

# Preventing Attribute Information Leakage in Automated Trust Negotiation

Keith Irwin  
North Carolina State University  
kirwin@ncsu.edu

Ting Yu  
North Carolina State University  
yu@csc.ncsu.edu

## ABSTRACT

Automated trust negotiation is an approach which establishes trust between strangers through the bilateral, iterative disclosure of digital credentials. Sensitive credentials are protected by access control policies which may also be communicated to the other party. Ideally, sensitive information should not be known by others unless its access control policy has been satisfied. However, due to bilateral information exchange, information may flow to others in a variety of forms, many of which cannot be protected by access control policies alone. In particular, sensitive information may be inferred by observing negotiation participants' behavior even when access control policies are strictly enforced.

In this paper, we propose a general framework for the safety of trust negotiation systems. Compared to the existing safety model, our framework focuses on the actual information gain during trust negotiation instead of the exchanged messages. Thus, it directly reflects the essence of safety in sensitive information protection. Based on the proposed framework, we develop *policy databases* as a mechanism to help prevent unauthorized information inferences during trust negotiation. We show that policy databases achieve the same protection of sensitive information as existing solutions without imposing additional complications to the interaction between negotiation participants or restricting users' autonomy in defining their own policies.

**Categories and Subject Descriptors:** K.6.5 [Management of Computing and Information Systems]: Security and Protection

**General Terms:** Security, Theory

**Keywords:** Privacy, Trust Negotiation, Attribute-based Access Control

## 1. INTRODUCTION

Automated trust negotiation (ATN) is an approach to access control and authentication in open, flexible systems such as the Internet. ATN enables open computing by as-

signing an access control policy to each resource that is to be made available to entities from different domains. An access control policy describes the attributes of the entities allowed to access that resource, in contrast to the traditional approach of listing their identities. To satisfy an access control policy, a user has to demonstrate that they have the attributes named in the policy through the use of digital credentials. Since one's attributes may also be sensitive, the disclosure of digital credentials is also protected by access control policies.

A trust negotiation is triggered when one party requests access to a resource owned by another party. Since each party may have policies that the other needs to satisfy, trust is established incrementally through bilateral disclosures of credentials and requests for credentials, a characteristic that distinguishes trust negotiation from other trust establishment approaches [2, 11].

Access control policies play a central role in protecting privacy during trust negotiation. Ideally, an entity's sensitive information should not be known by others unless they have satisfied the corresponding access control policy. However, depending on the way two parties interact with each other, one's private information may flow to others in various forms, which are not always controlled by access control policies. In particular, the different behaviors of a negotiation participant may be exploited to infer sensitive information, even if credentials containing that information are never directly disclosed.

For example, suppose a resource's policy requires Alice to prove a sensitive attribute such as employment by the CIA. If Alice has this attribute, then she likely protects it with an access control policy. Thus, as a response, Alice will ask the resource provider to satisfy her policy. On the other hand, if Alice does not have the attribute, then a natural response would be for her to terminate the negotiation since there is no way that she can access the resource. Thus, merely from Alice's response, the resource provider may infer with high confidence whether or not Alice is working for the CIA, even though her access control policy is strictly enforced.

The problem of unauthorized information flow in ATN has been noted by several groups of researchers [20, 22, 27]. A variety of approaches have been proposed, which mainly fall into two categories. Approaches in the first category try to "break" the correlation between different information. Intuitively, if the disclosed policy for an attribute is independent from the possession of the attribute, then the above inference is impossible. A representative approach in this category is by Seamons et al. [20], where an entity possessing a sensi-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CCS'05, November 7–11, 2005, Alexandria, Virginia, USA.  
Copyright 2005 ACM 1-59593-226-7/05/0011 ...\$5.00.

tive credential always responds with a cover policy of *false* to pretend the opposite. Only when the actual policy is satisfied by the credentials disclosed by the opponent will the entity disclose the credential. Clearly, since the disclosed policy is always *false*, it is not correlated to the possession of the credential. One obvious problem with this approach, however, is that a potentially successful negotiation may fail because an entity pretends to not have the credential.

Approaches in the second category aim to make the correlation between different information “safe”, i.e., when an opponent is able to infer some sensitive information through the correlation, it is already entitled to know that information. For example, Winsborough and Li [23] proposed the use of *acknowledgement policies* (“Ack policies” for short) as a solution. Their approach is based on the principle “discuss sensitive topics only with appropriate parties”. Therefore, besides an access control policy  $P$ , Alice also associates an Ack policy  $P_{Ack}$  with a sensitive attribute  $A$ . Intuitively,  $P_{Ack}$  determines when Alice can tell others whether or not she has attribute  $A$ . During a negotiation, when the attribute is requested, the Ack policy  $P_{Ack}$  is first sent back as a reply. Only when  $P_{Ack}$  is satisfied by the other party, will Alice disclose whether or not she has  $A$  and may then ask the other party to satisfy the access control policy  $P$ . In order to prevent additional correlation introduced by Ack policies, it is required that all entities use the same Ack policy to protect a given attribute regardless of whether or not they have  $A$ . In [23], Winsborough and Li also formally defined the safety requirements in trust negotiation based on Ack policies.

Though the approach of Ack policies can provide protection against unauthorized inferences, it has a significant disadvantage. One benefit of automated trust negotiation is that it gives each entity the autonomy to determine the appropriate protection for its own resources and credentials. The perceived sensitivity of possessing an attribute may be very different for different entities. For example, some may consider the possession of a certificate showing eligibility for food stamps highly sensitive, and thus would like to have a very strict Ack policy for it. Some others may not care as much, and have a less strict Ack policy, because they are more concerned with their ability to get services than their privacy. The Ack Policy system, however, requires that all entities use the same Ack policy for a given attribute, which deprives entities of the autonomy to make their own decisions. This will inevitably be over-protective for some and under-protective for others. And either situation will result in users preferring not to participate in the system.

In this paper, we first propose a general framework for safe information flow in automated trust negotiation. Compared with that proposed by Winsborough and Li, our framework focuses on modeling the actual information gain caused by information flow instead of the messages exchanged. Therefore it directly reflects the essence of safety in sensitive information protection. Based on this framework, we propose *policy databases* as a solution to the above problem. Policy databases not only prevent unauthorized inferences as described above but also preserve users’ autonomy in deciding their own policies. In order to do this, we focus on severing the correlation between attributes and policies by introducing randomness, rather than adding additional layers or fixed policies as in the Ack Policy system. In our approach, there is a central database of policies for each possession

sensitive attribute. Users who possess the attribute submit their policies to the database anonymously. Users who do not possess the attribute can then draw a policy at random from the database. The result of this process is that the distributions of policies for a given possession sensitive attribute are identical for users who have the attribute and users who do not. Thus, an opponent cannot infer whether or not users possess the attribute by looking at their policies.

