

Parameterized Authentication^{*}

Michael J. Covington, Mustaque Ahamad, Irfan Essa, and H. Venkateswaran

College of Computing, Georgia Institute of Technology, Atlanta, Georgia USA

Abstract. We describe an approach to sensor-based authentication that can adapt to accommodate incomplete, unreliable, or inaccurate input provided to the system. *Parameterized Authentication* moves beyond the traditional approach to security by acknowledging that identity verification cannot always produce perfect results. Our model addresses such inherent imperfections by introducing a metric, the Authentication Parameter, that captures the overall “quality” of authentication. We define authentication “quality” in terms of sensor trustworthiness and the accuracy of sensor measurements. Using the Authentication Parameter, we are able to enforce and enhance the principle of least privilege by ensuring that the authentication process provides credentials that are sufficient but not stronger than the access level required by the requested operation. This approach is particularly well-suited to meet the demands of a context-aware and pervasive computing environment in which authentication may be performed using passive and non-intrusive techniques. Our model supports the transparent capture of authentication-relevant information from the environment and provides a foundation for generating dynamic credentials for sources of requests. We present our model, discuss its contributions, and illustrate how it can be used to support rich access control policies.

1 Introduction

Authentication is a fundamental building block in any system that enforces a security policy; it enables “principals” to identify themselves to the system and provides a foundation for access control. For the purposes of this paper, the principals we consider are users, though their attributes such as location, role, or history may also be relevant in authorization decision-making. All authentication schemes follow the same basic approach: known identification information about a principal is compared with information received from the source claiming to be that principal. Authentication is successful if both pieces of information match; however, authentication failure will result if a match cannot be produced.

Pervasive computing environments strive to provide transparent access to resources and services. For example, the Aware Home [1], a prototype “home of the future” that has been built at Georgia Tech, is exploring a variety of emerging applications that range from remote appliance management (e.g., Cyberfridge [2]) to “awareness” and the active-monitoring of each resident’s activities and

^{*} This work was supported, in part, by NSF awards ITR-0081276 and ITR-0121643.

needs. The prototype home has a rich computation and communication infrastructure and will eventually be connected to other homes and institutions in the community. A variety of sensors are used to infer the activities of the home’s residents and various applications use this information to help improve the quality of life for residents.

Clearly, the assumption that a principal’s identity can be verified with absolute certainty is impractical in real world scenarios, even when explicit interaction is required. The Aware Home is an example of an environment in which sensors will be used for user identification and verification purposes. Homeowners are forced to balance authentication quality with financial limitations and a tolerance for being inconvenienced. For example, the Smart Floor [3] is currently being deployed into the Aware Home despite its less-than-perfect accuracy because it is non-intrusive and easy-to-use.

Non-binary authentication could be used to limit the damage that may result from an erroneous authentication. We have designed a model that can be used to produce an authentication measure from incomplete, unreliable, or inaccurate identification information that is provided by a set of sensors. We accomplish this by providing a quality measure for authentication. In addition, we provide a method for computing an authentication value by combining inputs from multiple sources; by reinforcing authentication and forming a consensus, our authentication framework is more robust and than those that rely on a single source for authentication. We refer to this approach as *parameterized authentication*.

This paper is organized as follows: Section 2 presents related work. In section 3, we discuss the various logical components that comprise our model, identify a set of design principles that guide the development of our model, and introduce the *Authentication Parameter*. Sect. 4 details our approach to managing trust and accuracy in the system and illustrates how these measures are used to produce the Authentication Parameter. We revisit our design principles in section 5 to discuss how well our model meets these principles. We discuss several outstanding issues and related research contributions in section 6.

2 Related Work

Our approach to computing an Authentication Parameter (AP) value makes use of techniques that have been explored in diverse research areas, including distributed authentication, sensor matching and fusion, and trust and reputation management. In this section, we introduce relevant background material necessary for understanding the research contributions described in the remainder of this document.

Distributed Authentication In sensor-based authentication, identification information is collected from multiple sensors that may be trusted to different degrees. In distributed system environments that span multiple trust domains, authentication may rely on certificates that are issued by certification authorities (CAs) that also have different levels of trust associated with them. Several researchers

have explored such models where an authentication measure based on the trust level of the CAs is derived. Beth et al. [4] present a model where trust can be combined from multiple CAs. Reiter and Stubblebine [5] explore the design principles that must be followed by such a model. Maurer [6], Jøsang [7] and others have explored additional techniques to compute an authentication metric based on paths or chains of CAs that are used for authentication. These techniques primarily focus on trust of the relevant CAs and do not address accuracy of identification information.

Sensor Matching and Fusion In interactive intelligent environments such as the home, biometric technologies are often used to obtain user identification with minimal explicit input. Unfortunately, many of the biometric devices widely available today cannot guarantee very high quality identification. This is typically a result of noise that interferes with sensor readings, limitations of the processing methods or the variability in both the biometric characteristic as well as its presentation [8]. For instance, biometric device test reports [9, 10] discuss biometric sensors that can be easily defeated, including a fingerprint sensor that can misinterpret imprints in a Gummi Bear candy as a valid fingertip scan and a vision system that can be defeated by a photograph of an authorized user. Such weaknesses in biometric technology create opportunities for an impostor to “mimic” the actions of a legitimate user and, in essence, trick the system into believing that they are someone else.

