Proc. of Internet Society INET '99

Electronic Commerce with Verifiable Audit Trails

Jon M. Peha
(peha@cmu.edu, http://www.ece.cmu.edu/~peha)
Carnegie Mellon University
USA

Abstract

Telecommunications networks are becoming an increasingly important medium for commerce, contractual agreements, and other important transactions. To support such transactions, a system must be devised that provides strong privacy protection for all parties, while simultaneously producing a detailed audit trail. Any attempt to falsify this audit trail should face significant risk of detection, even if multiple parties cooperate in the falsification. This system must also allow parties to present credentials relevant to a given transaction, and only those credentials. This paper presents an architecture that succeeds in making audit trails and credentials accessible as needed, without compromising privacy.

Table of Contents

Section 1: Introduction

The Internet and other electronic telecommunications networks are becoming a center of commercial transactions. Many of these transactions take place entirely over the network, with no physical goods changing hands. Contracts, software, news articles, technical designs, music, video-recordings, subpoenas, land deeds, stocks, airline ticket confirmations, and money can be transferred across a telecommunications network without a single piece of paper changing hands. Such transactions require two critical features that are often in conflict: auditability and privacy [17]. This paper demonstrates that it is possible for a system to provide both without compromise.

In the physical world, we often balance privacy and auditability by generating papers which are inherently hard to forge. For example, it is possible to purchase items from a store using cash while maintaining complete anonymity. A cash register provides auditability by recording every purchase on paper. These papers make it possible for store owners to catch a dishonest sales clerk, and for tax collectors to catch a dishonest store owner. In addition, the customer is given a receipt for the transaction. Receipts are also on paper, making them difficult to forge. If a system cannot provide (at least) the same degree of auditability, it may not meet the demands of users to be protected from fraud[17], or the legal requirements of tax collectors [16,17]. This auditability must be achieved through a new mechanism, as bits in computer memory are easy to change. There has been considerable attention on the development of a suitable payment system for electronic commerce [2,19], but not on the complementary system to produce and verify an audit trail.

Even one-party transactions can require both privacy and auditability. For example, consider an inventor racing to patent an invention. If an auditor can determine that the invention's documentation was complete by the stated date, then the inventor's rights are protected. However, the inventor clearly does not want to reveal her invention, i.e. she wants to maintain privacy.

Some transactions also require parties to present credentials. For example, those purchasing alcohol in the US must prove they are at least 21 years old. This is easy in person, but can be difficult over a network. This limitation first became a serious problem in the telecommunications context due to the Internet pornography market. Parents want to keep their children from these sites, and customers want to maintain their privacy. Vendors must somehow verify each customer's age without gathering enough information to reveal that customer's identity. which is generally not possible on today's Internet. A similar problem occurs in other circumstances, likewhen a vendor of strong encryption tries to comply with exportcontrols [17].

This paper will present a system that allows electronic transactions to be both private and auditable. Section 2 describes the properties required for such a system. Section 3 briefly summarizes current practices. The mechanisms of the system are presented in Section 4. Section 5 briefly highlights the system features that allow it to meet the requirements in Section 2. Finally, the paper is

summarized in Section 6.

Section 2: System Requirements

This section describes our design requirements more precisely. Section 2.1 discusses auditability. Section 2.2 discusses privacy. Section 2.3 discusses the need for verifiable credentials. Section 2.4 describes dependability requirements for the system.

Section 2.1: Auditable

An auditor should be able to retrieve a set of records associated with a given entity, and determine that those records contain the truth, the whole truth, and nothing but the truth. There should be a reasonable probability that any attempt to record incorrect, incomplete, or extra information will be detected. Thus, even though many transactions will never be scrutinized, the falsification of records is deterred. This section categorizes the many ways that records can be falsified.

Entity X has a transaction with an arbitrary number of other parties. Each should then record the agreed-upon transaction T into their records. X can falsify its records as follows.

X never records transaction T.

