Automated Trust Negotiation Using Cryptographic Credentials

Jiangtao Li, Ninghui Li, William H. Winsborough

Dept. of Computer Science

University of Texas

Presentation: Heikki Ollikainen /53089D

Presentation

- Introduction
- ATN (Automated Trust Negotiation)
    - example
- ETTG (Extended Trust Target Graph)
    - example

INTRODUCTION

- In ATN, two parties exchange digitally signed credentials that contain attribute information to establish trust and make access control decision.
    - Credentials are either transmitted entirely or not at all.

- Problem: ATN can fail at times, because a cyclic dependency makes neither negotiator want to reveal her credential before her opponent, since the opponent must be authorized for all attributes packaged together in a credential to receive any of them.

- Several cryptographic credential schemes and associated protocols have been developed to address these and other problems.

INTRODUCTION

⇒ Paper presents a framework for ATN in which the diverse credential schemes and protocols can be combined, integrated, and used as needed.

⇒ A new policy language is introduced that enables negotiators to specify authorization requirements that must be met by an opponent to receive various amounts of information about certified attributes and the credentials that contain it.

ATN

- Two parties exchange digitally signed credentials that contain attribute information to establish trust and make access control decision.

- The only way to use credential is to send it as whole, and disclose all the information in the credential.

- A digital credential is viewed as a black-box, and the information in a credential is disclosed in an all-or-nothing fashion.

- There is access control policy associated with each credential and can be disclosed if its access control policy has been satisfied.

ATN

- Limitations of ATN
  - Cyclic dependency among credentials and their policies, negotiations can fail unnecessarily.
  - Since attribute information is disclosed in all-or-nothing fashion, each attribute can be disclosed only when the policy governing the credential and its entire contents is satisfied, leading to unnecessary failure.
  - When one negotiator does not want to disclose detailed information about his policy and the other negotiator does not want to disclose too much information about her attributes, a negotiator can fail even thought the amount of information needs to be disclosed by each party is acceptable to both.

ATN

- ATN features:
  - ATN supports diverse credentials, including standard digital credentials (X.509)
  - ATN framework supports also attribute information that is not certified, in addition to attribute information stored in credentials.
  - ATN framework has a logic-based policy language that is called ATN Language (ATNL) which allows one to specify policies and govern the disclosure of partial information about sensitive attribute
  - ATN framework has a negotiation protocol that enables the various cryptographic protocols to be used to improve the effectiveness of ATN.

ATN

- Cryptographic credential schemes and their associated cryptographic tools:

  1) Separation of credential disclosure from attribute disclosure

  2) Selective show of attributes

  3) Zero-knowledge proof of attributes satisfying a policy

  4) Oblivious usage of a credential

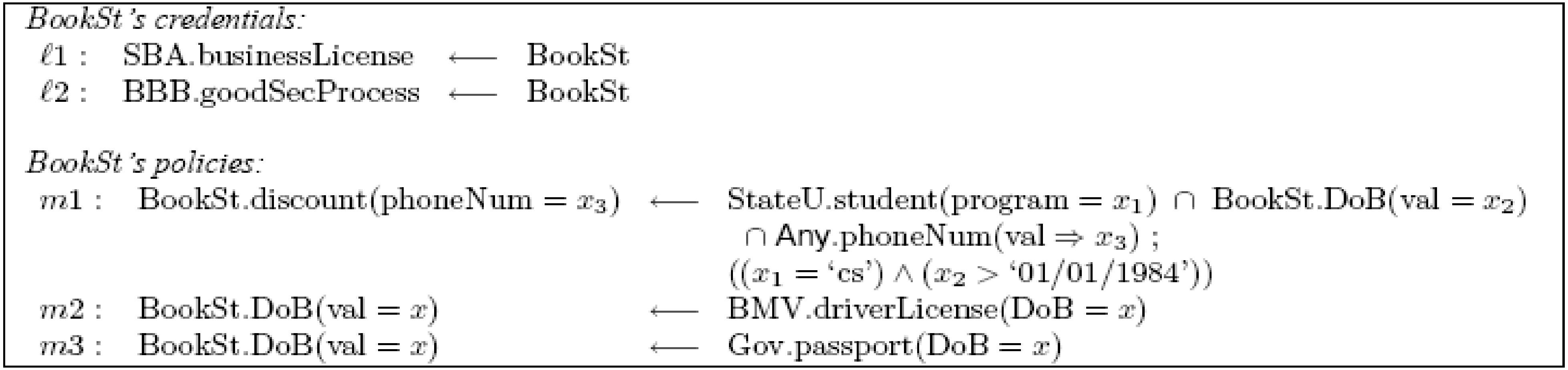
  5) Oblivious usage of an attribute

  6) Certified input private policy evaluation (CIPPE)