The rest of the paper is organized as follows. In section 2, we propose a formal definition of safety for automated trust negotiation. In section 3, we discuss the specifics of our approach, including what assumptions underlie it, how well it satisfies our safety principle, both theoretically and in practical situations, and what practical concerns to implementing it exist. Closely related work to this paper is reported in section 4. We conclude this paper in section 5

## 2. SAFETY IN TRUST NEGOTIATION

In [23], Winsborough and Li put forth several definitions of safety in trust negotiation based on an underlying notion of indistinguishability. The essence of indistinguishability is that if an opponent is given the opportunity to interact with a user in two states corresponding to two different potential sets of attributes, the opponent cannot detect a difference in those sets of attributes based on the messages sent. In the definition of deterministic indistinguishability, the messages sent in the two states must be precisely the same. In the definition of probabilistic indistinguishability, they must have precisely the same distribution.

These definitions, however, are overly strict. To determine whether or not a given user has a credential, it is not sufficient for an opponent to know that the user acts differently depending on whether or not that user has the credential: the opponent also has to be able to figure out which behavior corresponds to having the credential and which corresponds to lacking the credential. Otherwise, the opponent has not actually gained any information about the user.

*EXAMPLE 1. Suppose we have a system in which there is only one attribute and two policies,  $p_1$  and  $p_2$ . Half of the users use  $p_1$  when they have the attribute and  $p_2$  when they do not. The other half of the users use  $p_2$  when they have the attribute and  $p_1$  when they do not. Every user’s messages would be distinguishable under the definition of indistinguishability presented in [23] because for each user the distribution of messages is different. However, if a fraction  $r$  of the users have the attribute and a fraction  $1 - r$  do not, then  $\frac{1}{2} \cdot r + \frac{1}{2} \cdot (1 - r) = \frac{1}{2}$  of the users display policy  $p_1$  and the other half of the users display policy  $p_2$ . Since the policy displayed is independent of the attribute when users are viewed as a whole, seeing either policy does not reveal any information about whether or not the user in question has the attribute.*

As such, Winsborough and Li’s definitions of indistinguishability restrict a number of valid systems where a given user will act differently in the two cases, but an opponent cannot actually distinguish which case is which. In fact, their definitions allow only systems greatly similar to the Ack Policy system that they proposed in [22]. Instead we propose a definition of safety based directly on information gain instead of the message exchange sequences between the two parties.

Before we formally define safety, we first discuss what safety means informally. In any trust negotiation system, there is some set of objects which are protected by policies. Usually this includes credentials, information about attribute possession, and sometimes even some of the policies in the system. All of these can be structured as digital information, and the aim of the system is to disclose that information only to appropriate parties.

An obvious idea of safety is that an object's value should not be revealed unless its policy has been satisfied. However, we do not want to simply avoid an object's value being known with complete certainty, but also the value being guessed with significant likelihood.

As such, we can define the change in safety as the change in the probability of guessing the value of an object. If there are two secrets,  $s_1$  and  $s_2$ , we can define the conditional safety of  $s_1$  upon the disclosure of  $s_2$  as the conditional probability of guessing  $s_1$  given  $s_2$ . Thus, we define absolute safety in a system as being the property that no disclosure of objects whose policies have been satisfied results in any change in the probability of guessing the value of any object whose policy has not been satisfied regardless of what inferences might be possible.

There exists a simple system which can satisfy this level of safety, which is the all-or-nothing system, a system in which all of every user's objects are required to be protected by a single policy which is the same for all users. Clearly in such a system there are only two states, all objects revealed or no objects revealed. As such, there can be no inferences between objects which are revealed and objects which are not. This system, however, has undesirable properties which outweigh its safety guarantees, namely the lack of autonomy, flexibility, and fine-grained access control. Because of the necessity of protecting against every possible inference which could occur, it is like that any system which achieves ideal safety would be similarly inflexible.

Since there have been no practical systems proposed which meet the ideal safety condition, describing ideal safety is not sufficient unto itself. We wish to explore not just ideal safety, but also safety relative to certain types of attacks. This will help us develop a more complete view of safety in the likely event that no useful system which is ideally safe is found.

If a system does not have ideal safety, then there must be some inferences which can cause a leakage of information between revealed objects and protected objects. But this does not mean that every single object revealed leaks information about every protected object. As such, we can potentially describe what sort of inferences a system does protect against. For example, Ack Policy systems are motivated by a desire to prevent inferences from a policy to the possession of the attribute that it protects. Inferences from one attribute to another are not prevented by such a system (for example, users who are AARP members are more likely to be retired than ones who are not). Hence, it is desirable to describe what it means for a system to be safe relative to certain types of inferences.

Next we present a formal framework to model safety in trust negotiation. The formalism which we are using in this paper is based on that used by Winsborough and Li, but is substantially revised.

## 2.0.1 Trust Negotiation Systems

A Trust Negotiation System is comprised of the following

elements:

- A finite set,  $\mathcal{K}$ , of *principals*, uniquely identified by a randomly chosen public key,  $Pub_k$ . Each principal knows the associated private key, and can produce a proof of identity.
  - A finite set,  $\mathcal{T}$ , of *attributes*. An attribute is something which each user either possesses or lacks. An example would be status as a licensed driver or enrollment at a university.
  - A set,  $\mathcal{G}$ , of *configurations*, each of which is a subset of  $\mathcal{T}$ . If a principal  $k$  is in a configuration  $g \in \mathcal{G}$ , then  $k$  possesses the attributes in  $g$  and no other attributes.
  - A set,  $\mathcal{P}$ , of possible policies, each of which is a logical proposition comprised of a combination of *and*, *or*, and attributes in  $\mathcal{T}$ . We define an attribute in a policy to be true with respect to a principal  $k$  if  $k$  has that attribute. We consider all logically equivalent policies to be the same policy.
  - *Objects*. Every principal  $k$  has objects which may be protected which include the following:
    - A set,  $\mathcal{S}$ , of *services* provided by a principal. Every principal offers some set of services to all other principals. These services are each protected by some policy, as we will describe later. A simple service which each principal offers is a proof of attribute possession. If another principal satisfies the appropriate policy, the principal will offer some proof that he holds the attribute. This service is denoted  $s_t$  for any attribute  $t \in \mathcal{T}$ .
    - A set,  $\mathcal{A}$ , of *attribute status objects*. Since the set of all attributes is already known, we want to protect the information about whether or not a given user has an attribute. As such we formally define  $\mathcal{A}$  as a set of boolean valued random variables,  $a_t$ . The value of  $a_t$  for a principal  $k$ , which we denote  $a_t(k)$  is defined to be true if  $k$  possesses  $t \in \mathcal{T}$  and false otherwise. Thus  $\mathcal{A} = \{a_t | t \in \mathcal{T}\}$ .
    - A set,  $\mathcal{Q}$  of *policy mapping objects*. A system may desire to protect an object's policy either because of correlations between policies and sensitive attributes or because in some systems the policies themselves may be considered sensitive. Similar to attribute status objects, we do not protect a policy, per se, but instead the pairing of a policy with what it is protecting. As such, each policy mapping object is a random variable  $q_o$  with range  $\mathcal{P}$  where  $o$  is an object. The value of  $q_o$  for a given principal  $k$ , denoted  $q_o(k)$  is the policy that  $k$  has chosen to protect object  $o$ .
- Every system should define which objects are protected. It is expected that all systems should protect the services,  $\mathcal{S}$ , and the attribute status objects,  $\mathcal{A}$ . In some systems, there will also be policies which protect policies. Thus protected objects may also include a subset of  $\mathcal{Q}$ . We call the set of protected objects  $\mathcal{O}$ , where  $\mathcal{O} \subseteq \mathcal{S} \cup \mathcal{A} \cup \mathcal{Q}$ . If an object is not protected, this is equivalent to it having a policy equal to *true*.
- For convenience, we define  $\mathcal{Q}_{\mathcal{X}}$  to be the members of  $\mathcal{Q}$  which are policies protecting members of  $\mathcal{X}$ , where  $\mathcal{X}$  is a set of objects. Formally,  $\mathcal{Q}_{\mathcal{X}} = \{q_o \in \mathcal{Q} | o \in \mathcal{X}\}$ .
- Some subset of the information objects are considered to be *sensitive* objects. These are the objects about which we want an opponent to gain no information unless they have satisfied that object's policy. Full information about any object, sensitive or insensitive, is not released by the system until its policy has been satisfied, but it is acceptable for inferences to cause the leakage of information which is not considered sensitive.