1. All parties in the transaction agree not to record transaction T.
2. X does not record the transaction, but at least one party Y does. X's and Y's records are in conflict.
3. X does not record the transaction, but at least one party Y does. X used a false identity in the transaction T so that its records would not directly contradict those of Y.

X places an incorrect transaction record T' into its records at the time of the transaction instead of T.

4. All parties in the transaction agree to record the false transaction T'.
5. X records false information T', but some parties record T.

X invents and records a transaction that didn't occur.

6. All parties in the alleged transaction agree to support X's falsification.
7. X records the false transaction without support from all other parties.

X places incorrect information into the records after the fact. (This option is often particularly beneficial to perpetrators.)

8. X alters transaction T after it has been recorded.

9. X removes a prior transaction T from the records.
10. At the time of transaction T, X records multiple versions. X later selectively forgets all versions but one.
11. All parties agree to delay recording a transaction. They subsequently select false information to record, or decide not to record the transaction at all.
12. X retroactively records a transaction that never occurred.

Of the twelve approaches described above, three require all parties involved in a transaction (or alleged transaction) to conspire to falsify records at the time of the transaction (1, 4, and 6). No system can possibly detect such a conspiracy, whether the transaction takes place over a telecommunications network or in person. For example, if a store-owner and a customer agree to enter an incorrect price on a cash register, records will show no signs of falsification. However, in every other case listed above, there is an inconsistency that can potentially be observed. A system for electronic transactions should be as auditable as transactions in the physical world. Thus, in an auditable system, every other case should be detected, and it should be possible to identify the guilty parties. *This should be possible even if the parties in a transaction and operators of the system itself cooperate to falsify records.*

Section 2.2: Private

Unless entity Y is an authorized auditor, Y should not be able to determine

1. information about any of entity X's transactions to which Y was not a party.
2. the parties with whom X has had transactions.
3. whether X has had any transactions, and if so, how many

Moreover, it should be possible to audit entity X without examining the records of any other entity, including those with whom X had transactions. This protects the privacy of other entities. More importantly, it means audits are effective even when some entities lose their records, as might occur when a company goes out of business.

Section 2.3: Verifiable Credentials

As described in Section 1, it is sometimes necessary to provide some credentials before a transaction. This may include identity, age, nationality (e.g. to purchase encryption software), mailing address, credit rating, or tax identification number.

1. It should be possible for an entity to reveal the desired characteristics and only the desired characteristics.
2. False information should be detectable at the time of the transaction, and subsequently in an audit.

Section 2.4: System Dependability

The system as a whole should be available to handle transactions at all times, even when there are a large number of customers. To achieve this, the architecture must be distributed, with redundant components. There can be no performance bottleneck, and no single point of failure.

Section 3: Today's Systems

Designers typically address privacy or auditability. For two-party exchanges, this is unavoidable. Consider a customer with an established credit line at a software vendor. Through digital signatures, the customer and vendor can establish records that the other cannot later refute, provided that neither withheld information to guard privacy. Even then, nothing can prevent both from agreeing to falsify the record to avoid sales tax [16].

Inventory records can provide some corroborating evidence, e.g. the number of books sold on-line should equal the number of books leaving the warehouse, but this reveals nothing about price or date. (Shifting a sales date from December 31 to January 1 may greatly affect the tax burden.) Moreover, when pure information such as text, software, or music is sold on-line, it can be sold many times, so inventory is meaningless.

Another common payment paradigm is that of traditional checking, such as the system currently being devised by the Financial Services Technology Consortium [13,19]. Upon receiving a digitally signed "check," a bank will transfer funds from one account to the other. (Or if the two parties have accounts with different banks, they will arrange the transfer together.) The bank(s) maintain records of the cash transfers, but know nothing about the remaining details of the transaction.

Credit card companies often play the role of "trusted" third parties to transactions, and can keep more detailed records. Of course, customers and merchants must surrender their privacy to the credit card company, and often to each other. In some systems like First Virtual and Globe-ID [15,19], additional "trusted" third parties have complete access to transaction information, and they can maintain audit information. There is little to prevent any of these third parties from altering records in concert with their customers.

