Safety in Automated Trust Negotiation

William H. Winsborough
Center for Secure Information Systems
George Mason University
Fairfax, VA 22030-4444
wwinsborough@acm.org

Ninghui Li
Department of Computer Sciences and CERIAS
Purdue University
656 Oval Drive
West Lafayette, IN 47907-2086
ninghui@cs.purdue.edu

Abstract

Exchange of attribute credentials is a means to establish mutual trust between strangers wishing to share resources or conduct business transactions. Automated Trust Negotiation (ATN) is an approach to regulate the exchange of sensitive information during this process. It treats credentials as potentially sensitive resources, access to which is under policy control. Negotiations that correctly enforce policies have been called "safe" in the literature. Prior work on ATN lacks an adequate definition of this safety notion. In large part, this is because fundamental questions such as "what needs to be protected in ATN?" and "what are the security requirements?" are not adequately answered. As a result, many prior methods of ATN have serious security holes. We introduce a formal framework for ATN in which we give precise, usable, and intuitive definitions of correct enforcement of policies in ATN. We argue that our chief safety notion captures intuitive security goals under both possibilistic and probabilistic analysis. We give precise comparisons of this notion with two alternative safety notions that may seem intuitive, but that are seen to be inadequate under closer inspection. We prove that an approach to ATN from the literature meets the requirements set forth in the preferred safety definition, thus validating the safety of that approach, as well as the usability of the definition.

1. Introduction

In Attribute-Based Access Control (ABAC) systems, access control decisions are based on attributes of requesters. These attributes are often documented by digitally signed credentials. A principal proves that it has an attribute by showing an appropriate set of relevant credentials. Because attributes (such as financial or medical status) may be sensitive, they need protection just as other resources do. The goal of a growing body of work on *automated trust negotiation (ATN)* [7, 14, 15, 19, 20, 22, 23, 24, 26, 27] is to enable resource requesters and access mediators to establish trust in one another through cautious, iterative, bilateral disclosure of credentials. The distinguishing characteristic of ATN that differentiates it from most other trust establishment schemes (*e.g.*, [3, 6]) is that credentials themselves are treated as protected resources.

Prior work on ATN lacks an adequate notion of security. Fundamental questions such as "what needs to be protected in ATN?" and "what are the security requirements?" are not adequately answered. The main purpose of this paper is to answer some of these questions by providing a formal ATN framework with precise and appropriate security definitions. Let us illustrate the deficiencies of security definitions in existing ATN work.

In most ATN frameworks, each negotiator establishes access control (AC) policies to regulate the disclosure of credentials to negotiation opponents. A typical description in the literature of the safety requirement for AC-policy-based ATN is the one given in [27]: "given a sequence $\mathcal{Q} = \{C_1, \ldots, C_n\}$ of disclosures of protected resources, if each C_i is unlocked at the time it is disclosed, then we say $\mathcal{Q}$ is a *safe disclosure sequence*." Here, *unlocked* means that the AC policy for the credential is satisfied by credentials disclosed previously by the other party, and a credential is one kind of resource. This deceptively simple requirement turns out to be far from adequate in ensuring that an ATN system protects the privacy of sensitive attributes. Several groups of researchers have noted [15, 20, 25] that although early ATN designs satisfy the safety requirement for AC policies, they nonetheless fail to adequately protect the privacy of negotiators. So it is recognized that a problem exists with ATN's traditional notion of safety. The problem stems from the fact that the traditional notion is satisfied by ATN designs in which, although a sensitive credential itself is not transmitted until its associated AC policy is satisfied, the behavior of a negotiator reveals a great deal about the contents of these credentials. Indeed, most ATN designs

do just that. When a negotiator is asked to prove a sensitive attribute, the negotiator's behavior depends on whether it has the attribute or not. By observing the negotiator's behavior, the negotiator's opponent can infer whether the negotiator has a sensitive attribute or not. Thus, while the negotiator's opponent may not yet have proof of the authenticity of the attribute, the privacy of the attribute has certainly been compromised. In [15], some ad hoc solutions are proposed. For example, it is suggested that instead of transmitting the AC policy, a negotiator having a sensitive attribute could simply behave as though he did not, and just wait, hoping the opponent will happen to send enough credentials to satisfy the AC policy.

Trust negotiation is of little value if participants must mislead one another to protect sensitive information, since this would make many negotiations fail unnecessarily. Yet most prior negotiation techniques allow a negotiator's opponent to gain advantage just in case the negotiator is honest. As we show in this paper, one of the few existing ATN strategies that is immune from this problem is the eager strategy [22]. In it, each party transmits all credentials whose access control policies have already been satisfied, whether these credentials are related to the eventual negotiation goal or not. In the eager strategy, when a negotiator does not receive a given credential from the opponent, it does not know whether this is because the opponent does not have the credential, or because the negotiator simply has not satisfied the opponent's AC policy for that credential.

In [20], an approach was proposed for focusing the credential exchange while simultaneously protecting sensitive attributes of negotiators. The approach is based on the notion of an *acknowledgement policy* ("ack" policy, for short). An ack policy resembles an AC policy, though it is associated with an attribute, rather than with a credential proving the attribute. The key difference from AC policy is that one can associate an ack policy with an attribute one does not have. This makes it possible to provide the ack policy without in doing so indicating whether one satisfies the associated attribute. The intuitive goal of ack policies is that no one should learn through negotiation whether or not a negotiator N possesses an attribute without first satisfying its ack policy. This intuitive notion of safe enforcement of ack policies was not formalized in previous work using the concept [19, 20]. Therefore, it was impossible to prove that a given strategy using Ack policies is safe.

The present goal is to articulate a suitable definition of this notion that is precise, usable, and intuitive. The definition should be precise and usable so that one can prove security of negotiation strategies using the definition. The definition should also be intuitive in that ATN systems satisfying it should fulfill our expectations that sensitive attributes of the negotiator be protected from unintended disclosure. This goal is in keeping with the research tradition in information security and cryptography of finding security definitions for numerous problems and protocols that are suitably precise, usable, and intuitive.

The approach we take in this paper is to formalize the following intuition about safe enforcement of ack policies: unless N's negotiation opponent satisfies the ack policy for a sensitive attribute, N's behavior in the negotiation must give no indication of whether N possesses any credentials relating the sensitive attribute. As we will see, the details of the safety condition are somewhat intricate, and simply preventing an adversary from determining specific attributes is inadequate.

Accommodating diverse credential systems requires effort. In particular, we seek a notion of safety that can be supported by systems in which credentials can represent delegations of authority. Such credential systems support a limited form of deduction, which means we must prevent security being breached through deductive inference. The threat of probabilistic inference also influences the selection of an appropriate safety condition.

Let us outline our safety definition. We first formalize the ability of an adversary to distinguish between one negotiator and another. For each negotiator N and each adversary M, there is a set U of attributes whose ack policies are not satisfied by M. We define a strategy to be safe if any other negotiator N' who differs from N only in credentials that prove attributes in U is indistinguishable from N by M based on ATN. We discuss other definitions that capture similar, but different intuitions about safety, showing they are strictly weaker and inappropriate in various respects. We also discuss adequate safety definitions for access control policies.

The contributions of this paper are as follows:

1. A formal framework for trust negotiation and a precise definition of safety for enforcement of ack policies in that framework.
2. Proving that the eager strategy is safe based on this formal definition.
3. A formal analysis of the relationship between our safety definition and two alternative definitions that also seem intuitive.
4. An analysis that shows why our first safety definition is preferable to the two alternatives mentioned above.
5. Extensions to our (possibilistic) safety definition that handles probabilistic negotiation strategies.
6. A precise definition of safety for AC policies that can be used with cryptographic ATN protocols.

The rest of this paper is organized as follows. In Section 2, we discuss in details why previous notions of safety are inadequate. Section 3 is the heart of the paper, presenting contributions 1-5 from the list above. In Section 4, we discuss deficiencies of previous safety definition for AC

policies and give our definition. We discuss related work in Section 5 and conclude in Section 6. The appendix contains proofs of theorems.

