Security, Payment, and Privacy for Network Commerce

B. Clifford Neuman

(Invited Paper)

Abstract— **As the Internet is used to a greater extent in business, issues of protection and privacy will have more importance. Users and organizations must have the ability to control reads and writes to network accessible information, they must be assured of the integrity and confidentiality of the information accessed over the net, and they must have a means to determine the security, competence, and honesty of the commercial service providers with which they interact. They must also be able to pay for purchases made on the network, and they should be free from excessive monitoring of their activities. This paper discusses characteristics of the Internet that make it difficult to provide such assurances and surveys some of the techniques that can used to protect users of the network.**

I. Introduction

THE Internet is important as a facilitator of electronic commerce. Its use to conduct business between organizations is growing and it is now seen by many as an important means to reach the consumer. With such commercial use, the potential loss from unauthorized access and inappropriate disclosure, modification, and retention of information make the implementation of security and privacy measures critical.

Because attacks against security can be perpetrated at many points in the system, including the user's computer, the service provider's system, or at any point on the network between the two, security measures are needed throughout the system. This paper covers security measures applied to the exchange of messages between computer systems. These measures are implemented by computer systems which must themselves be protected from local compromise. Methods for protecting systems from local compromise have been discussed at length in the operating system literature [1], [2] and are not covered by this paper.

Firewalls [3] can be used to protect end systems from network attacks by screening network packets destined for systems on networks protected by the firewall. The firewall protects the end system against attacks on protocols and software that are blocked by the firewall, but it does not provide protection for applications and protocols that are allowed to communicate through the firewall. Most commerce applications require communication with outside systems and are thus not directly affected. A firewall does provide a line of defense against outside compromise of back-end systems, but because it only indirectly affects the security of commerce protocols, further discussion of firewalls is beyond the scope of this paper.

This paper begins with a discussion of common security requirements for conducting commerce in the physical world and the characteristics of distributed computer systems that make it difficult to meet those requirements. Among these requirements are confidentiality, integrity, authentication, authorization, payment, assurance, and privacy. Technical means for satisfying the confidentiality, integrity, authentication, authorization, and payment requirements are described. These solutions build upon one another with the basic technology that satisfies some requirements used as building blocks for meeting others.

Some of the requirements, like assurance and privacy, do not have purely technical solutions. To meet these requirements, one must make available the technical means for individuals and organizations to act responsibly and impose a legal or contractual obligation to do so. The technical measures supporting responsible use are described.

Though most of the technology needed to provide security for network commerce already exists, use of the technology is not widespread. The paper will conclude by discussing the scalability of some of the mechanisms, why the technology has not been widely deployed, the need for infrastructure supporting the technology, and the need for integration with network applications.

II. Security for "Real-World" Transactions

When we conduct business in the physical world, we hold certain beliefs about the privacy and security of the information involved. For example, when one gives a credit card number to a travel agent, one expects *confidentiality*—that it will not be disclosed to anyone other than those with a legitimate need to know, such as the airlines and the bank. Similarly, this situation requires *integrity*—that neither the amount of the purchase nor the itinerary will be inappropriately altered.

Manuscript received February 3, 1995; revised May 25, 1995. This research was supported in part by the Advanced Research Projects Agency under NASA Cooperative Agreement NCC-2-539 and through Ft. Huachuca under Contract no. DABT63-94-C-0034. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the funding agencies. Figures and descriptions in this paper were provided by the authors and are used with permission.

The author is with the Information Sciences Institute of the University of Southern California, Marina del Rey, CA 90292 USA.

IEEE Log Number 9414022.

Each party may require *authentication* of the other—assurance that a party is the one claimed. When dealing with a business in person, authentication of the business to the customer is often implicit, based on the place of business and permanence of the facilities. For transactions where the business needs to know the identity of the customer—e.g., accepting a personal check or granting access to medical records—authentication may be based on physical evidence like a photograph on a driver's license, or comparison of signatures. When conducting business over the phone, authentication is often accomplished by answering queries about private information like one's personal identification code, social security number, or mother's maiden name.

Once the parties to a transaction are known, *authorization* allows the party of whom the request was made to determine if the other party is allowed to perform the requested action. Authorization may be based on local information like a list of legitimate users of a service, or may be obtained by checking with a third party (e.g., credit card authorization, or a security guard calling someone before allowing entry to a building). Just as the party providing a service wants to know that the requester is authorized, the requester might require *assurance* that a service provider is competent and worthy of trust. Such assurances usually take the form of business licenses, endorsements, and surety bonds.

For some business transactions we want *privacy* of the transaction details. For example, an organization conducting research might purchase information on the network, but might not want competitors to learn what has been purchased. Ideally, information would be maintained only for the purposes of providing the service and for billing, and detailed information about the transactions would remain private.

