An Introduction to Automated Trust Establishment

Marianne Winslett

winslett@uiuc.edu

Abstract: The last decade of improvement in service offerings over the Internet offers the hope that many kinds of sensitive interactions between strangers can be carried out electronically, without requiring physical transmission of paper credentials to establish trust. In this short paper, I describe one way of converting the current paper-based approach to establishing trust into an electronic approach that minimizes human intervention. I also describe the theoretical and systems issues that are raised by this approach.

1 What kinds of sensitive interactions do we want to accomplish electronically?

Last weekend I bought twelve shirts for my husband at a going-out-of-business sale. The lady behind the counter took my credit card, looked at the hologram on the front of the card, ran the card through her scanner, and waited while the computer behind the scanner checked to see that the card had not been revoked and that sufficient credit was available for me to buy all those shirts. Then the cash register printed out a receipt, which I signed. The lady behind the counter compared my signature to the signature on the back of the credit card. After a long pause, she said dubiously, "I guess they're the same."

In this manner, we two strangers carried out a face-to-face business transaction. The system we used is far from perfect, as it is vulnerable to attack at several points along the line. For example, credit cards can be forged (hence the hologram), stolen (hence the automatic phone call to a remote center to check for revocation, the example signature on the back, and the signature comparison), or expired (hence the automatic check for expiration). My privacy is not well served by the system, since the credit card company knows about all my purchases and likes to sell that information to third parties. Still, with all its weaknesses, the system seems to work fairly well.

I would like to see the same ease of interaction between strangers on line, when they come together to carry out a transaction. In this case, the interaction could be between an individual and (a representative of) an organization, as in my previous example; or it could be between two individuals, or two organizations. The scenario could involve an ordinary purchase, as in the example above; participation in an on-line auction; a request to access medical records, military data pertinent to a joint exercise, or any other kinds of sensitive documents; registration for school, for a voters' card, library card, marriage license, visa,

passport, etc.; proof that all requirements for an adoption, change of citizenship, work permit, etc., have been met; or any other scenario where two strangers today must disclose paper credentials to carry out an interaction with some degree of sensitivity.

2 How can we use automated trust establishment in these interactions?

Perhaps the most obvious starting point for this quest is to attempt to digitize our current system, while perhaps plugging a few of its most blatant vulnerabilities at the same time. For example, what are the requirements for a digital version of the shirt-purchase transaction?

1. I will need a digital version of my credit card, and some way to prove that it is *my* credit card. The digital version of my credit card should be verifiable and unforgeable.
2. The store needs to be able to recognize my credit card: it must be able to tell that the credit card was issued by someone it trusts, such as VISA, and correctly read and interpret the fields of the card. It must also know how to verify that I own the card.
3. The store needs a policy for credit card acceptance. This policy will need to specify the acceptable credit card issuers, require that ownership of the card be demonstrated, check that the card has not expired, and require that the card issuer be contacted to check that the card is not revoked and that the new charges do not exceed the available credit for the card.
4. If my credit card is to be shown automatically when needed, I will require a policy that specifies the conditions under which I am willing to show my card. For example, I might require the merchant to belong to the Better Business Bureau of the Internet, or I might want to check certain aspects of the merchant's privacy policy. To prevent my kindergartner from using my credit card on-line to buy toys, I might want my computer to check the identity of the person at the console, either through checking the login or using another authentication procedure.
5. The merchant and I need a protocol that will allow us the opportunity to show each other the credentials that are relevant for the purchase, and perhaps also to find out which credentials are relevant for the purchase.

Clearly there are ways to satisfy most of these requirements.

1. X.509 certificates could be used for digital credentials, or we could move to a more modern credential format that offers improved guarantees for privacy, non-forgeability, single- versus multiple-use, and so on (e.g., [BK02, Bra00, PV00]). We might choose to use one or more standard languages for expressing credential contents, such as XML [BS00].

2. Public Key Infrastructures (PKI) can be used to establish domains of trust. For example, VISA International could be the root authority for a PKI hierarchy devoted to VISA cards. The Better Business Bureau could be the root authority for a PKI hierarchy that includes all the chapters of the Better Business Bureau. My employer can create its own hierarchy, used for employee ID cards and other purposes.

