Technical Note
Programming Techniques and Data Structures

R. Rivest
Editor

Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms

David L. Chaum
University of California, Berkeley

A technique based on public key cryptography is presented that allows an electronic mail system to hide who a participant communicates with as well as the content of the communication—in spite of an unsecured underlying telecommunication system. The technique does not require a universally trusted authority. One correspondent can remain anonymous to a second, while allowing the second to respond via an untraceble return address.

The technique can also be used to form rosters of untraceable digital pseudonyms from selected applications. Applicants retain the exclusive ability to form digital signatures corresponding to their pseudonyms. Elections in which any interested party can verify that the ballots have been properly counted are possible if anonymously mailed ballots are signed with pseudonyms from a roster of registered voters. Another use allows an individual to correspond with a record-keeping organization under a unique pseudonym which appears in a roster of acceptable clients.

Key Words and Phrases: electronic mail, public key cryptosystems, digital signatures, traffic analysis, security, privacy

CR Categories: 2.12, 3.81

Introduction

Cryptology is the science of secret communication. Cryptographic techniques have been providing secrecy of message content for thousands of years [3]. Recently, some new solutions to the "key distribution problem" (the problem of providing each communicant with a secret key) have been suggested [2, 4], under the name of public key cryptography. Another cryptographic problem, "the traffic analysis problem" (the problem of keeping confidential who converses with whom, and when they converse), will become increasingly important with the growth of electronic mail. This paper presents a solution to the traffic analysis problem that is based on public key cryptography. Baran has solved the traffic analysis problem for networks [1], but requires each participant to trust a common authority. In contrast, systems based on the solution advanced here can be compromised only by subversion or conspiracy of all of a set of authorities. Ideally, each participant is an authority.

The following two sections introduce the notation and assumptions. Then the basic concepts are introduced for some special cases involving a series of one or more authorities. The final section covers general purpose mail networks.

Notation

Someone becomes a user of a public key cryptosystem (like that of Rivest, Shamir, and Adleman [5]) by creating a pair of keys K and K^{-1} from a suitable randomly generated seed. The public key K is made known to the other users or anyone else who cares to know it; the private key K^{-1} is never divulged. The encryption of X with key K will be denoted $K(X)$, and is just the image of X under the mapping implemented by the cryptographic algorithm using key K. The increased utility of these algorithms over conventional algorithms results because the two keys are inverses of each other, in the sense that

$$K^{-1}(K(X)) = K(K^{-1}(X)) = X.$$

A message X is *sealed* with a public key K so that only the holder of the private key K^{-1} can discover its content. If X is simply encrypted with K, then anyone could verify a guess that $Y = X$ by checking whether $K(Y) = K(X)$. This threat can be eliminated by attaching a large string of random bits R to X before encrypting. The sealing of X with K is then denoted $K(R, X)$. A user *signs* some material X by prepending a large constant C (all zeros, for example) and then encrypting with its private key, denoted $K^{-1}(C, X) = Y$. Anyone can verify that Y has been signed by the holder of K^{-1} and determine the signed matter X, by forming $K(Y) = C, X$, and checking for C.

Assumptions

The approach taken here is based on two important assumptions:

This work was partially supported by the National Science Foundation under Grant MCS 75-23739 and by the Air Force Office of Scientific Research under Contract F49620-79-CO173.

Author's present address: Computer Science Division, Electrical Engineering and Computer Sciences Department, University of California, Berkeley, California 94720. (415) 642-1024.

(1) No one can determine anything about the correspondences between a set of sealed items and the corresponding set of unsealed items, or create forgeries without the appropriate random string or private key.

(2) Anyone may learn the origin, destination(s), and representation of all messages in the underlying telecommunication system and anyone may inject, remove, or modify messages.

Mail System

The users of the cryptosystem will include not only the correspondents but a computer called a *mix* that will process each item of mail before it is delivered. A participant prepares a message M for delivery to a participant at address A by sealing it with the addressee's public key K_a, appending the address A, and then sealing the result with the mix's public key K_1. The left-hand side of the following expression denotes this item which is input to the mix:

$$K_1(R_1, K_a(R_0, M), A) \rightarrow K_a(R_0, M), A.$$

The $\rightarrow$ denotes the transformation of the input by the mix into the output shown on the right-hand side. The mix decrypts its input with its private key, throws away the random string R_1, and outputs the remainder. One might imagine a mechanism that forwards the sealed messages $K_a(R_0, M)$ of the output to the addressees who then decrypt them with their own private keys.