Sensor fusion refers to the combining of multiple identification “inputs” in order to produce a single identification metric. For example, one research group has recently incorporated speaker identification and speech recognition systems with a person tracking system to accurately locate a speaker and identify the speaker and what they are saying [11]. This form of sensor fusion yields a more reliable metric in case of uncertainty from an individual sensor. In addition to combined input, sensor fusion allows the system to reason about the fidelity of the composite information. This measure can be used to enhance the strength of the authentication service.

Despite the “stronger” authentication results produced through sensor fusion, they still do not reflect the overall quality of the authentication process. Instead, the results are binary, thus allowing the user to either receive all access rights or none at all. Parameterized authentication provides a more novel approach by incorporating a notion of trust (in individual sensors) into the authentication process and, ultimately, providing a metric that indicates the quality of the authentication process. This metric can be used to adjust the level of authentication. For instance, a user can be authenticated into a role that is based on the strength of her identification, thus ensuring that a user is never allowed to have more access than the evidence provided by them for authentication.

Trust and Reputation Management Trustworthiness is often viewed as the expectation of cooperative behavior and can be based on previous experiences with the same party. However, it is often necessary to evaluate the trustworthiness of an entity without having any prior direct interactions. In such situations, a

participant can place trust based on the latter’s “reputation” among others in the system. This approach bases reputation on the collection of evidence that supports certain claims of good or bad behavior. In parameterized authentication, reputation or sensor trustworthiness can be based on whether a sensor’s input led to correct authentication or a breach of security.

eBay and other similar Internet communities are practical examples of reputation management systems. On eBay’s site, for example, sellers receive feedback (+1, 0, −1) for their reliability in an online auction. Reliability is computed using the feedback values that are collected over a period of several months.

In similar work, Kamvar et al. [12] describe a trust-based algorithm that identifies malicious entities in a peer-to-peer environment and isolates them from the network. Their reputation system, called EigenTrust, identifies inauthentic files on a network and even handles conditions where malicious peers cooperate in an attempt to deliberately compromise the system. Likewise, work by Beth et al. [4] presents a method for the valuation of trustworthiness based on experiences and recommendations.

The mathematical foundations for reputation management are firmly rooted in probability and statistics. Our work draws heavily from Bayesian statistics in which a mechanism for combining evidence is presented.

3 System Model

Our model for parameterized authentication is based on information obtained from a distributed network of sensors. The model is composed of the following logical components: users, sensors, a user-attribute database, an attribute-matching service, a trust analysis engine, an audit service and the authentication service. A high-level overview of our adaptive authentication architecture is given in figure 1. In the following sections, we detail the functionality of these components and describe how they interact with one another.

3.1 Users

Our approach to user authentication assumes an open-world model in which there exist two classes of *Users* – those that are “known” (identifiable) by the system, and those that are not. A user is defined by a collection of *traits*, or properties, that are either non-intrusively captured by sensors or explicitly provided as input to the system. While some users may have similar traits, it is the collection of properties that define a user. By definition, no two collections are equal. Our model can make use of four fundamental trait types: physiological, knowledge-based, object-based and historical.

3.2 Sensors

Our system model consists of distributed sensors that collect identity information that is ultimately provided to an authentication service. Sensors are mechanisms

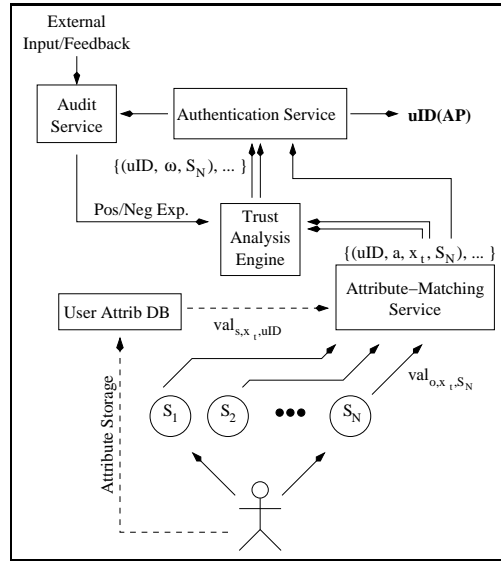


Fig. 1. Overview of Authentication Model

designed to observe and capture user-specific traits from the environment. Some sensors require explicit user participation to complete the capture, while others are less intrusive and can record information without the user’s knowledge or active participation. These sensors observe user traits and forward relevant values to the authentication service where the information is interpreted and used to authenticate users. For example, in figure 1, sensor S_N provides a measured value of trait x_t .

