Authentication Confidences

Gregory R. Ganger
ganger@ece.cmu.edu
April 28, 2001

April 2001
CMU-CS-01-123

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

"Over the Internet, no one knows you're a dog," goes the joke. Yet, in most systems, a password submitted over the Internet gives one the same access rights as one typed at the physical console. We promote an alternate approach to authentication, in which a system fuses observations about a user into a probability (an ***authentication confidence****) that the user is who they claim to be. Relevant observations include password correctness, physical location, activity patterns, and biometric readings. Authentication confidences refine current yes-or-no authentication decisions, allowing systems to cleanly provide partial access rights to authenticated users whose identities are suspect.*

We thank the members and companies of the Parallel Data Consortium (at the time of this writing: EMC Corporation, Hewlett-Packard Labs, Hitachi, IBM Corporation, Intel Corporation, LSI Logic, Lucent Technologies, Network Appliances, Panasas, Inc., Platys Communications, Seagate Technology, Snap Appliances, Sun Microsystems and Veritas Software Corporation) for their insights and support.

Keywords: security, authentication, biometric authentication, system access.

1. The Case for Authentication Confidences

Access control decisions consist of two main steps: authentication of a principal's digital identity and authorization of the principal's right to perform the desired action. Well-established mechanisms exist for both. Unfortunately, authentication in current computer systems results in a binary yes-or-no decision, building on the faulty assumption that an absolute verification of a principal's identity can be made. In reality, no perfect (and acceptable) mechanism is known for digital verification of a user's identity, and the problem is even more difficult over a network. Despite this, authorization mechanisms accept the yes-or-no decision fully, regardless of how borderline the corresponding authentication. The result is imperfect access control.

This white paper promotes an alternative approach in which the system remembers its confidence in each authenticated principal's identity. Authorization decisions can then explicitly consider both the "authenticated" identity and the system's confidence in that authentication. Explicit use of authentication confidences allows case-by-case decisions to be made for a given principal's access to a set of objects. So, for example, a system administrator might be able to check e-mail when logged in across the network, but not be able to modify sensitive system configurations. The remainder of this section discusses various causes of identity uncertainty and existing mechanisms for dealing with it. The following section discusses how authentication confidences might be added to systems.

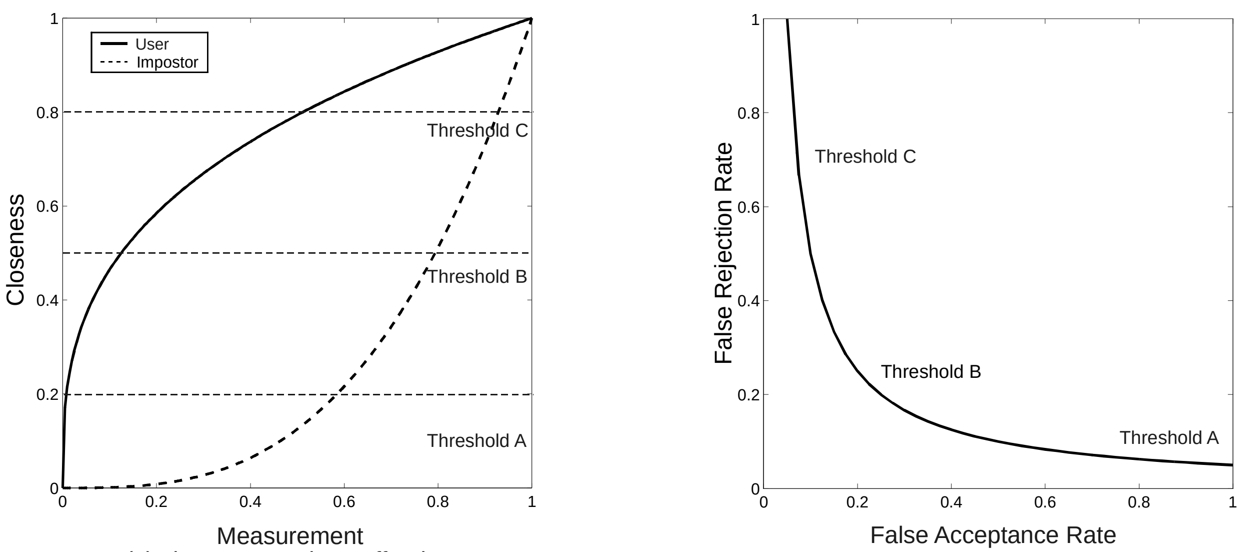
1.1. Human identification and confidence

In current computer systems, authentication of a user's digital identity relies on one or more mechanisms from three categories:

- **Something one knows.** The concept here is that if the user knows a pre-determined secret, it must be the right person. The common type of secret is a password, though other schemes like images [5] and patterns are being explored. The conventional wisdom is that since it is a secret, no additional information about the likelihood of true identity is necessary or available. We disagree. For example, a system's confidence in the provided password could certainly depend upon the location of its source — the likelihood of an imposter providing my password from my office is much lower than the likelihood of them providing it over the network (especially from the Internet or the dormitories). As well, a gap of idle time between when the password was provided and a session's use might indicate that the real user has left their workstation and an intruder has taken the opportunity to gain access.