2. Prior Unsatisfactory Notions of Safety

Most existing negotiation strategies are safe according to the limited definitions laid out for them by their designers. However, as we show in this section, they are not safe in the sense of protecting the content of credentials, which is arguably the central goal of ATN: if credential content did not need protection, requesters could simply push all their credentials to the access mediator for evaluation. This leads us to the inevitable conclusion that the definition of safety set forth in prior work is inadequate, thereby motivating our introduction of adequate definitions in Section 3.

In the ATN literature, access control (AC) policies are associated with credentials as well as with resources. Let us consider an example in which Bob obtains a credential from the Internal Revenue Service (IRS) documenting his low-income status. Such a credential might be useful, for example, with a nonprofit organization that offers a service preparing free living wills over the Internet for people with low incomes. Suppose Bob uses an AC policy recommended by the IRS for protecting this credential, which says that Bob will show his IRS.lowIncome credential to organizations that document they are registered with the IRS as nonprofits. Bob can use his ATN-enabled browser to contact an ATN-enabled service provided by a nonprofit to obtain a living will, and Bob's browser and the service's access mediator will negotiate successfully.

Now suppose another Web user, Alice, does not have a low income-status credential. Alice and Bob each visit the web site of an unfamiliar real estate service, SwampLand.com. When Alice and Bob each request information about listed properties, the SwampLand access mediator initiates a negotiation requesting Alice and Bob prove they have low-income status, which is not an appropriate requirement. If Alice and Bob use a typical ATN strategy, such as the TrustBuilder1-Relevant Strategy [27], this request induces Bob to present his AC policy for his low-income credential. The same request causes the negotiation with Alice to fail, since Alice does not have the requested credential. SwampLand.com can easily observe the difference between these two behaviors, and deduce Bob's low-income status, even though Bob's AC policy indicates he does not want to share that information with for-profit companies. Granted, SwampLand.com does not obtain proof that Bob is low-income. However, this should provide Bob little comfort in using the ATN strategy, as SwampLand.com's unauthorized inferences are accurate just in case he adheres to the protocols faithfully. Similarly, SwampLand.com can deduce that Alice does not have the credential, though Alice may also not wish this either.

This unsafe behavior, which characterizes most ATN strategies, occurs because the strategies transmit AC policies, or information derived from them, in an effort to focus exchanges on credentials that are relevant to enabling the negotiation to succeed. This focus aims to reduce message size and other resource utilization, as well as to avoid distributing sensitive information needlessly. Assuming ATN strategy should not fail when success is possible, the competing goal of protecting sensitive attributes and this goal of focused disclosure seem to be at odds with one another. This is because of the nature of AC policies, namely that they are associated only with attributes that the negotiator satisfies. An alternative is to introduce a form of policy that can be associated with an attribute whether or not the negotiator satisfies it. In keeping with [19, 20], we call such policies *ack* policies. We now use a simple example to show how ack policies can be used to stop information leakage.

Example 1 Bob has a credential from the IRS documenting his low-income status, and adopts the IRS's recommended ack policy that he will discuss the matter of low-income status only with nonprofit organizations registered with the IRS. Alice does not have the low-income status credential, but also considers information about her income status sensitive, so she also adopts the same ack policy.

When Alice and Bob each visit the web site of SwampLand.com, and both are asked to prove they have low-income status, both Alice and Bob will ask SwampLand.com to prove that it is nonprofit first. Therefore, SwampLand.com only learns that both Alice and Bob considers their income status sensitive, but not whether they are low-income or not.

It has been argued [25] that the use of ack policies is unworkable because people who feel they have nothing to hide with respect to a given attribute will not bother to use ack policies for those attributes, casting suspicion on those who do. However, anybody wishing to protect some of his own sensitive attributes by using ack policies needs to enforce ack policies on some attributes about which he has nothing to hide. Otherwise the fact that he protects the attribute probably indicates he holds it. Now, given that he needs to protect some non-held, sensitive attributes, if there were a straightforward mechanism for obtaining suitable ack policies for all such attributes, a negotiator would have little incentive not to enforce them uniformly. After all, if exceptions were to be made, they would have to be specified. If appropriate ack policies were widely available, the simplest course of action for the negotiator would be to always apply them. We argue that defining appropriate ack policies and making them available should be part of attribute vocabulary design. Using some mix of natural and formal language, the vocabulary designer is expected to explain the at-

tributes he names. Characterizing the appropriate recipients of the named information can be viewed as part of that explanation. By using the designer-recommended policy, the negotiator obtains not only convenience, but uniformity in his behavior with respect to that of other negotiators.

Unintended collisions among attribute names must be avoided; one convenient solution to this problem can also be used to solve the problem of publishing ack policies. Name collisions can be prevented by making each attribute name include a reference, such as a URL, to a document describing an attribute vocabulary of which it is a part. Names containing different vocabulary references cannot collide. Using this scheme, when a policy requests that a negotiator prove a certain attribute, the policy also provides a reference to an ack policy recommended by the premier expert in the meaning of the attribute, which the negotiator can then use in his response.

Although ack policies have been previously introduced, and were shown in an informal manner to protect sensitive information, precise security definitions for them have not been introduced. Therefore, one cannot prove that a strategy using ack policies is safe. In fact, defining safety for ack policies is quite tricky. One difficulty comes from the fact that credentials may contain rules for deriving principals' attributes. Such rules are necessary in credential systems that express delegation of authority, as is essential in decentralized environments and is common in most access control languages designed for distributed access control. When credentials may contain delegations, having one attribute may imply having another attribute. Suppose for instance that a credential asserts that anyone who has attribute t_1 also has attribute t_2. Then, the following two kinds of inference can be made.

- **forward positive inference:** If the opponent M knows that N has attribute t_1, then M infers that N also has attribute t_2 (*i.e.*, modus ponens).
- **backward negative inference:** If the opponent M knows that N does not have attribute t_2, then M infers that M does not have t_1 either (*i.e.*, modus tollens).

Furthermore, sometimes the only way of having the attribute t_2 is by having attribute t_1. In that case M can perform the following two kinds of inference as well.

- **backward positive inference:** If M knows that N has attribute t_2, then M infers that N also has attribute t_1.
- **forward negative inference:** If M knows that N does not have attribute t_1, then M infers that N does not have attribute t_2 either.

Because of the possibility of these (and maybe other) inferences, it is not obvious what the precise safety requirement for ack policies should be. Although previous work develops techniques to try to defend against these inferences, it is not clear whether these techniques satisfy the intended security requirements, since such requirements have not been defined in a precise way.

3. A Formal Framework for Trust Negotiation

In this section, we present a formal framework for automated trust negotiation and precise definitions for safety in this framework. In Section 3.1, we set up the framework and in Section 3.2 we give the definition of the safety requirement for a negotiation strategy. In Section 3.3, we discuss two alternative safety notions that appeal to different intuitions and show that they are weaker than the definition in Section 3.2. We also present reasons why we ultimately dismiss each of these alternatives as inadequate. We extend our safety notion by giving it a probabilistic interpretation in Section 3.4. In Section 3.5, we discuss applying the framework when the credential system supports delegation and in Section 3.6 we briefly discuss work reported elsewhere that extends a strategy in the literature to obtain a family of probabilistic strategies that satisfy our probabilistic safety notion.

3.1. The Framework

The elements in the setup of the basic ATN model are as follows.

- A countable set $\mathcal{K}$ of principals. Each *principal* is identified with a public key.
- A countable set $\mathcal{T}$ of attributes. Each attribute t is identified by a pair containing an attribute authority (which is a principal) and an attribute name (which is a string over some standard alphabet).
- A universe of potential credentials. Each credential e contains a principal K (called the subject). A credential proves that K has a finite, nonempty set of attributes $T(e)$; we also say that K possesses or holds these attributes. In addition to supporting credentials that explicitly aggregate attributes, the capacity of credentials to entail more than one attribute will be useful when we introduce delegation in Section 3.5. Possession of attributes in $\mathcal{T}$ may be considered sensitive, and the goal is to protect this information.