Our authentication framework gathers information from a variety of sensors to infer user characteristics. Examples of sensor technologies being deployed into the Aware Home include the Smart Floor, vision systems (ranging from facial recognition to movement sensors) and voice-recognition. Such sensors can non-intrusively gather user information, albeit less-than-perfect, from the environment.

3.3 User-Attribute Database

We assume the existence of a user-attribute database that maintains, for each user in the system, a collection of traits that define that user. We can think of the traits for user u as a t -dimensional trait vector x_u , where $x_u = (x_{u1}, x_{u2}, \dots, x_{ut})$. In the Aware Home, trait vectors are used to create profiles for each user of the Smart Floor. Traits, in this scenario, are derived from a biomechanics measure known as the ground reaction force (GRF). The trait vector consists of ten footstep profile features that are later used to match unknown footsteps against user’s configured in the database.

3.4 Attribute-Matching Service

The Attribute-Matching Service is responsible for collecting and assembling a trait vector from sensor output, and then computing accuracy by comparing the assembled trait vector with the stored definition for a user. If a collection of sensors assemble a trait vector y , it can be compared directly to the associated values stored in x to determine how well the observed features match those stored in the system.

3.5 Trust Analysis Engine

We define *trust* as a measure of the system's confidence in a particular sensor; it reflects the quality of the information provided by a given sensor in the system. Clearly, this measure can only be refined over time, through a series of experiences that are recorded with the sensor.

We have identified three possible outcomes that can result from an authentication process. *Positive user identification* captures instances in which the correct user is identified or in which an unauthorized user is prevented from obtaining credentials from the system. A *denial of service* results when a legitimate user is prevented from being authenticated due to malicious tampering with sensors or authentication data. Similarly, a *compromise* has taken place when a user obtains credentials belonging to another user. Any interaction that leads to a denial of service or a system compromise is considered to be a negative experience and the trust measures that contribute to a negative experience are subsequently degraded. Likewise, the system attempts to reinforce positive experiences by assigning higher trust values to sensors that consistently provide data to the authentication service which leads to correct authentication.

3.6 Audit Service

On-line scrutiny of all authentication decisions may not be possible for a number of reasons. Therefore, selective authentication decisions may be logged to enable subsequent off-line examination. For example, if a security violation is detected, a log can be used to reconstruct the sequence of interactions that led up to the violation. This would allow the system to revise the experience values associated with the sensors that contributed to the incorrect authentication; instead of contributing to a positive experience, the sensors would receive a negative rating for the interaction. In addition, the audit service would be responsible for processing log information to produce feedback that is sent to the Trust Analysis Engine. This feedback is used to maintain evolving trust values for each of the sensors.

3.7 Authentication Service

The authentication service receives input from the Trust Analysis Engine and the Attribute-Matching Service that is combined to derive the Authentication Parameter. The output from this service is based on input supplied by one

or more sensors. We identify a set of useful design principles that guide the derivation of an Authentication Parameter from the information received.

Principle 1: Accuracy of a sensor. Accuracy measures the similarity between a stored trait value and the value of a trait observed by a single sensor. The accuracy measure should address the need for normalized results so comparisons can be made.

Principle 2: Evolution of Trust through experience. Trust is a measure of confidence held by the authentication service in a particular sensor; it represents a reputation that is established through consistent behavior and observed over a series of experiences, both positive and negative. Trust should increase slowly to allow reputation to build between the authentication service and sensor. Likewise, once negative experiences are detected, trust should decrease quickly to defeat malicious or compromised sensors that attempt to improve their reputation through a short-run of positive performance.

Principle 3: Combining Trust and Accuracy. When trying to determine a user's authenticity, the authentication service analyzes input provided by each sensor. The authentication service forms an opinion by taking sensor input and adjusting it to reflect the current level of trust associated with each sensor. This opinion, generated for each sensor, should reflect both certainties and uncertainties with regard to user identity, giving both trust and accuracy important roles in generating the opinion.

Principle 4: Consensus of Opinions. Once a set of individual opinions have been collected, they must be combined to generate the authentication parameter. When sensor opinions are in agreement, certainty should increase. Conflicts in opinion should be indicated through increased lack of confidence in the authentication decision.

Principle 5: Evaluation order independence. The derived conclusions of the model, namely the Authentication Parameter, should be independent of the order in which sensor input is considered. This principle allows additional sensor inputs to be factored into an Authentication Parameter that has already been computed.

Principle 6: Property dependence and feedback. The value of the model's trust and accuracy parameters should impact the feedback cycle in which trust is recomputed and returned to the system.

Principle 7: Robustness. The authentication parameter should be designed to be resilient to long-term manipulation of its model by misbehaving entities, and its sensitivity to various forms of misbehavior should be made explicit.

4 Deriving the AP

The authentication service builds an authentication opinion by collecting input from one or more sensors with information relevant to the current user or request. An opinion is formed for each source of input and has two measures that impact its outcome. The first measure, *accuracy*, measures the similarity between a user trait that is observed by a sensor and the value of the same trait that is

stored in a user signature. The second measure, *trust*, represents the reputation a sensor has with the authentication service; it measures consistency between past results and the output of an individual sensor. In this section, we further describe accuracy and trust and show how they can be combined to produce the Authentication Parameter.