3. Access control policies for *every* credential and *every* service (such as my VISA card and the shirt purchase) can be codified so that they can be checked and enforced automatically whenever possible, to minimize manual intervention by the user. If a need arises while trust is being established, I should also be able to export my access control policies in a format that can be understood by strangers trying to gain access to my local resources, so that strangers understand what the requirements are for gaining access to the resources of interest to them. For example, I may need to know which credit cards are accepted by this merchant, and the merchant may need to find out about the access control policy that he will have to satisfy before I will disclose my VISA card to him.

4. Each party can decide which PKI authorities it trusts (VISA and the Better Business Bureau, in the shirt purchase transaction), and write its access control policies accordingly. Parties can cache credentials of interest to them personally (e.g., their employee ID, library card, credit card, certification that their employer is a state agency, and so on), so that they will be available to show to strangers as needed. Similarly, for the parts of an access control policy that require the policy's owner to actively seek out credentials (e.g., the merchant's check that my credit card is not revoked, and my check that my child is not trying to use my credit card), the party can contact the appropriate authority on line in real time.

3 An example of automated trust establishment in action

To make these ideas more concrete, let's look at a simple example of how things might work. The example involves a request to view Susan Jones's medical record at Busey Hospital. Busey Hospital's access control policy for that record says that the requester must be a physician at Busey Hospital, or else be the patient. The hospital also has an institutional ID that it is willing to show to anyone.

The requester, Susan Jones, has two credentials of interest. She is a patient at Busey Hospital, and she also works there as a staff physician. She will show her employee ID only to Busey Hospital. Her access control policy for her patient ID says that it can only be shown to people who work at Busey Hospital, Blue Cross Blue Shield of Illinois, or to Busey Hospital itself. For the sake of concreteness, the exact policies are given below, although the reader may prefer to skip over their contents because there are no formal definitions of the access control policy language or credential representations in this paper.

Server (Busey Hospital) resources and their access control policies:
institutionID:
$true$
medicalRecord:
(x.type = "patient ID" $\wedge$ x.issuerPublicKey = publicKey("Busey Hospital")
$\wedge$ x.patientNumber = "12345" $\wedge$ requesterAuthenticatesTo(x.ownerPublicKey))
$\vee$
(y.type = "employee ID"
$\wedge$ y.issuerPublicKey = publicKey("Busey Hospital")
$\wedge$ y.jobtitle = "Staff Physician" $\wedge$ requesterAuthenticatesTo(y.ownerPublicKey))

Client (patient Susan Jones and employee Dr. Sue Jones) resources and their access control policies:
employeeID:
x.type = "institutional ID"
$\wedge$ x.issuerPublicKey = publicKey("Busey Hospital")
$\wedge$ x.ownerPublicKey = publicKey("Busey Hospital")
$\wedge$ requesterAuthenticatesTo(publicKey("Busey Hospital"))
patientID:
(x.type = "employee ID"
$\wedge$ x.issuerPublicKey = publicKey("Busey Hospital")
$\wedge$ requesterAuthenticatesTo(x.ownerPublicKey))
$\vee$
(y.type = "employee ID"
$\wedge$ y.issuerPublicKey = publicKey("Blue Cross Blue Shield
of Illinois") $\wedge$ requesterAuthenticatesTo(y.ownerPublicKey))
$\vee$
(z.type = "institutional ID"
$\wedge$ z.issuerPublicKey = publicKey ("Busey Hospital")
$\wedge$ requesterAuthenticatesTo(publicKey("Busey Hospital")))

This example uses descriptive names for credentials, to help the reader. In the real world, credential and access control policy names must not reveal useful information, else the disclosure of an access control policy can reveal sensitive information about the credential it protects. Also for realism, the example uses a *RequesterAuthenticatesTo* predicate, whose truth or falsity is determined at run time by whether or not the requester can demonstrate knowledge of the private key associated with the public key specified in the credential (or by another suitable authentication approach). Finally, the example uses a function *publicKey*, whose interpretation is, in practice, supplied by a call to a certification authority that publishes public keys.