ATN language

l1: Bookstore got credentials by Small Business Administration

l2: bookstore is certified by Better Business Bureau to have good security process



BookSt's credentials:

$\ell 1:\quad \mathrm{SBA.businessLicense} \longleftarrow \mathrm{BookSt}$

$\ell 2:\quad \mathrm{BBB.goodSecProcess} \longleftarrow \mathrm{BookSt}$

BookSt's policies:

$m1:\quad \mathrm{BookSt.discount(phoneNum} = x_3) \longleftarrow \mathrm{StateU.student(program} = x_1) \cap \mathrm{BookSt.DoB(val} = x_2)$
$\cap\ \mathrm{Any.phoneNum(val} \Rightarrow x_3)\ ;$
$((x_1 = \text{‘cs’}) \wedge (x_2 > \text{‘01/01/1984’}))$

$m2:\quad \mathrm{BookSt.DoB(val} = x) \longleftarrow \mathrm{BMV.driverLicense(DoB} = x)$

$m3:\quad \mathrm{BookSt.DoB(val} = x) \longleftarrow \mathrm{Gov.passport(DoB} = x)$

m1: the requester should be certified by StateU to be a student majoring Computer Science under 21 and willing to provide a phone number

m2: bookstore consider a driver license from BMW issued by the government to be valid document for date of birth (DoB)

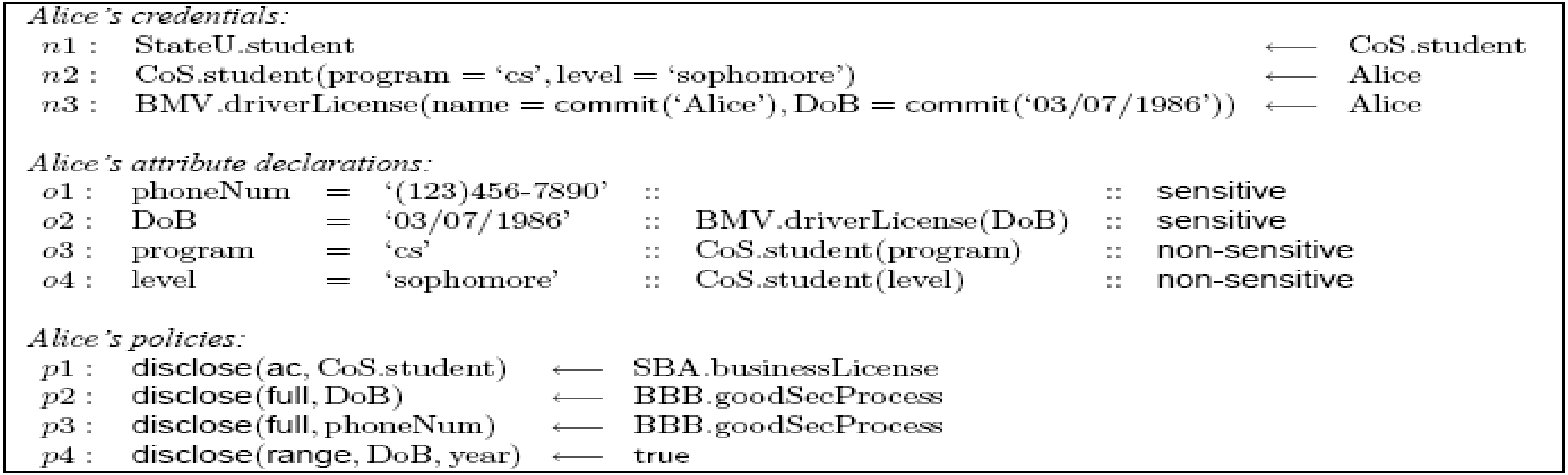
m3: bookstore consider a passport issued by the government to be valid document for date of birth (DoB)

ATN language

n1: Credential issued by StateU and delegates to Collage of Science (CoS) the authority to certify students

n2: Alice’s student certificate issued by CoS

n3: Alice’s digital drivers license issued by BMW



Alice's credentials:

$n1$:	StateU.student	⟵	CoS.student
$n2$:	CoS.student(program = ‘cs’, level = ‘sophomore’)	⟵	Alice
$n3$:	BMV.driverLicense(name = commit(‘Alice’), DoB = commit(‘03/07/1986’))	⟵	Alice

Alice's attribute declarations:

$o1$:	phoneNum	=	‘(123)456-7890’	::		::	sensitive
$o2$:	DoB	=	‘03/07/1986’	::	BMV.driverLicense(DoB)	::	sensitive
$o3$:	program	=	‘cs’	::	CoS.student(program)	::	non-sensitive
$o4$:	level	=	‘sophomore’	::	CoS.student(level)	::	non-sensitive

Alice's policies:

$p1$:	disclose(ac, CoS.student)	⟵	SBA.businessLicense
$p2$:	disclose(full, DoB)	⟵	BBB.goodSecProcess
$p3$:	disclose(full, phoneNum)	⟵	BBB.goodSecProcess
$p4$:	disclose(range, DoB, year)	⟵	true

p1: student certificate *sensitive* will be provide only to those with valid business license from SBA.

p2-p3: DoB and phone number *sensitive*, revealed only for organizations whose security practices are adequate to provide reasonable privacy.

p4: Year of birth available for everyone.

