Safety in Automated Trust Negotiation

William H. Winsborough
Center for Secure Information Systems
George Mason University

Joint work with:
Ninghui Li, Purdue University

Attribute-based Access Control

- The Big Goal
  - Flexible and scalable access control for decentralized, collaborative environments and open systems
- The Approach
  - Authorization decision is based on attributes of requester
  - Credentials carry cryptographically signed statements about a principal's attributes & rules for deriving them
  - Requestor and provider may be strangers
  - Automated Trust Negotiation protects sensitive attributes

Protecting Sensitive Attributes While Using Credentials for Authorization

- Goal of Automated Trust Negotiation (ATN)
  - Provide information about sensitive attributes only to authorized entities
- Approach
  - Credentials are potentially protected resources
  - Bilateral exchange of attribute credentials
  - Establish mutual trust incrementally

Eager Strategy

- Negotiators take turns sending all unlocked credentials
- If policy governing requested resource is satisfied, negotiation succeeds
- Else, when no more credentials flow, fails
- Results
  - Completeness
  - Privacy (Correctness)
  - Efficiency
- [Winsborough, Seamons, and Jones. DISCEX 2000]

Subsequent ATN Strategy Designs

- Parsimoneous Strategy: a linear strategy with focused disclosures
  - [Winsborough, Seamons, and Jones. DISCEX 2000]
- Prunes: a quadratic backtracking strategy
  - [Yu, Ma, and Winslett. CCS 2000]
- Policy graphs: protecting policy content as a sensitive resource
  - [Seamons, Winslett, and Yu. NDSS 2001]
- Interoperable strategies: closed strategy families
  - [Yu, Winslett, and Seamons. CCS 2001]
  - [Yu, Winslett, and Seamons. TICSEC 2003]
- Trust Target Graph (TTG): Integrating trust management, credential discovery, privacy for sensitive attributes into ATN
  - [Winsborough and Li. Policy 2002]
  - [Winsborough and Li. WPES 2002]
- UniPro: protecting policy content
  - [Yu and Winslett. Oakland 2003]

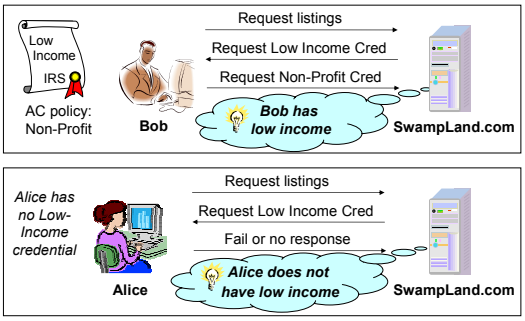
Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

Protecting Sensitive Credentials

- Prior notion of "Safety" is inadequate:
  - Def: a credential's access control (AC) policy must be satisfied before the credential is disclosed
  - Issue: what does "disclose" mean?
- Most prior ATN strategies do not adequately protect information in credentials
  - Negotiator's behavior depends on the credentials he has, no matter who he is negotiating with
  - Arises in strategies that share policy information in an effort to avoid unnecessary credential flow

AC Provides Inadequate Safety



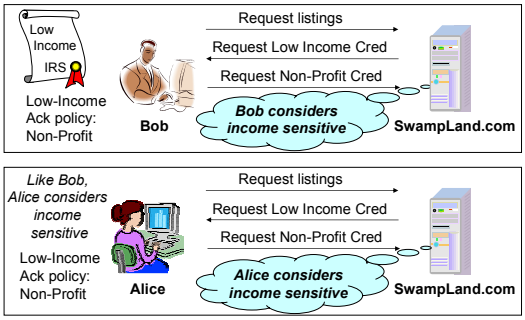
Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

How to Safely Guide Disclosures?

- AC policies are associated with credentials
- Introduce acknowledgement (ack) policies
  - Negotiator can associate ack policy with attribute, whether or not he has the attribute
  - If one satisfies an attribute's ack policy, one is authorized to know whether the negotiator has the attribute
  - By providing an ack policy, a negotiator indicates only that the attribute is sensitive

Ack Policy for all Sensitive Attributes



How are Ack Policies Workable?