- A set,  $\mathcal{N}$ , of *negotiation strategies*. A negotiation strategy is the means that a principal uses to interact with other principals. Established strategies include the eager strategy [24] and the trust-target graph strategy [22]. A negotiation strategy,  $n$ , is defined as an interactive, deterministic, Turing-equivalent computational machine augmented by a random tape. The random tape serves as a random oracle which allows us to discuss randomized strategies.

A negotiation strategy takes as initial input the public knowledge needed to operate in a system, the principal's attributes, its services, and the policies which protect its objects. It then produces additional inputs and outputs by interacting with other strategies. It can output policies, credentials, and any additional information which is useful. We do not further define the specifics of the information communicated between strategies except to note that all the strategies in a system should have compatible input and output protocols. We refrain from further specifics of strategies since they are not required in our later discussion.

- An adversary,  $M$ , is defined as a set of principals coordinating to discover the value of sensitive information objects belonging to some  $k \notin M$ . Preventing this discovery is the security goal of a trust negotiation system. We assume that adversaries may only interact with principals through trust negotiation and are limited to proving possession of attributes which they actually possess. In other words, the trust negotiation system provides a means of proof which is resistant to attempts at forgery.
- A set,  $\mathcal{I}$ , of all *inferences*. Each inference is a minimal subset of information objects such that the joint distribution of the set differs from the product of the individual distributions of the items in the set.<sup>1</sup>

These then allow a partitioning,  $\mathcal{C}$ , of the information objects into *inference components*. We define a relation  $\circ$  such that  $o_1 \circ o_2$  iff  $\exists i \in \mathcal{I} | o_1, o_2 \in i$ .  $\mathcal{C}$  is the transitive closure of  $\circ$ .

In general, we assume that all of the information objects in our framework are static. We do not model changes in a principal's attribute status or policies. If such is necessary, the model would need to be adapted.

It should also be noted that there is an additional constraint on policies that protect policies which we have not described. This is because in most systems there is a way to gain information about what a policy is, which is to satisfy it. When a policy is satisfied, this generally results in some service being rendered or information being released. As such, this will let the other party know that they have satisfied the policy for that object. Therefore, the effective policy protecting a policy status object must be the logical *or* of the policy in the policy status object and the policy which protects it.

### 2.0.2 The Ack Policy System

To help illustrate the model, let us describe how the Ack Policy system maps onto the model. The mapping of opponents, the sets of principals, attributes, configurations, and policies in the Ack Policy system is straightforward.

In an Ack Policy system, any mutually compatible set of negotiation strategies is acceptable. There are policies for

<sup>1</sup>A system need not define the particulars of inferences, but should discuss what sort of inferences it can deal with, and hence what sort of inferences are assumed to exist.

protecting services, protecting attribute status objects, and protecting policies which protect attribute proving services. As such, the set of protected objects,  $\mathcal{O} = \mathcal{S} \cup \mathcal{A} \cup \mathcal{Q}_S$ .

According to the definition of the Ack Policy system, for a given attribute, the policy that protects the proof service for that attribute is protected by the same policy that protects the attribute status object. Formally,  $\forall t \in \mathcal{T}, \forall k \in \mathcal{K}, q_{a_t}(k) = q_{q_{s_t}}(k)$ . Further, the Ack policy for an attribute is required to be the same for all principals. Thus we know  $\forall t \in \mathcal{T} \exists p \in \mathcal{P} \forall k \in \mathcal{K} | q_{a_t}(k) = p$ .

Two basic assumptions about the set of inferences,  $\mathcal{I}$ , exist in Ack Policy systems, which also lead us to conclusions about the inference components,  $\mathcal{C}$ . It is assumed that inferences between the policy which protects the attribute proving service,  $q_{s_t}(k)$ , and the attribute status object,  $a_t(k)$ , exist. As such, those two objects should always be in the same inference component. Because Ack Policies are uniform for all principals, they are uncorrelated to any other information object and they cannot be part of any inference. Hence, each Ack Policy is in an inference component of its own.

### 2.0.3 Safety in Trust Negotiation Systems

In order to formally define safety in trust negotiation, we need to define the specifics of the opponent. We need to model the potential capabilities of an opponent and the information initially available to the opponent. Obviously, no system is safe against an opponent with unlimited capabilities or unlimited knowledge.

As such, we restrict the opponent to having some *tactic*, for forming trust negotiation messages, processing responses to those messages, and, finally, forming a guess about the values of unrevealed information objects. We model the tactic as an interactive, deterministic, Turing-equivalent computational machine. This model is a very powerful model, and we argue that it describes any reasonable opponent. This model, however, restricts the opponent to calculating things which are computable from its input and implies that the opponent behaves in a deterministic fashion.

The input available to the machine at the start is the knowledge available to the opponent before any trust negotiation has taken place. What this knowledge is varies depending on the particulars of a trust negotiation system. However, in every system this should include the knowledge available to the principals who are a part of the opponent, such as their public and private keys and their credentials. And it should also include public information such as how the system works, the public keys of the attribute authorities, and other information that every user knows. In most systems, information about the distribution of attributes and credentials and knowledge of inference rules should also be considered as public information. All responses from principals in different configurations become available as input to the tactic as they are made. The tactic must output both a sequence of responses and, at the end, guesses about the unknown objects of all users.

We observe that an opponent will have probabilistic knowledge about information objects in a system. Initially, the probabilities will be based only on publicly available knowledge, so we can use the publicly available knowledge to describe the a priori probabilities.

For instance, in most systems, it would be reasonable to assume that the opponent will have knowledge of the odds

that any particular member of the population has a given attribute. Thus, if a fraction  $h_t$  of the population is expected to possess attribute  $t \in \mathcal{T}$ , the opponent should begin with an assumption that some given principal has a  $h_t$  chance of having attribute  $t$ . Hence,  $h_t$  represents the a priori probability of any given principal possessing  $t$ . Note that we assume that the opponent only knows the odds of a given principal having an attribute, but does not know for certain that a fixed percentage of the users have a given attribute. As such, knowledge about the value of an object belonging to some set of users does not imply any knowledge about the value of objects belonging to some other user.