Negotiation process:

1. Alice request discount, BookSt responds with his discount policy (m1)
2. Alice discloses driver license (n3), an OACERT
3. Alice wants BookSt to show valid business license to protect her phone number and student certificate
4. After bookSt shows l1 and l2, Alice reveals her student certificate chain (n1,n2) and phone number (o1).
5. Alice uses zero-proof protocol to prove BookSt that her DoB is between 1/1/86 and 12/31/86, since according to p4 everyone can see it
6. BookSt knows that Alice is under 21 -> Alice proves to be entitled to discount

ETTG

- A trust negotiation protocol that can take advantage of ATNL and the cryptographic protocols

- A trust negotiation process involves the two negotiators together to construct a trust-target graph (TTG)

- Two negotiators working together to construct a trust-target graph (TTG)

- TTG is a direct graph, each node of which is a trust target

ETTG process

- When requester wants to access the resources, the access mediator and the requester enter into a negotiation process. The access mediator creates a TTG containing one target (*primary target*)

- The access mediator then tries to process the primary target by decomposing the question that it asks and expanding the TTG accordingly.
    - It then sends partially processed TTG to the requester
    - In each round, one negotiator receives new information about the changes to the TTG, verifies that the changes are legal and justified and updates its local copy of the TTG accordingly
    - The negotiator then tries to process some nodes, making its own changes to graph which it sends to the other party, completing the round.

- The negotiation succeeds when primary target is satisfied; it fails when the primary target is failed, or when a round occurs in which neither negotiator changes the graph

Nodes and edges in the TTG

Nodes:

- Role target
- Policy target
- Intersection target
- Trivial target
- Attribute goal

⇒Each target has satisfaction state (satisfied, failed, unknown)

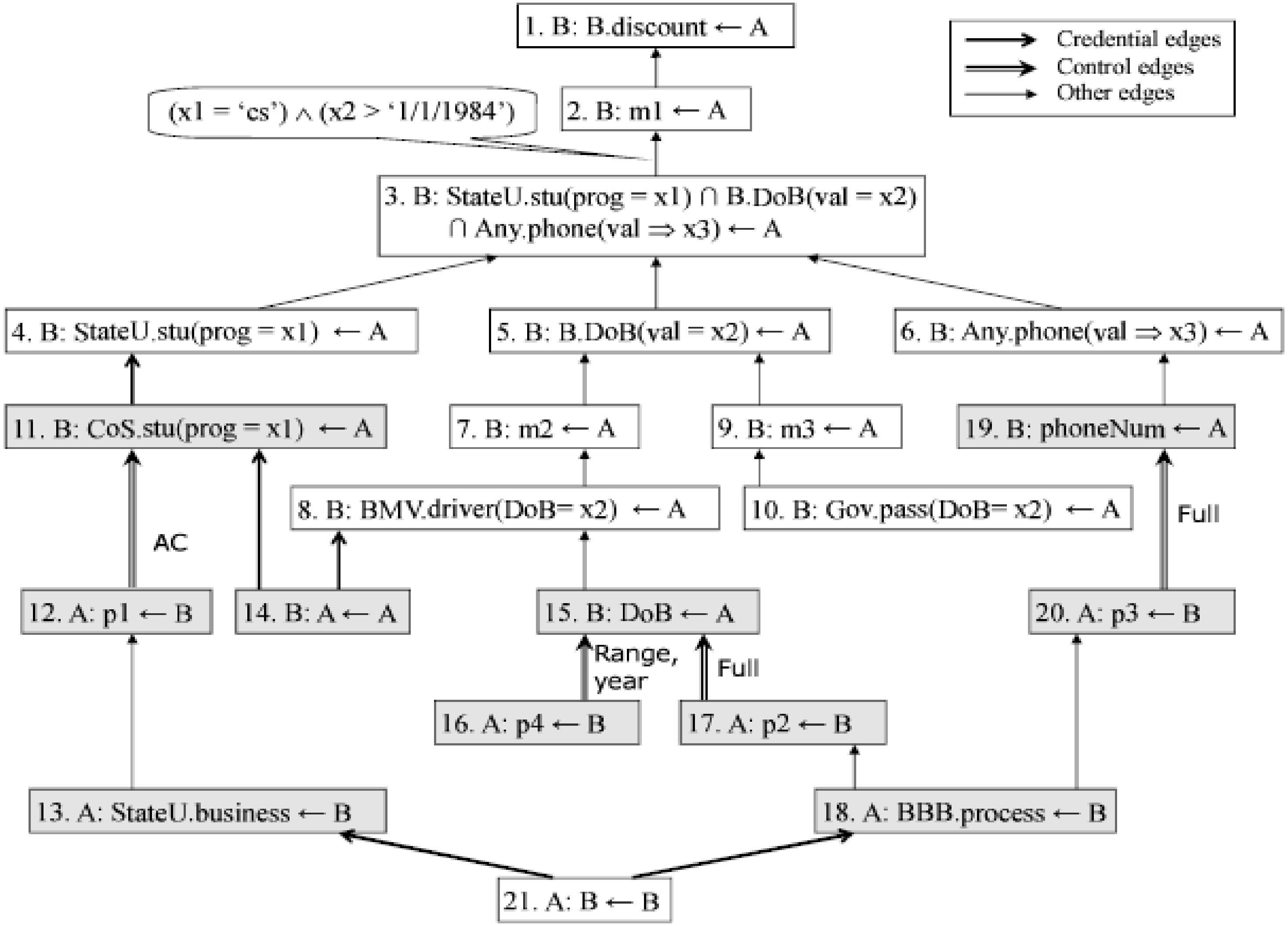
⇒ the value is determined inductively depending on the containing TTG structure and credentials present