In this model, a participant in the ATN system is characterized by a finite *configuration* G, which is given by $G = \langle K_G, E_G, \mathsf{Policy}_G, \mathsf{Ack}_G \rangle$. (We drop the subscripts when G is clear from context.)

- K is the principal *controlled by* the participant; this means that the participant has access to the private key that corresponds to K, enabling the participant to prove itself to be the (presumably unique) entity controlling the key.

- E is a set of credentials. Every credential in E has K as the subject. We assume that across all configurations G', the presence in $E_{G'}$ of credentials proving disjoint attribute sets is uncorrelated.
- Policy is a *policy table* that consists of a list of entries. Each *entry* has a unique identifier and a policy, which is a positive propositional logical formula in which the propositions are attributes in $\mathcal{T}$. We say that a policy is satisfied by a principal if the attributes that the principal possesses make the formula true.
- Ack is a partial function mapping some attribute in $\mathcal{T}$ to a policy in Policy. Attributes in the domain of Ack are called sensitive attributes. Technically, Ack is given by a function that associates with each sensitive attribute an identifier in Policy[1]. We write $\mathsf{Ack}[t]$ for the ack policy of the attribute t. Ack can associate an ack policy with an attribute whether K possesses the attribute or not.

For a set of credentials E (all having the same subject), the set of attributes induced by E is $T(E) = \bigcup_{e \in E} T(e)$. Thus, each attribute in $T(E)$ follows from an individual credential $e \in E$. Attributes defined in terms of conjunctions of other attributes cannot be protected in the system we present. Thus, we do not support credentials that enable one to infer one attribute from two or more other attributes, though we can and do allow conjunctions of attribute to be required in policies.

Example 2 Consider the scenario described in Example 1. Bob's configuration is $G_B = \langle K_B, E_B, \mathsf{Policy}_B, \mathsf{Ack}_B \rangle$, in which K_B is Bob's public key, $E_B = \{\text{IRS.lowIncome} \longleftarrow K_B\}$ and $\mathsf{Ack}_B[\text{IRS.lowIncome}] = \text{IRS.nonprofit}$. Alice's configuration is $G_A = \langle K_A, E_A, \mathsf{Policy}_A, \mathsf{Ack}_A \rangle$, in which $E_A = \{\}$ and $\mathsf{Ack}_A[\text{IRS.lowIncome}] = \text{IRS.nonprofit}$.

We assume that before a trust negotiation process starts, the two negotiators have established a secure connection and have authenticated the principals they each control. This enables one to protect sensitive information that one may disclose during the course of the negotiation. One way to achieve this is for the two parties to establish a TLS/SSL connection using self-signed certificates.

A negotiation process starts when one participant (called the *requester*) sends a request to another participant (called the *access mediator*) requesting access to some resource. The access mediator identifies the policy protecting that resource and then starts the negotiation process. The negotiation proceeds by two negotiators exchanging messages. Each negotiator maintains a local state during the negotiation process. The details of the messages and the local states are not mandated in the abstract model described in the current section. However we assume there are two predefined states: $success$, and $failure$. A negotiation process fails when one of the two negotiators enters into the $failure$ state. (The negotiator might send a message notifying the opponent about the failure; we choose not to include such a message in the model here for technical convenience.) A negotiation process succeeds when the access mediator enters into the $success$ state. A negotiation process stops when it succeeds or when it fails.

A negotiation strategy determines the structure of states and messages and what a negotiator does in a negotiation process. More specifically, a *negotiation strategy* consists of the following four deterministic functions:

- The function $\mathsf{strat.init}(G)$ takes a configuration G, and returns an extended configuration $\overline{G}$ whose type remains opaque in the abstract model. This represents an initialization phase the negotiator does before entering into any negotiation process. If no initial processing is needed in a particular system, $\overline{G}$ has the same type as G, and $\mathsf{strat.init}(G)$ returns G unchanged.
- The function $\mathsf{strat.rstart}(\overline{G}, K_O)$ takes an extended configuration $\overline{G}$ and a principal K_O and outputs a state st. This represents the negotiator entering a negotiation process as a requester. This function is called after the negotiator sends the request to its opponent (who uses K_O in the negotiation); the negotiator uses st as its initial local state.
- The function $\mathsf{strat.start}(\overline{G}, pid, K_O)$ takes an extended configuration $\overline{G}$, a policy identifier pid, and a principal K_O and outputs $\langle st, msg \rangle$ (a state and a message). This represents the negotiator entering a negotiation process as an access mediator. This function is called when the negotiator receives a resource request from its opponent (who uses K_O in the negotiation) and has determined that the identifier of the policy protecting the requested resource is pid. The access mediator uses st as its initial local state and, when $st \notin \{success, failure\}$, sends msg to the opponent.
- The function $\mathsf{strat.respond}(\overline{G}, st, msg)$ takes an extended configuration $\overline{G}$, the current state st, and a message msg, and outputs $\langle st', msg' \rangle$. Upon receiving a message msg during the negotiation process, the negotiator calls the respond function, then changes the current state to st' and (when $st' \notin \{success, failure\}$) sends msg' to the other negotiator.

3.2. Safety of Ack-Policy Enforcement

We now define what it means when we say a negotiation strategy is safe. Intuitively, a strategy is safe if the ack poli-

1 The policy table may contain entries that represent policies that are not used in ack policies. These policies may correspond to access control policies for resources that the participant controls and protects. Thus our model does not represent resources explicitly, but we assume a participant can determine the (identifier of the) policy protecting each resource under his control.

cies are correctly enforced when using the strategy. What does it mean to say that a negotiator N's ack policies are correctly enforced? The definition we will present uses the following intuition: no adversary M, using observations it can make in negotiation processes with N, can make any inference about credentials proving the attributes of N it is not entitled to know (*i.e.*, attributes whose ack policies are not satisfied by M).

To make the above intuition precise, we first model the ability of adversaries. An *adversary* is given by a set of principals it controls and a set of credentials for each of the principals. This models the ability of entities controlling different principals to collude. We assume each such set contains all credentials potentially available to the principal for use in trust negotiation. (If an adversary controls a principal that is an attribute authority of an attribute t, then credentials about t are available to the adversary.) We assume that an adversary only interacts with a participant N through trust negotiation. We allow the adversary M to initiate negotiation with N, by sending N a request, as well as to wait for N to initiate a negotiation process by sending a request to M. An adversary is limited by the credentials available to it, which determine the attributes possessed by the principals it controls. We assume that no adversary can efficiently compute credentials not available to it. That is, we assume that it is infeasible to forge signatures without knowing the private keys.

We next formalize the observations an adversary can make about a negotiator's configuration by engaging in negotiation processes. The key notion we need is that an adversary cannot distinguish between two configurations. Intuitively, if an adversary cannot distinguish between two configurations, one of which has an attribute and the other of which does not, then the adversary cannot infer whether the negotiator has the attribute or not. The notion of indistinguishability we give in this section is suitable for deterministic negotiation strategies. We extend our treatment of safety to support nondeterministic negotiation strategies in Section 3.4.

Definition 1 (Indistinguishability) Given an adversary M, a negotiation strategy strat, and two configurations G and G', G and G' are *indistinguishable under* strat *by* M if for every attack sequence seq that is feasible for M, the response sequence induced by seq from G is the same as the response sequence induced by seq from G'.