4.1 Accuracy

In our model, *accuracy* is defined as a similarity measure between a stored trait value and an observed trait value. When a sensor provides input to the system, that input data is compared with stored identity data to determine if a match exists. In order to account for variations between stored and observed data, a perfect match is not always required by the system. The closeness or quality of a match, therefore, is reflected in the accuracy value.

In order to assess how well an observed trait x^o matches a stored trait x^s , we consider a distance measure $d(x^o, x^s)$ such that

$$d(x^o, x^s) = \begin{cases} \textit{large} & \text{when } x^o, x^s \Rightarrow \textit{mismatched traits} \\ \textit{small} & \text{when } x^o, x^s \Rightarrow \textit{similar traits} \end{cases}$$

The most obvious measure of similarity (or dissimilarity) between two measurements is the distance between them. Similarity and distance measures have been explored in a variety of domains (e.g., [13, 14] and can be used to compare one or more identifying features. Some comparison studies exist among similarity measures and indicate that different similarity measures perform best when coupled with the appropriate set of attributes. Clearly, the method for computing accuracy is implementation-specific.

An example of accuracy measurements being used in the Aware Home can be found in the Smart Floor. In modeling each individual’s footsteps, the Smart Floor designers chose ten footstep profile features to use as markers in an overall user profile. References are made to a combination of user, foot, and shoe type as a condition (e.g., “Joe’s left foot while he was wearing tennis shoes”). One instance of a user’s footstep constitutes one cluster of data in a training set. This cluster is then used to calculate a Euclidean distance from an unidentified footstep. The identity of the cluster with the lowest average distance is chosen as the identity of the unknown footstep.

There are many approaches for computing distances in order to obtain an accuracy measure. Some instances will require that the attribute-matching service produce a distance measure between only two points. Other instances will require that a collection of identifiers or traits be used in the computation. For instance, a vision-based sensor will attempt to collect a series of traits that are combined to produce a single user identifier. This collection forms a vector that is compared with one stored by the system. Other sensors will produce more primitive output that will require less intensive comparisons.

4.2 Trust

The authentication service, when taking input from multiple sensors, may be presented with conflicting information that could either represent a compromised sensor, a failed sensor or a malicious agent trying to access the system. We make use of techniques from Bayesian statistics to derive and maintain the trust metrics for individual sensors. Bayes' theorem is based on the subjective definition of probability as "degrees of belief." This approach assigns probabilities to hypotheses, allowing for the combination of *a priori* judgments and experimental information. Trust, as defined in our model, is not a static value; it involves a set of uncertainties that are refined over time through experiences and interactions. This makes Bayesian logic a natural mechanism for evolving trust values in our model.

After defining the initial hypothesis and the associated probability for each sensor, Bayes' approach attempts to refine trust by calculating the effect of a correlated event on the original hypothesis. Our model focuses on an event E that captures the outcome of an authentication experience. We define E to be a binary event that is either *positive* or *negative*. The experience is positive if the sensor produces output that is correct (e.g., correctly identifies a user) and is negative if the sensor produces output that is incorrect (e.g., identifies an intruder as a known, authorized user). We then compute the following probability using Bayes' theorem, where T is the hypothesis or current trust value, and E is the value of the experience:

$$P(T|E) = \frac{P(E|T) \cdot P(T)}{P(E|T) \cdot P(T) + P(E|\neg T) \cdot P(\neg T)}$$

If the event is positive then $P(E|T) = P(+E|T)$ which is defined to be a , where a is the accuracy associated with the reading. Likewise, for a negative experience, $P(E|T) = P(-E|T)$ which is defined to be $(1 - a)$.

Similarly, we define $P(E|\neg T)$ in terms of the experience. If the experience is positive we define $P(+E|\neg T) = \alpha$. If the experience is negative we define $P(-E|\neg T) = (1 - \alpha)$. The value for $P(E|\neg T)$ reflects the probability that a (positive/negative) event will occur, even when the sensor is *not* trustworthy (e.g., when the hypothesis is invalid). The value for α is predetermined and indicates a lower-bound threshold for untrusted activity. Our model assumes that a compromised sensor will provide some positive experiences as it attempts to evade detection and exclusion. A high value for α implies that when a sensor is compromised, its malicious behavior will be caught quickly and it will no longer be used in authentication decisions. Thus, if a compromised sensor wants to damage the system over a longer period of time, it must limit incorrect results that it provides and behave more like a good sensor.

Once $P(T|E)$ is computed, it provides a revised trust value for a given sensor. This updated measure replaces the current trust value, $P(T)$, and is used in future refinements under the Bayes approach. We previously defined Design Principle 6 in which we detailed a "feedback cycle" for recomputing trust and returning

it to the system. Our approach to evolving trust by computing $P(T) = P(T|E)$ provides this essential feedback cycle and allows us to react accordingly when sensors malfunction or misbehave.