We now show how Susan Jones and Busey Hospital can establish trust. The actual negotiation is conducted between Susan's and the hospital's security agents, without intervention from Susan or the hospital; for simplicity, we describe the negotiation as though Susan and the hospital were negotiating directly with one another.

1. The negotiation is triggered when Susan Jones requests to read her *medicalRecord* over the web. After she requests to read the record, Busey Hospital sends Susan the access control policy for the hospital's *medicalRecord* resource.

2. Susan determines that her employee ID satisfies the access control policy for *medicalRecord*, and discloses her access control policy for *employeeID* to the hospital.

3. Busey Hospital determines that its institutional ID satisfies Susan's access control policy for *employeeID*, and discloses *institutionID*.

4. Susan determines that *institutionID* satisfies the access control policy for *employeeID*, and discloses her employee ID.

5. At this point, Busey Hospital receives Susan Jones's *patientID*, which satisfies the access control policy for *medicalRecord*. Thus Busey Hospital grants access to *medicalRecord*.

The negotiation could have followed a different path, if Susan or the hospital had used different strategies. For example, Susan might have preferred to satisfy the hospital's policy by disclosing her patient ID, rather than her employee ID. If she is very eager to establish trust, she might prefer to disclose both of them.

Suppose that Susan is accessing her medical record in order to check her home address, to see if it is correct. After examining the address, she may see that it does not reflect her recent move, and ask to update the record. The access control policy for updating the address information in *medicalRecord* may state that only the patient is allowed to update the patient's address. In this case, Susan will have to learn about this new access control policy and reestablish trust, based on her patient ID rather than on her employee ID: the two IDs refer to different identities, and the hospital does not know that Susan possesses both identities.

4 Theoretical and practical issues raised by automated trust establishment

A number of interesting theoretical and practical questions arise if we adopt this approach to establishing trust between strangers, revolving around such issues as autonomy, scalability, and vulnerability to attack. For example:

1. **Delegation.** The digital credentials described here are *general-purpose*. For example, my employer, the University of Illinois, issues student and staff ID cards for its own purposes. However, these cards can be used outside the university, for purposes that the university never contemplated. For example, I can use my staff ID to get the government room rate at hotels. Our local independent hardware store gives a 15% discount to people who show a student or staff ID card from the university. The general-purpose nature of these credentials sets them apart from the kinds of

credentials used in KeyNote [BFIK99]. More precisely, general-purpose credentials cannot be used in KeyNote policies that control delegation; the attributes in these credentials are not intended to specify conditions of use. Extra machinery will be needed to handle delegation using general-purpose credentials.

2. **Policy and credential capture and interpretation.** The central role of access control policies in this approach raises many software engineering and knowledge representation issues. We need good languages for expressing access control policies, as well as tools to help people write and update them. We need standard schemas for popular types of credentials, such as employee IDs and passports, so that strangers will be able to interpret their contents correctly.

3. **Architectures.** Can we use a trusted third party to establish trust between strangers, in a manner that does not leave the third party vulnerable to attack? Can we use a zero-knowledge approach, if there is no trusted third party?

4. **Strategies for establishing trust.** To establish trust without a trusted third party or a zero-knowledge approach, strangers will have to disclose some of their credentials, and possibly some of their access control policies as well. However, there will be many decision points while trust is being established: out of all the credentials and access control policies that I *could* disclose next (i.e., their own access control policies are satisfied), which should I actually disclose? For example, I am theoretically willing to show almost all of the credentials in my purse to a stranger (i.e., their access control policies are satisfied). However, unless the stranger demonstrates a *need to know*, I will not disclose them. For example, I did not show my driver's license or my daughter's baby pictures to the store clerk when I bought the shirts. Thus there is a strategic decision in choosing which credentials to disclose at each step of trust establishment. What is the possible range of strategic choices? Further, to support autonomy, we should not require all participants to make the same strategic choices. How can we ensure that two strangers will be able to establish trust, while still giving them autonomy in their strategic choices?