There is a need for third parties that can be trusted because they are subject to truly effective audits. Indeed, as the next section will show, multiple parties should play this role so that no one entity can compromise the privacy of others.

Section 4: Proposed Architecture

Conceptually, the system works as follows. All parties agree on the specifics of a transaction, and create a record of it. Each party receives a copy of the record signed by all parties. A party that is subject to audits then *notarizes* its copy in a manner that allows subsequent audits.

In addition to the customers engaging in auditable transactions, the system consists of *verifiers, notaries,* and *auditors.* It is

assumed that each customer can locate one or more of each of these critical entities. Verifiers are responsible for checking the identity of all parties, and verifying their credentials to interested parties. Every transaction record passes through a notary, who establishes a timestamp, and must somehow insure that nothing is altered after the transaction record is notarized. Auditors oversee customers, as well as verifiers and notaries. All of these entities make use of public key encryption [6]. It is assumed that no party has deliberately or accidentally revealed its secret key.

Verifier and notary functions are separated deliberately. Verifiers typically know the true identity of a customer. Notaries know whether an entity is notarizing transactions, and perhaps some information about those transactions. An entity that served as both a verifier and a notary (like a typical credit card company) would therefore know that a given customer is processing transactions, violating privacy objectives.

Sections 4.1 through 4.4 describe four critical processes: registering a customer with verifiers and notaries, creating transaction records, submitting records to a notary, notarizing transaction records, and conducting an audit.

Section 4.1: Registering a Customer

Each customer must register with one or more verifiers before using the system. To register, a customer informs the verifier of its public key. The customer has the option of providing additional information, which it may designate as either public or private. Public information can be used as credentials during transactions. Private information may later be accessed by authorized auditors. The verifier is responsible for checking the veracity of all optional information. The verifier then assigns this customer a unique account number, and makes the account number, public key, and any optional public information accessible to everyone. Prior public keys are also displayed, along with the times in which that public key was in use. With this approach, any one can look up the public key and public credentials of the customer with account A on verifier V, without observing any private information.

Commercial companies have been providing some of these services. When establishing an account, they investigate the veracity of the client's identity. On-line services can then certify an entity's identity and public key. They have not allowed customers to provide an arbitrary set of information and choose which to display publicly, as is proposed here. However, the International Telecommunications Union (ITU) and the International Standards Organization (ISO) are providing an enhanced X.509 standard for authentication that supports extensible certificates [11]. Certificates will be encoded in Abstract Syntax Notation (ASN.1) [10], making it easier to support certificates with an arbitrary number of fields of various types. This could facilitate the development of suitable verifiers as proposed in this paper.

For example, a registering customer provides his name and social security number as private information, and his nationality as public information. This is done in person to simplify verification. His nationality, public key, and account number are placed on the verifier's web site. He can now purchase software without showing identification, including software with export control restrictions. Auditors can determine his identity if necessary, but software vendors know only his account number. No one else,

including the vendor, knows his private key.

For each verifier account [V,A], a customer also establishes a relationship with one or more notaries. Each transaction from account [V,A] must be processed by one (and only one) of these notaries. Finally, the customer informs the auditor of all of its verifier accounts, and respective relationships with notaries. (This information may change over time as accounts are opened and closed.) The registration process is summarized in Figure 1.

There are several reasons why a customer must be allowed to register with multiple verifiers and notaries. (1) A customer can have multiple identities. Each identity may reveal different information to the public. For example, one account shows nationality, and another shows age, so a customer need not reveal more information than is necessary for a given transaction. (2) If a customer registered with only one verifier, that verifier could count the number of times others verified the identity of that customer, possibly compromising privacy. If a customer had only one notary, that notary could count the number of items it notarized. (3) With redundancy, the customer can continue to operate even when a verifier or notary is down.