The purpose of a mix is to hide the correspondences between the items in its input and those in its output. The order of arrival is hidden by outputting the uniformly sized items in lexicographically ordered batches. By assumption (1) above, there need be no concern about a cryptoanalytic attack yielding the correspondence between the sealed items of a mix's input and its unsealed output—if items are not repeated. However, if just one item is repeated in the input and is allowed to be repeated in the output, then the correspondence is revealed for that item.

Thus, an important function of a mix is to ensure that no item is processed more than once. This function can be readily achieved by a mix for a particular batch by removing redundant copies before outputting the batch. If a single mix is used for multiple batches, then one way that repeats aross batches can be detected is for the mix to maintain a record of items used in previous batches. (Records can be discarded once a mix changes its public key by, for example, announcing the new key in a statement signed with its old private key.) A mix need not retain previous batches if part of each random string R_1 contains something—such as a time-stamp—that is only valid for a particular batch.

If a participant gets signed receipts for messages it submits to a mix, then the participant can provide substantial evidence that the mix failed to output an item properly. Only a wronged participant can supply the receipt $Y\,(=K_1^{-1}(C, K_1(R_1, K_a(R_0, M), A)))$, the missing output $X\,(=K_a(R_0, M), A)$, and the retained string R_1, such that $K_1(Y) = C, K_1(R_1, X)$. Because a mix will sign each output batch as a whole, the absence of an item X from a batch can be substantiated by a copy of the signed batch.

The use of a *cascade*, or series of mixes, offers the advantage that any single constituent mix is able to provide the secrecy of the correspondence between the inputs and the outputs of the entire cascade. Incrimination of a particular mix of a cascade that failed to properly process an item is accomplished as with a single mix, but only requires a receipt from the first mix of the cascade, since a mix can use the signed output of its predecessor to show the absence of an item from its own input. An item is prepared for a cascade of n mixes the same as for a single mix. It is then successively sealed for each succeeding mix:

$$K_n(R_n, K_{n-1}(R_{n-1}, \ldots, K_2(R_2, K_1(R_1, K_a(R_0, M), A)) \cdots)) \rightarrow.$$

The first mix yields a lexicographically ordered batch of items, each of the form

$$K_{n-1}(R_{n-1}, \ldots, K_2(R_2, K_1(R_1, K_a(R_0, M), A)) \cdots) \rightarrow.$$

The items in the final output batch of a cascade are of the form $K_a(R_0, M), A$, the same as those of a single mix.

Return Addresses

The techniques just described allow participant x to send anonymous messages to participant y. What is needed now is a way for y to respond to x while still keeping the identity of x secret from y. A solution is for x to form an untraceable return address $K_1(R_1, A_x), K_x$, where A_x is its own real address, K_x is a public key chosen for the occasion, and R_1 is a key that will also act as a random string for purposes of sealing. Then, x can send this return address to y as part of a message sent by the techniques already described. (In general, two participants can exchange return addresses through a chain of other participants, where at least one member of each adjacent pair knows the identity of the other member of the pair.) The following indicates how y uses this untraceable return address to form a response to x, via a new kind of mix:

$$K_1(R_1, A_x), K_x(R_0, M) \rightarrow A_x, R_1(K_x(R_0, M)).$$

This mix uses the string of bits R_1 that it finds after decrypting the address part $K_1(R_1, A_x)$ as a key to re-encrypt the message part $K_x(R_0, M)$. Only the addressee x can decrypt the resulting output because x created both

R_1 and K_x. The mix must not allow address parts to be repeated—for the same reason that items of regular mail must not be repeated. This means that x must supply y with a return address for each item of mail x wishes to receive. Also notice that conventional as opposed to public key cryptography could be used for both encryptions of M.

With a cascade of mixes, the message part is prepared the same as for a single mix, and the address part is as shown in the following input:

$$K_1(R_1, K_2(R_2, \ldots, K_{n-1}, (R_{n-1}, K_n(R_n, A_x))\cdots)),$$
$$K_x(R_0, M) \rightarrow.$$

The result of the first mix is

$$K_2(R_2, \ldots, K_{n-1}(R_{n-1}, K_n(R_n, A_x))\cdots),$$
$$R_1(K_x(R_0, M)) \rightarrow,$$

and the final result of the remaining $n - 1$ mixes is

$$A_x, R_n(R_{n-1} \cdots R_2(R_1(K_x(R_0, M))) \cdots).$$

Untraceable return addresses allow the possibility of *certified* mail: They can provide the sender of an anonymous letter with a receipt attesting to the fact that the letter appeared intact in the final output batch. The address A that is incorporated in a certified letter is expanded to include not only the usual address of the recipient, but also an untraceable return address for the sender. When this return address appears in the output batch of the final mix, it is used to mail the sender a signed receipt which includes the message as well as the address to which it was delivered. The receipt might be signed by each mix.