Our beliefs about the security of transactions in the physical world are based on our assumptions about the infrastructure we use to conduct them. For the typical transaction, we take for granted that that our telephone lines are not tapped, we expect to detect changes to and forgeries of physical documents, and we trust the service providers that we deal with face to face. These assumptions hold because the benefit to an attacker of compromising routine transactions is often small compared to the effort required and the risk of being caught. For more critical transactions, additional security measures must be taken, often similar to those that we describe in this paper for use on computer networks.

III. The Security of Computer Networks

In 1983 Voydock and Kent described attacks that occur on computer networks, and surveyed countermeasures [4]. These threats include eavesdropping, modification of messages, impersonation, traffic analysis, and denial of service. The nature of computer networks makes many of these attacks trivial. In particular, when a message is sent across a computer network, it passes through many computer systems, including systems of other network users. The broadcast nature of many network links makes it even easier for others to eavesdrop on network communications.

As use of the network for commerce grows, business will be conducted more frequently between parties that have never met one another. This reduces the customer's confidence in the legitimacy of the merchant, and makes it harder for the merchant to verify the customer's identity. Also, information sent over computer networks can be quickly scanned by programs looking for sensitive information like passwords and credit card numbers. Further, electronic documents are easily copied, and a copy is identical to the original. Finally, modifications to electronic documents cannot be detected unless the document is electronically signed.

IV. Protecting Electronic Commerce

Electronic commerce and communication on the Internet require the same protections we expect in traditional commerce, but the means employed to provide the protections are often different.

A. Confidentiality, Integrity, and the Use of Cryptography

The best way to protect the confidentiality of data accessed though computer networks is with encryption. Encryption is a transformation of data that varies based on a secret parameter called an encryption key. Using symmetric (or *conventional*) cryptosystems like the Data Encryption Standard (DES) [5], triple-DES, and IDEA, data transformed (or *encrypted*) using an encryption key is scrambled in such a way that it can only be unscrambled (or *decrypted*) by a similar transformation using the same encryption key. The scrambled data is called *ciphertext*, and the original or subsequently unscrambled data is called *plaintext*. When the sender and receiver share an encryption key known only to them, data can be encrypted before transmission, and decrypted after transmission, protecting the data from disclosure to eavesdroppers.

Besides protecting the confidentiality of data, encryption also protects data integrity. Because knowledge of the encryption key is required to produce ciphertext that will yield a predictable value when decrypted, modification of the data by someone who does not know the key can be detected by attaching a checksum to the data before encryption and requiring that the receiver verify the checksum after decryption.

When using symmetric cryptosystems, both the party encrypting the data and the party decrypting the data must share the same encryption key. This means that a user needs a different key for every other user or service provider with which it exchanges information or messages, and each service provider would have to maintain a key for every potential customer. This limitation can be addressed in two ways: through the use of asymmetric (or *public key*) cryptography, or by using a mutually trusted intermediary to generate a new encryption key which is distributed to both parties.

In asymmetric cryptography, encryption and decryption are performed using a pair of keys such that knowledge of one key does not provide knowledge of the other key in the pair [6]. One key, called the *public key* is published, and the other key, called the *private key*, is kept private. The principal advantage of asymmetric cryptography is that secrecy is not needed for the public key, meaning that only a single key pair needs to be generated for each user or service provider, and dissemination of the keys is easier.

The principal disadvantage of asymmetric cryptography is its performance. Existing asymmetric cryptosystems are significantly slower than their symmetric counterparts. With common key sizes of 512 to 1024 bits, encryption of a single block with a private key using the RSA algorithm [7] can take on the order of 10th of a second or longer to complete on computers in common use today. The public key operation using the Digital Signature Algorithm (DSA) [8] has similar performance.

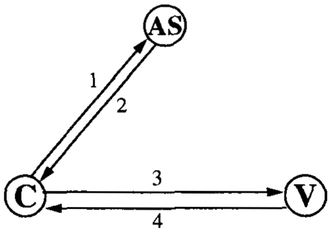
Because of the performance issues, asymmetric cryptography is rarely used in isolation. Instead, it is used to encrypt a symmetric encryption key and checksum, which are in turn used to protect the actual data. This technique is used in systems that protect electronic mail, including Privacy Enhanced Mail (PEM) [9], Pretty Good Privacy (PGP) [10], and in secure versions of protocols for the World Wide Web including Secure HTTP (SHTTP) [11] and Secure Sockets Layer (SSL) [12].

While using asymmetric cryptography in this manner reduces the cost of encrypting large messages and documents, at least one encryption operation using an asymmetric algorithm is required for each signed document and for the exchange of a symmetric key between any new pair of users. For some applications, particularly transaction processing applications that handle many operations per second on behalf of different clients, the transaction rate precludes the use of asymmetric encryption. Asymmetric encryption is well suited for use in store and forward applications such as electronic mail and information dissemination applications where documents can be signed before they are stored (e.g., many web documents).