In the following, we define *feasible attack sequences* and the response sequences they induce. An attack sequence can be either active or passive. An *active attack sequence* has the form $[K_A, pid, a_1, a_2, \ldots, a_k]$, in which K_A is a principal, pid is a policy identifier, and $a_1, a_2, \ldots, a_k$ are messages. This corresponds to the case in which the adversary starts the negotiation by using K_A, a principal it controls, to request access to a resource protected by the policy that has identifier pid, and then sends $a_1, a_2, \ldots, a_k$ one by one in the negotiation. Given a configuration G and a strategy strat, the *response sequence induced by an active attack sequence* $[K_A, pid, a_1, a_2, \ldots, a_k]$ is the sequence of messages: $[m_1, m_2, \ldots, m_\ell]$ that satisfies the following conditions:

1. $\overline{G} = \mathsf{strat.init}(G)$
2. $\langle st_1, m_1 \rangle = \mathsf{strat.start}(\overline{G}, pid, K_A)$
3. $\forall i \in [2, \ell], \langle st_i, m_i \rangle = \mathsf{strat.respond}(\overline{G}, st_{i-1}, a_{i-1})$
4. $\forall i \in [1, \ell - 1], st_i \notin \{success, failure\}$,
5. either $\ell = k + 1$ or $1 \leq \ell \leq k \wedge st_\ell \in \{success, failure\}$ (in the latter case, the negotiation ends before the complete attack sequence is used)

A *passive attack sequence* has the form $[K_A, a_1, a_2, \ldots, a_k]$, in which K_A is a principal and $a_1, a_2, \ldots, a_k$ are messages. This corresponds to the case that the negotiator sends a resource request to the adversary, who responds by sending the messages of the attack sequence. Given a configuration G and a strategy strat, a *response sequence induced by a passive attack sequence* $[K_A, a_1, a_2, \ldots, a_k]$ is the sequence of messages: $[m_1, m_2, \ldots, m_\ell]$ that satisfy the following conditions:

1. $\overline{G} = \mathsf{strat.init}(G)$
2. $st_0 = \mathsf{strat.rstart}(\overline{G}, K_A)$
3. $\forall i \in [1, \ell], \langle st_i, m_i \rangle = \mathsf{respond}(\overline{G}, st_{i-1}, a_i)$
4. $\forall i \in [1, \ell - 1], st_i \notin \{success, failure\}$,
5. either $\ell = k + 1$ or $1 \leq \ell \leq k \wedge st_\ell \in \{success, failure\}$.

Given an adversary M, an attack sequence seq is *feasible* for M if K_A is controlled by M and the messages can be efficiently computed by M. (This formalizes the notion that seq does not contain credentials not available to M.)

Definition 2 (Unacknowledgeable Attribute Set) Given a configuration G and an adversary M, we say that an attribute t is *acknowledgeable* to M if there exists a principal that is controlled by M and that possesses attributes that satisfy $\mathsf{Ack}_G[t]$. We further define $\mathsf{UnAcks}(G, M)$ to be the set of attributes that are not acknowledgeable to M.

Intuitively, ATN should not enable an adversary M to learn any information about $\mathsf{UnAcks}(G, M)$ that M would not otherwise be able to learn. Given such a set of unacknowledgeable attributes, the negotiator's credentials can be divided into those that can be released to M and those that cannot.

Definition 3 (Releasable and Unreleasable Credentials) Given a set of credentials E and a set of unacknowledgeable attributes U, the set of *unreleasable credentials* consists of those that define unacknowledgeable attributes, and is

given by $\mathsf{unreleaseable}(E, U) = \{e \in E \mid T(e) \cap U \neq \emptyset\}$. The remaining elements of E are *releasable credentials*: $\mathsf{releaseable}(E, U) = E - \mathsf{unreleaseable}(E, U) = \{e \in E \mid T(e) \cap U = \emptyset\}$.

Equipped with this terminology, we can now state that if U is the set of attributes that cannot be acknowledged to M and if two negotiators using the same strategy have the same set of releasable credentials with respect to U, then they should behave the same from the point of view of M. Thus we now formalize this intuition in the central definition of the paper, which requires that an ATN strategy hide all information about credentials representing unacknowledgeable attributes.

Definition 4 (Credential-Combination Hiding) A negotiation strategy strat is *credential-combination-hiding safe* if for every pair of configurations $G = \langle K, E, \mathsf{Policy}, \mathsf{Ack}\rangle$ and $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack}\rangle$, and every adversary M, if $\mathsf{releaseable}(E, \mathsf{UnAcks}(M, G)) = \mathsf{releaseable}(E', \mathsf{UnAcks}(M, G))$, then G and G' are indistinguishable under strat by M.

One aspect of Definition 4 that differs from prior notions of safety is that it is concerned only with the attributes M has, and not with the ones M proves in the negotiation. This simplifies matters and is entirely justified because our objective is to ensure that information flow is authorized, not that it is matched by a compensatory flow in the reverse direction.

Next, we discuss the eager strategy and observe that it satisfies Definition 4. A negotiator using the eager strategy sends all credentials as soon as the attributes they define have their ack policies satisfied by credentials received from the opponent. The two negotiators take turns exchanging all credentials that are unlocked, *i.e.*, that define attributes whose ack policies have been satisfied by credentials disclosed previously by the opponent. In the first transmission, one negotiator sends all credentials defining unprotected attributes. The other negotiator then sends all credentials defining unprotected attributes or attributes whose ack policies were satisfied in the first transmission. The negotiators continue exchanging credentials until either the policy governing the desired resource has been satisfied by credentials sent by the requester, in which case the negotiation succeeds, or until a credential exchange occurs in which no new credentials become unlocked, in which case the negotiation fails.

Definition 5 The eager strategy eager uses a state of the form $\langle \mathsf{opCreds}, \mathsf{locCreds}, K_O, pid\rangle$, in which $\mathsf{opCreds}$ and $\mathsf{locCreds}$ are the sets of credentials disclosed thus far by the opponent and the negotiator, respectively. The operations are shown in Figure 1.

Theorem 1 The eager strategy is credential-combination-hiding safe.

$\mathsf{eager.init}(G)$ = return G

$\mathsf{eager.rstart}(G, K_O)$ =
  $\mathsf{startState} = \langle \emptyset, \emptyset, K_O, null\rangle$
  return $\mathsf{startState}$

$\mathsf{eager.start}(G, pid, K_O)$ =
  $\mathsf{publicCreds} = \{e \in E \mid$ each policy in $\mathsf{Ack}_G[T(e)]$ is trivially satisfied$\}$
  $\mathsf{startState} = \langle \emptyset, \mathsf{publicCreds}, K_O, pid\rangle$
  return $\langle \mathsf{startState}, \mathsf{publicCreds}\rangle$

$\mathsf{eager.respond}(G, \langle \mathsf{opCreds}, \mathsf{locCreds}, K_O, pid\rangle, msg)$ =
  $\mathsf{opCreds}_{+1} = \mathsf{opCreds} \cup msg$
  if local negotiator is resource provider (*i.e.*, $pid \neq null$) and $\mathsf{opCreds}_{+1}$ proves K_O satisfies $\mathsf{Policy}(pid)$
    return $\langle success, null\rangle$
  $\mathsf{locCreds}_{+1} = \{e \in E \mid \mathsf{opCreds}$ proves K_O satisfies each policy in $\mathsf{Ack}_G[T(e)]\}$
  $msg_{+1} = \mathsf{locCreds}_{+1} - \mathsf{locCreds}$
  if $msg_{+1} = \emptyset$ return $\langle failure, null\rangle$
  return $\langle\langle \mathsf{opCreds}_{+1}, \mathsf{locCreds}_{+1}, K_O, pid\rangle, msg_{+1}\rangle$

Figure 1. Operations of the eager strategy.

The proof is in Appendix A.

Example 3 If Alice and Bob have the configurations shown in Example 2, and each one negotiates with SwampLand.com, which has no credentials, both negotiations start with SwampLand sending an empty message and then immediately fail, with no further messages flowing. For the sake of illustration, if we assume that $K_A = K_B$, $\mathsf{Policy}_A = \mathsf{Policy}_B$, and $\mathsf{Ack}_A = \mathsf{Ack}_B$, then SwampLand.com obtains no basis on which to distinguish Alice from Bob[2].

It should be acknowledged that the eager strategy does not take advantage of the distinguishing characteristic of ack policies, *viz.*, that they can be defined for attributes the negotiator does not possess, and therefore can be revealed without disclosing whether the negotiator has the attribute. As we discuss further in Section 3.6, we have elsewhere [21] presented and proven safe a strategy that takes advantage of the fact that ack policies can be safely disclosed, enabling the strategy to use them to focus the exchange on relevant credentials.