- Detractors' argument:
  - People with nothing to hide will not bother to use ack policy, casting suspicion on those who do
- But, anyone wishing to hide a sensitive attribute must hide some he does not hold
  - If suitable ack policies were widely available, the simplest approach would be to enforce them all
- Ack policy design should be part of attribute vocabulary design
  - References to attribute include URI of vocabulary
  - So credential request contains pointer to ack policy

Safe Enforcement of Ack Policies

- Credential systems are often inferential
  - Delegation is often modeled as a rule: anyone who has attribute t_1 also has t_2.
  - An adversary that knows this rule can make several kinds of inference
    - **Forward positive:** if M knows N has t_1, M infers N has t_2
    - **Backward negative:** if M knows N does not have t_2, M infers N does not have t_1
- Intuitive goal: Unless N's opponent satisfies the ack policy for t_1, N's negotiation behavior must not depend on whether N has t_1

Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

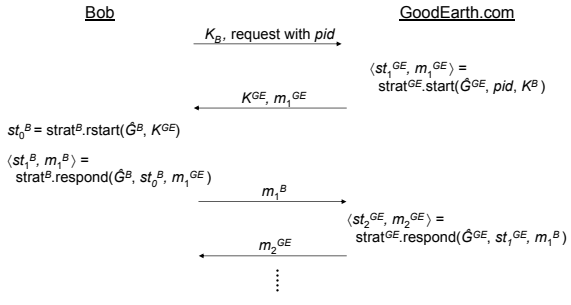
Formal Framework

- Each principal K is identified by a public key
- Each attribute t is identified by an attribute authority (a principal) and an attribute name (a string)
- Each credential e contains a subject K and a set of attributes $T(e)$, which e proves K *possesses*
- Each negotiator has a *configuration* $G = \langle K, E, \text{Policy}, \text{Ack} \rangle$
  - K is the principal (public key) controlled by the negotiator
  - E is a set of credentials
  - Policy associates policy identifiers with positive formulas over attributes
  - Ack associates attributes with policies in Policy

Negotiation Strategies

- Determines structure of:
  - Messages
  - Local state
    - Except always have *success* and *failure*
- Strategy gives four deterministic functions:
  - strat.init(G) returns $\hat{G}$, extended configuration
  - strat.rstart($\hat{G}$, K_O) returns *st*, initial local state
    Used by requester when the opponent is principal K_O
  - strat.start($\hat{G}$, *pid*, K_O) returns $\langle st, msg \rangle$
    Used by access mediator when opponent is K_O
  - strat.respond($\hat{G}$, *st*, *msg*) returns $\langle st', msg' \rangle$
    Used by either negotiator upon receiving *msg* in state *st*

Negotiating with Good Guys



Modeling Adversaries

- An adversary M is given by a set of principals and the credentials available to each principal
- Attack sequences
  - Active attack sequence: $[K_A, pid, a_1, a_2, \ldots a_k]$
    adversary with principal K_A requests resource governed by policy with identifier *pid* and then sends $a_1, a_2, \ldots a_k$
  - Passive attack sequence: $[K_A, a_1, a_2, \ldots a_k]$
    adversary with principal K_A responds to a resource request by sending $a_1, a_2, \ldots a_k$
  - Attack sequence *seq* is *feasible* for M if K_A is controlled by M and the messages can be efficiently computed by M (meaning *seq* is based only on credentials available to M)

Indistinguishability: What the Adversary Can't See

- Two configurations G and $\hat{G}$ are *indistinguishable under* strat *by* M if for every attack sequence seq that is feasible for M, the response sequence induced from G by seq is the same as the one induced from G.
- The response sequence *induced* from G by $[K_A, pid, a_1, a_2, \ldots a_k]$ is the sequence of messages $[m_1, m_2, \ldots, m_\ell]$ satisfying:
  - $\hat{G}$ = strat.init(G)
  - $\langle st_1, m_1 \rangle$ = strat.start($\hat{G}$, pid, K_A)
  - $\langle st_i, m_i \rangle$ = strat.respond($\hat{G}$, st_{i-1}, a_{i-1}), for $2 \le i \le \ell$
  - $st_i \notin \{success, failure\}$, for $1 \le i \le \ell$-1
  - ℓ = k+1 or $1 \le \ell \le$ k and $st_\ell \notin \{success, failure\}$