- **Something one has.** The concept here is that if a user has a pre-configured item, it must be the right person. The common item is some kind of smart card or ID badge. The conventional wisdom is that anyone who has the token should have full access and that no other information is needed. Again, we disagree. As with the password example, location of token and time since session use can both affect the confidence that a system should have in the corresponding authentication. More radical out-of-band information, such as the owner's expected location based on scheduled appointments, could also provide insight.

- **Something one is.** The concept here is that the system compares measured features of the user to pre-recorded values, allowing access if there is a match [1]. Commonly, physical features (e.g., face shape or fingerprint) are the focus of such schemes, though researchers continue to look for identifying patterns in user activity. Identifying features are boiled down to numerical values called "biometrics" for comparison purposes. Biometric values are inherently varied, both because of changes in the feature itself and because of changes in the measurement environment. For example, facial biometrics can vary during a day due to acne appearance, facial hair growth, facial expressions, and ambient light variations. More drastic changes result when switching between eyeglasses and contact lenses or upon breaking one's nose. Similar sets of issues exist for other physical features. Therefore, the decision approach used is to define a "closeness of match" metric and to set some cut-off value — above the cut-off value, the system accepts the identity, and below it, not. When setting the cut-off value, an administrator makes a trade-off between the likelihood of false positives (allowing the wrong person in) and false negatives (denying access to the right person). Figure 1 illustrates this process and the corresponding trade-off. Note that we are not suggesting elimination of the cut-off. Instead, we are suggesting that the amount by which the observed value exceeds this cut-off be remembered as part of confidence.



(a) closeness and cut-off points (b) false accept/reject trade-off

Figure 1: Illustrative example of closeness-of-match thresholds for biometric-based authentication and the corresponding trade-off between false acceptance rate and false rejection rate. On the left, (a) shows possible distributions of closeness values for a user and an imposter. Notice that each cut-off threshold will sometimes reject the real user and sometimes accept the imposter. Specifically, at a given cut-off threshold, false accepts are to the right of the dashed line and false rejects are to the left of the solid line. As biometric accuracy improves, the area beneath the user's distribution will increase and that beneath the imposter's curve will decrease. On the right, (b) illustrates the trade-off between false acceptance rate and false rejection rate more directly with the common "Receiver Operating Characteristic" curve. Better biometric accuracy would reduce the space beneath this curve.

Confidence in identity can be enhanced by combining multiple mechanisms. The simplest approach is to apply the mechanisms independently and then combine their resulting confidences, but more powerful fusing is also possible. For example, merged lip reading and speech processing [3] can be better than either alone. Likewise for password checks and typing patterns [8]. Note that if the outcomes conflict, this will reduce confidence, but will do so appropriately.

1.2. Propagating identity among distributed systems

Solid mechanisms exist for propagating authenticated identity across systems [2], but they assume that intermediary nodes can be trusted to SPEAK-FOR the original party appropriately. Recent work [4] refines this SPEAK-FOR logic to put limits on what a remote system can say as the principal, but any statements within the set are still made as though 100% confident in identity. Arguably, the more nodes through which the SPEAK-FOR relationship is passed, the lower the confidence value associated with it should be. This is particularly true when the security of any of those nodes is questionable.

1.3. Alternatives to authentication confidences

Few alternatives exist to allow a given user differing degrees of access depending on the quality of authentication. One option is simply not to allow access when sufficient confidence can not be gained — this is still not an uncommon practice for commercial settings, where traveling employees are denied access to systems on internal company networks. This tends to reduce productivity and induce insecure workarounds. An alternate approach is to provide users with several digital personalities, giving different access rights to each. This approach creates management headaches and results in security problems when users choose the same password for all accounts. Properly used, we think that authentication confidences provide a flexible, intuitive mechanism for expressing to systems the varying levels of access for a given user.

1.4. Authentication confidences in the real world

Authentication confidences are common practice in the real world. For example, a sentry might allow someone dressed in the right uniform to approach, but then deny entry if the right passphrase is not given. A civilian example is practiced by credit card companies, which may block suspect purchases or attempt to increase their confidence by calling the card owner. Proven useful in the real world, we think explicit use of authentication confidences in computer authorization decisions is worth exploring.