Digital Pseudonyms

A digital *pseudonym* is a public key used to verify signatures made by the anonymous holder of the corresponding private key. A *roster*, or list of pseudonyms, is created by an authority that decides which applications for pseudonyms to accept, but is unable to trace the pseudonyms in the completed roster. The applications may be sent to the authority anonymously, by untraceable mail, for example, or they may be provided in some other way.

Each application received by the authority contains all the information required for the acceptance decision and a special unaddressed digital letter (whose message is the public key K, the applicant's proposed pseudonym). In the case of a single mix, these letters are of the form $K_1(R_1, K)$. For a cascade of n mixes, they are of the form $K_n(R_n, \ldots, K_2(R_2, K_1(R_1, K)) \cdots)$. The authority will form an input batch containing only those unaddressed letters from the applications it accepts. This input batch will be supplied to a special cascade whose final output batch will be publically available. Since each entry in the final output batch of the cascade is a public key K from an accepted applicant, the signed output of the final mix is a roster of digital pseudonyms.

Notification of applicants can be accomplished by also forming a roster for unaccepted applications and then using the technique of certified mail to return a single batch of receipts to both sets of applicants. Of course, repeats must not be allowed within or across batches.

If only registered voters are accepted for a particular roster, then it can be used to carry out an election. For a single mix, each voter submits a ballot of the form $K_1(R_1, K, K^{-1}(C, V))$, where K is the voter's pseudonym and V is the actual vote. For a cascade of mixes, ballots are of the form $K_n(R_n, \ldots, K_2(R_2, K_1(R_1, K, K^{-1}(C, V))) \cdots)$. The ballots must be processed as a single batch, as were the letters used to form rosters. Items in the final lexicographically ordered output batch are of the form $K, K^{-1}(C, V)$. Since the roster of registered voters is also ordered on K, it is easy for anyone to count the votes by making a single pass through both batches at once. Each ballot is counted only after checking that the pseudonym K which forms its prefix, is also contained in the roster and that the pseudonym properly decrypts the signed vote V.

An individual might be known to an organization only by a pseudonym that appears in a roster of acceptable clients. Clients can correspond with the organization via untraceable mail and the organization can correspond with the clients using untraceable return addresses. If applicants identify themselves in their applications, or if they sign applications with pseudonyms that appear in a roster issued by an authority that requires identification, then the organization is assured that the same client cannot come to it under different pseudonyms. Under special circumstances, such as default of payment, a particular pseudonym could be shown to correspond to a particular application (without revealing any other correspondences) if each mix in turn supplied the appropriate R_i.

General Purpose Mail Systems

One way to construct a general purpose, untraceable mail system is to require that every message pass through a cascade. Of course, mixes can operate continuously or periodically, and long messages will be encrypted first and then split into multiple items. In order to hide the number of messages sent, each participant supplies the same number of messages to each batch (some of which might be randomly addressed dummies). In order to hide the number of messages received, each participant privately searches the entire output for messages directed to it.

Such a system may prove too costly for some participants. One way to reduce the cost is to allow mail to be addressed to subsets of participants, such as a local net. Participants that take advantage of such arrangements need search only the mail addressed to a particular subset. Another way to economize is for a participant to send for each batch only the number of dummy messages suggested by a random value (chosen from some suitable distribution), as opposed to always sending the maximal number of messages. This can substantially reduce message traffic and consequently, the size of output batches. While these techniques may open the door to some kinds of statistical attack, the system size that necessitated them may reduce the effectiveness of such attacks.

In a large, general purpose mail system with many mixes, it may be impractical for every message to pass through every mix. In such a case, a sequence of mixes will be selected for each message, perhaps on the basis of network topology or trust. Notice that if a participant can choose mixes it trusts with its traffic volume data as early members of its sequences, then these mixes can discard dummies they receive from the participant and deliver small, fixed-sized batches (padded with dummies) directly to the participant.

A new kind of mix will be presented here that allows a sequence of mixes to be selected for each message. It also (a) hides the number and identity of the mixes a message must pass through, (b) allows incrimination of a mix that does not properly forward items, and (c) makes no distinction between regular mail and mail sent by untraceable return address. It is based on the idea that every item of mail is composed of the same number of fixed-sized blocks.

The operations performed by this new kind of mix are always the same. First it removes the first block and adds a random block J of junk to the end, to maintain the item's length of l blocks. Then, using its private key, the mix decrypts the block removed during the first step. This yields a key R, which the mix uses to encrypt each of the l blocks of the item (using either public key or conventional cryptography). It also yields the address A (either of a recipient or of another mix) to which the item will be forwarded.