What the Adversary Shouldn't See

- Unacknowledgeable Attributes: UnAcks(G,M)
  - Given configuration G and adversary M, an attribute t is *acknowledgeable* to M if some principal controlled by M possesses attributes that satisfy $Ack_G[t]$.
- Releasable Credentials: releasable(E, U)
  - Given a set of credentials E and a set of unacknowledgeable attributes U, the *releasable credentials* are those that define no unacknowledgeable attributes: releasable(E, U) = $\{e \in E \mid T(e) \cap U = \emptyset\}$

Credential-Combination Hiding

- A strategy strat is *credential-combination-hiding* safe if for every pair of configurations $G = \langle K, E, \text{Policy}, \text{Ack} \rangle$ and $G' = \langle K, E', \text{Policy}, \text{Ack} \rangle$ and every adversary M, if releasable(E, UnAcks(G,M)) = releasable(E', UnAcks(G',M)) then G and G' are indistinguishable under strat by M

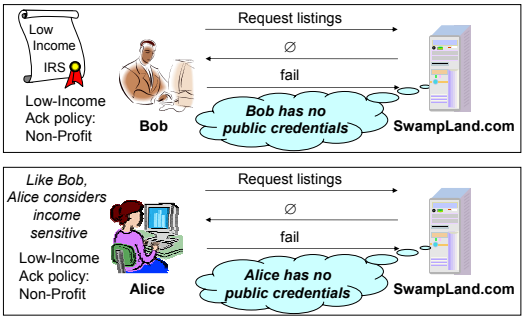
Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

The Definition is Usable

- Theorem: The eager strategy is credential-combination-hiding safe

Eager Strategy is C-C-H Safe



Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

Motivations for TTG Work

- Support a trust management policy language suited to collaborative environments and open systems
- Discover distributed credential chains
- Protect sensitive attribute information
  - Protocols, procedures, and strategies for ATN
  - Safety results

Policy Language Requirements

- Clear, monotonic semantics
- Decentralized attribute authority
- Delegation of attribute authority:
  - To specific entities,
  - To entities with certain attributes
- Inference of attributes
- Intersection

Role-based Trust Management (*RT*)

- A family of credential / policy languages
  - Simplest, RT_0, satisfies these requirements
- RT_0 example: ReliefNet
  - MedixFund.purchasingA ← Alice
  - ReliefNet.provisioner ← MedixFund.purchasingA
  - MedSup.discount ← ReliefNet.provisioner

Implications for ATN

- Negotiators must discover and collect distributed credential chains
- The potential for inference of attributes makes protection of attributes tricky

Distributed Credential Chain Discovery

- Distributed credential collection techniques
  - Chain discovery algorithm
  - Credential type system that ensures chains of distributed credentials can be located
- Paper
  - Distributed Credential Chain Discovery in Trust Management. Li, Winsborough, and Mitchell. *Journal of Computer Security*, February 2003

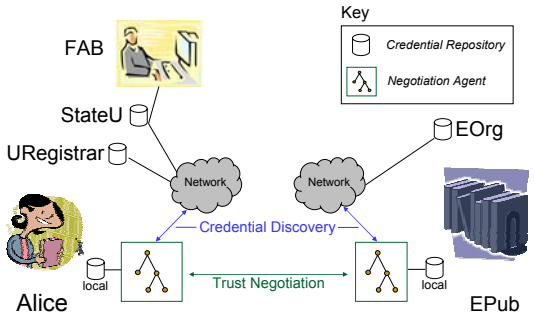
Example: Student ACM Discount

- EPub.studentACM ← EOrg.student ∩ ACM.member
- EOrg.student ← EOrg.university.student
- EOrg.university ← FAB.accredited
- FAB.accredited ← StateU
- StateU.student ← URegistrar.fulltimeLoad
- StateU.student ← URegistrar.parttimeLoad
- URegistrar.parttimeLoad ← Alice
- ACM.member ← Alice

Credential Discovered by EPub in Backwards Direction

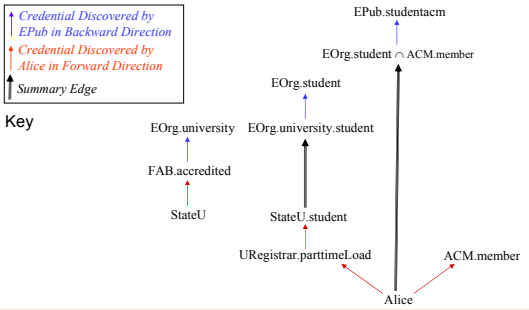
Credential Discovered by Alice in Forwards Direction

