions against certificates.

- **P()**: credential type
- **C**: set of conditions

Policy expressed as

\[ R \leftarrow P_1(c_1), P_2(c_2) \]

Resource which the policy refers to

Requested certificates

The basic representation of a disclosure policy

- Disclosure policies can be expressed as logic rules by specifying:
  - Specific credentials
    - Report\(\downarrow\)SchoolId(), Passport()
  - Conditions against attributes
    - Report\(\downarrow\)Passport\(\downarrow\)CountryofIssue=USA), SchoolId()
  - Attributes which may appear in any credential
    - Report\(\downarrow\)X(Job=Professor), Y(Citizenship=USA)
  - High level properties to be mapped into credentials
    - \(\{\text{Report}, \{\text{Position, Citizenship}\}, \{\text{Country=USA, Position=Professor}\}\}\)
Disclosure policy extended definition

\[ (r, <\text{pol\_prec\_set, sugg, priv}> ) \]

\[ \text{CredName}(\text{Attr=A}) \]
\[ \text{X}(\text{Attr=A}) \]
\[ \text{CredName()} \]

\[ R \uparrow x_1, \ldots, x_q \]

\[ \text{list, op} \]
\[ x_1, \ldots, x_n \]
\[ \{+, -\} \]

\[ p_3p \]

Policy Identifier

An example of disclosure policy

\[ p_2=(r, <\text{pol\_prec\_set, sugg, priv}> ) \]

Report
Passport(\text{Citizenship=USA}), SchoolId()

\[ \{p, p_1\} \]

\[ \text{LibraryNet}, - \]

\[ \text{<POLICY>} \]
\[ \text{<STATEMENT>} \]
\[ \text{<DATA-GROUP>} \]
\[ \text{<CATEGORIES>} </\text{CATEGORIES}> \]
\[ \text{<DATA>} \]
\[ \text{<ACCESS>_contact\_and\_other>></ACCESS>_contact\_and\_other>} \]
\[ \text{<PURPOSE>_resolution-type="independent">}</PURPOSE>_resolution-type="independent>)} \]
\[ \text{<current>_develop/}</current>_develop/)} \]
\[ </PURPOSE> \]
\[ </RECIPIENT> </same/> </RECIPIENT> \]
\[ </RETENTION> </stated-purpose/> </RETENTION> \]
\[ </STATEMENT> </STATEMENT> \]
\[ </POLICY> \]
1. Upon receiving a disclosure policy the Trust-X party determines if it can be satisfied by any certificate of its X-profile.
2. Then the compliance checker checks in the Policy base the protection needs associated with the certificates, if any.
3. If a set of credentials and associated policies are actually found, a counterpolicy is sent.

θ If a set of credentials and associated policies are actually found, a counterpolicy is sent.

The state of the negotiation is anyway updated by the tree manager, which records whether new policies and credentials have been involved or not.
Modeling a trust negotiation

- And/or tree to keep track of the progress of the policy evaluation phase
- Each node denotes a request, which can assume different states, depending on
  - the strategy adopted
  - the current state of the negotiation
- Edges denote policies
  - **Simple edge** (policy requiring a single request $R \leftarrow P(C)$)
  - **Multi edge** (policy requiring a set of requests $R \leftarrow P_1(C_1) P_2(C_2)$)

The state of the nodes in a negotiation tree can be
- Undel
- Open
- Cred_Header
- Attribute
- Del, Del

Negotiation tree- example

HELEN: $R$
ALICE: $R \leftarrow P_1(C_1), P_2(C_2)$
ALICE: $R \leftarrow P_3(C_3)$
HELEN: $R \leftarrow P_4(C_4)$
HELEN: $P_1 \leftarrow P_5(C_5)$
HELEN: $P_2 \leftarrow P_6(C_6)$
HELEN: $P_3 \leftarrow \text{UNDEL}$
ALICE: $P_4 \leftarrow \text{DEL}$
ALICE: $P_6 \leftarrow \text{DEL}$

The goal of a negotiation is to determine a valid view of the tree, which is composed of nodes denoting credentials/attributes that comply with both parties trust requirements.
**Views compatibility**

- **Completeness:** If parties have compatible requirements and policies, the algorithm will find a valid view.
  - the same set of nodes is included
  - the nodes have an associated compatible state

- The selected views are mapped into one trust sequence of items to be disclosed.