3.3. Weaker Notions of Safety

In this section we discuss two weaker notions of safety that seem natural to consider, one of which in particular seemed quite appealing to us at first. However, as we explain at the end of this section, it turns out that both

2 In practice, Alice and Bob would not have the same key; the point is that SwampLand.com cannot distinguish someone who has the low-income attribute from someone who does not.

are inadequate. These two alternative notions of safety are strictly weaker than credential-combination hiding; in this section we prove their logical relationship to credential-combination hiding and to one another.

A strategy that violates Definition 4 may not actually enable an adversary to make any inferences about the negotiator's unacknowledgeable attributes. A violation means that there exist configurations G and $G' = \langle K_G, E', \mathsf{Policy}_G, \mathsf{Ack}_G \rangle$ and an adversary M such that the releasable credentials of G and G' are the same, but G and G' can be distinguished by M. This means that M can infer that certain combinations of unreleasable credentials are not candidates for being the exact set held by G; however it does not ensure M can rule out any combination of unacknowledgeable attributes. For example, suppose that the low-income status can also be proved by another credential issued by the IRS, a strategy that violates Definition 4 may enable an adversary to rule out that a negotiator does not have one credential, but it cannot infer that a negotiator does not have the low-income attribute.

Thus, a weaker notion of safety in which we ensure only that M cannot rule out any combination of attributes seems natural to consider. The goal of the following weaker safety notion, which we call attribute-combination hiding, is to preclude negotiation enabling the adversary to make any inferences that certain attribute combinations are impossible. However, when there are interdependencies among attributes, anyone familiar with the credential scheme can rule out certain attribute combinations. For instance, if every credential proving one attribute t_1 also proves another attribute t_2, it is impossible to have t_1 but not t_2. Therefore, the definition only precludes the adversary inferring anything he does not already know.

Definition 6 (Attribute-Combination Hiding) A negotiation strategy strat is *attribute-combination-hiding safe* if for every configuration $G = \langle K, E, \mathsf{Policy}, \mathsf{Ack} \rangle$, for every subset U of $\mathcal{T}$, and for every expressible subset U' of U, there exists a configuration $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack} \rangle$ such that (a) E' induces every attribute in U', but none of the attributes in $U - U'$ (*i.e.*, $T(E') \cap U = U'$) and (b) for every adversary M such that $\mathsf{UnAcks}(G, M) \supseteq U$, G and G' are indistinguishable under strat by M.

Given a set U of attributes, U' is an *expressible subset* of U if there exists a set of credentials E_0 such that $T(E_0) \cap U = U'$. By "exists" here, we mean hypothetically; the credentials in E_0 need never actually have been issued.

Definition 6 says that if N uses strategy strat, then from M's point of view, N could have any expressible combination of attributes in U. If the definition is violated, then there is a configuration G, a set of attributes U, and a $U' \subseteq U$ such that there exists 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 $\mathsf{UnAcks}(G, M) \subseteq U$. In other words, M can determine that $T(E') \cap U \neq U'$, thereby ruling out U' as a candidate for the combination of unacknowledgeable attributes held by N.

The following still weaker notion, which we call attribute hiding, only prevents the adversary learning whether specific attributes are satisfied. It says that an adversary cannot determine through ATN whether or not the negotiator has any given unacknowledgeable attribute.

Definition 7 (Attribute Hiding) A negotiation strategy strat is *attribute-hiding safe* if, for every configuration $G = \langle G, E, \mathsf{Policy}, \mathsf{Ack} \rangle$ and every attribute t, there exists a $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack} \rangle$ that differs from G in t (*i.e.*, G induces t and G' does not, or vice versa) and, for every adversary M, if t in $\mathsf{UnAcks}(G, M)$, G' is indistinguishable from G by M.

A violation of attribute hiding means that some M can use ATN to determine whether or not N satisfies a particular unacknowledgeable attribute, which is clearly something that any reasonable safety definition must preclude. The following theorem verifies that both credential-combination hiding and attribute-combination hiding do so.

Theorem 2 The relative strength of the safety definitions is as follows:

1. If strat is credential-combination-hiding safe, then it is attribute-combination-hiding safe.
2. If strat is attribute-combination-hiding safe, then it is attribute-hiding safe.

The proof is in Appendix B.

Attribute hiding by itself is not sufficient as a safety requirement because it does not preclude the adversary M inferring that N does not have a certain combination of attributes. For example, a strategy could be attribute-hiding safe while enabling the adversary to infer N has either a CIA credential or an NSA credential, so long as M cannot determine which of these is the case. Since even this imprecise information clearly may be damaging, this makes attribute hiding an unacceptable standard for ATN security. This problem is prevented by attribute-combination hiding, illustrating that it is strictly stronger than attribute hiding.

The problem with attribute-combination hiding is revealed when we consider probabilistic inferencing of attributes. Assume that the opponent has some prior knowledge about the probability that each *credential combination* occurs; the opponent can easily infer information about the probability that each *attribute combination* occurs. Given a set U of unacknowledgeable attributes, safety should mean that after any number of negotiations, the opponent has no

basis on which to improve his estimate of the probability that the negotiator has any given attribute combination in U. (This is why we need the assumption that credentials proving disjoint attributes are uncorrelated.) To make this more concrete, suppose that several configurations each induce a given set of unacknowledgeable attributes U' and that all but one of them are distinguishable from the negotiator's actual configuration G. This does not violate the requirement of attribute-combination hiding. However, it does mean that the opponent can rule out many configurations. So, for instance, if the one indistinguishable configuration is very rare, the adversary can learn that N's unacknowledgeable attributes are very unlikely to be exactly U'. In the CIA and NSA credential example above, learning that the negotiator *probably* does not have a certain combination (*e.g.*, none) of the credentials can be detrimental, even if that knowledge is not entirely certain. Credential-combination hiding does not have this problem because all configurations with the same releasable credentials are indistinguishable, so it does not permit the opponent to rule out any of the configurations that induce U'.

3.4. Probabilistic Indistinguishability

In this section we present a natural, probabilistic notion of indistinguishability that leads to an appealing variant of the credential-combination hiding notion of safety. If we relax our assumption that the functions defining a strategy are deterministic, we can no longer apply the notion of indistinguishability that is given in Definition 1. In particular, if the strategy's behavior is probabilistic, we need a new notion, such as the following:

Definition 8 (Probabilistic Indistinguishability) Given an adversary M, a negotiation strategy strat, and two configurations G and G', G and G' are *probabilistically indistinguishable under* strat *by* M if for every attack sequence seq that is feasible for M, the probability distribution of response sequences induced by seq from G is the same as the probability distribution of response sequences induced by seq from G'.

The above definition of probabilistic indistinguishability is information-theoretic. Weaker variants of probabilistic indistinguishability include statistical indistinguishability, which requires the two probability distributions to be sufficiently similar to require a very large sample size to distinguish them.

We can now consider the notion of credential-combination hiding obtained by interpreting indistinguishability in Definition 4 as probabilistic indistinguishability. What we get is a requirement that the probability distribution of response sequences induced by G and G' be indistinguishable provided the two configurations differ only in unreleasable credentials.

3.5. Applying the Model with Delegation

We now discuss the application of our model to credential systems that support forms of delegation common in trust management languages. Delegation credentials enable decentralization of authority over attributes, and support administrative scalability. They are essential to the traditional trust-management approach to authorization [1], where they allow a single attribute, such as an access right, to be delegated from one principal to another. However they can be more general [10, 11], specifying that having attribute t_1 implies having attribute t_2. Here the authority on t_2 is delegating to the authority on t_1 some control over who satisfies t_2.

In terms of the framework given in Section 3.1, when the credential system supports delegation, we capture this by presuming that the credentials directly represented in the model are those that assign attributes directly to principals specified in the credential. These are the only credentials that appear in the configuration of a negotiator. In the environment, there is also a set L of delegation credentials that do not belong to a specific negotiator, since they can be used in many proofs showing various principals have an attribute. In general, a delegation credential $\ell \in L$ asserts that one attribute implies another attribute. So, to handle the inferencing problems raised in Section 2, when there are delegation credentials, $\mathsf{init}(G)$ will return a $\overline{G}$ in which $\mathsf{Ack}_{\overline{G}}$ protects more attributes than does Ack_G.