When using asymmetric cryptography to exchange symmetric encryption keys or to sign checksums, each party must know the other party's public key *a priori*, or rely on a trusted third party to certify the other user's public key. Without a trusted third party it becomes possible for an attacker to replace the public key of a participant with a different key for which the corresponding private key is known by the attacker. This allows the attacker to decrypt messages encrypted using the fictitious key and generate messages signed by the key. Similarly, when using purely symmetric cryptography, a trusted third party intermediary with whom both parties share an encryption key can generate and distribute a new key, called a session key, to be used between parties that do not share a key directly. The use of such third parties for the exchange and certification of encryption keys is closely tied to authentication.

B. Authentication

Network service providers require the ability to identify the user making a request accurately. In traditional systems, the user's identity is verified by checking a password typed during login; the system records the identity and uses it to determine what operations may be performed. Password-based authentication is not suitable for use on computer networks. Passwords sent across the network can be intercepted and subsequently used by eavesdroppers to impersonate the user. While this vulnerability has long been known, it was recently demonstrated dramatically with the discovery of planted password collecting programs at critical points on the Internet [13].



1. as_req: c, v, $time_{exp}$, n
2. as_rep: $\{K_{c,v}$, v, $time_{exp}$, n, ...$\}K_c$, $\{T_{c,v}\}K_v$
3. ap_req: $\{$ts,ck, $K_{subsession}$, ...$\}K_{c,v}$ $\{T_{c,v}\}K_v$
4. ap_rep: $\{$ts$\}K_{c,v}$ (optional)

$T_{c,v} = K_{c,v}$, c, $time_{exp}$ ...

Fig. 1. Basic Kerberos authentication protocol (simplified).

To address this problem, one uses an authentication protocol to prove knowledge of a password, without actually sending the password across the network. This can be accomplished by using an encryption key in place of a password and proving knowledge of the encryption key. Because knowledge of the encryption key is required to produce ciphertext that will yield a predictable value when decrypted, knowledge of the encryption key can be demonstrated by encrypting a known, but nonrepeating value, and sending the encrypted value to the party verifying the authentication. As was the case with encryption for confidentiality and integrity, unless each party knows the other party's key *a priori*, a trusted third party is required to certify or distribute the keys.

Such an authentication and key distribution protocol was described by Needham and Schroeder in 1978 [14]. The Kerberos system [15] is based in part on the symmetric version of the Needham and Schroeder authentication protocol, with changes that reduce the number of messages needed for basic authentication and the addition of facility for subsequent authentication without re-entry of the user's password.

The Kerberos protocol is shown in Fig. 1. When a client (C) wishes to communicate with a service provider (the verifier, V) it contacts the Kerberos authentication server (AS), sending its own name, the name of the server to be contacted, and additional information (1). The Kerberos server randomly generates a *session* key ($K_{c,v}$) and returns it to the client encrypted in the key derived from the user's password (K_c) and registered in advance with the Kerberos server (2). The encrypted session key is returned together with a *ticket* ($T_{c,v}$) that contains the name of the client and the session key, all encrypted in the service provider's key (K_v).

The session key and ticket received from the Kerberos server are valid until $time_{exp}$ and are cached by the client, reducing the number of requests to the Kerberos server. Additionally, the user's secret key is only needed when initially logging in. Instead of using the user's secret key, subsequent requests during the same login session use the session key returned by the Kerberos server in response to the initial request.

To prove its identity to a service provider, the client forwards the ticket together with a timestamp encrypted in the session key from the ticket (3). The service provider decrypts the ticket and uses the session key contained therein to decrypt the timestamp. If the timestamp is recent, the server knows that the message was sent by someone who knew the session key. Since the session key was only issued to the user named in the ticket, this authenticates the client. If the client requires authentication from the server, the server extracts the timestamp, re-encrypts it using the session key and returns it to the client (4).

It is important to note that users and service providers only have to register encryption keys in advance with the Kerberos server itself, and not with each party with which they will eventually communicate. The Kerberos server, as a trusted intermediary, generates a session key when needed, distributes it to the client, and places it in the ticket where it can be subsequently recovered by the service provider. This session key can then be used directly by the client and the service provider for encrypted communication as described in the previous paragraphs.

The Kerberos authentication protocol is based on symmetric cryptography, but authentication (and key distribution) can also be accomplished using asymmetric cryptography. Asymmetric cryptography has several advantages over symmetric cryptography when used for authentication. These include more natural support for authentication to multiple recipients, better support for nonrepudiation (repudiation is a party's denial of having sent a message), and the elimination of secret encryption keys from the central authentication server.