2. Using Authentication Confidences

There are two main issues with incorporating authentication confidences into systems: establishing confidence values and using them. The latter is simpler, in theory, so we will discuss it first. To retain confidence information, the “user ID” normally associated with a session should be paired with a corresponding confidence value. Likewise, any authorization structure [7] (e.g., an ACL entry) should be extended to keep a confidence value with each user ID field. To allow access, an authorization check verifies two things: that the authenticated user ID matches and that the measured confidence matches or exceeds the recorded requirement. Capabilities are somewhat more difficult than ACLs, since a system's confidence in a given authentication could drop after the corresponding session was granted a capability. However, if the required confidence were recorded in the capability, then a usage time check could be made to ensure proper access control.

Note that migration to authentication confidences can be straightforward and incremental. Current systems essentially use a global value for all decisions. Systems can continue to have a global default, but also provide support for specifying specific confidence requirements. When an authorization decision is made, the authorization mechanism can check both the identity and the confidence in that identity, independent of whether the required confidence is a default or a specific value.

Now, we discuss the more difficult issue of establishing the confidence value. This issue requires an algorithm for converting the system's relevant observations into a value. We currently envision authentication confidences as percentage values between 0 and 100, as in "I am 90% sure that this is Fred." Doing this will involve pre-configured confidence value settings for binary observations (e.g., password checks and local/remote console). More continuous values, such as biometric "closeness of match" comparisons, can be included mathematically. Table 1 gives an example.

Password-Based Information	Biometric-Based Information	Authentication Confidence
Failed check	Don't care	No access granted
Passed check (remote console)	No check	60% confidence
Passed check (local console)	No check	80% confidence
Passed check (local console)	Failed check	30% confidence
Passed check (local console)	Barely passed check	85% confidence
Passed check (local console)	Strongly passed check	95% confidence

Table 1: Illustrative example of authentication confidence values for a hypothetical system. This system always requires a successful password check. Its confidence in user identity is also improved by using the local console and by passing a biometric-based identity check at the local console. Failing the biometric check reduces authentication confidence, but does not reject the user entirely; so, clearly, the administrators of this system do not fully trust biometric-based authentication. The highest confidence is reached when all signs are positive.

As suggested earlier, a system's confidence in a given authentication may vary over the duration of a session. There are several possible causes of such variation. For example, a lengthy period of idle time may reduce confidence due to the potential of another person sitting down at an abandoned workstation. On the other hand, the system may observe positive signs of identity, such as activities common to the user in question (for example, the author looks at the Dilbert web page early each day). Continuous biometric authentication checks (as could be done with video-based checks) could also increase or decrease the system's confidence in a user's claimed identity [6]; this would be particularly true if the video camera observes the original user leaving the workstation. Finally, if the confidence drops too low, the system could insist that the user re-authenticate before continuing the session.

3. Summary

Authentication confidences are an interesting approach to embracing (rather than hiding) the uncertainty inherent to authentication decisions. By explicitly subsetting the privileges of a principal based upon authentication confidence, one could more cleanly handle the difficulties involved with specifying access rights that vary based on how the principal authenticates to the system. Clearly, experience is needed to determine exactly how to best realize authentication confidences in practice, but we believe that the notion is worth exploring.

References

[1] Biometrics. IEEE Computer, February 2000.

[2] Michael Burrows, Martín Abadi, and Roger Needham. A logic of authentication. ACM Transactions on Computer Systems, 8 (1):18-36, February 1990.

[3] Tsuhan Chen. Audio-Visual Speech Processing, http://amp.ece.cmu.edu/projects/AudioVisualSpeechProcessing/.

[4] J. Howell and D. Kotz. End-to-end authorization. Symposium on Operating Systems Design and Implementation. San Diego, CA, 23-25 October 2000, pages 151-164. USENIX Association, 2000.

[5] Ian Jermyn, Alain Mayer, Fabian Monrose, Michael K. Reiter, and Aviel D. Rubin. The design and analysis of graphical passwords. 8th Security Symposium. Washington DC, 23-26 August 1999, pages 238, 1-14. USENIX Association, 1999.

[6] Andrew J. Klosterman and Gregory R. Ganger. Secure Continuous Biometric-Enhanced Authentication. CMU-CS-00-134. Carnegie Mellon University School of Computer Science Technical Report, May 2000.

[7] B. W. Lampson. Protection. Princeton Symposium on Information Sciences and Systems, pages 437-443, 1971.

[8] Fabian Monrose, Michael K. Reiter, and Susanne Wetzel. Password hardening based on keystroke dynamics. ACM Conference on Computer and Communications Security. Kent Ridge Digital Labs, Singapore, November 2-4. Published as Proceedings of ACM Conference on Computer and Communications Security, pages 73-82. ACM Press, November 1999.