We make the simplifying assumption that all delegation credentials are available to the negotiator. If we assume only that delegation credentials are available to principals that satisfy the attributes defined in the credentials, negotiators cannot safely protect attributes they do not have. When having t_1 implies having t_2, it is not possible to hide not having t_1 unless one also hides not having t_2, so the negotiator must be aware of the implication. Thus, it appears to be inherent that a negotiator cannot effectively negotiate while protecting all information about an attribute without knowing whether it is at least possibly related to other attributes he may be asked about in the course of the negotiation.

The assumption that delegation credentials are available is typically justified when attributes are characteristics of subjects or roles that they occupy within their organizations. For instance, it is unlikely to be private information that a university delegates to its registrar authority for identifying students. However, when attributes are capabilities to access specific resources, there may be times when delegation of those capabilities are sensitive. If the negotiator does not have access to all delegation credentials, but has an upper bound for the set, he can still negotiate safely. However, if this is done, negotiation may fail in some cases where it would succeed if the negotiator had perfect knowledge of the delegation credentials. For instance, although a negotiator may not know it, it may be that an attribute representing

a given permission can depend on other attributes representing the same permission, but cannot depend on attributes representing something else. Without having this information, safety would require the negotiator to protect all attributes as strongly as it does the permission. Thus it seems that our assumption can be relaxed only at the cost of having some negotiations fail that would otherwise succeed.

3.6. Applying the Model to the Trust-Target-Graph ATN Protocol

In work reported elsewhere [21] we present a family of ATN strategies based on the credential language RT_0 [11, 12], which supports delegation. Credentials, ack policies, and AC policies are all expressed using statements in this language. That work extends the family of trust-target-graph (TTG) strategies [20] to obtain the first family of probabilistic ATN strategies. Unlike with the eager strategy, negotiators using these strategies exchange information about their ack policies so as to focus their credential disclosures on credentials that are relevant to the negotiation. We show [21] that these strategies provide credential-combination hiding with probabilistic indistinguishability. This result supports our contention that Definition 4 with the probabilistic interpretation of indistinguishability is a useful definition of safety for ATN.

4. Safety of Access-Control-Policy Enforcement

In this paper, we use ack policies, but not AC policies for protecting credentials and their attribute-information content. AC policies may be useful to have in a system as well, for instance, if the signed credential is considered more sensitive than its unsigned content. It is straightforward to add AC policies to our formal model of ATN for additional protection of credentials. We now discuss the deficiencies in prior work of the traditional definition of safety for AC policies and present a definition following the spirit of providing meaningful notions of safety.

The existing safety definition of AC policies is inadequate even when not considering the leaking of attribute information. The requirement that "credentials should not flow until AC policies for them are satisfied" is acceptable only for ATN systems of certain kinds, *i.e.*, those that use credentials only by directly transmitting them. It is inadequate for ATN systems where one takes advantage of the fact that credentials are structured objects, *e.g.*, by using the signatures to compute messages in a protocol without transmitting the signatures themselves [9, 8].

There are two parts of the requirement that are imprecise. First it is undefined what it means that a "credential flows." Clearly, sending the exact bit-string of a credential should be viewed as the credential flowing. What if one does not send the exact bit-string, but sends something (presumably derived from the bit-string) that enables everyone to verify that the credential exists? For example, if σ is the signature, then one could send the content (but not the signature) of the credential and $\theta = 2\sigma$; the receiver can recover the signature easily. One may argue that in this case the receiver recovers the complete credential, thus the credential flows. Now consider the case that some value derived from the signature is sent to the opponent, enabling the opponent to verify that the signature exists but not to recover the signature. (Such a value is easily constructed for RSA signatures [13].) Whether this constitutes a flow of a credential is not so clear. This becomes even less clear in the case that one uses a zero-knowledge protocol to convince the opponent that one holds the credential, but the opponent cannot use the communication transcript to convince any other party of this. We believe that a suitable notion of AC-policy enforcement should not permit any of these forms of credential flow to unauthorized recipients. To capture all such forms of credential flow, the precise definition of "a credential does not flow" should be "the same communication transcripts can be generated efficiently without having access to the credential." Note that we do not require such transcripts be generated by negotiators during trust negotiation; we only require that there exists an algorithm that can generate such transcripts efficiently. Since the transcripts can be generated without access to the credential, clearly the credential does not flow. This is similar to the notion of simulations and zero-knowledge proofs used in the cryptography literature.

Another place where the requirement is imprecise is "until AC policies are satisfied." This is related to the discussion above; how is AC policy satisfied? Does one have to see credential bit-strings, or is it sufficient to be convinced that the credential exists? We argue that a straightforward definition is that "credentials do not flow to parties who do not satisfy the corresponding AC policies." The flow of a credential does not violate security so long as the opponent holds the necessary credentials to satisfy the credential's AC policy.

To summarize, the AC safety requirement should be as follows.

Definition 9 (Safety of Access Control Policies) A negotiation strategy is AC-safe if for every configuration G, for every adversary M, and for every feasible attack sequence seq, the response sequence induced from G by seq can be efficiently computed without credentials whose AC policy is not satisfied by M.

The notion of credentials not flowing is formalized here by saying that it is not necessary to have access to the credentials to efficiently play the negotiator's part in the negotiation. Also note that, instead of making requirements on

the order of events, we simply require that to receive credentials governed by an AC policy, an opponent must possess credentials satisfying that AC policy.

5. Related Work

Automated trust negotiation was introduced by Winsborough et al. [22], who presented two negotiation strategies, an eager strategy in which negotiators disclose each credential as soon as its access control policy is satisfied, as well as a "parsimonious" strategy in which negotiators disclose credentials only after exchanging sufficient policy content to ensure that a successful outcome is ensured. The former strategy has the problem that many irrelevant credentials may be disclosed; the latter, that negotiators reveal implicitly, and in an uncontrolled way, which credentials they hold, by transmitting access control policy content for them. The length of negotiations in both strategies is at most linear in the number credentials the two parties hold. Yu et al. [24] introduced the quadratic "prunes" strategy, which requires negotiators to explicitly reveal arbitrary attributes with no protection.

Yu et al. [27] developed families of strategies called disclosure tree protocols that can interoperate in the sense that negotiators can use different strategies within the same family. Seamons et al. [14] and Yu and Winslett [26] studied the problem of protecting contents of policies as well as credentials. These previous works did not address the leaking of sensitive attribute information.

On the aspect of system architecture for trust negotiation, Hess et al. [7] proposed the Trust Negotiation in TLS (TNT) protocol, which is an extension to the SSL/TLS handshake protocol by adding trust negotiation features. Winslett et al. [23] introduced the TrustBuilder architecture for trust negotiation systems.

The problem of leaking attribute information was recognized by Seamons et al. [15] and Winsborough and Li [20]. Winsborough and Li [20, 19] introduced the notion of ack policies to protect this information and studied various inferencing attacks that can be carried out. However, precise notion of security was not provided in this work.

Yu and Winslett [25] have introduced a technique called policy migration that seeks to make it more difficult for the adversary to infer information about a negotiators attributes based on AC policies. In the versions of credential AC policies disclosed during ATN, the technique moves requirements from policies governing credentials defining sensitive attributes to those of other credentials that are also required by the ATN. This approach obscures the information carried in the ATN about the negotiator's sensitive attributes, but it does not hide it entirely. For instance, by observing multiple negotiations, an adversary can observe that the AC policies presented for a given credential are not always the same and then infer that the negotiator has another credential that the adversary has requested. Moreover, the technique can sometimes cause negotiation to fail when success is possible. For these reasons, it seems clear that policy migration is not an adequate solution to the problem.