In practice, determining the quality of E is difficult. We currently rely on both consensus results and an out-of-band audit analysis to determine the quality of an experience. For instance, if multiple sensors produce output for a single authentication request and a majority of those sensors support a similar user profile, those in the minority will be flagged as providing a negative experience. Likewise, if an out-of-band audit log review determines that a compromise has occurred, the sensors that contributed to the false authentication will have their trust values degraded through a negative experience.

Furthermore, evaluating and revising T after every authentication event may be impractical or impossible. If a sensor is overwhelmed with incoming authentication requests, recomputing T after each one may slow response time. Furthermore, it is difficult to determine the quality of an experience in real-time as the system may require information from outside sources (e.g., audit logs).

Using our function for trust evolution, the trust after n experiences (t_n), out of which k are positive and m are negative, can be written as follows. The base trust value t_0 is assigned to each individual sensor by the security administrator.

$$t_n = \frac{(\prod_{i=1}^k a_i) \cdot (\prod_{j=1}^m (1 - a_j)) \cdot t_0}{(\prod_{i=1}^k a_i) \cdot (\prod_{j=1}^m (1 - a_j)) \cdot t_0 + \alpha^k \cdot (1 - \alpha)^m \cdot (1 - t_0)}$$

4.3 Authentication Parameter

Jøsang defines a framework for artificial reasoning called Subjective Logic [7] that consists of a belief model called opinion space. Subjective Logic was developed to mathematically describe and manipulate subjective beliefs; it is an extension of standard logic that uses continuous uncertainty and belief parameters instead of only discrete truth values. We have used the Subject Logic framework for Parameterized Authentication as it provides a foundation for the handling of uncertainties and the forming of conclusions based on insufficient evidence.

Similar to Jøsang’s approach, we assume that knowledge about the world (obtained through a sensor) is never perfect and it may be impossible to verify a user’s identity with absolute certainty. Given this imperfect knowledge, it is impossible to know authoritatively whether a user has been properly identified, so a sensor can only have an opinion about the observation. For a single opinion about a user’s authentication, we assume that

$$b + d + u = 1, \quad \{b, d, u\} \in [0, 1]$$

where b , d , and u represent belief, disbelief and uncertainty respectively. A situation in which there is zero uncertainty is equivalent to the traditional probability model.

Our method for assigning values to the $\{b, d, u\}$ -tuple differs from that proposed by Jøsang. His approach involves mapping the opinion space to an evidence

space that consists of a probability certainty density function. Our evidence space, however, consists of trust and accuracy measures obtained from sensors. Therefore, we let $\omega_x = \{b_x, d_x, u_x\}$ be a single sensor’s opinion about the authentication of user x . We now define ω_x as a function of trust and accuracy measures that have been obtained from the sensor:

$$\omega_x = \begin{cases} b_x = t \cdot a \\ d_x = t \cdot (1 - a) \\ u_x = (1 - t) \end{cases}$$

Here, t is a measure of trust for the sensor that is providing the authentication data and a is the accuracy of the observed trait. Our definition for ω_x makes it clear that belief and disbelief are functions of both accuracy and trust, whereas uncertainty results from the lack of trustworthiness. For example, belief will be high only when both match accuracy and sensor trust are high.

Subjective Logic contains several different operations, with the most relevant to our work being *consensus*. A consensus opinion consists of combining two or more independent opinions about the same proposition (e.g., “The user’s identity as *Bob* can be verified”) into a single opinion.

The Authentication Parameter is found in the final ω_x after consensus has been computed for all relevant opinions. The value of interest is the b_x that reflects the overall belief in the user’s authenticity based on input from multiple sensors. However, this value alone is insufficient to describe the quality of authentication. The entire $\{b, d, u\}$ -tuple is retained for a variety of reasons, including the addition of supplemental input through the consensus operation and the comparison of results.

5 Validation

Ideally, validation is done by using actual sensors deployed in a real environment in which users are authenticated based on sensor-obtained data. Unfortunately, we currently do not have access to such an infrastructure. We discuss the validity of our approach by examining how well it met the original design principles that were presented by us in section 3.7.

Design Principle 1: Accuracy of a sensor The accuracy measure we define is the distance measure between an observed trait and a stored trait. The open method presented in section 4.1 to compute authentication accuracy meets the guidelines presented in our model’s design principles. We do not restrict the design of an accuracy function and acknowledge that implementations will vary widely depending on the sensor technology being used.

Design Principle 2: Evolution of Trust through Experience The following figures (figure 2 for positive experiences and figure 3 for negative experiences) demonstrate how trust evolves for various sensor “types” as experiences are encountered over a period of time. In figures 2 and 3, a sensor is defined with an

initial trust value of $t = 0.9$, a sensor reading rated with accuracy $a = 0.9$, and an administrative α -value set to $\alpha = 0.6$. The solid line in the plot shows how trust changes when experiences are encountered with the sensor. Likewise, the dotted line shows a similar evolution process for trust with a lower setting of the initial trust value.