Customer opens one or more accounts with verifiers.
For each account [V,A]:

1. Customer sends verifier V the customer's public key, optional public information, and optional private information.
2. Verifier V creates unique account number A, sends back to customer.
3. Verifier V checks the veracity of all optional information.
4. Verifier V puts on display: Account A, public key, optional public information.
5. Customer establishes relationship between [V,A] and one or more notaries.
6. Customer registers account [V,A] and notary relationships with auditor.

Figure 1: The registration process.

Section 4.2: Creating a Transaction Record

All parties in a transaction create a description D. For example, if this transaction is a software purchase, D might include a description of the software, the price, the date, and the time. All parties apply their respective digital signatures to D, and each party gets a copy. A transaction record T consists of D, D encrypted with each of the digital signatures, the account [V,A] of each party, and a unique transaction identification number. An auditor can later determine and verify the identity of each party by obtaining the

public key that [V,A] was using at the time and date specified in D from verifier V.

In electronic commerce transactions, it is already common to construct signed and encrypted transaction records that can also be adapted to this purpose. Credit card companies obtain complete signed records. Intermediaries like NetBill collect this information during each transaction as a means of providing atomicity [20]. Signed digital checks (such as NetCheque [13] and the Financial Services Technology Consortium system) must include some of this information, and customers wanting warranties demand the rest. Such records are not routinely collected for anonymous e-cash [3,4,12] transactions, but there are advantages to doing so: resolving cases where there are multiple attempts to redeem the same electronic token, or tracing criminal movements of cash [7].

This transaction record T proves who was involved, and that there was a time when they all agreed to D. However, it does not prove that D has escaped modification. This is the responsibility of the notary.

Section 4.3: Submitting Transaction Records to a Notary

Notarizing transaction T makes it difficult to alter the records retroactively. Each party in a transaction notarizes its version of the transaction record T. (The various versions of T differ only in the order in which the fields are stored.) This makes it possible to audit one customer without viewing the records of other customers.

It is not sufficient to simply submit T to the notary, for two reasons. First, submitting T would reveal the specifics of the transaction. Second, even if notarizing T somehow protected it from fraudulent manipulation, what would prevent the customer from "forgetting" that this transaction ever occurred. Equivalently, a customer could notarize multiple transactions and later choose which one to remember.

To protect its privacy, the customer runs the transaction T through a one-way hash function H(T). A one-way hash has the property that determining H(T) from T is trivial, but determining T from H(T) in a reasonable amount of time is computationally infeasible. H(T) can be submitted to the notary, rather than T. H(T) also typically requires less storage space than T. Where appropriate, multiple transactions can be submitted in one batch, $H(T_1,T_2,T_3, ...)$. (Thus, even micropayment systems [8,18,20] can be supported.) In practice, the customer might use a one-way hash function provided by the associated notary or the associated verifier, who would be responsible for recording the history of which hash functions were used and when.

The key to preventing customers from selectively forgetting some transactions is to require them to include verifiable information on the author's identity with every submission. Thus, the notaries with whom a given customer has registered can collectively produce every single transaction that the customer has notarized. To enable this, the customer sends the notary H(T), V, A, and H(T) encrypted with the private key that is associated with account [V,A]. One can check the veracity of the author information by applying [V,A]'s public key at the time indicated by the timestamp to the encrypted string, and comparing the results with H(T).

Section 4.4: Protecting Notarized Records

The notarization process must insure that each transaction is recorded immediately, and that the record is not altered thereafter, even if there is a conspiracy to do so.

A notary design was presented in [9], and the authors subsequently launched a company (Surety) to make this service available. Their service is useful, but does not meet the requirements described in this paper. In particular, it must be possible to reliably audit one customer without accessing the records of other customers, and it must be possible for an auditor to retrieve *all* records associated with a given customer. To do this, the notary must request identity information, and store this information in a way that it cannot be altered by the notary or observed by other customers.

Our approach is as follows. A customer submits a record to be notarized. The notary appends a timestamp and any associated data. The entire string is manipulated such that any further modifications can be detected. The notary stores results from this manipulation, along with the account [V,A] of the customer. The latter is essential because it makes it possible for the notary to identify all records from a given account during an audit. The notary sends some kind of receipt back to the customer which can be used to verify that the record has been notarized.