Credential Discovery



Credential Graph Organizes Discovery

EPub gives a double subscription discount



Storage Type System

- Storage type of role name determines where credential is stored: with issuer or with subject
- Well-typing ensures credentials are stored where they can be found by tracing the credential graph
- In ATN, types determine which negotiator discovers and provides which credentials during ATN

Credentials	Attribute Name	Type	Credential Stored by
MedSup.discount 1) ReliefNet.provisioner	discount	backward-traceable	MedSup
2) MedixFund.purchasingA	provisioner	forward-traceable	MedixFund
3) Alice	purchasingA	forward-traceable	Alice

Trust Target Graph Strategies

- Outline
  - Trust Target Graph (TTG) negotiation protocol
  - Negotiation procedure: enforcing acknowledgement policies
  - Safety result
- Papers
  - Towards Practical Automated Trust Negotiation. William H. Winsborough and Ninghui Li. *IEEE 3rd Intl. Workshop on Policies for Distributed Systems and Networks*, June 2002
  - Protecting Sensitive Attributes in Automated Trust Negotiation. William H. Winsborough and Ninghui Li. *Workshop on Privacy in the Electronic Society*, Nov. 2002

Alice's Policies

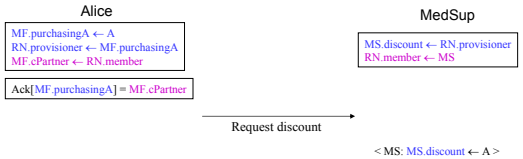
- Ack[MedixFund.purchasingA] = MedixFund.cPartner

Trust Target Graph Protocol

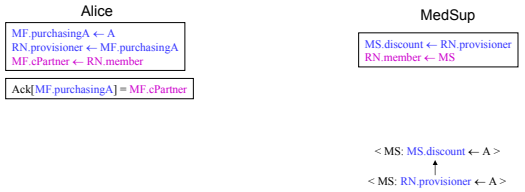
- Protocol uses TTG to represent negotiation state
  - Nodes are (unique) trust targets:
    - < MedSup: MedSup.discount ← Alice >
    - < Alice: MedixFund.cPartner ← MedSup >
  - Edges represent implication, control, etc.
  - Each negotiator keeps a local copy of TTG
  - Negotiators take turns extending the TTG
  - Each transmits edges added during current round
  - Also transmits credentials that justify implication edges