As commonly implemented with asymmetric cryptography, the trusted intermediary is called a certification authority (or CA) and resides off-line since it need not be contacted at the time authentication occurs. Instead, the CA digitally signs a ''certificate'' binding the name of a client or server to a public key. This certificate can be presented by the client during authentication, or it can be stored in a directory service and retrieved on demand by the verifier.

Running the CA off-line improves the security of the system since a compromise of the CA would be devastating, and since it is easier to protect a system that is not directly connected to the network. But, when run off-line, certifications must last for an extended period, usually on the order of years, making recovery from compromised user keys more difficult. This ''revocation'' problem is often addressed by re-introducing a trusted on-line authority that is consulted in conjunction with the credentials issued by the off-line CA, diminishing, but not eliminating, the advantages of using an off-line authority.

As discussed in Section IV-A, the principal disadvantage of asymmetric cryptography is its performance. This problem is usually addressed by using hybrid protocols that combine symmetric and asymmetric encryption.

Encryption and authentication technology has been available for years, yet our networks are still not secure. This is because widespread deployment of these technologies requires deployment of infrastructure supporting them (i.e., the trusted third party authentication servers and CA's) and integration of the technologies with applications. Infrastructure and integration are described in Section VI.

C. Authorization

Though an important security service, authentication is only the means to determine who the user is, so that a decision can be made about what the user is allowed to do. Authorization is the process of deciding whether the user is allowed to perform a particular operation. Authorization in existing systems is usually based on information local to the server. This information is present in access control lists associated with files or directories, files listing individuals authorized to login to an account (e.g., .rhosts, or .k5login files), and sometimes files read over the network.

There are several efforts underway to develop distributed authorization services that support the maintenance of authorization information, such as group membership and access control lists, separate from the services that use them [16], [17]. These approaches use certificates signed by an authorization service to assert information such as group membership, or the authority to perform a specific operation. Upon receiving such a certificate, a service provider verifies the signature of the authorization server, and checks to make sure the rights conveyed by the certificate allow the operation requested by the user. Implementation of the signed authorization certificates depends on the integrity and authentication services described earlier.

Distributed authorization services will be critical for electronic commerce. Such services provide better management of authorization information and support the sharing of authorization information across computers and applications. For example, service providers may make a service available to members of particular groups. Without a distributed authorization service, each service provider would have to maintain its own list of group members and adding or deleting a member from a group would require an update to all service providers.

D. Payment

Money is of critical importance in traditional commerce, and the same is true for electronic commerce. Widespread commercial use of the Internet will require secure payment services. Recently proposed, announced, and implemented Internet payment mechanisms can be grouped into three broad classes: electronic currency systems, credit-debit systems, and systems supporting secure presentation of credit card numbers.

In electronic currency systems like Chaum's DigiCash system [18] and USC-ISI's NetCash system [19], both of which are undergoing noncommercial trials on the Internet, customers purchase electronic currency certificates from a currency server, paying for the certificates through an account established with the currency server in advance or through other forms of payment. Once issued, the electronic currency represents value, and may be spent with merchants who deposit the certificates in their own accounts or spend the currency elsewhere. The principal advantage of electronic currency is its potential for anonymity. The principal disadvantage of electronic currency mechanisms is the need to maintain a large database of past transactions to prevent double spending (the use of indistinguishable copies of currency).

In payment mechanisms based on the credit-debit model, including CMU's NetBill [20], First Virtual's Internet payment system [21], and USC-ISI's NetCheque system [22], customers maintain accounts on a payment server and authorize charges against those accounts. Payment services supporting the credit-debit model rely on authorization services like those described in the previous section: NetBill uses Kerberos directly, while the NetCheque system is layered over a proxy-based authorization service based on Kerberos. An important advantage of NetBill and NetCheque are their low transaction cost. This is critical if such systems are to support small payments (called *micropayments*) on the order of pennies that one is likely to see as payments for database queries and royalties for access to individual documents.

Secure credit card transactions constitute the third class of network payment services. This model is used in the initial system offered by CyberCash, and it is the model used today by NetScape and most secure variants of Mosaic. When using this model for payment, the customer's credit card number is encrypted using a key distributed using asymmetric cryptography so that it can only be read by the merchant, or in some approaches by a third party payment processing service. The biggest advantage of this approach is that the customer does not need to be registered with a network payment service; all that is needed is a credit card number. This provides a much larger customer base for merchants accepting this method of payment. Encryption using this approach prevents an eavesdropper from intercepting the customer's credit card number. It is important to note, however, that without advance registration of customers, the encrypted credit card transaction does not constitute a signature; anyone with knowledge of the customer's credit card number can create an order for payment, just as they can fraudulently place an order over the telephone. Because of the cost of clearing credit card payments through the existing financial infrastructure, this model of payment is not suited for micropayments where the transaction cost would be many times the payment amount.