**DEFINITION 1.** *A trust negotiation system is safe relative to a set of possible inferences if for all allowed mappings between principals and configurations there exists no opponent which can guess the value of sensitive information objects whose security policies have not been met with odds better than the a priori odds over all principals which are not in the opponent, over all values of all random tapes, and over all mappings between public key values and principals.*

Definition 1 differs from Winsborough and Li’s definitions in several ways. The first is that it is concerned with multiple users. It both requires that the opponent form guesses over all users and allows the opponent to interact with all users. Instead of simply having a sequence of messages sent to a single principal, the tactic we have defined may interact with a variety of users, analyzing incoming messages, and then use them to form new messages. It is allowed to talk to the users in any order and to interleave communications with multiple users, thus it is more general than those in [23]. The second is that we are concerned only with the information which the opponent can glean from the communication, not the distribution of the communication itself. As such, our definition more clearly reflects the fundamental idea of safety.

We next introduce a theorem which will be helpful in proving the safety of systems.

**THEOREM 1.** *There exists no opponent which can beat the a priori odds of guessing the value of an object,  $o$ , given only information about objects which are not in the same inference component as  $o$ , over all principals not in  $M$  and whose policy for  $o$   $M$  cannot satisfy, over all random tapes, and over all mappings between public keys and principals.*

The formal proof for this theorem can be found in Appendix A. Intuitively, since the opponent only gains information about objects not correlated to  $o$ , its guess of the value of  $o$  is not affected.

With theorem 1, let us take a brief moment to prove the safety of the Ack Policy systems under our framework. Specifically, we examine Ack Policy systems in which the distribution of strategies is independent of the distributions of attributes, an assumption implicitly made in [23]. In Ack Policy systems the Ack Policy is a policy which protects two objects in our model: an attribute’s status object and its policy for that attribute’s proof service. Ack Policies are required to be uniform for all users, which ensures that they are independent of all objects.

Ack Policy systems are designed to prevent inferences from an attribute’s policy to an attribute’s status for attributes which are sensitive. So, let us assume an appropriate set of inference components in order to prove that Ack

Policy systems are safe relative to that form of inference. As we described earlier, each attribute status object should be in the same inference component with the policy which protects that attribute’s proof service, and the Ack policy for each attribute should be in its own inference component. The Ack Policy system also assumes that different attributes are independent of each other. As such, each attribute status object should be in a different inference group.

This set of inference components excludes all other possible types of inferences. The set of sensitive objects is the set of attribute status objects whose value is *true*. Due to Theorem 1, we know then that no opponent will be able to gain any information based on objects in different inference components. So the only potential source of inference for whether or not a given attribute’s status object,  $a_t$ , has a value of *true* or *false* is the policy protecting the attribute proof service,  $s_t$ .

However, we know that the same policy,  $P$ , protects both of these objects. As such unauthorized inference between them is impossible without satisfying  $P$ .<sup>2</sup> Thus, the odds for  $a_t$  do not change. Therefore, the Ack Policy system is secure against inferences from an attribute’s policy to its attribute status.

### 3. POLICY DATABASE

We propose a new trust negotiation system designed to be safe under the definition we proposed, but to also allow the users who have sensitive attributes complete freedom to determine their own policies. It also does not rely on any particular strategy being used. Potentially, a combination of strategies could even be used so long as the strategy chosen is not in any way correlated to the attributes possessed.

This system is based on the observation that there is more than one way to deal with a correlation. A simple ideal system which prevents the inference from policies to attribute possession information is to have each user draw a random policy. This system obviously does not allow users the freedom to create their own policies. Instead we propose a system which looks like the policies are random even though they are not.

This system is similar to the existing trust negotiation systems except for the addition of a new element: the policy database. The policy database is a database run by a trusted third party which collects anonymized information about the policies which are in use. In the policy database system, a user who has a given sensitive attribute chooses their own policy and submits it anonymously to the policy database for that attribute. The policy database uses pseudonymous certificates to verify that users who submit policies actually

<sup>2</sup>Except that one of these is a policy mapping object which is being protected by a policy. As such, we have to keep in mind that there exists a possibility that the opponent could gain information about the policy without satisfying it. Specifically, the opponent can figure out what attributes do not satisfy it by proving that he possesses those attributes. However, in an Ack Policy system, the policy protecting an attribute proof object of an attribute which a user does not hold is always *false*. No opponent can distinguish between two policies which they cannot satisfy since all they know is that they have failed to satisfy them. And we are unconcerned with policies which they have satisfied. Thus, we know that the opponent cannot gain any useful information about the policies which they have not satisfied, and hence cannot beat the a priori odds for those policies.

have the attribute, in a manner that will be discussed later in section 3.2. Then users who do not have the attribute will pull policies at random from the database to use as their own. The contents of the policy database are public, so any user who wishes to can draw a random policy from the database.

In our system, each user uses a single policy to protect all the information objects associated with an attribute. They neither acknowledge that they have the attribute nor prove that they do until the policy has been satisfied. This means that users are allowed to have policies which protect attributes which they do not hold. The policy in our system may be seen as the combination of the Ack policy and a traditional access control policy for attribute proofs.

The goal of this system is to ensure that the policy is in a separate inference component from the attribute status object, thus guaranteeing that inferences between policies and attribute status objects cannot be made.

This system is workable because of the following. We know that policies cannot require a lack of an attribute, thus users who do not have a given attribute will never suffer from their policy for that attribute being too strong. Changes in the policy which protects an attribute that they do not have may vary the length of the trust negotiation, but it will never cause them to be unable to complete transactions which they would otherwise be able to complete. Also, we deal only with possession sensitive attributes. We do not deal with attributes where it is at all sensitive to lack them. As such, users who do not have the attribute cannot have their policies be too weak. Since there is no penalty for those users for their policies being either too weak or too strong, they can have whatever policy is most helpful for helping disguise the users who do possess the attribute.

This also means that users who do not have the attribute do not need to trust the policy database since no policy which the database gives them would be unacceptable to them. Users who have the attribute, however, do need to trust that the policy database will actually randomly distribute policies to help camouflage their policies. They do not, however, need to trust the policy database to act appropriately with their sensitive information because all information is anonymized.

### 3.1 Safety of the Approach of Policy Databases

Let us describe the Policy Database system in terms of our model. Again the opponent and the sets of principals, attributes, configurations, and policies need no special comment. Because we only have policies protecting the services and attribute status objects, the set of protected objects,  $\mathcal{O} = \mathcal{S} \cup \mathcal{A}$ . Also, each attribute proving service and attribute status object are protected by the same policy.  $\forall t \in \mathcal{T}, \forall k \in \mathcal{K}, q_{a_t}(k) = q_{s_t}(k)$ .

This system is only designed to deal with inferences from policies to attribute possession, so we assume that every attribute status object is in a different inference component. If the policies do actually appear to be completely random, then policies and attribute status objects should be in separate inference components as well.

The obvious question is whether Policy Database systems actually guarantee that this occurs. The answer is that they do not guarantee it with any finite number of users due to the distribution of policies being unlikely to be absolutely, precisely the same. This is largely due to a combination of

rounding issues and the odds being against things coming out precisely evenly distributed. However, as the number of users in the system approaches infinity, the system approaches this condition.

