M.I.T. Laboratory for Computer Science

Request for Comments No. 270
February 4, 1985

Authentication for Inter-Organization Networks

by Deborah L. Estrin

1. Introduction

This section describes authentication requirements and protocols for Inter-Organization Networks (IONs). I discuss how Needham-Schroeder type authentication tools can be used to satisfy the authentication requirements outlined in my usage control model [1]. The primary ideas presented here are that internal authentication mechanisms need not necessarily be modified to comply with inter-organization requirements, and that multiple classes of authentication are desirable.

Organizations interconnect their internal computer networks to support automated transactions, communication, and resource sharing; referred to as *Inter-Organization Networks* (IONs). Typically, these relationships are accompanied by standard business contracts. Additional controls may be employed to protect participants from new vulnerabilities introduced by the powerful transaction mechanisms. In order to comply with contract agreements, and to enforce the desired policies and controls, each organization must be able to authenticate the other. The main purpose of authentication in this domain is to assure *accountability* should some behavior transpire that is in violation of contracts.

There are two types of authentication required:

- First. when one organization contacts another for the first time. the organizations must authenticate that each is legitimate. For example, when a new client contacts a vendor, the vendor typically checks the client's credit rating just as the client has checked the

WORKING PAPER – Please do not reproduce without the author's permission and do not cite in other publications.

vendor's credibility in the market. In this case, the new computer-based transaction mechanisms should allow organizations to assess one another via third parties in the same formal way that is done currently via telephone and paper.

~ Second, each time an organization contacts another, it must authenticate that it is the organization that it claims to be. For example, when an established client contacts a vendor to reorder some item by telephone or paper mail, both parties typically have informal or formal procedures for assuring each other that they are who they claim to be. For example, purchasing agents recognize one another's voices, or rely on the letterhead of invoices and letters, or call back the requester at the claimed organization. In this case, the new mechanisms must substitute for what are often informal procedures via telephone or postal mail.

In both cases, different levels of authentication are appropriate for different organizations and types of transactions. For example, the larger the proposed purchase, the more confident the vendor will want to be that the customer has an adequate credit rating and that it is who it claims to be. Similarly, the larger the purchase, the more confident the client will want to be that the vendor will be able to uphold its end of the agreement--delivery date, quantity, quality, service, etc.

The goal of this discussion is to describe how two or more organizations can make use of trusted third parties to authenticate one another without having to modify internal systems and protocols, with the exception of the ION gateway. The methods proposed fit well with the model of usage controls proposed in [1]. The discussion begins with a list of assumptions about organizations' and third party facilities. Initiation authentication is then described, followed by transaction authentication, and multiple levels of service for both types of authentication.

2. Assumptions

Internal Facilities:

1. Each organization,[1] A, has an internal authentication server, AS_a, that it, A, trusts to authenticate individuals *within* organization A. Contracts between ION participants specify that an organization, A, is responsible for the integrity of the information provided by AS_a.

2. An organization should be able to participate in ION authentication using existing internal authentication mechanisms. Although the organization might choose to beef up

[1] An organization is defined here as a set of entities that are willing to trust and be represented to other organizations by a single authentication server and gateway. This usually coincides with a common administration.

such mechanisms in the presence of new liabilities for correctness, it remains an internal decision.

3. Organizations have a known and small number of ION gateways (we will assume 1 for simplicity). All packets that enter or exit an organization's internal network must pass through one of these official ION gateways. Much of the function described below for the ION gateways can be offloaded to special policy servers to improve the gateway's packet-forwarding performance. However, for simplicity all functions will be treated as a part of a single logical gateway, even though they may be physically separated.

Third Parties:

1. Each organization has many ION-supported relationships each of which is governed by a separate contract. If no third party is employed, authentication must be handled on a pairwise basis. Since authentication fundamentally depends on sharing a secret [], each organization would have to keep track and guard as many secrets as there are organizations it communicates with. The benefit of employing a third party is not the traditional space considerations, but rather the liability associated with guarding each of the secrets. In addition, minimizing the number of organizations that one trusts with a secret makes it is easier to certify that the secret is being kept. Also, if the communicating organizations are competitors or otherwise mistrustful of one another, the third party can act as a buffer between them.

2. The function of the third party is twofold. The first is to provide information about organizations to one another when they interact for the first time. The second function as ION Authenticator is to certify that a particular transaction/connection/message/packet is from the *organization* that it says it is from. It is left to the source organization's AS to certify that the packet/message/connection is from the claimed individual, i.e., x, within the organization, A.

3. Third parties are available to authenticate organizations (ION participants) to one another. Different levels of service (of *guarantee*) are available for different types of organizations, transactions, and relationships. Any two (or more) organizations that want to be able to authenticate messages from one another must agree on a single mutually-trusted third party.[2]

[2] Actually, the scheme below could be extended to allow the participants to use different third parties [3] but for simplicity we will assume that they agree on a single one.