Conceptually, the simplest way to implement the above is to use a "trusted" notary. This is best achieved by implementing a notary on a chip, embedding public and private keys, and using "tamper-proof" packaging. Such a system would be built for honest operation by carefully selected and supervised manufacturers, as was once proposed for the controversial US Clipper encryption standard [5,14]. A system auditor can periodically query a notary to make sure that it still knows the correct time and its private key, i.e. the chip is functioning and has not been replaced. This trusted notary chip would apply its digital signature to the record being notarized and associated timestamp, thereby proving that the trusted entity saw the record in this form. Of course developing hardware components that everyone will trust is a difficult process, especially when there could be tremendous financial incentive to develop impostor chips. This was a major criticism of the Clipper proposal, and has been problematic in other cases [1].

An alternative is to make the notary trusted by forcing it to operate in view of the public, or at least the auditors. This must be done in a manner that protects the records, timestamps, and author information, but never compromises privacy. To this end, each notary maintains a database of records with the following fields. The k'th record in the database has

- the index k,
- the customer's account [V,A],
- the hashed transaction H(T),
- H(T) encrypted with [V,A]'s private key,
- timestamp information, and
- verification information V_k.

V_k is the output of a one-way hash function, where the input consists of the other fields associated with record k, and possibly additional information as well. The values of k and V_k are placed on open display, perhaps on a web site. It may also help to display V_k with the notary's digital signature to preclude impersonation of a notary. Any subsequent change of information on public display risks detection. If any of the other fields are changed, the displayed value of V_k will no longer be correct. However, no private information is revealed by displaying V_k alone.

All of the fields associated with a new record in this database are transmitted back to the customer as a receipt. Anyone with a copy of the receipt can verify that the information is correct by running the receipt through the same one-way hash used by the notary, and verifying that the result equals the V_k on display. For example, this receipt can now serve as a proof of purchase in an electronic commerce transaction.
Several precautions are possible to prevent notaries from deliberately creating incorrect timestamps. First, new records in the database are added only at the end, making the index order k the same as timestamp order. This makes it difficult to assign timestamps out of order without detection. Second, the "timestamp information" field could include unpredictable and externally verifiable information, like the current temperature or current stock market values. This makes it even harder to assign future timestamps without detection.

X sells article to Y over the Internet. X records the transaction as follows.

1. A description D of the sale indicates the article sold, the price, date, time, seller X's account [V,A], and buyer Y's account [V,A].
2. Transaction record T contains D, D encrypted with X's private key, D encrypted with Y's private key, X's account [V,A], Y's account [V,A], and transaction identification number.
3. X runs T through X's one-way hash $H_X(T)$.
4. X submits to notary N: $H_X(T)$, account [V,A], and $H_X(T)$ encrypted with the private key associate with account [V,A].
5. The notary N appends index k and timestamp to the submitted information to form receipt R_k. The receipt R_k is stored in N's database, and is transmitted back to X.
6. The notary N runs receipt R_k through its one-way hash $H_N(R_k)$ to produce V_k.
7. The notary places k and V_k on public display.

Figure 2: Recording a transaction.

Section 4.5: Conducting an Audit

To audit an individual customer, the auditor asks the customer for every transaction T. The auditor already knows the notaries, verifiers, and account numbers used by this customer. He can therefore ask all the notaries to specify which records in the database came from the customer, under any of the possible account numbers that the customer may have been using. There should be a one-to-one correspondence between the transactions provided by the customer and those provided by the notaries, proving that all transactions were notarized once and only once. The auditor then runs the transactions through the appropriate one-way hash functions and verifies that the results equal the values received from the notaries. This shows that no data has been modified since it was notarized. Each transaction record T has a timestamp that was agreed upon by the parties involved, and a second timestamp that was applied by the notary. Any differences in these timestamps should fall within established limits. This helps demonstrate that the parties did not deliberately delay the notary process so they could retroactively change the transaction record before it was notarized.