V. Nontechnical Solutions

Constraints on the use of a system fall into two categories: 1) clearly defined access controls that must be enforced by strong security services, and 2) customary practices that all users are expected to follow, but that cannot be enforced without knowledge of the intent of an action (e.g., does an action constitute personal or commercial use, is a message advertising or informational, etc.). The first kind of constraints are enforced by the security services already described. The second can be addressed by establishing a framework for services in which accepted behavior is automatic. While it may still be possible for users and businesses to violate such constraints, doing so will demonstrate willful intent and may be dealt with by the judicial system.

A. Assurance

As the conduct of electronic commerce increasingly occurs between parties that have no pre-existing relationships with one another, customers will have a harder time assessing the character and competence of service providers. In the ''real world'' customers may rely on licensing, endorsements, liability insurance and surety bonding to compensate for such lack of confidence. For example, in the United States, the Better Business Bureau provides information about local businesses to offer customers assurance that the business is reputable. In computer systems, such assurances can be represented as electronic certificates digitally signed by a licensing authority, endorser, or insurance provider. Such *assurance credentials* would be granted to a service provider after it meets the requirements set by the organization issuing the credentials [23].

These assurance credentials can be implemented in a manner similar to the way authorization credentials are implemented, except that it is the customer that verifies the credentials of the service provider. Though there are no technical safeguards ensuring that a service provider in possession of such assurance credentials acts in a safe and honest manner, the service provider has a contractual and legal obligation to do so, and depending on the level of assurance conferred by the credential, the service provider may be subject to periodic inspection by the assurance provider. Repeated complaints by customers can result in the loss of certification.

B. Intellectual Property Protection

There is a large collection of information owned by publishers and information providers that is not yet accessible on the network. Information providers are concerned that once they make information available electronically, copies will propagate and they will lose the ability to collect royalties. Though some have proposed elaborate technical mechanisms to prevent illegitimate copying, it is our view that such attempts are bound to fail (consider the failure of software copy-protection mechanisms). A better approach is to make it easy for users to follow copyright law, with royalty payments occurring automatically, and to resort to the legal system when users choose to violate the law. Evasion of the payment of royalties, though not technically difficult, would require willful action, demonstrating intent.

The Corporation for National Research Initiatives has an effort underway to support the registration of documents for copyright protection, and to provide a service for clearing royalty payments for access to such documents. One way to make it easy for users to follow copyright law is to attach such a copyright clearance tag to each document, and to propagate the tag with copies of the document. Access to the document would be through applications that interpret the tag and automatically pay the fee, or ask the user if they really want to retrieve the document.

While such voluntary compliance is the only practical solution to controlling the copying of copyrighted materials on end systems, information services could enforce the payment of royalties during retrieval. Such servers would track and aggregate payments and periodically submit payment to the copyright clearance center with an accounting of the clearance codes and the amount to be credited to each.

C. Privacy and Audit

A great deal can be learned about an individual from transaction records, especially those pertaining to the queries made by a user or the documents read. While there are often legitimate reasons to track access to particular resources, it is also often appropriate to protect from disclosure information about what a user is reading, or the purchases that have been made. For example, an organization conducting research does not want its competitors to learn which publications have been purchased, since such knowledge may divulge information about the research. In any event, it should be possible for the user to determine when such records are maintained and to specify limits on the detail of the information recorded, and its use.

While not possible to control what a service does with the transaction details in its possession, by allowing users to tag requests with an indication of the extent to which transaction details may be recorded, excessive abuse of such information will either be reduced or exposed. Users wanting to assert their privacy rights could configure their system to always limit the information that can be collected. Service providers would define policies about how much detail they wish to record, and the minimum they insist on recording (e.g., information required for billing). If a service insists on recording more detail than the client allows, the client's request would be denied.

While this approach cannot prevent dishonest service providers from recording such transaction details counter to the user's request, such use would make the service provider vulnerable to legal action. Users will learn how much detail is reasonable as they experiment to determine the lowest threshold they can set without incurring frequent interruption from service denials. Competition between service providers will encourage less intrusive logging by servers.

VI. Deployment

This paper has described much of the technology that is needed to support the conduct of electronic commerce on the Internet. Unfortunately, though available in the laboratory, the technology has not been widely deployed. Security technologies have traditionally been deployed within individual organizations. Yet, for use in electronic commerce, the technologies must be deployed on a network-wide basis.

Most of the security mechanisms described are scalable in design and implementation but they are not scalable in isolation; a network-wide security infrastructure connecting individual security servers needs to be established, and the security mechanisms need to be integrated with applications.

A. Scalability

For our discussion of scalability, we concentrate on the scalability of authentication services. Scalability of authorization and payment services depends on the scalability of the underlying authentication methods.