3. Initiation Authentication

If transactions are carried out *online* it makes economic sense for organizations to be able to initiate relationships with one another online as well. For example, a computer manufacturer may buy a certain chip by sending online price queries to a collection of suppliers and initiating a purchase with the lowest bidder. In this case, the selected supplier will want to check the credit rating of the new client just as it does when a first-time purchase is proposed over the phone or on paper.

In the paper and voice world a wide range of requirements and corresponding procedures exist for evaluating the legitimacy or credit of a new client. We will discuss this further in section 5. For now, we will assume that the third parties that a supplier traditionally checks with are accessible online. If they are not, then the supplier must use traditional media for evaluating new clients.

3.1. Protocol

I will describe the general approach to initiation authentication in terms of a new client, A, proposing to purchase something from a supplier, B.

When the supplier receives a message it checks the destination to see if authentication is required. If so, the supplier checks the source listed on the order against a list of known entities, i.e., initiated clients. If the source does not appear on the list, the supplier sends an authentication request to one of several third parties employed for this purpose. A may send a suggested third party's name along with the original message if A anticipates the need for initiation authentication. Along with the name of the claimed entity, A, B includes the criteria according to which the AS should evaluate A, e.g., credit rating. B may set the criteria according to the destination of the message (i.e., the level of risk or value of information or product control residing at the destination), or the size of the request.[3] If the source is not registered with any of the third parties employed by the supplier the purchase order may be rejected or a message returned saying that registration with third party X is required. It is then up to the customer to reinitiate the purchase after establishing its identity with X. If the third party does have the client registered, the third party returns its evaluation of the client (e.g., credit rating, or perhaps just an assurance that the client is a real company) to the

[3] In the latter case, gw_b would have to pass the purchase order to some service in order to determine the appropriate evaluation criteria since it is based on something other than the source and destination of the message; which is the only information the gateway has direct access to.

supplier. The supplier ads the client to its list of initiated clients along with the evaluation. The supplier also records the name of the third party that was able to provide the information about the client. From this point on the client is initiated until the supplier decides to recheck the evaluative information.

Following is an example of a dialog that could be used to implement initiation authentication as described above:

1. A ---> B: purchase order
2. B ---> AS: A, evaluation criteria
3. AS ---> B: A, evaluation
4. { B ---> A: m, register with AS }
5. B adds A, evaluation, AS address to known-entity list

At this stage the organization that the purchase order *claims* to be from is initiated as a legitimate entity to do commerce with. However, the supplier still needs to know that the purchase order in fact came from that organization. In addition, in the future, when the initiated client sends other purchase orders, the supplier must be able to authenticate that the purchase orders are from the claimed client for which the supplier maintains credit rating information, etc. What is needed is a mechanism for authenticating that a particular transaction is from the claimed party. I refer to this as *transaction authentication* and describe our approach in the following section.

4. Transaction Authentication

Assuming that a client has been initiated and is now a registered client with the vendor, each transaction must be authenticated. I outline the approach and describe a simple protocol for transaction authentication and implementation issues.

4.1. Protocol

The protocol for ION authentication will be described for two organizations, A and B, who want to authenticate messages from one another. However, we assume that both organizations communicate with many other organizations as well so that the approach must scale well. After each organization has registered itself and a secret key with a common third party, a Needham and

Schroeder protocol is used to authenticate the organizations and provide communicating pairs with session keys so that they can authenticate messages from one another [2].

Before describing the protocol, I should emphasize why a third party is employed in this dynamic phase of authentication. As long as each organization is maintaining information about the other, each pair of communicating organizations could exchange a secret key with which to authenticate one another. Our rationale for employing a third party is that there is significant oveıhead in protecting a secret. Given that organizations have many correspondents (i.e., other organizations that they transact with), it is significantly more manageable for an organization to safeguard a single key to communicate with a third party than it is to safeguard n keys, one for each of its n correspondents. Note that the concern is not for space, since as I mentioned, some contract or other information is already stored for almost every correspondent. Rather, the concern is for the nuisance associated with safeguarding secrets. For this reason, a third party is employed for transaction authentication.

The protocol begins when an individual x in organization A sends a message to y in organization B; x and y may be people, machines, etc. The message header lists the source and destination organizations and individuals. All messages travel in and out of A and B via gw_a and gw_b, respectively. If B considers there to be no need (i.e., no risk associated with open access to y), it may forward the message to y unauthenticated. However, if B wants wants to control external access to internal resource, y, then for this discussion we will assume that B uses non-discretionary controls and assigns category labels to incoming messages, as is described in [1]. Because B assigns a category label according to the source of the message, B wants to authenticate the source, i.e., make sure that the source listed in the message header really generated the message. Functionally, this means that the organization listed in the header will take responsibility for the message.