To retrieve and verify information on a given transaction from customer X.
X knows the receipt R_k from notary N and the transaction record T.

1. For each party with account [V,A] in the transaction T: retrieve the public key associated with account A from verifier V. Apply the key to D. Make sure that the results match the values in T. *(Thus, all parties agreed to D.)*
2. Run R_k through N's one-way hash $H_N(R_k)$. Make sure results match V_k, which notary N has on public display. *(Thus, R_k has not been altered.)*
3. Run T through X's one-way hash $H_X(T)$. Make sure results match the $H_X(T)$ in receipt R_k. *(Thus, T has not been altered since it was notarized.)*
4. Make sure that the difference between the timestamp in D and the timestamp in receipt R_k are within acceptable limits. *(Thus, the transaction was notarized promptly.)*
5. D is retrieved from transaction record T. D contains the desired information

Figure 3: Verifying a transaction.

To audit a customer.

1. Customer provides auditor with all transaction records.
2. For each account customer was using, and all notaries with whom customer had a relationship, auditor requests all of the customer's receipts. *(Thus, customer cannot exclude any notarized transaction.)*
3. For each receipt, the auditor applies the public key associated with account [V,A] to the encrypted $H_X(T)$. Make sure the result matches the unencrypted $H_X(T)$. *(Thus, the customer cannot disavow responsibility for any receipt.)*
4. Customer indicates which receipts correspond with which transaction records.
5. Auditor verifies each transaction, as described in Figure 3.

Figure 4: Verifying a transaction.

Occasionally, i.e. with a non-negligible probability, the audit of one customer should lead to a limited audit of others. More specifically, it may lead to a request that one of the notaries be audited, to insure that the notary did provide all of the records associated with the given customer. Otherwise, a customer and notary could conspire to "forget" some transaction records. In addition, when auditing a customer A that has had transactions with customer B, the auditor may request a limited audit of customer B. The purpose is not to advance the audit of customer A; as described in Section 2.2, it must be possible to audit one customer without accessing the records of other customers. The purpose is to deter B from failing to notarize the transaction.

Verifiers and notaries must also be audited. Auditors will regularly retrieve information from the verifiers' and notaries' web pages. They may or may not keep this information, but the notary or verifier has no way of knowing what they do and do not keep. The information kept can be checked later, to make sure that notaries and verifiers do not ever change this information. Verifiers, notaries, and auditors may also notarize their own records, the same way customers notarize transactions, thereby making record alterations detectable to auditors.
From time to time, auditors must also check that a notary's one-way hash function still yields the displayed verification information V_k: "k, that the timestamps are in increasing order with respect to index k, and that no index numbers are missing. This deters notaries from altering the data that is not observable by the public.
Verifiers and notaries can also notarize their own records using the same mechanisms as customers do, and auditors can view this. Similarly, auditors notarize their own records that indicate the verifier and notary accounts used by each customer. All of this is

subject to audit.
Once a transaction has been notarized, it is not possible to alter records of that transaction without risking detection from the auditor, even if buyer, seller, verifier, and notary conspire to do so. Thus, the best way to falsify records is to involve the auditor. To address this, there will be multiple auditors. Generally, one auditor will not know about the actions of other auditors, so it is possible that the same customer will be audited twice. At this point, if one finds fraud and the other did not, the matter would be investigated further, possibly revealing any auditor malfeasance.
Note that there might be a different set of auditors supervising individual customers, as opposed to notaries and verifiers, as the latter requires more invasive investigative powers. For example, investor groups may hire accountants to audit individual companies, while the government audits verifiers and notaries.

Section 5: Key Features

This section briefly reviews some of the crucial features of the proposed system, and how they serve to meet the design requirements in Section 2. Section 2.1 describes twelve methods that a customer might use to falsify records. A well designed system should prevent all but three of them (methods 1, 4, and 6). This system succeeds for the following reasons.