In an ideal system, the distribution of policies would be completely random. If an opponent observes that some number of principals had a given policy for some attribute, this would give them no information about whether or not any of those users had the attribute. However, in the Policy Database system, every policy which is held is known to be held by at least one user who has the attribute. As such, we need to worry about how even the distributions of different policies are.

We can describe and quantify the difference which exists between a real implementation of our system and the ideal. There are two reasons for a difference to exist. The first is difference due to distributions being discrete. For example, let us say that there are five users in our system, two of which have some attribute and three who do not. Let us also say that the two users with the attribute each have different policies. For the distributions to be identical, each of those policies would need to be selected by one and a half of the remaining three users. This, obviously, cannot happen. We refer to this difference as rounding error.

The second is difference due to the natural unevenness of random selection. The distributions tend towards evenness as the number of samples increases, but with any finite number of users, the distributions are quite likely to vary some from the ideal.

These differences can both be quantified the same way: as a difference between the expected number of principals who have a policy and the actual number. If the opponent knows that one half of the principals have an attribute and one half do not, and they observe that among four users, there are two policies, one of which is held by three users and the other by one user, then they can know that the user with the unique policy holds the attribute. In general, any time the number of users who share a policy is less than the expectation, it is more likely that a user who has that policy also has the attribute. Information is leaked when there is a difference between the expected number of principals who have a policy and the actual number of principals who have that policy in proportion to the ratio between them.

**THEOREM 2.** *The limit of the difference between the expected number of principals who have a policy and the actual number of principals who have the policy as the number of users goes to infinity is 0.*

The proof of Theorem 2 can be found in Appendix B. The intuition behind it is that as the number of samples grows very large, the actual distribution approaches the ideal distribution and the rounding errors shrink towards zero.

### 3.2 Attacks and Countermeasures

Until now, we have only proven things about a system which is assumed to be in some instantaneous unchanging state. In the real world we have to deal with issues related to how policies change over time and multiple interactions.

Therefore, we also want the policy which a given user randomly selects from the database to be persistent. Otherwise an adversary would simply make multiple requests to the same user over time and see if the policy changed. If it did, especially if it changed erratically, it would indicate that the

user was repeatedly drawing random policies. Instead, the user should hold some value which designates which policy the user has.

An obvious answer would be to have the user hold onto the policy itself, but this would open the user up to a new attack. If users lacking a given attribute simply grabbed onto a policy and never changed it, this itself could be a tell. If there were some event which occurred which made having a given attribute suddenly more sensitive than it used to be, then rational users who have the attribute would increase the stringency of their policies. For example, if a country undertook an action which was politically unpopular on a global scale, holders of passports issued by that country would likely consider that more sensitive information now and would increase their policies appropriately. The result would then be that the average policy for people who had cached a previously fetched policy would then be less stringent than those who were making their own policies.

Instead of a permanent policy, it would be more sensible for a principal to receive a cookie which could get it the policy from a particular principal so that when principals who possess the attribute changed their policies, principals who do not possess it would too.

We also need to guard against stacking the deck. Obviously we can restrict the database to users who actually have the attribute by requiring the presentation of a pseudonymous certificate [6, 7, 8, 9, 10, 18] which proves that they have the attribute. However, we also need to assure that a legitimate attribute holder cannot submit multiple policies in order to skew the set of policies. To this end, we require that each policy be submitted initially with a one-time-show pseudonymous credential [8]. The attribute authorities can be restricted so that they will only issue each user a single one-time-show pseudonymous credential for each Policy Database use. Then we can accept the policy, knowing it to come from a unique user who holds the attribute, and issue them a secret key which they can later use to verify that they were the submitter of a given policy and to replace it with an updated policy.

This does not prevent a user who has the attribute from submitting a single false policy, perhaps one which is distinctly different from normal policies. The result would be that users who draw that policy would be known to not have the attribute. However, under the assumptions of our system, not having the attribute is not sensitive, so this does not compromise safety.

### 3.3 Limitations

We assume that for the attribute being protected, it is not significantly sensitive to lack the attribute. This assumption means that our system likely cannot be used in practice to protect all attributes. Most notably it fails when lacking an attribute implies having or being highly likely to have some other attribute. For example, not having a valid passport probably means that you are a permanent resident of the country you are currently in (although users could be an illegal immigrants or citizens of a defunct nation).

It also fails when the lack of an attribute is more sensitive than having it. For instance, few people are going to wish to prevent people from knowing that they have graduated from high school, but many would consider their lack of a high school graduation attribute to be sensitive. However, we argue that no system can adequately handle such a case

because those who do have the attribute would likely be unwilling to accept any system which would result in them having to not disclose the attribute when it was useful for them to do so. And if they still easily disclose their attribute, then it becomes impossible for those without to disguise their lack.

Similarly to the Ack Policy system, policy databases also do not generally handle any form of probabilistic inference rule between attributes. The existence of such a rule would likely imply certain relationships between policies which most users would enforce. If the possession of a city library card suggested with strong probability that the user was a city resident, then perhaps all users who have both would have a policy protecting their library card which is stricter than the policy protecting their city residency. However, as there is variety in the policies of individuals, a user could pick a random pair of policies which did not have this property. That would then be a sure tell that he did not actually have both of those attributes.

Another drawback of the system is that it requires a policy database service be available on-line. This decreases the decentralized nature of trust negotiation. However, our approach is still less centralized than Ack Policies, which require that users cooperate to determine a universally accepted Ack policy. And this centralization may be able to be decreased by decentralizing the database itself. Although we discuss the database as if it were a single monolithic entity, it could be made of a number of different entities acting together. The only requirement is that it accepts policies from unique users who have the attribute and distributes them randomly.

## 4. RELATED WORK

The framework of automated trust negotiation was first proposed by Winsborough et al. [24]. Since then, great efforts have been put forward to address challenges in a variety of aspects of trust negotiation. An introduction to trust negotiation and related trust management issues can be found in [25]. As described in detail there, a number of trust negotiation systems and supporting middleware have been proposed and/or implemented in a variety of contexts (e.g., [3, 4, 11, 12, 14, 17, 19]). Information leakage during trust negotiation is studied in [13, 5, 15, 20, 21, 22, 23]. The work by Winsborough and Li has been discussed in detail in previous sections. Next, we discuss several other approaches.

In [20], non-response is proposed as a way to protect possession-sensitive attributes. The basic idea is to have Alice, the owner of a sensitive attribute, act as if she does not have the attribute. Only later when the other party accidentally satisfies her policy for that attribute will Alice disclose that attribute. This approach is easy to deploy in trust negotiation. But clearly it will often cause a potentially successful negotiation to fail because of Alice's conservative response.

Yu and Winslett [26] introduce a technique called policy migration to mitigate the problem of unauthorized inference. In policy migration, Alice dynamically integrates her policies for sensitive attributes with those of other attributes, so that she does not need to explicitly disclose policies for sensitive attributes. Meanwhile, policy migration makes sure that "migrated" policies are logically equivalent to original policies, and thus guarantees the success of the negotiation

whenever possible. On the other hand, policy migration is not a universal solution, in the sense that it may not be applicable to all the possible configurations of a negotiation. Further, it is subject to a variety of attacks. In other words, it only seeks to make unauthorized inference harder instead of preventing it completely.