Protected Resource is Requested

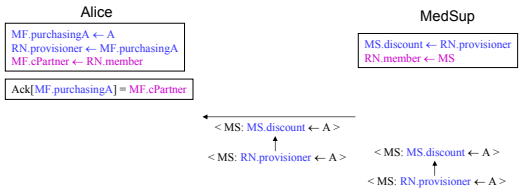
MedSup Initializes Local TTG



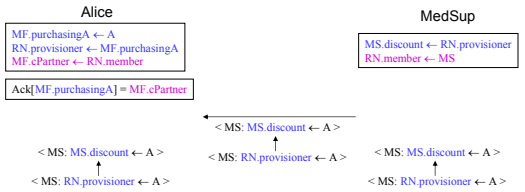
MedSup Extends TTG



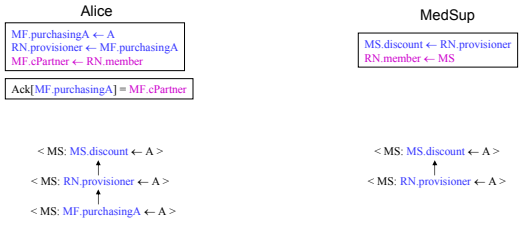
MedSup Transmits TTG



Alice Initializes Her Copy of TTG

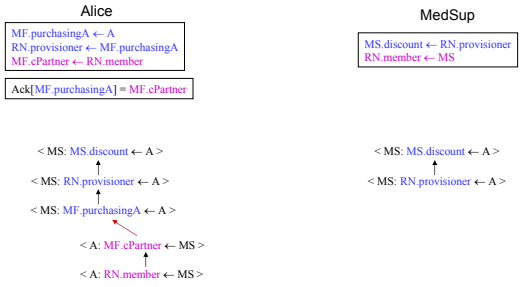


Alice Extends TTG

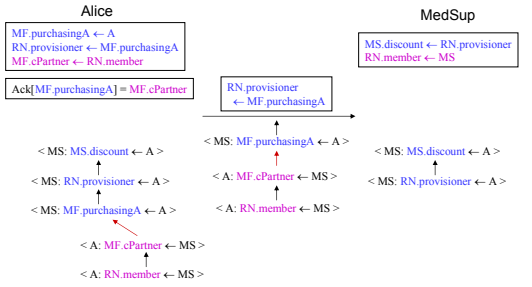


Alice Extends TTG

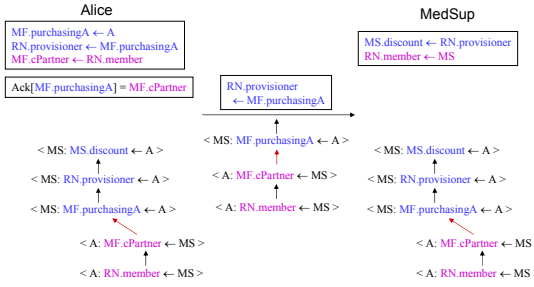
Alice Extends TTG



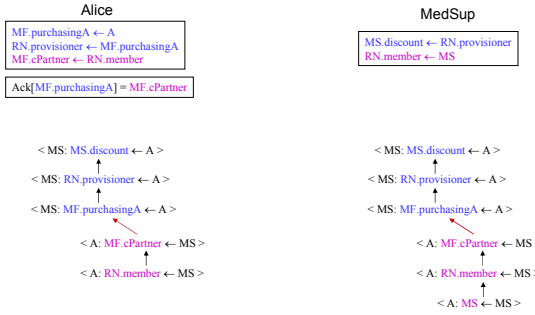
Alice Transmits Changes and Creds



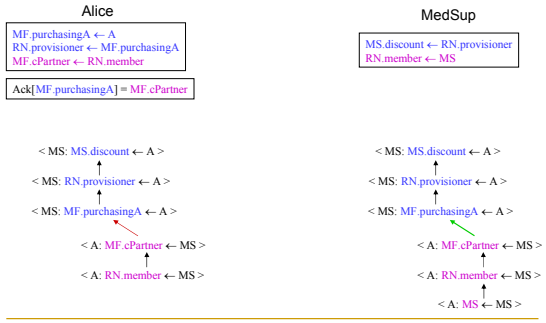
MedSup Updates Local TTG



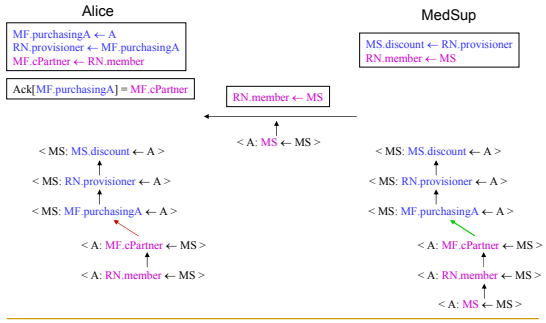
MedSup Proves TT



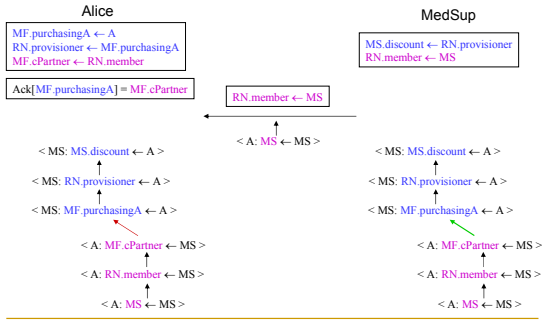
Satisfaction Propagates



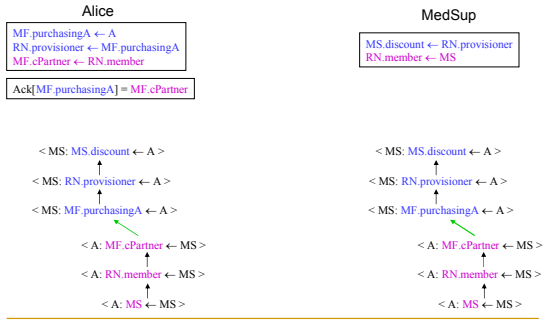
MedSup Transmits Proof



Alice Checks Proof

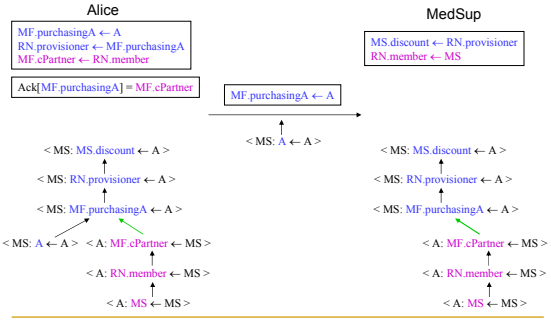


Alice Unlocks Sensitive TT

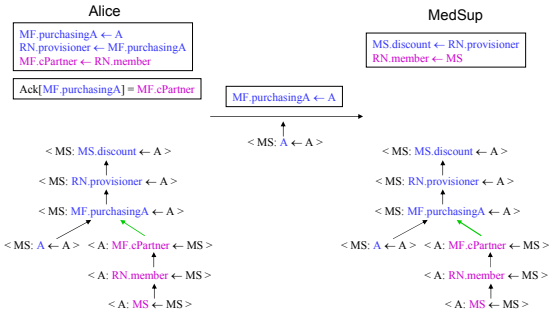


Alice Proves Original TT

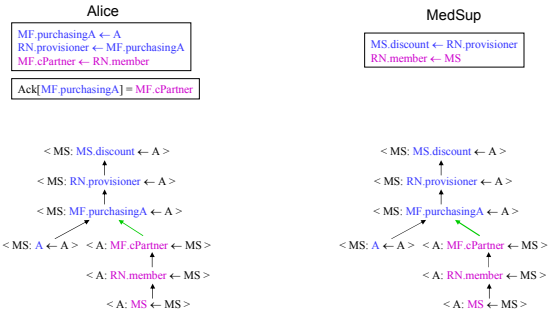
Alice Transmits Proof



MedSup Checks Proof



Primary TT is Now Proven

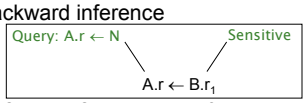


TTG Satisfaction and Failure

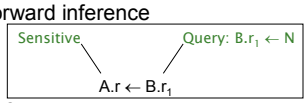
- Satisfaction propagates up from "trivial" TTs
  - Unlocks sensitive attributes
  - Negotiation succeeds when root is satisfied
- Failure propagates up from dead ends
- Negotiation fails when failure reaches root or TTG cannot be extended
  - Latter can happen if there is a cyclic dependence

Enforcing Ack Policy

- Defending against deductive breach of Ack policy
  - Backward inference



    - Solution: Step-by-step, refutation-like TTG search procedure