**Trust negotiation strategies**

*The approach a negotiating party can adopt in carrying on a negotiation.*

- **Strongly Suspicious Strategy**
- **Suspicious Strategy**
- **Trusting Strategy**
- **Standard Strategy**

Privacy protection
Trusting strategy

- Commonly used credentials are regulated by standard off-the-shelf policies, which are public (Credit Card, BBB).
- Avoids computational waste.
- Protects from inference.
- Protects widely used resources.
- Preserves privacy.
- Allows one party to drive the negotiation steps.

Suspicious strategy

- Prove possession of the credentials as soon as the corresponding policy is satisfied.
- Address the “going-first” problem.
Properties of the strategies

- Correctness and completeness
- Interoperability and flexibility
  - Two parties... two strategies
  - The same party... mixed strategies
- Trust Requirement satisfiability
  - Satisfaction of policies for all the involved credentials
  - Compliance of privacy policies
  - Minimum credential disclosure
- Freedom from cycles
  - Detection of repeated credentials and dependencies among policies

Trust-X logical framework

[Diagram showing the Trust-X logical framework with labeled components and arrows indicating the flow of data or processes.]
The Trust-X prototype

Trust-X is composed of a Web service supporting the operations to carry on a trust negotiation according to the different negotiation strategies proposed, and of a client application that invokes the Web service operations.

Some experimental evaluation results

♦ Complexity of the disclosure policies
  - The overall number of credentials involved is kept constant, and 3 simple policies are replaced with a complex policy requesting three credentials.

♦ Number of negotiation rounds
  - Overall number of credential kept constant to 106, exchanged during negotiations of 4, 6 and 8 rounds.
Test results on policy complexity

Test results on negotiation rounds
The Trust-X client application

A policy wizard helps in defining policies and credentials, which are valid XML documents.

Outline
- Foundations
- State of the Art
- Break
- The Trust-X Project
- Open issues
  - Research challenges
  - Future Trends
Trust Negotiation Summary

- Trust establishment among strangers
- Targeted for open systems
- Attribute-based
- Exchange of credentials and policies
- Credentials assert attributes about owner
- Policies express trust requirements

What is left to be done?

**Q1: Scalability**

- How can we implement trust negotiations in a modular, scalable and portable manner?

**Q2: Vulnerabilities**

- What kind of attacks is a trust negotiation vulnerable to?
- How can we mitigate the danger?
Future Research Directions

- **Wide deployment** of trust negotiations
  - What are the social barriers?
  - Why is Trust negotiation *underused*?

- Can we increase **security** of trust negotiation protocols without heavily depend on cryptography?
  - Crypto is sometime unpractical!!
  - Several cryptographic based negotiations are incompatible among each other.

Evolution of trust negotiation models

Q3: How can we effectively deploy collaborations using this model?
- Trust is an essential component for collaborative activities....is the assumption of crypto credentials too strong?

Q4: Can we replace issued credentials with other reliable statements?
- What if there is no such CA that guarantees trust?

Q5: Can we support **multi party** negotiations?
- What are the technical challenges?
- What are the possible domains and applications?
Other applications

- Web services architecture
- Grid Computing systems
- Identity management systems [Bhargav-Spantzel, Squicciarini, 2006]
- Mobile computing [Squicciarini, Bertino, Ferrari, 2004]

Thank you

Questions?
acs20@psu.edu
http://asquicciarini.ist.psu.edu/trustx/

Thanks to Adam J. Lee and Danfeng Yao for the borrowed material and insightful comments.