The notion of credential-combination-hiding is similar to the notion of noninterference [5], which considers a system that has inputs and outputs of different sensitivity levels. A system can be defined as noninterference secure if low-level outputs do not depend upon high-level inputs. The definition for credential-combination-hiding safety says that the behavior the adversary can observe, (i.e., low-level outputs) does not depends on credentials proving unacknowledgeable attributes (i.e., high-level inputs). The notion of attribute-combination-hiding is similar to the notion of nondeducibility [17], which requires that low-level outputs be compatible with arbitrary high-level inputs. Our definitions deal with a system that involves communication between the two parties, and we want to ensure that one party cannot tell the state of another party. Our notions of indistinguishable configurations are also reminiscent of security definitions for cryptographic protocols.

Inference control has received a lot of attention, particularly in the context of multilevel databases [16], statistical databases [4, 18] and, to a lesser extent, in deductive databases [2]. Most of this work focuses on limiting the information that can be deduced from answers to multiple queries. Such schemes require that history information be maintained allowing multiple interactions with the same party to be correlated, which is a very strong assumption in our context of open systems, an assumption that we do not make. As a result, our approach is quite different.

6. Conclusion

Although many ATN schemes have previously been proposed, precise security goals and properties were lacking. In this paper, we have introduced a formal framework for ATN in which we have proposed a precise and intuitive definition of correct enforcement of policies in ATN. We call this safety notion credential-combination hiding, and have argued that it captures the natural security goals desired under both possibilistic and probabilistic analyses. We have stated two alternative, weaker safety notions that seem somewhat intuitive, and identified flaws that make them unacceptable. We have formulated the eager strategy using our framework and shown that it meets the requirements set forth in our safety definition, thus supporting our contention that the framework and safety definition are usable. In the technical-report version of this paper [21], we present a family of probabilistic ATN strategies that support a credential system with delegation. There we show that these strategies provide credential-combination hiding with probabilistic indis-

tinguishability. This result further supports our contention that credential-combination hiding with the probabilistic interpretation of indistinguishability is a useful definition of safety for ATN.

Appendix

A. Proof of Theorem 1

Theorem 1 The eager strategy is credential-combination-hiding safe.

Proof. Consider any pair of configurations $G = \langle K, E, \mathsf{Policy}, \mathsf{Ack}\rangle$ and $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack}\rangle$ such that $\mathsf{releaseable}(E, \mathsf{UnAcks}(M, G)) = \mathsf{releaseable}(E', \mathsf{UnAcks}(M, G))$. For any given active attack sequence, $[K_A, pid, a_1, a_2, \ldots, a_k]$, we show that the response sequence it induces given G, $[m_1, m_2, \ldots, m_\ell]$, is the same as the response sequence it induces given G', $[m'_1, m'_2, \ldots, m'_{\ell'}]$. For this, we use induction on the steps in the eager-strategy construction of the response sequence to show that $\langle st_i, m_i\rangle = \langle st'_i, m'_i\rangle$ for all $i \in [1, \ell]$. Referring to the construction of $\langle st_1, m_1\rangle = \mathsf{eager.start}(G, pid, K_A)$, clearly $\mathsf{publicCreds} \subseteq \mathsf{releaseable}(E, \mathsf{UnAcks}(M, G))$. By our choice of G and G', it follows that in the construction using G', $\mathsf{publicCreds}' = \mathsf{publicCreds}$. (We use primes to indicate values of local variables in the construction using G' and unprimed versions of the variables for the values in the construction using G.) It follows that $\mathsf{startState} = \mathsf{startState}'$, completing the proof in the base case.

Now we assume $\langle st_i, m_i\rangle = \langle st'_i, m'_i\rangle$ for $i \in [1, \ell]$, and show that it holds for $i + 1$. It is easy to see by inspection of $\mathsf{eager.respond}$ that $st_{i+1} = success$ if and only if $st'_{i+1} = success$. Since $\mathsf{opCreds}_i$ consists of credentials held by M, it follows that $\mathsf{locCreds}_{i+1} \subseteq \mathsf{releaseable}(E, \mathsf{UnAcks}(M, G))$. Similarly, $\mathsf{locCreds}'_{i+1} \subseteq \mathsf{releaseable}(E', \mathsf{UnAcks}(M, G'))$. Clearly $\mathsf{UnAcks}(M, G) = \mathsf{UnAcks}(M, G')$, so, since $\mathsf{opCreds}_i = \mathsf{opCreds}'_i$ by induction hypothesis, $\mathsf{locCreds}_{i+1} = \mathsf{locCreds}'_{i+1}$. It now follows easily that $\langle st_{i+1}, m_{i+1}\rangle = \langle st'_{i+1}, m'_{i+1}\rangle$, as required to complete the induction.

Note that it cannot be that $\ell' > \ell$ because either $\ell = k+1$ or $st'_\ell \in \{success, failure\}$, which terminates the response sequence by definition. Thus the two response sequences are identical, as desired. When the attack sequence is passive, essentially the same proof applies. ∎

B. Proof of Theorem 2

Before we present the proof of this theorem, we note several identities that follow from Definition 3.

1. $T(E) \cap U = T(\mathsf{unreleaseable}(E, U)) \cap U$.
$$\begin{aligned} T(E) \cap U &= (\cup_{e\in E} T(e)) \cap U = \cup_{e\in E}(T(e) \cap U) \\ &= \cup_{e\in E \wedge T(e)\cap U\neq\emptyset}(T(e) \cap U) \\ &= \cup_{e\in \mathsf{unreleaseable}(E,U)}(T(e) \cap U) \\ &= \big(\cup_{e\in \mathsf{unreleaseable}(E,U)} T(e)\big) \cap U \\ &= T(\mathsf{unreleaseable}(E, U)) \cap U \end{aligned}$$

2. $T(\mathsf{releaseable}(E, U)) \cap U = \emptyset$.
$$\begin{aligned} & T(\mathsf{releaseable}(E, U)) \cap U \\ &= \big(\cup_{e\in \mathsf{releaseable}(E,U)} T(e)\big) \cap U \\ &= \cup_{e\in E\wedge T(e)\cap U=\emptyset}(T(e) \cap U) \\ &= \cup_{e\in E\wedge T(e)\cap U=\emptyset}\emptyset \\ &= \emptyset \end{aligned}$$

3. $\mathsf{releaseable}(E_1 \cup E_2, U)$
$$= \mathsf{releaseable}(E_1, U) \cup \mathsf{releaseable}(E_2, U)$$
$\mathsf{releaseable}(E_1 \cup E_2, U)$
$$\begin{aligned} &= \{e \in (E_1 \cup E_2) \mid T(e) \cap U \neq \emptyset\} \\ &= \{e \in E_1 \mid T(e) \cap U \neq \emptyset\} \\ &\quad \cup \{e \in E_2 \mid T(e) \cap U \neq \emptyset\} \\ &= \mathsf{releaseable}(E_1, U) \cup \mathsf{releaseable}(E_2, U) \end{aligned}$$

4. For all $U' \supseteq U$, $\mathsf{releaseable}(\mathsf{unreleaseable}(E, U), U') = \emptyset$.
$\mathsf{releaseable}(\mathsf{unreleaseable}(E, U), U')$
$$\begin{aligned} &= \{e \in \{e \in E \mid T(e) \cap U \neq \emptyset\} \mid T(e) \cap U' = \emptyset\} \\ &= \{e \in E \mid T(e) \cap U \neq \emptyset \wedge T(e) \cap U' = \emptyset\} \\ &= \emptyset \end{aligned}$$

5. For all $U' \supseteq U$, $\mathsf{releaseable}(\mathsf{releaseable}(E, U), U') = \mathsf{releaseable}(E, U')$.
$\mathsf{releaseable}(\mathsf{releaseable}(E, U), U')$
$$\begin{aligned} &= \{e \in \{e \in E \mid T(e) \cap U = \emptyset\} \mid T(e) \cap U' = \emptyset\} \\ &= \{e \in E \mid T(e) \cap U = \emptyset \wedge T(e) \cap U' = \emptyset\} \\ &= \{e \in E \mid T(e) \cap U' = \emptyset\} \\ &= \mathsf{releaseable}(E, U') \end{aligned}$$

Theorem 2 The relative strength of the safety definitions is as follows:

1. If strat is credential-combination-hiding safe, then it is attribute-combination-hiding safe.
2. If strat is attribute-combination-hiding safe, then it is attribute-hiding safe.