  - Forward inference



    - Solution: impose closure property on Ack policy making Ack policy for $B.r_1$ as strong as Ack policy for A.r

The Definition of C-C-H is Usable

- Theorem: The TTG strategy is credential-combination-hiding safe

Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

A Weaker Notion of Safety

- A strategy violates C-C-H safety if there are *G*, *G'*, and *M* such that the releasable credentials of *G* and *G'* are the same, yet *M* can distinguish *G* and *G'*.
- Thus *M* can infer that the unreleasable credentials held by the negotiator are not those of *G'*
- Yet, *M* still may be unable to rule out the negotiator's having any combination of unacknowledgeable attributes
- Example:
  - Suppose low-income can be proven by either of two credentials
  - A strategy violating C-C hiding may enable *M* to rule out one of these credentials without *M* being able to infer the negotiator does not have low-income

Attribute-Combination Hiding

- A strategy strat is *attribute-combination-hiding* safe if
  for every configuration $G = \langle K, E, \text{Policy}, \text{Ack}\rangle$,
  for every set of attributes *U*, and
  every expressible subset *U'* of *U*,
  there exists a configuration $G' = \langle K, E', \text{Policy}, \text{Ack}\rangle$ such that
  - *E'* induces every attribute in *U'*, but no attribute in *U* - *U'*, and
  - For every adversary *M* such that $\text{UnAcks}(G,M) \supseteq U$, *G* and *G'* are indistinguishable under strat by *M*
- *U'* is an *expressible subset* of *U* if there is a (hypothetical) set of credentials E_0 such that $T(E_0) \cap U = U'$