A system is said to be scalable when the addition of users and resources can be handled without a significant decrease in performance or increase in administrative complexity. A system's scale has three components: numerical, geographic, and administrative [24]. Where trust is involved, it is the administrative component, i.e., the number of organizations that are part of a system, that is hardest to deal with.

For use on a world-wide basis, it is not practical to require all users to register with a single authentication server or certification authority. Instead, users will register with authentication servers or certification authorities maintained by their own organization. Yet, it must be possible for users from one organization to prove their identities to services in other organizations.

Both symmetric and asymmetric authentication systems support multiple registration authorities and these registration authorities certify one another. In Kerberos, the administrative locale over which a particular server has authority is called a "realm." As long as a "certification path" exists between the principal trying to prove its identity and an authority with which the verifier has *a priori* knowledge of an encryption key, authentication will succeed.

When using asymmetric cryptography, the user can present a chain of certificates starting from a common "root" certification authority whose public key is known by all verifiers [25]. The first certificate is signed by the root CA and certifies the public key of the CA that signs the next certificate. Each certificate in turn certifies the public key of the CA at the next lower level until the final certificate certifies the public key of the end user. The verifier can check the signatures in the chain without interacting with CA's, yet each CA must be trusted to certify only legitimate keys for registered users.

Similarly, Kerberos realms are organized hierarchically with higher level authentication servers distributing keys for use with lower level authentication servers along a path between the client and the verifier. With Kerberos, intermediate authentication servers must be contacted whenever authentication is required with a server in a new realm. This requirement can be loosened by allowing authentication servers to cache encryption keys for use between realms that frequently communicate.

B. Infrastructure

For authentication to work, authentication servers and certification authorities must be established and users and service providers must have their encryption keys registered. The process of establishing these registries has been slow because of liability and trust issues associated with running such authorities, especially those that certify other authorities. These issues have, to date, hindered widespread availability of such services.

Recently, as organizations have seen how profitable such certification services might be, there is renewed interest in resolving these issues. Unfortunately, though the profit motive will lead to more interest in a solution, it may also lead to competition instead of cooperation and delay deployment of the infrastructure. What is likely to happen is that there will be several certifiers, and users will have to register with each certifier and decide which certifications they trust. This is middle ground between requiring a universally trusted root of a certification hierarchy [25], and the model employed by PGP

[10], a grass-roots tool for public key encryption. In PGP, users certify the encryption keys of other users, without globally trusted certification authorities.

Because off-line certification authorities are possible when using asymmetric cryptography, because secrecy is not required for the public key, and because knowledge of the key presented to the CA for registration does not allow impersonation of the user to whom that key is registered, issues of liability are more easily dealt with in an asymmetric system. For this reason, we expect that the certification infrastructure that will be used for registering users and connecting organizations will be based on assymetric cryptography. Extensions to Kerberos have been proposed to use the public key infrastructure for initial user authentication and to exchange inter-realm keys between authentication servers [26]. For performance reasons, authentication to application servers using Kerberos may continue to be based on symmetric cryptography.

C. Integration

While the lack of a universally accepted infrastructure is hindering deployment, it is not the only problem. Deployment also requires integration of these security services with applications and operating systems. Security services can be integrated with protocols at several layers.

There are efforts underway in the Internet Engineering Task Force (IETF) to add security services at the IP layer [27]. With these extensions, computer systems will be able to authenticate to one another, and communication between the systems can be encrypted. Integrating security services at this layer does not provide authentication of the individual users of the system to the remote service providers, and thus, does not by itself meet the requirements for authentication (in support of access control) by many applications. It does, however, improve the confidentiality and integrity of communications by applications running on those systems, including applications which have not been modified to use application level security services.

Integration of security services can also occur at the application layer, and changes at the application layer are necessary for services where the operations allowed depend on the identity of the user. Integrating security at this layer can be cumbersome, requiring changes to the application protocol for each application. The Common Authentication Technology Working Group of the Internet Engineering Task Force has developed the Generic Security Services Application Programming Interface (GSS-API) [28] to facilitate the integration of security services at the application layer. When using the GSS-API, applications make calls to authentication, confidentiality, and integrity services in a manner that is independent of the underlying security services.

Integration of security services is easier for applications that run on top of remote procedure call and similar transport mechanisms. When running on top of such protocols, user authentication, confidentiality, and integrity can be provided at the transport or session layers. Though the application must still be modified to ask the right questions, and to use the answers as a basis for authorization, such changes to the application are less intrusive than changes to the application protocol itself. Security services have been integrated at the RPC layer for the Open Software Foundation's DCE RPC [29], and Sun's ONC RPC [30].