5. **Authentication of multiple identities.** Under this approach to automated trust establishment, each party can have many identities, each corresponding to the identity that a particular credential issuer uses to designate that individual. For example, the identities in my purse today include my patient number, my driver's license number, my employee ID number, my credit card number, my library card number, and so on. I may be asked to prove that I possess several of these identities during trust establishment. How can I do so, in a manner that prevents or penalizes collusion? How can I make the actions I take under different identities unlinkable?

6. **Obtaining and storing credentials.** How can I obtain the credentials I need? How should I store them to keep them safe from prying eyes? If I need to find credentials that are not cached locally, how can I do so in real time while trust is being negotiated?

7. **Scalability.** How can trust establishment be automated in a highly scalable manner? Can it be made ubiquitous?

8. **Attacks.** What kinds of attacks is trust negotiation vulnerable to, and to what degree can they be mitigated? What parts of trust negotiation software must be trusted, and to what degree?

5 Conclusions

For a number of years, a small set of researchers has been exploring the issues listed in the previous section. We have made progress on many fronts, both theoretical and practical, though many issues have not yet been addressed. Further, a new result often suggests several additional intriguing issues to investigate. Of the theoretical and practical issues listed above, we have made the most progress in the following areas:

1. Understanding the range of possible strategies for establishing trust and addressing issues related to autonomy in the choice of a trust negotiation strategy and correct interoperation of the strategies chosen by two strangers (e.g., [YWS01]).
2. Understanding how to obtain credentials in real time, and how to map credentials to roles in a role-based access control system (e.g., [LWM01, LMW02]).
3. Testbed implementations of trust negotiation in a variety of scenarios (e.g., HTTP and TLS [HMM+00, HJM+02]).

The interested reader is invited to look at the publications cited above and in the bibliography (or, more likely, their follow-on work) to see the directions we have pursued and the results that have been obtained. I do not summarize the results here, because they quickly become extremely technical.

Acknowledgements

Other people have been instrumental in exploring issues related to automated trust establishment. During the past several years, I have participated in a joint project with Kent E. Seamons of Brigham Young University. Kent and his students have been building trust negotiation prototypes and exploring systems issues that arise in implementing trust negotiation in a scalable manner. Closer to home, most of the theoretical results for which I have been a coauthor over the past few years have been proved by Ting Yu. We have also benefitted from the closely related work done by Will Winsborough of NAI Labs and Will's colleagues.

My portion of this work has been supported by DARPA through AFRL contract number F33615-01-C-0336 and through Space and Naval Warfare Systems Center San Diego grant number N66001-01-18908.

References

[BFIK99] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis. The KeyNote Trust Management System Version 2. In *Internet Draft RFC 2704*, September 1999.

[BK02] J. Biskup and Y. Karabulut. A Hybrid PKI Model with an Application for Secure Mediation. *Submitted for publication*, 2002.

[Bra00] S. Brands. *Rethinking Public Key Infrastructures and Digital Certificates.* MIT Press, 2000.

[BS00] P. Bonatti and P. Samarati. Regulating Service Access and Information Release on the Web. In *ACM Conference on Computer and Communications Security*, Athens, November 2000.

[HJM+02] 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.

[HMM+00] 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.

[LMW02] N. Li, J. C. Mitchell, and W. H. Winsborough. Design of a Role-based Trust-management Framework. In *IEEE Symposium on Security and Privacy*, Oakland, May 2002.

[LWM01] N. Li, W. Winsborough, and J. Mitchell. Distributed Credential Chain Discovery in Trust Management. In *Conference on Computer and Communication Security*, Philadelphia, PA, November 2001.

[PV00] P. Persiano and I. Visconti. User Privacy Issues Regarding Certificates and the TLS Protocol. In *ACM Conference on Computer and Communications Security*, Athens, Greece, 2000.

[YWS01] T. Yu, M. Winslett, and K. Seamons. Interoperable Strategies in Automated Trust Negotiation. In *ACM Conference on Computer and Communication Security*, Philadelphia, PA, November 2001.