A Still Weaker Notion of Safety

- A strategy violates A-C-H safety if there is a *G*, a *U,* and a $U' \subseteq U$ such that there is a credential set *E'* that agrees with *U'* on *U* (*i.e.* $T(E') \cap U = U'$), and every such *E'* is distinguishable from E_G by some adversary *M* with $\text{UnAcks}(G, M) \supseteq U$
- Thus, possibly by colluding, adversaries can determine that $T(E_G) \cap U \neq U'$, thereby ruling out *U'* as a candidate for the combination of unacknowledgeable attributes held by *N*
- Still, *M* may be unable to determine whether *N* holds individual unacknowledgeable attributes

Attribute-Hiding Safety

- A strategy strat is *attribute-hiding* safe if
  for every configuration $G = \langle K, E, \text{Policy}, \text{Ack}\rangle$,
  and every attribute *t*, there exists a configuration $G' = \langle K, E', \text{Policy}, \text{Ack}\rangle$ that differs from *G* in *t* and,
  for every adversary *M*, if *t* is in $\text{UnAcks}(G,M)$, *G* and *G'* are indistinguishable under strat by *M*

Relative Strength of Definitions

- Theorem:
  - If strat is credential-combination-hiding safe, then it is attribute-combination-hiding safe
  - If strat is attribute-combination-hiding safe, then it is attribute-hiding safe

Why Attribute-Hiding is Insufficient

- Does not preclude *M* inferring that *N* does not have a certain combination of attributes
- Example: *M* might infer *N* has either a CIA credential or an NSA credential. This is A-H safe as long as *M* cannot tell which one *N* has
- This is prevented by attribute-combination-hiding

Why Attribute-Combination-Hiding is Insufficient

- Probabilistic inferencing
  - Negotiation should not enable an adversary to improve his estimation of the probability that *N* has any given attribute combination in *U*
- Example
  - Suppose several configurations each induce a given set of unacknowledegable attributes *U'* and that all but one of them are distinguishable from *G*. If the one is very rare, *M* can infer it is unlikely that *N's* unacknowledgeable attributes are exactly *U'*
  - For instance, *M* may be able to infer that *N probably* has either a CIA credential or an NSA credential

Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

Probabilistic Indistinguishability

- Suppose strategies define functions whose output is not deterministic, but probabilistic
- *G* and *G' are probabilistically indistinguishable under* strat *by M* if for every attach sequence seq that is feasible for *M*, the probability distribution over response sequences induced by seq from the two configurations is the same
- *Statistical indistinguishability* allows the distribution of induced response sequences to differ by an amount that is statistically insignificant without a very large sample
- Corresponding versions of safety are induced

Talk Outline

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

Safety of Access Control Enforcement

- Want a definition that is adequate for strategies that do not simply transmit credentials, but use credential signatures to compute messages
- A strategy is *AC-safe* if for every *G*, every *M*, and every attack sequence seq that is feasible for *M*, the response sequence induced from *G* by seq can be efficiently computed without credentials whose AC policy is not satisfied by *M*

Talk Outline (Summary)

- Problem:
  - Original notion of correctness ("safety") for ATN does not achieve goal of protecting sensitive credentials
- Background:
  - An alternative approach sought to protect attributes, but had no formal safety requirement
- Contributions:
  - Formalization of an intuitive safety requirement for protecting attributes
  - Notion is usable: satisfied by the *eager* strategy
  - Notion is usable: satisfied by the *TTG* strategy
  - Formal comparison with two intuitive alternative requirements, that are, in the end, less satisfactory
  - Extension of safety definition to accommodate probabilistic negotiation strategies
  - Formalization of an adequate safety requirement for protecting signed credentials

Paper

- Except for the TTG material, this talk is based on the following paper:
  - Safety in Automated Trust Negotiation. W. Winsborough and N. Li. To appear in: *IEEE Symposium on Security and Privacy*. Oakland, CA. May, 2004.