NetScape's Secure Socket Layer (SSL) [12] integrates security services at the session layer for applications that use a socket (TCP style) transport interface. SSL provides an encrypted and integrity-protected layer over which HTTP and other protocols may be transmitted. It is important to note that SSL provides security for the HTTP connection, and thus protects the integrity of documents as they are transmitted to the user, but it does not protect the integrity of documents while stored on the server, which would require that the document be signed by the author.

Security for the World Wide Web can also be provided at the application layer. Secure HTTP (SHTTP) [11] allows a web browser to request a digital signature for a retrieved document. The digital signature is a checksum of the document encrypted using the private key of the signator—usually the document's author. The checksum can be decrypted with the public key of the signator and the integrity of the document verified. Links to documents under SHTTP identify the signator of the document whose public key must be used to verify the signature.

Like SSL, the SHTTP protocol provides for the protected transmissions of data in forms sent by the user to the server. This is accomplished by encrypting the data stream using a key that is itself sent encrypted in the public key of the HTTP server, preventing eavesdropping. The primary use of this feature is to allow users to safely provide servers with passwords or credit card numbers.

Similar efforts are underway to integrate support for payment into information service protocols. At present, each payment system provides its own method for requesting payment from the user and allowing the user to respond. Different extensions are used by First Virtual, CyberCash, DigiCash, NetCheque, NetBill, and others. OpenMarket has developed a web server that supports multiple payment options through a payment switch [31] that runs on an OpenMarket server. Because it runs on a system operated by OpenMarket, new payment methods are more readily integrated with the payment switch, while the merchant interface remains unchanged.

Most of the methods for integrating payments with the web were designed to work with minimal or no modifications to existing web browsers. More efficient integration is possible when changes can be made to the browsers, and the HTTP, SHTTP, and SSL protocols. Discussion between players is underway to ensure that such extensions are payment-method independent so that browsers and servers are not tied to individual payment methods.

VII. Unresolved Issues

Most of the security measures described in this paper depend for their security on the protection from disclosure of the user's encryption key. Just as infrastructure is required for registering users and their encryption keys, integration and hardware support are needed to allow users to enter their keys securely from the systems they use. As points of connection to the Internet become ubiquitous, users will begin to connect to the network from computers that are available in public locations. If authentication, authorization, and payment require the user to enter a password or encryption key, then the software at the point of connection might be modified to keep a copy of the password and provide it to an attacker.

One-time passcode devices can be used to address this problem. The user would carry a device that accepts a challenge entered by the user and returns a passcode that is to be used for authentication. This passcode would be used in combination with a password or encryption key for authentication. Because the passcode would change on each use, an attacker who obtained a passcode from a compromised workstation would not be able to use it for subsequent impersonation of the user. However, once a user has logged in to a compromised workstation, that workstation could originate requests on behalf of the user for the duration of the initial authentication period.

Where one time passcode devices are not sufficient, the user can carry a smartcard that is connected to the computer through a smartcard reader or a PCMCIA slot. The smartcard is hardware trusted by the user to maintain the user's encryption keys and sign messages on behalf of the user without divulging the keys to the computer device to which it is connected. When using a smartcard, a compromised workstation could not itself generate messages signed directly by the user, but the workstation could try to trick the smartcard into signing inappropriate messages.

VIII. Conclusion

The Internet is being used increasingly for commerce, and with such use we will encounter more attacks on the security of the system for monetary gain. When compared with commerce in the "real world," network commerce affords reduced personal contact, ease of eavesdropping, the ability of attackers to automatically extract sensitive information from messages, and easy copying and modification of data. As these weaknesses are increasingly exploited we will see an increased emphasis on the integration of security mechanisms with applications and network services.

Much of the technology needed to protect network systems already exists. Cryptographic techniques can be applied in support of authentication, authorization, integrity, confidentiality, assurance, and payment. Technology can also be applied in conjunction with legal and contractual obligations to provide protection for intellectual property and individual privacy. To be useful, however, the infrastructure supporting these technologies must be put in place, and the technology must be integrated with applications and protocols for electronic commerce.

Acknowledgment

The author wishes to acknowledge the suggestions and assistance of C. Anderson, S.-M. Cheang, R. Kaplan, D. Kutsikos, A. Medvinsky, S. Rao, B. Tung, and J. Yamazaki in the preparation of this paper. Special thanks are also in order for M. Schwartz whose efforts to coordinate the input of many reviewers was extremely helpful.

NETCHEQUE and NETCASH are service marks of the University of Southern California. All other product or service names mentioned herein are trademarks or service marks of their respective owners.

References

[1] J. H. Saltzer and M. D. Schroeder, "The protection of information in computer systems," *Proc. IEEE*, vol. 63, no. 9, Sept. 1975.