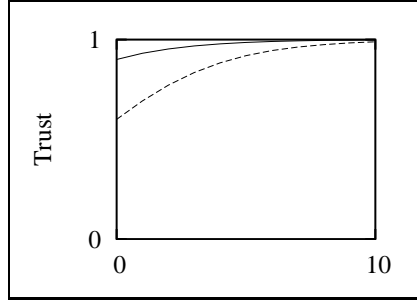


Fig. 2. 10 Positive Experiences

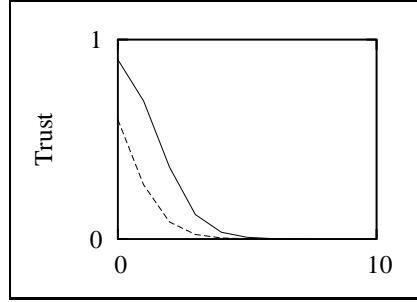


Fig. 3. 10 Negative Experiences

As illustrated through these figures, trust is slow to build and quick to degrade. These results follow in line with the expectations set forth in the second design principle. These properties allow for trust reputation to build over time, while quickly punishing any malicious or compromised components.

Figure 4 provides a look at trust evolution when both positive and negative experiences are encountered in a single sensor. The solid line illustrates how trust is increased when a series of 5 consecutive positive experiences are encountered and, subsequently, how trust is quickly degraded when it is followed by 5 consecutive negative experiences. The dotted line shows how trust is decreased during the time period when a random collection of experiences occur (e.g., an equal number of alternating positive and negative experiences). As above, these properties remain consistent with our design principles: trust is slow to build, quick to fall, and our model always errs on the side of caution – random experiences reflect an inconsistency with the sensor and cause trust to degrade.

Design Principle 3: Combining Trust and Accuracy By utilizing the Subjective Logic framework, our model is capable of describing and manipulating subjective beliefs, or sensor opinions. Our evidence space is comprised of the trust values maintained for each sensor and the accuracy of the each sensor output.

The consensus operation described in section 4.3 allows for the combination of two independent opinions. Clearly, an opinion consists of more than just trust and accuracy. By generating belief, disbelief and uncertainty values, the resulting tuple reflects a multitude of information that could not be conveyed with a single value. Combining trust and accuracy in this manner reflects both certainties and uncertainties, without placing any undue weight or emphasis on a particular

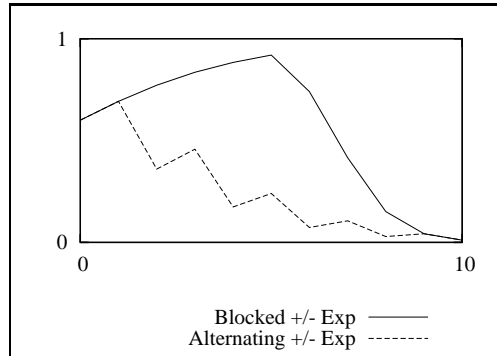


Fig. 4. 10 Alternating Positive/Negative Experiences

component of the model. These properties are aligned with the design principles that govern the combination of trust and accuracy.

Design Principle 4: Consensus of Opinions Using Jøsang’s model for consensus [7] we can compute how an authentication parameter is effected when more sensors are used to compute an authentication decision. A “good” sensor is one with a high trust, or belief, value. A “bad” sensor is one that has low trust and high distrust. With both sensor types, uncertainty results from the lack of perfect trust in sensors and perfect accuracy from their readings.

The results from the consensus operation are consistent with design principle 4 in that certainty only goes up when the majority of opinions are in agreement. Any conflicts or increased disbelief cause a decline in confidence and, as a result, a lowering of the belief value in the authentication parameter.

Design Principle 5: Evaluation order independence. The fifth design principle can be validated by showing that order does not matter in evaluating trust. In section 4.2, we demonstrated that the trust after n experiences can be expressed as a function, with k positive and m negative experiences. This approach enables us to produce a trust value that is based on multiple experiences and computed in any order. Since trust is based on the collection of experiences, regardless of order, this design principle applies to our model and ensures that trust can be computed efficiently.

Design Principle 6: Property dependence and feedback. Our approach for managing trust using Bayesian statistics takes into account current trust values and accuracy readings. In addition, the model provides a feedback loop that evaluates the quality of an experience and incorporates this into an evolving trust value. By allowing trust parameters to evolve and accuracy readings to impact feedback cycles, our model is protected from misbehaving and malfunctioning sensors.