Most existing trust negotiation frameworks [16, 17, 28] assume that the appropriate access control policies can be shown to Bob when he requests access to Alice's resource. However, realistic access control policies also tend to contain sensitive information, because the details of Alice's policy for the disclosure of a credential  $C$  tends to give hints about  $C$ 's contents. More generally, a company's internal and external policies are part of its corporate assets, and it will not wish to indiscriminately broadcast its policies in their entirety. Several schemes have been proposed to protect the disclosure of sensitive policies. In [4], Bonatti and Samarati suggests dividing a policy into two parts – prerequisite rules and requisite rules. The constraints in a requisite rule will not be disclosed until those in prerequisite rules are satisfied. In [19], Seamons et al. proposed organizing a policy into a directed graph so that constraints in a policy can be disclosed gradually. In [26], access control policies are treated as first-class resources, thus can be protected in the same manner as services and credentials.

Recently, much work has been done on mutual authentication and authorization through the use of cryptographic techniques that offer improved privacy guarantees. For example, Balfanz et al. [1] designed a secret-handshake scheme where two parties reveal their memberships in a group to each other if and only if they belong to the same group. Li et al. [15] proposed a mutual signature verification scheme to solve the problem of cyclic policy interdependency in trust negotiation. Under their scheme, Alice can see the content of Bob's credential signed by a certification authority CA only if she herself has a valid certificate also signed by CA and containing the content she sent to Bob earlier. A similar idea was independently explored by researchers [5, 13] to handle more complex access control policies. Note that approaches based on cryptographic techniques usually impose more constraints on access control policies. Therefore, policy databases are complementary to the above work.

## 5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed a general framework for safety in automated trust negotiation. The framework is based strictly on information gain, instead of on communication. It thus more directly reflects the essence of safe information flow in trust negotiation. We have also shown that some of the existing systems are safe under our framework. Based on the framework, we have presented policy databases, a new, safe trust negotiation system. Compared with existing systems, policy databases do not introduce extra layers of policies. Thus it does not introduce complications to the negotiation between users. Further, policy databases preserve user's autonomy in defining their own policies instead of imposing uniform policies across all users. Therefore it is more flexible and easier to deploy to prevent unauthorized information flow in trust negotiation.

Further, we have discussed a number of practical issues which would be involved in implementing our system.

In the future, we plan to address how our system can be used in the presence of delegated credentials. And we plan to

attempt to broaden the system to account for probabilistic inferences rules which are publicly known.

## 6. REFERENCES

- [1] D. Balfanz, G. Durfee, N. Shankar, D. Smetters, J. Staddon, and H. Wong. Secret Handshakes from Pairing-Based Key Agreements. In *IEEE Symposium on Security and Privacy*, Berkeley, CA, May 2003.
- [2] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The KeyNote Trust Management System Version 2. In *Internet Draft RFC 2704*, September 1999.
- [3] M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures. In *Security Protocols Workshop*, Cambridge, UK, 1998.
- [4] P. Bonatti and P. Samarati. Regulating Service Access and Information Release on the Web. In *Conference on Computer and Communications Security*, Athens, November 2000.
- [5] R.W. Bradshaw, J.E. Holt, and K.E. Seamons. Concealing Complex Policies in Hidden Credentials. In *ACM Conference on Computer and Communications Security*, Washington, DC, October 2004.
- [6] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*. The MIT Press, 2000.
- [7] J. Camenisch and E.V. Herreweghen. Design and Implementation of the *Idemix* Anonymous Credential System. In *ACM Conference on Computer and Communications Security*, Washington D.C., November 2002.
- [8] J. Camenisch and A. Lysyanskaya. Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation. In *EUROCRYPT 2001*, volume 2045 of *Lecture Notes in Computer Science*. Springer, 2001.
- [9] D. Chaum. Security without Identification: Transactions Systems to Make Big Brother Obsolete. *Communications of the ACM*, 24(2), 1985.
- [10] I.B. Damgård. Payment Systems and Credential Mechanism with Provable Security Against Abuse by Individuals. In *CRYPTO'88*, volume 403 of *Lecture Notes in Computer Science*. Springer, 1990.
- [11] A. Herzberg, J. Mihaeli, Y. Mass, D. Naor, and Y. Ravid. Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers. In *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2000.
- [12] A. Hess, J. Jacobson, H. Mills, R. Wamsley, K. Seamons, and B. Smith. Advanced Client/Server Authentication in TLS. In *Network and Distributed System Security Symposium*, San Diego, CA, February 2002.
- [13] J. Holt, R. Bradshaw, K.E. Seamons, and H. Orman. Hidden Credentials. In *ACM Workshop on Privacy in the Electronic Society*, Washington, DC, October 2003.
- [14] W. Johnson, S. Mudumbai, and M. Thompson. Authorization and Attribute Certificates for Widely Distributed Access Control. In *IEEE International Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 1998.
- [15] N. Li, W. Du, and D. Boneh. Oblivious Signature-Based Envelope. In *ACM Symposium on Principles of Distributed Computing*, New York City, NY, July 2003.
- [16] N. Li, J.C. Mitchell, and W. Winsborough. Design of a Role-based Trust-management Framework. In *IEEE Symposium on Security and Privacy*, Berkeley, California, May 2002.
- [17] N. Li, W. Winsborough, and J.C. Mitchell. Distributed Credential Chain Discovery in Trust Management. *Journal of Computer Security*, 11(1), February 2003.
- [18] A. Lysyanskaya, R. Rivest, A. Sahai, and S. Wolf. Pseudonym Systems. In *Selected Areas in Cryptography, 1999*, volume 1758 of *Lecture Notes in Computer Science*. Springer, 2000.
- [19] K. Seamons, M. Winslett, and T. Yu. Limiting the Disclosure of Access Control Policies during Automated Trust Negotiation. In *Network and Distributed System Security Symposium*, San Diego, CA, February 2001.
- [20] K. Seamons, M. Winslett, T. Yu, L. Yu, and R. Jarvis. Protecting Privacy during On-line Trust Negotiation. In *2nd Workshop on Privacy Enhancing Technologies*, San Francisco, CA, April 2002.
- [21] W. Winsborough and N. Li. Protecting Sensitive Attributes in Automated Trust Negotiation. In *ACM Workshop on Privacy in the Electronic Society*, Washington, DC, November 2002.
- [22] W. Winsborough and N. Li. Towards Practical Automated

Trust Negotiation. In *3rd International Workshop on Policies for Distributed Systems and Networks*, Monterey, California, June 2002.

- [23] W. Winsborough and N. Li. Safety in Automated Trust Negotiation. In *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2004.
- [24] W. Winsborough, K. Seamons, and V. Jones. Automated Trust Negotiation. In *DARPA Information Survivability Conference and Exposition*, Hilton Head Island, SC, January 2000.
- [25] M. Winslett, T. Yu, K.E. Seamons, A. Hess, J. Jarvis, B. Smith, and L. Yu. Negotiating Trust on the Web. *IEEE Internet Computing, special issue on trust management*, 6(6), November 2002.
- [26] T. Yu and M. Winslett. A Unified Scheme for Resource Protection in Automated Trust Negotiation. In *IEEE Symposium on Security and Privacy*, Oakland, CA, May 2003.
- [27] T. Yu and M. Winslett. Policy Migration for Sensitive Credentials in Trust Negotiation. In *ACM Workshop on Privacy in the Electronic Society*, Washington, DC, October 2003.
- [28] T. Yu, M. Winslett, and K. Seamons. Supporting Structured Credentials and Sensitive Policies through Interoperable Strategies in Automated Trust Negotiation. *ACM Transactions on Information and System Security*, 6(1), February 2003.