Proof.
Part 1 Given a credential-combination-hiding safe strategy strat, for every configuration $G = \langle K, E, \mathsf{Policy}, \mathsf{Ack}\rangle$, for every subset U of $\mathcal{T}$, and for every expressible subset U' of U, we can construct a configuration $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack}\rangle$ as follows. By the assumption that U' is expressible, there exists E_0 such that $T(E_0) \cap U = U'$. Let $E' = \mathsf{unreleaseable}(E_0, U) \cup \mathsf{releaseable}(E, U)$.

We now show **(1a)**: E' induces the desired set of unacknowledgeable attributes, i.e., $T(E') \cap U = U'$. From Identities 1 and 2, we have the following:

$$
\begin{aligned}
T(E') \cap U &= (T(\mathsf{unreleaseable}(E_0, U)) \cup \\
&\quad T(\mathsf{releaseable}(E, U))) \cap U \\
&= (T(\mathsf{unreleaseable}(E_0, U)) \cap U) \cup \\
&\quad (T(\mathsf{releaseable}(E, U)) \cap U) \\
&= (T(E_0) \cap U) \cup \emptyset = U'
\end{aligned}
$$

We now use credential-combination-hiding safety to show the following **(1b)**: for every M such that $\mathsf{UnAcks}(G, M) \supseteq U$, G and G' are indistinguishable under strat by M. Let U'' be the set of attributes that are unacknowledgeable to M; we have $U'' \supseteq U$. It is sufficient to show that $\mathsf{releaseable}(E, U'') = \mathsf{releaseable}(E', U'')$, since by the credential-combination-hiding safety property of strat, M cannot distinguish G and G'. This equality follows from Identities 3, 4, and 4 as follows:

$$
\begin{aligned}
&\mathsf{releaseable}(E', U'') \\
&\quad = \mathsf{releaseable}(\mathsf{unreleaseable}(E_0, U) \cup \\
&\qquad \mathsf{releaseable}(E, U), U'') \\
&\quad = \mathsf{releaseable}(\mathsf{unreleaseable}(E_0, U), U'') \cup \\
&\qquad \mathsf{releaseable}(\mathsf{releaseable}(E, U), U'') \\
&\quad = \emptyset \cup \mathsf{releaseable}(E, U'') = \mathsf{releaseable}(E, U'')
\end{aligned}
$$

Part 2: Given an attribute-combination-hiding safe strategy strat, for every configuration $G = \langle K, E, \mathsf{Policy}, \mathsf{Ack} \rangle$, for every attribute t, we need to show that there exists G' that differs from G in t (*i.e.*, G induces t and G' does not, or vice versa) and for every adversary M, if t in $\mathsf{UnAcks}(G, M)$, G' is indistinguishable from G by M. Case one: if G induces t, i.e., $t \in T(E)$, then let $U = \{t\}$ and $U' = \{\}$. Clearly, U' is an expressible subset to U. By attribute-combination-hiding safety of strat, there exists a configuration $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack} \rangle$ that satisfies the above requirement. Case two: if $t \notin T(E)$, then let $U = \{t\}$ and $U' = \{t\}$. Clearly, U' is an expressible subset of U. (By the setup of the framework, every attribute has at least one credential to prove it.) Again, by attribute-combination-hiding safety of strat, there exists a configuration $G' = \langle K, E', \mathsf{Policy}, \mathsf{Ack} \rangle$ that satisfies the above requirement. ∎

Acknowledgements

Both authors are supported by NSF ITR grant CCR-0325951 (BYU). We thank the anonymous reviewers for their helpful suggestions.

References

[1] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pages 164–173. IEEE Computer Society Press, May 1996.

[2] P. Bonatti, S. Kraus, and V. S. Subrahmanian. Foundations of secure deductive databases. *Knowledge and Data Engineering*, 7(3):406–422, 1995.

[3] P. Bonatti and P. Samarati. Regulating service access and information release on the web. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS-7)*, pages 134–143. ACM Press, Nov. 2000.

[4] J. Domingo-Ferrer, editor. *Inference Control in Statistical Databases, From Theory to Practice*, volume 2316 of *Lecture Notes in Computer Science*. Springer, 2002.

[5] J. Goguen and J. Meseguer. Security policies and security models. In *In Proceedings of the 1982 IEEE Symposium on Security and Privacy*, pages 11–20. IEEE Computer Society Press, Apr. 1982.

[6] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pages 2–14. IEEE Computer Society Press, May 2000.

[7] A. Hess, J. Jacobson, H. Mills, R. Wamsley, K. E. Seamons, and B. Smith. Advanced client/server authentication in TLS. In *Network and Distributed System Security Symposium*, pages 203–214, Feb. 2002.

[8] J. E. Holt, R. W. Bradshaw, K. E. Seamons, and H. Orman. Hidden credentials. In *Proceedings of the 2nd ACM Workshop on Privacy in the Electronic Society*, Oct. 2003.

[9] N. Li, W. Du, and D. Boneh. Oblivious signature-based envelope. In *Proceedings of the 22nd ACM Symposium on Principles of Distributed Computing (PODC 2003)*. ACM Press, July 2003.

[10] N. Li, B. N. Grosof, and J. Feigenbaum. Delegation Logic: A logic-based approach to distributed authorization. *ACM Transaction on Information and System Security (TISSEC)*, 6(1):128–171, Feb. 2003.

[11] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.

[12] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, Feb. 2003.

[13] R. L. Rivest, A. Shamir, and L. M. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21:120–126, 1978.

[14] K. E. Seamons, M. Winslett, and T. Yu. Limiting the disclosure of access control policies during automated trust negotiation. In *Proceedings of the Symposium on Network and Distributed System Security (NDSS'01)*, February 2001.

[15] K. E. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting privacy during on-line trust negotiation. In *2nd Workshop on Privacy Enhancing Technologies*. Springer-Verlag, Apr. 2002.

[16] J. Staddon. Dynamic inference control. In *Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 94–100. ACM Press, 2003.

[17] D. Sutherland. A model of information. In *Proceedings of the 9th National Computer Security Conference*, pages 175–183, Sept. 1986.

[18] L. Wang, D. Wijesekera, and S. Jajodia. Cardinality-based inference control in data cubes, 2003. To appear in: *Journal of Computer Security*.

[19] W. H. Winsborough and N. Li. Protecting sensitive attributes in automated trust negotiation. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pages 41–51. ACM Press, Nov. 2002.

[20] W. H. Winsborough and N. Li. Towards practical automated trust negotiation. In *Proceedings of the Third International Workshop on Policies for Distributed Systems and Networks (Policy 2002)*, pages 92–103. IEEE Computer Society Press, June 2002.

[21] W. H. Winsborough and N. Li. Safety in automated trust negotiation. Technical Report CSIS-TR-1-04, Center for Secure Information Systems, George Mason University, Mar. 2004.

[22] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition*, volume I, pages 88–102. IEEE Press, Jan. 2000.

[23] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. Negotiating trust on the web. *IEEE Internet Computing*, 6(6):30–37, November/December 2002.

[24] T. Yu, X. Ma, and M. Winslett. Prunes: An efficient and complete strategy for trust negotiation over the internet. In *Proceedings of the 7th ACM Conference on Computer and Communications Security (CCS-7)*, pages 210–219. ACM Press, Nov. 2000.

[25] T. Yu and M. Winslett. Policy migration for sensitive credentials in trust negotiation. In *Proceedings of the ACM Workshop on Privacy in the Electronic Society*, pages 9–20. ACM Press, Oct. 2003.

[26] T. Yu and M. Winslett. Unified scheme for resource protection in automated trust negotiation. In *Proceedings of IEEE Symposium on Security and Privacy*, pages 110–122. IEEE Computer Society Press, May 2003.

[27] T. Yu, M. Winslett, and K. E. Seamons. Supporting structured credentials and sensitive policies through interoperable strategies for automated trust negotiation. *ACM Transactions on Information and System Security (TISSEC)*, 6(1):1–42, Feb. 2003.