To authenticate the source organization, B sends a message to the third party that it has listed as the one to use to authenticate messages from A; we will call this third party authenticator AS_{ab}.[4] B asks AS_{ab} for a key with which to authenticate A and subsequent messages sent by A during this session. B also returns the message, m, to gw_a saying that authentication is required. When gw_a

[4] We assume that during initiation authentication described above, the two parties identified a mutually trusted third party.

receives the returned, unauthenticated message from gw_b it asks its internal AS_a to authenticate x. B also authenticates y through a conversation with its internal AS_b.

AS_{ab} sends gw_b a session key, E_{ab}, along with the session key encrypted in A's private key, E_a; included also is a timestamp and an identifier of B. The entire message includes a time-stamp or nonce and is encrypted under B's private key, E_b.[5] B then sends A the session key encrypted in A's secret key. B does not have A's secret key, but was given the encrypted session key by AS_{ab}. B is guaranteed by AS_{ab} that only A will be able to read this message. Similarly, A is guaranteed by AS_{ab} that any message identifying B along with a session key encrypted under A's secret key must have originated with AS_{ab} and that only B has been given a copy of the session key. A and B now each have a copy of the session key and are guaranteed by AS_{ab} that any message encrypted under that key can be read by the other organization, only. Finally, to protect against replays by an intruder, A and B carry out a simple handshake, e.g., exchanging the current date and time.

Both gateways store the session key and gw_a resends the message, m, from x encrypted with the key. Both gateways encrypt all subsequent communication between x and y with the session key until the session ends or either party decides to reauthenticate. gw_b is assured that any messages arriving under that key came from gw_a and gw_a relies on internal authentication to assure that the message came from party x within A. Similarly, gw_a is assured that only someone in B can receive the message, since only gw_b can decrypt the message, and gw_b relies on its internal authentication to assure that the message goes to y, only.

The dialog that corresponds to this protocol is listed below.

1. x-->B: m

2. B-->AS_{ab}: (B,A)

3. B-->A: m, error-unauthenticated

4. gw_a-->AS_a: x
   gw_b-->AS_b: y

5. AS_{ab}-->B: $E_b(A,E_{ab},T,E_a(B,E_{ab},T))$

[5] Both organizations' addresses and private keys have been stored with AS_{ab} previously when A and B registered with AS_{ab}. AS_{ab} uses these secret keys to authenticate the organizations.

6. B-->A: $E_a(B,E_{ab},T)$

7. A-->B: $E_{ab}(I)$ B-->A: $E_{ab}(I\text{-}1, J)$ A-->B: $E_{ab}(J\text{-}1)$

In summary, using their secret keys (e.g., E_a and E_b), each organization can authenticate itself to the trusted third party in order to request a session key. The gateways use this session key to authenticate the source and destination organizations of each message. The organizations take responsibility for authenticating the destination within their respective organizations, based on existing internal authentication mechanisms. Consequently, AS_{ab} is liable if organization A or B is incorrectly authenticated, whereas AS_a and AS_b are liable if x or y are not who they claim to be. This characteristic is significant because it allows an organization with tight physical security to dispense completely with internal authentication if it so chooses.

4.2. Implementation

The following changes are required to implement this protocol among organizations with heterogeneous internal networks:

Third party:

1. A method for distributing keys between organizations and trusted third parties is needed so that the trusted third party can authenticate the organization.

ION gateway:

1. The gateway must maintain a list of trusted third parties so that when an unauthenticated message arrives from another organization, the gateway knows where to go to request authentication. The gateway must also store the private key used to authenticate its organization to trusted third parties. In addition the gateway maintains the known-entities list which includes evaluation information and mutually-trusted third party for each initiated organization.

2. Encryption in the gateway. NO internal entities need to encrypt messages for the purpose of authentication. Each gateway must store the session keys and associate them with the appropriate incoming and outgoing packets; e.g., by assigning the source, destination pair and the key to a virtual or physical port.

3. The gateway must be able to ask the authentication mechanism to authenticate the source of an outgoing message (i.e., generated internally).

Note that the individual persons or machines that originate messages need not be concerned with this procedure other than responding to authentication challenges from the internal AS. The gateway handles external authentication requests, retransmission of the first message in a session, as well as all encryption.