## APPENDIX

### A. PROOF OF THEOREM 1

Our goal is to prove the following theorem:

There exists no opponent which can beat the a priori odds of guessing the value of an object,  $o$  given only information about objects which are not in the same inference component as  $o$ , over all principals not in  $M$  and whose policy for  $o$   $M$  cannot satisfy, over all random tapes, and over all mappings of public key values to principals.

Now it follows that if the opponent can beat the a priori odds of guessing the value of an object,  $o$ , then the opponent can beat the a priori odds of guessing the parity of  $o$ . Hence, if no opponent can beat the a priori odds of guessing the parity of an object, then none can beat the odds of guessing the value of the object.

LEMMA 1. *There exists no opponent which can beat the a priori odds of guessing the parity of an object,  $o$  given only information about objects which are not in the same inference component as  $o$ , over all principals not in  $M$  and whose policy for  $o$   $M$  cannot satisfy, over all random tapes, and over all mappings of public key values to principals.*

In order to prove this, we begin with an assumption that there exists some tactic which can successfully guess the parity of  $o$  with odds better than the a priori odds for at least some public key mappings. We are going to prove that any such tactic cannot beat the a priori odds on average across all mappings because there must be more mappings where it fails to beat the a priori odds than where it beats them.

Just to be clear, the tactic is allowed to interact with principals whose policy for  $o$  it can satisfy. It just does not get to guess about the value of  $o$  for those principals, as it is entitled to beat the a priori odds for them. Hence, doing so is not considered a leakage in the system.

Because the tactic is a deterministic Turing-equivalent computational machine, when it outputs its final guesses, it must output them in some order. We will define  $n$  to be the number of users,  $|\mathcal{K}|$ . We will number the series of principals  $k_1, k_2, \dots, k_n$ . Without loss of generality, we can

assume that every principal's strategy's random tape has some fixed value, resulting in them behaving in a strictly deterministic manner. Therefore, as the tactic and strategies are deterministic, the only remaining variable is the mapping of public keys to principals.

Next we will fix the sequence of public keys. Because public keys are randomly chosen to begin with, and we are varying over the set of all public-key to user mappings, we can do this without loss of generality. The order in which guesses are made must in some way depend only on the a priori knowledge, the public keys, and the communications which the tactic has with the strategies. So, if all of these things are kept constant, the guesses will not change.

Let us suppose that a fraction  $h$  of the population whose policy for  $o$  has not been satisfied has one parity value, and a fraction  $1 - h$  of the population has the other. Without loss of generality, we assume that  $h \geq 1 - h$ . We determine  $h$  by calculating the relative a priori probabilities given the distribution of the values of the object.

The a priori probability of successfully guessing which parity a given user's object has is  $h$ . Now, if there exists some order of interaction,  $i$  which beats the a priori odds, then its number of correct guesses must be expressible as  $hn + \Delta$  for some  $\Delta > 0$ .

We can break the set of users whose policies for  $o$   $M$  cannot meet down into a group of sets according to the values of the objects which are in inference components other than the one which contains  $o$ . We will define a set of sets,  $VG$  such that  $vg \in VG$  is a set of users all of which have the same values for all objects in all inference components other than the one which contains  $o$ .

Now, let us consider the possibility of rearranging the public keys of members of this group. Because the strategies in use are defined to be deterministic with respect to the policies governing the attributes which distinguish the two configurations and because the opponent is defined to be deterministic: it follows that if we were to rearrange user's public keys from the original mapping to create a new mapping, the communication would be the same in both. Since the communication would be the same, it follows that the tactic would make the same guesses relative to the order of users because it is a deterministic machine and must produce the same output given the same input, the end result of which is that switching two users both of whom are members of the same value group will result in the guesses of the parity of those two users switching as well.

We can then consider the set of all arrangements of public keys formed by switching principals around within their value groups, which we shall call  $I$ . So the question at hand, then, is whether or not the expected value of  $\Delta$  across all members of  $I$  is positive. If we can demonstrate that it is not, then no successful opponent can exist.

Here we introduce another lemma. Proof of this lemma is now sufficient to establish our earlier lemma.

LEMMA 2. *The expected value of  $\Delta$  across all public key mappings is less than or equal to zero.*

If we have some quantity of extra correct guesses,  $\Delta$ , for some public key mapping  $i$ , then these guesses must be distributed over some set of value groups. If  $\Delta$  is to be positive on average, then at least some value groups must average a number of correct guesses above the a priori probability over all arrangements in  $I$ .

Let us assume that we have one such group  $vg$ . Because the distributions of values of items in other inference components are defined to be precisely independent of  $o$ , we can know that in each group, there must be a fraction  $h$  of the members which have one parity and  $1 - h$  which have the other. So, in  $vg$  there will be  $x = h|vg|$  principals with the first parity and  $y = (1 - h)|vg|$  principals with the second, and the a priori expected number of correct guesses would be  $x$ .

If, for some mapping,  $i$ , the tactic is successful, then there must be some number of correct guesses  $x + \delta$  where  $\delta > 0$ . We also know that  $\delta \leq y$  simply because the tactic is limited in total correct guesses to  $|vg| = x + y$ . As the number of correct guesses is  $x + \delta$ , it must follow that the number of incorrect guesses is  $y - \delta$ .

Further, we need to note that the tactic must make some quantity of first parity guesses and some quantity of second parity guesses. Obviously, these quantities need to add up to  $|vg|$ , but need not match up with  $x$  and  $y$ . Every extra first parity or second parity guess guarantees at least one mistake, but even with several mistakes, it is quite possible to beat the a priori odds for some arrangements. So we define  $x + c$  to be the number of first parity guesses and  $y - c$  to be the number of second parity guesses.

Now, we know that each increase of one in  $|c|$  guarantees at least one wrong guess, so we have a bound of  $\delta + |c| \leq y$ . Further, we know that since  $c$  is fixed (as it is not dependent on the arrangement, only the guesses which are unchanging), the only way to gain a wrong guess is to swap a first parity principal with a second parity principal, which must necessarily create two wrong guesses. So we can quantify the number of wrong first parity guesses and the number of wrong second parity guesses using the terms we have set up. Specifically, there must be  $\frac{1}{2}(y - \delta + c)$  incorrect first parity guesses, and  $\frac{1}{2}(y - \delta - c)$  incorrect second parity guesses.

Now we can determine the number of arrangements of principals which will create  $x + \delta$  correct guesses. Specifically, we look at the total number of principals which are first parity and choose a way to arrange them to match up with incorrect second parity guesses and we look at the total number of principals which are second parity and choose a way to arrange them to match up with incorrect first parity guesses. Then we multiply that by the number of permutations of first parity principals and the number of permutations of second parity principals. And we arrive at  $\binom{x}{\frac{1}{2}(y - \delta - c)} \binom{y}{\frac{1}{2}(y - \delta + c)} x! y!$ .