- A customer who records false information without support from other parties (as occurs in methods 3, 5, and 7) can be detected because the digital signatures of all parties involved in a transaction are captured in the transaction record. The customer cannot forge these signatures since it does not know the private keys of the other parties.
- Any alterations of a transaction record after the fact (methods 8 and 9) will cause a detectable anomaly in notary records. An auditor can detect a change in the information on public display in its random viewings. If this public verification information is not changed but a receipt is changed, then running the receipt through the notary's one-way hash function will no longer produce the displayed information. If the receipt is not changed but the transaction record is, then running the transaction record through the other one-way hash will no longer produce the values stored in the receipt.
- It is not possible for customers to "forget" transactions (as occurs in methods 9 and 10) at audit time, because notaries collect verifiable information on the identity of their customers. Thus, notaries can produce all receipts associated with a given account. Because of the registration process, the auditor knows all accounts associated with a customer, and all notaries used.
- Customers cannot safely delay notarization (as occurs in methods 11 and 12). Transaction records contain one timestamp and the notary applies another when the transaction record is notarized, so auditors can detect when these differ significantly. Thus, to delay notarization without detection, a customer would have to convince the notary to later create an outdated timestamp. This is also detectable. For a given notary, timestamps of successive transactions must be in increasing order with respect to index. This limits the extent to which a notary can falsify timestamp, especially when the time between notary transactions is small. The auditor may also detect anomalies by noting which transactions are and are not on display during random viewings, i.e. if evidence of a transaction is not on display at time t, it cannot later be added with a timestamp

less than t. Further protection is also possible if needed.

- A customer who never notarizes a given transaction record when another party X does (as occurs in method 2) risks detection when X is audited. The auditor may launch a new audit of this customer, and will see that the transaction with X is missing.

An important aspect of this system is the number of independent entities. This redundancy gives the system the needed dependability, as described in Section 2.4. Customer privacy is protected because notary and verifier functionality is separated, and because there are multiple independent verifiers and notaries acting in parallel. Thus, a customer need not reveal much information to any single entity, thereby meeting the requirements in Section 2.2. The presence of multiple verifiers makes it possible for customers to reveal credentials as desired, and nothing more, as required in Section 2.3. Because there are multiple auditors, even dishonest auditors risk detection, and it only takes one honest auditor to catch fraud.

Section 6: Summary

Many emerging network applications such as electronic commerce must meet apparently contradictory requirements. Transactions must produce a verifiable audit trail, such that attempts to falsify records can be detected with reasonable probability. Even a conspiracy to falsify records must be detectable. Moreover, it must be possible for parties involved in some transactions to check the credentials of other parties and be certain of their veracity. For example, a vendor selling software on credit may want to know the buyer's credit rating. However, auditability and the visibility of credentials when needed must be achieved without compromising the privacy of any of the parties. Other than authorized auditors, no one should be able to access any information on others except those credentials deliberately declared for a given transaction. It should even be impossible to determine whether some one else has engaged in transactions, or with whom they have engaged in transactions. As a final requirement, architectures should be distributed with sufficient redundancy to allow parallelism and dependability.

This paper has demonstrated that it is possible to meet all of these requirements. The architecture presented here divides system functions among three types of entities: notaries, verifiers, and auditors, and allows multiple actors of each type to operate independently. Notaries have the crucial responsibility of making the retroactive modification of all records detectable. They achieve this by capturing data on each transaction, and then appending timestamp information. As part of this process, notaries must capture verifiable information on who is notarizing the transactions, so that a notary can later identify every transaction notarized for a given account. One way a notary can make alterations of the records detectable without compromising privacy is to place indicia of the records in public. Variations of digital signatures are used routinely to verify identities, and one-way hash functions to protect privacy. A system with these features can support transactions that are both private and auditable.
This architecture implies the creation of many new businesses offering verification and notarization services over the Internet. Their presence will be as important for tomorrow's electronic commerce as banks and insurance companies are for today's international trade.