Several of the functions that logically are done in the gateway when a session is first authenticated may be offloaded to different hardware in order to improve the efficiency of forwarding packets that belong to ongoing sessions. However, if the level of authentication is such that sessions consist of one message only (e.g., authenticating electronic mail), there is little savings. On the other hand, if each packet in a mail, remote login, or file transfer session is authenticated individually, the overhead may be great and warrant offloading. Therefore, the appropriate engineering depends on the level of interconnection, i.e., whether the gateway is a packet forwarding gateway or an application level gateway in which application protocols are terminated.

To offload this function to a server, the protocol would be modified as follows. When the first packet in a session arrives it is assumed to be unauthenticated and is forwarded to the ION policy server which sits in the destination organization (B in the above example). The policy server carries out the protocol listed above for the gateway (gw_b in the above example). The gw automatically forwards all unauthenticated incoming packets to the policy server during this dialog with the third party ION authenticator (AS_{ab}). Once the source organization is authenticated and the session key is obtained, the ION policy server sets the port in the gateway to authenticated and sets the session key. From then on packets arriving to that port in that key will be forwarded to the destination(s) for which they were authorized (determined by the rights assigned to the source organization, see [1]), until the session is closed or until either side decides to reauthenticate. In either event, the policy server resets the port and session key entries. The policy server could also handle the initiation protocols for authenticating new clients.

5. Multiple Levels of Service

Different types of transactions require different degrees of confidence in the credit or authenticity of the client. And, different strengths of authentication require different types of equipment and facilities. When the highest level of authentication is not available, some lower level of authentication may be adequate. If a purchase order arrives for $10,000 worth of goods, the supplier must be relatively confident that the client is legitimate and in fact made the order, before the order is acted upon; the cost associated with incorrect authentication is high. However, if a smaller client sends a purchase order for $100 worth of goods, relatively little authentication may be necessary and the facilities needed for the protocol described above may not be available. Therefore, it would be nice to support intermediate services, i.e., multiple levels of service.[6]

One method for offering a "second-class" authentication scheme is to rely solely on initiation type authentication. I describe this approach below.

5.1. Protocol

The protocol begins when A sends a message to B. We assume that A has no encryption capabilities at all.

Initiation authentication is only slightly affected by the lack of encryption capabilities. If A is not on B's known-entity list then B contacts a (set of) third party(ies) to authenticate the existence of organization A and to evaluate it. Assuming B contacts a third party that does have A registered, that third party returns to B values of the requested evaluation criteria along with a flag indicating the level of authentication that A can support; for example, first-class to indicate that A has encryption capabilities and can carry out the protocol described earlier, and second-class to indicate that A has no capabilities and must rely on passwords sent in the clear to authenticate itself to the third party.

Transaction authentication can no longer rely on a Needham-Schroeder protocol since A has no encryption or decryption capabilities. Therefore, when B asks the third party to authenticate a particular transaction or message from A (either the first transaction or later ones), the third party informs B that only second-class transaction authentication is available. One procedure that the third party could use in the absence of encryption would be to ask the source of the message to B (presumably A) to resend the password that it submitted upon registration. If the resent password matches A's registered password, the third party could send a message to B indicating that the third party believes the source of the message is in fact A. Similarly, the third party could authenticate B and inform A that the third party believe that the destination is B. In both cases, the third party must include the authentication level rating, second-class. A and B can then decide whether to accept or reject this level of authentication for the proposed transaction. The primary risks are that

[6] This feature was suggested by J. Saltzer.

there is no session key for the parties to authenticate themselves to one another directly and there is no control over an impostor intervening in the transaction after it has begun. In addition, passwords are subject to intervention because they are sent to the third party in the clear.

For certain types (low risk) of transactions and communications, this limited level of assurance may be acceptable, and preferable to no authentication at all to the extent casual impostors are detected or discouraged. However, it is vital that both parties keep track of the level of authentication in use. For example, if in the middle of a transaction A proposes to increase a purchase order by an order of magnitude, B should know that only second-class authentication is being used and reject the suggestion if it sees fit.

6. Conclusion

In summary, organizations can initiate relationships with one another using third parties to authenticate one another's identity and desired credit information, can carry out transactions using third parties to authenticate that the transaction request travels from and to the claimed party, and finally, both of these activities can be carried out at the appropriate authentication cost level.

References

[1]

Estrin, D.
Non-Discretionary Controls for Inter-Organization Networks.
In *Proceedings of the 1985 Symposium on Security and Privacy.* IEEE Computer Society Press, Silver Spring, MD, 1985.
Accepted for Publication

[2]

Needham, R., Schroeder, M.
Using Encryption for Authentication in Large Networks of Computers.
Communications of the ACM 21(12):993-999, December, 1978.

[3]

Routhier, S.
An Improved Authentication Server For Inter-Computer Communication.
B.S. Thesis, Massachusetts Institute of Technology, Dept. of Electrical Engineering and Computer Science, June, 1983.