[2] M. Branstad, H. Tajalli, F. Mayer, and D. Dalva, "Access mediation in a message passing kernel," in *Proc. 1989 IEEE Symp. Security and Privacy*, May 1989.

[3] S. M. Bellovin and W. R. Cheswick, "Network firewalls," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 50–57, Sept. 1994.

[4] V. L. Voydock and S. T. Kent, "Security mechanisms in high-level network protocols," *ACM Computing Surveys*, vol. 15, no. 2, pp. 135–171, June 1983.

[5] National Bureau of Standards, "Data Encryption Standard," Federal Information Processing Standards Publication 46, Jan. 1977.

[6] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Trans. Inform. Theory*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[7] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, Feb. 1978.

[8] National Institute of Standards and Technology, "Digital signature standard," Federal Information Processing Standards Publication 186, May 1994.

[9] S. T. Kent, "Internet privacy enhanced mail," *Commun. ACM*, vol. 36, no. 8, pp. 48–60, Aug. 1993.

[10] P. Zimmermann, *PGP(tm) User's Guide*, vol. 1 and 2, 1994, Distributed with PGP 2.6; ftp to net-dist.mit.edu:/pub/PGP/.

[11] E. Rescorla and A. Schiffman, "The secure hypertext transfer protocol," Internet draft rescorla-shttp-0, Dec. 1994.

[12] K. E. B. Hickman, "The SSL protocol," Internet draft hickman-netscape-ssl-00, Apr. 1995.

[13] Computer Emergency Response Team, "Ongoing network monitoring attacks," CERT Advisory CA-94:01, 3 Feb. 1994.

[14] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.

[15] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Commun.*, vol. 32, no. 9, pp. 33–38, Sept. 1994.

[16] B. C. Neuman, "Proxy-based authorization and accounting for distributed systems," in *Proc. 13th Int. Conf. Distributed Computing Systems*, May 1993, pp. 283–291.

[17] M. E. Erdos and J. N. Pato, "Extending the OSF DCE authorization system to support practical delegation," in *Proc. PSRG Workshop Network and Distributed System Security*, Feb. 1993.

[18] D. Chaum, "Achieving electronic privacy," *Scientific Amer.*, pp. 96–101, Aug. 1992.

[19] G. Medvinsky and B. C. Neuman, "NetCash: A design for practical electronic currency on the Internet," in *Proc. First ACM Conf. Computer and Communications Security*, Nov. 1993.

[20] M. Sirbu and J. D. Tygar, "NetBill: An electronic commerce system optimized for network delivered information and services," in *Proc. IEEE Compcon '95*, Mar. 1995, pp. 20–25.

[21] M. T. Rose and N. S. Borenstein, "The simple green commerce protocol (SGCP)," First Virtual Holdings Incorporated, Oct. 1994.

[22] B. C. Neuman and G. Medvinsky, "Requirements for network payment: The NetCheque perspective," in *Proc. IEEE Compcon '95*, Mar. 1995, pp. 32–36.

[23] C. Lai, G. Medvinsky, and B. C. Neuman, "Endorsements, licensing, and insurance for distributed system services," in *Proc. Second ACM Conf. Computer and Communications Security*, Nov. 1994.

[24] B. C. Neuman, "Scale in distributed systems," in *Readings in Distributed Computing Systems*. IEEE Computer Society Press, 1994.

[25] CCITT, "Recommendation X.509: The directory authentication framework," Dec. 1988.

[26] B. C. Neuman, B. Tung, and J. Wray, "Public key cryptography for initial authentications in kerberos," Internet draft ietf-cat-kerberos-pk-init-00, Mar. 1995.

[27] R. Atkinson, "Security architecture for the internet protocol," Internet draft ietf-ipsec-arch-02, May 1995.

[28] J. Linn, "Generic security service application program interface," Internet RFC 1508, Sept. 1993.
[29] Open Software Foundation, "Security in a distributed computing environment," White paper OSF-O-WP11-1090-3, Jan. 1992.
[30] B. Jaspan, "GSS-API security for onc rpc," in *Proc. ISOC Symp. Network and Distributed System Security*, Feb. 1995.
[31] D. Gifford, A. Payne, L. Stewart, and W. Treese, "Payment switches for open networks," in *Proc. IEEE Compcon '95*, Mar. 1995, pp. 26–31.

B. Clifford Neuman is a scientist at the Information Sciences Institute of the University of Southern California (USC). After receiving the S.B. degree from the Massachusetts Institute of Technology in 1985 he spent a year working for Project Athena where he was one of the principal designers of the Kerberos authentication system. He received the M.S. and Ph.D. degrees from the University of Washington, where he designed the Prospero Directory Service which is widely used to locate information from Internet archive sites.

Dr. Neuman's recent work in the security area includes the development of a security infrastructure supporting authorization, accounting, and electronic payment mechanisms.