Design Principle 7: Robustness. This design principle aims to produce a resilient authentication parameter that is not subject to long-term manipulations by misbehaving entities. Figure 5 shows the result of a consensus operation that has been performed by one form of misbehavior – a “random,” or inconsistent, sensor. In this example, a random sensor is defined as one with an initial trust value of $t = 0.5$ and an accuracy reading of $a = 0.5$. These values yield a “random opinion” of:

$$\omega_x = \begin{cases} b_x = t \cdot a = 0.25 \\ d_x = t \cdot (1 - a) = 0.25 \\ u_x = (1 - t) = 0.50 \end{cases}$$

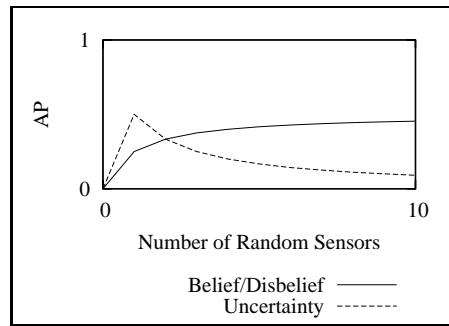


Fig. 5. 10 Random Sensors forming Consensus

As indicated in the figure, uncertainty decreases as more random sensors contribute an opinion to the authentication process. However, the randomness of the sensors results in an equal, but slow, rise of belief and disbelief values. Given the relatively low accuracy and trust values associated with a random sensor, the consensus operation does not allow these sensors to have a significant impact on the overall AP that results. In fact, the AP value does not increase beyond 0.5 even when many sensors are polled.

6 Discussion

We have introduced the concept of Parameterized Authentication and detailed the process for computing a measure that captures the overall quality of an authentication process. In this section, we present some noteworthy aspects of parameterized authentication that did not receive sufficient discussion in previous sections.

6.1 Authentication Paradigms and Historical Data

We believe that the use of historical state information would supplement and enhance the accuracy of a security subsystem. In essence, we propose an *authen-*

tication cache that would allow authentication decisions to be based on identification information currently stored in the system. Traditional implementations of caching technology were able to store often-requested information at the edge of the network, therefore speeding up site performance and lowering connection costs. Given the overwhelming success of caching, we propose that similar technology be applied to the security services in interactive environments such as the home.

Such a cache would pull user identification information from sensors in the environment and use it when making future access control decisions. For example, in the Aware Home users will be identified using a number of devices and technologies. Suppose that *Alice* swipes a smart card in order to gain access to the house via the front door. As she enters, a camera is able to identify her with 86% accuracy and the Smart Floor is able to track her movements and maintain a positive identification as she moves throughout the house. When *Alice* attempts to access a restricted resource, the system is able to use all of the historical state information, including the data obtained via the smart card’s key signature, the camera, and the Smart Floor, to obtain a positive identification, and therefore an accurate authentication, for *Alice*, a legitimate resident and authorized user of the resource.

To our knowledge, the use of historical information in security-related decisions has never been included as an enhancement to the overall system. Instead, historical data has been viewed as stale and inconsequential because security parameters and policies are subject to dynamic change and variability. However, in the case of the Aware Home and other computationally rich environments, historical data can offer something that is highly desirable – more timely and efficient access to critical information. In addition, historical information can act as a supplement in cases where the current or active data set is inaccurate or incomplete.

6.2 Authentication Refinement

Other authentication services, such as Kerberos [15], enforce session limits that force tickets or credentials to expire after a specified period of time has elapsed. We provide a mechanism to enforce similar session limitations, but use a method of enforcement that compliments our model for parameterized authentication. Since the user credential, or AP, is actually a consensus of input from multiple sources that could potentially be collected over a period of time, credentials cannot simply timeout. Also, it may be desirable to have a credential based on a lower AP value expire before one that was based on more reliable sensor input.

To address these concerns, our model provides a decay function that decrease the authentication value over time to enforce session limits and ensure the authenticity of user credentials. The effect of this decay on an AP’s value can be modeled using a decay function $f(n)$ such that the AP decreases after some specified time has passed; the rate of decay should increase as more time passes. Higher AP values (e.g., those based on more trustworthy or accurate sensor inputs) are typically more reliable than lower AP values. After t time-periods, the

AP value will be equal to y . This approach guarantees that access to restricted resources is provided only to users with sufficient authentication credentials that are current, accurate and trustworthy.

6.3 Application Scenario

To best demonstrate the functionality of our approach, we have used it to build a secure implementation of AudioIM, an instant messaging (IM) application that has been deployed in the Aware Home. AudioIM is one member of a family of applications [16] that are being used to explore how technology can enhance distributed conversation among family and close friends. Similar to text-based IM, which is used in the office to start semi-synchronous talks, AudioIM extends IM into the home with a touchscreen interface and audio messaging. AudioIM strives to provide instant messaging services where desktops are not suitable.

In order to identify users interacting with the system, the original AudioIM application provides a password verification scheme. The login component requires the user to authenticate herself with a username and password combination. This approach uses a desktop-centric authentication model for an application that was designed specifically for pervasive computing environments.

The enhanced application, supporting parameterized authentication, no longer requires the user to log in. Instead, the AudioIM console provides a collection of functions that are available for use. Should one of them be protected by an access control policy, the authentication support would be activated in the background. The authentication service would, in turn, be responsible for collecting sensor output from the appropriate location in the home. If adequate sensor readings for the current user can be implicitly gathered, no explicit communication is necessary to obtain user credentials.