Edges:

- Credential edge
- Policy edge
- Policy control edge
- Policy expansion edge
- Intersection edges
- Attribute edge
- Attribute control edge

⇒The optional tag on policy edge is used the constrain portion of the policy statement identified by policy-id.

ETTG protocol and example


1. B: B.discount ← A
Credential edges
Control edges
Other edges
(x1 = 'cs') ∧ (x2 > '1/1/1984')
2. B: m1 ← A
3. B: StateU.stu(prog = x1) ∩ B.DoB(val = x2) ∩ Any.phone(val ⇒ x3) ← A
4. B: StateU.stu(prog = x1) ← A
5. B: B.DoB(val = x2) ← A
6. B: Any.phone(val ⇒ x3) ← A
11. B: CoS.stu(prog = x1) ← A
7. B: m2 ← A
9. B: m3 ← A
19. B: phoneNum ← A
AC
8. B: BMV.driver(DoB= x2) ← A
10. B: Gov.pass(DoB= x2) ← A
Full
12. A: p1 ← B
14. B: A ← A
15. B: DoB ← A
20. A: p3 ← B
Range, year
Full
16. A: p4 ← B
17. A: p2 ← B
13. A: StateU.business ← B
18. A: BBB.process ← B
21. A: B ← B

Example

BookSt (white nodes):

- BookSt wants to see the proof of BookSt.discount => BookSt creates the *primary target* (node 1) for negotiation (state: *unknown*) => If node 1 becomes state: *satisfied*, the negotiation is succeeded
- In BookSt policy base, there is policy statement (m1) for discount, BookSt creates policy target (node 2) and adds a policy *edge* between node 1 and node 2
- BookSt reveals the policy by adding a policy expansion child (node 3) and a constrain *tag* between the parent node 2 and child node 3
- BookSt wants (policy m1) to see Alice’s phone number and if the program and DoB satisfy his constraint. BookSt then creates node4, 5, 6 and adds them as intersection children to node 3
- BookSt adds policy target (node 7) as the policy child to node 6. BookSt adds then a policy expansion child (node 8) to node 7. Similarly, BookSt adds node 9 and 10 => BookSt wants to see Alice’s DoB and driver license or passport.

⇒ Now the BookSt cannot process the TTG anymore

Eaxmple

Alice (grey nodes):

1) Disclose credential n1 and adds credential node 11
2) Alice cant disclose her student credential (n2) immediately, since there is an AC policy (p1) for n2
3) Alice adds policy target node 12 and expands it with a role target node 13
4) Attribute control edge between nodes 11 and 12. Means Alice can disclose her student credential if node 12 is satisfied
5) Alice reveals her digital driver license to BookSt (without revealing DoB), Alice creates trivial target node 14 and adds a credential edge between node 8 and node 14
6) Now Alice notifies that she need to prove that she is younger than 1/1/1984 and reveal her phone number => Alice adds an attribute goal node 15 for her DoB and another goal for the phone number
7) Alice also expands the TTG by addin node 16, 17, 18 and 20
8) Alice proves that she is born in 1986 (as node 16 is trivially satisfied and DoB follows up from node 8 to 3))

⇒ BookSt shows Business Certificate that triggers satisfaction in nodes 12 and 20

⇒ Alice then reveals her student credential (n2) and her certified phone number

⇒ The values follow up to node 3, where BookSt verifies than Alice’s attributes satisfy the constraints

⇒ Finally, The *primary target (node 1)* is satisfied and the negotiation is over

Thank You!