Now, similarly, we can calculate the number of arrangements which will result in  $x - \delta$  correct answers. And if for all  $\delta$  there are at least as many arrangements which produce  $x - \delta$  correct answers as produce  $x + \delta$  of them then the average of  $\delta$  cannot exceed 0. Now, if there are  $x - \delta$  correct answers, then there must be  $y + \delta$  incorrect ones. And we can use the same reasoning to establish that there must be  $\frac{1}{2}(y + \delta + c)$  incorrect first parity guesses and  $\frac{1}{2}(y + \delta - c)$  incorrect second parity guesses, and hence  $\binom{x}{\frac{1}{2}(y + \delta - c)} \binom{y}{\frac{1}{2}(y + \delta + c)} x! y!$  arrangements which result in  $x - \delta$  correct guesses. So if we can prove that this is no less than the previous quantity then our proof will be complete.

$$\begin{aligned} \binom{x}{\frac{1}{2}(y + \delta - c)} \binom{y}{\frac{1}{2}(y + \delta + c)} x! y! &\leq \binom{x}{\frac{1}{2}(y + \delta - c)} \binom{y}{\frac{1}{2}(y + \delta + c)} x! y! \Leftrightarrow \\ \binom{x}{\frac{1}{2}(y + \delta - c)} \binom{y}{\frac{1}{2}(y + \delta + c)} &\leq \binom{x}{\frac{1}{2}(y + \delta - c)} \binom{y}{\frac{1}{2}(y + \delta + c)} \Leftrightarrow \\ \frac{x!}{\frac{1}{2}(y + \delta - c)! (x - \frac{1}{2}(y + \delta - c))!} \frac{y!}{(\frac{1}{2}(y + \delta + c))!} &\leq \end{aligned}$$

$$\begin{aligned} \frac{x!}{(\frac{1}{2}(y + \delta - c))! (x - \frac{1}{2}(y + \delta - c))!} \frac{y!}{(\frac{1}{2}(y + \delta + c))! (y - \frac{1}{2}(y + \delta + c))!} &\Leftrightarrow \\ \frac{1}{(\frac{1}{2}(y + \delta - c))! (x - \frac{1}{2}(y + \delta - c))!} \frac{1}{(\frac{1}{2}(y + \delta + c))! (y - \frac{1}{2}(y + \delta + c))!} &\leq \\ \frac{1}{(\frac{1}{2}(y + \delta - c))! (x - \frac{1}{2}(y + \delta - c))!} \frac{1}{(\frac{1}{2}(y + \delta + c))! (y - \frac{1}{2}(y + \delta + c))!} &\Leftrightarrow \\ \frac{1}{(x - \frac{1}{2}(y + \delta - c))! (\frac{1}{2}(y - \delta + c))!} \frac{1}{(x - \frac{1}{2}(y + \delta - c))! (\frac{1}{2}(y + \delta + c))!} &\Leftrightarrow \\ (x - \frac{1}{2}(y - \delta - c))! (\frac{1}{2}(y - \delta + c))! \geq (x - \frac{1}{2}(y + \delta - c))! (\frac{1}{2}(y + & \\ \delta + c))! \Leftrightarrow \frac{(x - \frac{1}{2}(y - \delta - c))!}{(x - \frac{1}{2}(y + \delta - c))!} \geq \frac{(\frac{1}{2}(y + \delta + c))!}{(\frac{1}{2}(y - \delta + c))!} \Leftrightarrow \frac{(x - \frac{1}{2}(y - \delta - c))!}{(x - \frac{1}{2}(y - \delta - c) - \delta)!} \geq & \\ \frac{(\frac{1}{2}(y + \delta + c))!}{(\frac{1}{2}(y + \delta + c) - \delta)!} & \end{aligned}$$

We define a function  $f(a, k) = a! / (a - k)!$ , i.e. the product starting from  $a$  going down  $k$  integers. And obviously  $a \geq b \Rightarrow f(a, k) \geq f(b, k), \forall b \geq k \geq 0$ .

Then we can rewrite the last inequality as  $f(x - \frac{1}{2}(y - \delta - c), \delta) \geq f(\frac{1}{2}(y + \delta + c), \delta)$ , which, noting that  $\delta \geq 0$  and  $y \geq \delta + |c| \Rightarrow y \geq \delta - c \Rightarrow y + c + \delta \geq 2\delta \Rightarrow \frac{1}{2}(y + c + \delta) \geq \delta$ , is implied by  $x - \frac{1}{2}(y - \delta - c) \geq \frac{1}{2}(y + \delta + c) \Leftrightarrow x - \frac{1}{2}y \geq \frac{1}{2}y \Leftrightarrow x \geq y \Leftrightarrow h|vg| \geq (1 - h)|vg| \Leftrightarrow h \geq (1 - h)$  which we know to be true from our assumption at the start of the proof.

So we have proven lemma 2, and this completes the proof.

## B. PROOF OF THEOREM 2

We define  $n$  to be the number of users,  $|K|$ . Because we assume that this system is in a fixed state, every user  $k$  is in some configuration  $g_k$ . Now let us examine some particular attribute,  $t$ . We know that a fraction  $h$  of users have that attribute and  $1 - h$  do not. Let us define a set of policies  $L = \{p | \forall t \in \mathcal{T}, k \in K \exists q_{s_t} \in \mathcal{Q} \text{ such that } p = q_{s_t}(k)\}$ . We also need to know the fraction of users who have each policy in  $L$ . As the number of users grows towards infinity, the number of possible policies stays finite, so multiple users with the attribute will wind up sharing the same policy. For every member  $l \in L$ , we define  $f_l$  to be the fraction of users with attribute  $t$  who have policy  $l$ .  $\sum_{l \in L} f_l = 1$ . We assume that as  $n$  approaches infinity,  $f_l$  approaches some fixed quantity  $\hat{f}_l$  for every  $l \in L$ . Essentially, what we are assuming is that there is a fixed fraction of users with the attribute who will chose any given policy. The particular number will vary at any given time, but over time, we will approach this fraction. We should then know that for some particular policy  $l$ , the odds of a user without the attribute drawing policy  $l$  are also  $f_l$  because policies are handed out with the same distribution that they are submitted.

The distribution which describes how many users we are actually going to have with this policy is a binomial distribution. The variance of a binomial distribution is  $\sigma^2 = n(1 - h)f_l(1 - f_l)$ . The difference between the actual and the ideal is the square root of the variance divided by the expected number of users who have a given policy, which is  $n f_l$ . Hence, the expected difference between our practical system and the ideal system is  $\frac{\sqrt{n(1-h)f_l(1-f_l)}}{n f_l} = \sqrt{\frac{(1-h)(1-f_l)}{n f_l}}$ .

$1 - h$  is a constant term, and  $f_l$  will approach  $\hat{f}_l$ , which is a fixed quantity. So  $\lim_{n \rightarrow \infty} \frac{(1-h)(1-f_l)}{n f_l} = 0$ , and we have proven that our system approaches the ideal as the number of users goes to infinity.