7 Conclusion

We have introduced a new model for user authentication and have described how it can be used to secure dynamic applications in pervasive computing environments. The major benefit provided by this authentication model is its ability to provide quantitative measure for an authentication parameter when faced with incomplete, inaccurate or unreliable information.

We introduced the concept of *parameterized authentication* and have illustrated how it can allow continued functionality in a damaged, compromised or fundamentally inaccurate system. Furthermore, our authentication scheme is not limited by being binary in nature – users who cannot be fully authenticated by the system may still be given access rights based on their role and the level of confidence the system has in their identification.

Our notion of parameterized authentication provides a metric, the authentication parameter, that is based on *trust* and *accuracy* values associated with each authentication sensor and the output provided by those sensors. The authentication parameter is used to provide knowledge of the authentication process to

an authorization service as a single, well-understood metric. We have presented several design principles that guided the design of our model and have used those principles to evaluate the model itself.

Our ongoing work focuses on building a more robust model for parameterized authentication. We intend to extend this work by investigating outstanding issues such as the impact of byzantine adversaries on our model, session management in pervasive computing environments, and mechanisms to provide a more timely feedback cycle. Furthermore, our approach is limited in the sense that we look only at “local” experiences at the sensor level. It would be desirable to consider rule-based systems that can detect malicious activity similar to those found in Intrusion Detection Systems (IDS). By utilizing a variety of mechanisms to identify malicious activity in the system, we expect that the authentication service could be less reactive and more aggressive in preventing negative experiences from occurring. In its current implementation, the authentication service can only detect malicious activity after it has occurred.

References

1. Georgia Tech Broadband Institute: The Aware Home research initiative (1999-2004) <http://www.cc.gatech.edu/fce/ahri/>.
2. Mankoff, J., Abowd, G.: *Domisilica: Providing ubiquitous access to the home*. Technical Report GIT-GVU-97-17, College of Computing, Georgia Institute of Technology (1997)
3. Orr, R., Abowd, G., Atkeson, C., Essa, I., Gregor, R.: *The smart carpet: A mechanism for user identification and location tracking*. Georgia Tech Graphics, Visualization, and Usability (GVU) Center Seed Grant (1998)
4. Beth, T., Borchering, M., Klein, B.: *Valuation of trust in open networks*. In Coppersmith, D., ed.: *Proceedings of ESORICS*. Lecture Notes in Computer Science 875, Springer-Verlag (1994) 3–18
5. Reiter, M.K., Stubblebine, S.G.: *Authentication metric analysis and design*. ACM Transactions on Information and System Security (1999)
6. Maurer, U.: *Modelling a public-key infrastructure*. In Bertino, E., ed.: *Proceedings of ESORICS*, Lecture Notes in Computer Science (LNCS), Springer-Verlag (1996)
7. Jøsang, A.: *Artificial reasoning with subjective logic*. In: *Australian Workshop on Commonsense Reasoning, In conjunction with the Tenth Australian Joint Conference on Artificial Intelligence* (1997)
8. Pankanti, S., Jain, A.: *Biometrics: The future of identification*. In: *IEEE Computer*. IEEE (2000) 46–49
9. Thalheim, L., Krissler, J., Ziegler, P.M.: *Körperkontrolle — Biometrische Zugangssicherungen auf die Probe gestellt*. *c't* **8** (2002) 114 English translation: <http://heise.de/ct/english/02/11/114/>.
10. Matsumoto, T., Matsumoto, H., Yamada, K., Hoshino, S.: *Impact of artificial gummy fingers on fingerprint systems*. In: *Proceedings of SPIE*. Volume 4677, *Optical Security and Counterfeit Deterrence Techniques IV*. (2002)
11. Gardner, A., Essa, I.: *Prosody analysis for speaker affect determination*. In: *Proceedings of Perceptual User Interfaces Workshop*. (1997)
12. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: *The EigenTrust algorithm for reputation management in P2P networks*. In: *Proceedings of the Twelfth International World Wide Web Conference*. (2003)

13. Chellappa, R., Wilson, C., Sirohey, S.: Human and machine recognition of faces: A survey. *Proceedings of IEEE* **83** (1995) 705–740
14. Monroe, F., Reiter, M.K., Wetzel, S.G.: Password hardening based on keystroke dynamics. *International Journal on Information Security* **1** (2002) 69–83
15. Kohl, J.T., Neuman, B.C., T'so, T.Y.: The evolution of the Kerberos authentication system. *Distributed Open Systems* (IEEE Computer Society Press) (1994)
16. Nagel, K., Kidd, C.D., O'Connell, T., Dey, A., Abowd, G.D.: The family intercom: Developing a context-aware audio communication system. In Abowd, G.D., Brumitt, B., Shafer, S.A.N., eds.: *UbiComp, Lecture Notes in Computer Science (LNCS)*, Springer-Verlag (2001) 176–183