Government also has a role to play. For example, appropriate government standards are needed on electronic commerce record-keeping, and limited oversight of verifiers and notaries is essential to insure honest operation [17]. Government, industry, and academia must work closely together to insure that electronic commerce is supported by government, but not micro-managed by government.

References

1. R. Anderson and M. Kuhn, "Tamper Resistance - A Cautionary Note," *Proc. 2nd Usenix Workshop On Electronic Commerce,* Nov. 1996.
2. N. Asokan, P. A. Janson, M. Steiner, and M. Waidner, "The State of the Art in Electronic Payment Systems," *IEEE Computer,* Vol. 30, No. 9, Sept. 1997, pp. 28-35.
3. D. Chaum, A. Fiat, and N. Naor, "Transaction Systems To Make Big Brother Obsolete," *Communications of the ACM,* Vol. 28, No. 5, Oct. 1985, pp. 1030-44.
4. D. Chaum and S. Brands, "Minting Electronic Cash," *IEEE Spectrum,* Vol. 34, No. 2, Feb. 1997, pp. 30-4.
5. D. E. Denning and M. Smid, "Key Escrowing Today," *IEEE Communications,* Vol. 32, No. 9, Sept. 1994, pp. 58-68.
6. W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Trans. Information Theory,* Vol. 22, Nov. 1976.
7. P. S. Gemmell, "Traceable E-cash," *IEEE Spectrum,* Vol. 34, No. 2, Feb. 1997, pp. 35-7.
8. S. Glassman, M. Manasse, M. Abadi, P. Gauthier, and P. Sobalvarro, "The Millicent Protocol For Inexpensive Electronic Commerce," *Proc. Fourth Intl. World Wide Web Conference,* Dec. 1995, pp. 603-18.
9. S. Haber and W. S. Stornetta, "How To Time-Stamp a Digital Document," *Journal of Cryptology,* Vol. 3, 1991, pp. 99-111.
10. International Standards Organization (ISO), *ASN.1 and its Encoding Rules,* IS 8824/5, 1988.
11. ITU-T Recommendation X.509, ISO/IEC 9594-8:1997, *Information Technology - Open Systems Interconnection - The Directory: Authentication Framework,* 1997.
12. G. Medvinsky and B. C. Neuman, "Netcash: A Design for Practical Electronic Currency on the Internet," *Proc. ACM Conference on Computer and Communications Security,* Nov. 1993, pp. 102-6.
13. B. C. Neuman and G. Medvinsky, "Requirements for Network Payment: The NetCheque Perspective," *Proc. IEEE Compcon,* March 1995.
14. National Institute for Standards and Technology (NIST), "Escrowed Encryption Standard (EES)," Federal Information Processing Standards Publication (FIPS PUB) 185, Feb. 9, 1994.
15. P. Pays and F. de Comarmond, "An Intermediation and Payment System Technology," *Computer Networks and ISDN Systems,* Vol. 28, 1996, pp. 1197-1206.
16. J. M. Peha and R. P. Strauss, "Changing Information Technology and the Fisc," *National Tax Journal,* Vol. L, No. 3, Sept. 1997, pp. 608-21. Available at http://www.ece.cmu.edu/~peha/ecommerce.html
17. J. M. Peha, "Making the Internet Fit for Commerce: New Policies to Enforce Tax Laws, Protect Privacy, Deter Fraud, and Prevent Illegal Sales ,"to appear in *Issues in Science and Technology,* National Academy Press, Winter 1999-2000. Available

at http://www.ece.cmu.edu/~peha/ecommerce.html
18. R. L. Rivest and A. Shamir, "Payword and MicroMint: Two Simple Micropayment Schemes," *Proc. Eurocrypt,* 1996.
19. M. A. Sirbu, "Credits and Debits on the Internet," *IEEE Spectrum,* Vol. 34, No. 2, Feb. 1997, pp. 23-9.
20. M. A. Sirbu and J. D. Tygar, "NetBill: An Internet Commerce System Optimized for Network Delivered Services," *IEEE Personal Communications,* Vol. 24, Aug. 1995, pp. 34-9.