The left-hand side of the following shows how an item is prepared to pass through a single mix:

$$\begin{aligned} A_1: &[K_{A_1}(R_{A_1}, A)], [R_{A_1}^{-1}(M_1)], [R_{A_1}^{-1}(M_2)], \ldots, \\ &[R_{A_1}^{-1}(M_{l-1})] \rightarrow A: [M_1], \ldots, [M_{l-1}], [R_{A_1}(J_{A_1})], \end{aligned}$$

where square brackets show the extent of each block, and the sealed message $K_a(R_0, M)$ is divided into pieces M_i, such that $K_a(R_0, M) = M_1, M_2, \ldots, M_{l-n}$. The A_1: indicates that the left-hand side is delivered to mix A_1, while the A: means that the right-hand side is delivered to address A. Items with the same first block should be regarded as repeats.

A message prepared to be passed through mixes A_1 through A_n has the form

$$\begin{aligned} A_1: &[K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots, \\ &[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)], \\ &[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots], \ldots, \\ &[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)] \rightarrow. \end{aligned}$$

The result leaving A_1 is

$$\begin{aligned} A_2: &[K_{A_2}(R_{A_2}, A_3)], [R_{A_2}^{-1}(K_{A_3}(R_{A_3}, A_4))], \ldots, \\ &[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A)) \cdots)], \\ &[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_1) \cdots)], \ldots, \\ &[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_n}^{-1}(M_{l-n}) \cdots)], [R_{A_1}(J_{A_1})] \rightarrow, \end{aligned}$$

and the final result leaving A_n is

$$\begin{aligned} A: &[M_1], [M_2], \ldots, [M_{l-n}], \\ &[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})]. \end{aligned}$$

An intermediate mix always knows which mix it received its input from—by assumption (2)—but if a mix broadcasts copies of its fixed-sized output batches, then only individual recipient mixes need be able to recognize their own input in a broadcast batch.

The untraceable return address $\mathbf{x}$ sends to $\mathbf{y}$ contains the key $K_\mathbf{x}$ that $\mathbf{y}$ uses to encrypt the message part. It also includes, in the case of a single mix, what $\mathbf{y}$ will use as the first block of the item it submits to the mix:

$$\begin{aligned} &A_1: [K_{A_1}(R_{A_1}, A_\mathbf{x})], [M_1], \ldots, [M_{l-1}] \rightarrow \\ &A_\mathbf{x}: [R_{A_1}(M_1)], \ldots, [R_{A_1}(M_{l-1})], [R_{A_1}(J_{A_1})], \end{aligned}$$

where $K_\mathbf{x}(R_0, M) = M_1, M_2, \ldots, M_{l-n}$. Only $\mathbf{x}$ can decrypt the item it receives since it created R_{A_1} and $K_\mathbf{x}$. When a message is to pass through n mixes, the untraceable return address contains the first n blocks:

$$\begin{aligned} A_1: &[K_{A_1}(R_{A_1}, A_2)], [R_{A_1}^{-1}(K_{A_2}(R_{A_2}, A_3))], \ldots, \\ &[R_{A_1}^{-1}(R_{A_2}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A_\mathbf{x})) \cdots)], \\ &[M_1], [M_2], \ldots, [M_{l-n}] \rightarrow. \end{aligned}$$

After being operated on by mix A_1 it will have the form

$$\begin{aligned} A_2: &[K_{A_2}(R_{A_2}, A_3)], \ldots, \\ &[R_{A_2}^{-1}(R_{A_3}^{-1} \cdots R_{A_{n-1}}^{-1}(K_{A_n}(R_{A_n}, A_\mathbf{x})) \cdots)], [R_{A_1}(M_1)], \\ &[R_{A_1}(M_2)], \ldots, [R_{A_1}(M_{l-n})], [R_{A_1}(J_{A_1})] \rightarrow, \end{aligned}$$

and the final result leaving A_n is

$$\begin{aligned} A_\mathbf{x}: &[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(M_1) \cdots)], \ldots, \\ &[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(M_{l-n}) \cdots)], \\ &[R_{A_n}(R_{A_{n-1}} \cdots R_{A_1}(J_{A_1}) \cdots)], \ldots, [R_{A_n}(J_{A_n})]. \end{aligned}$$

Summary and Conclusion

A solution to the traffic analysis problem has been presented that allows any single intermediary to provide security for those messages passing through it. In addition, the solution allows messages to be sent or received anonymously. Through the notion of a roster of pseudonyms, it also provides some new and interesting kinds of limited anonymity.

Acknowledgments. I owe a great deal to R. Fabry's outstanding and multifaceted support. Special thanks are due C. Sèquin, who has read my work with great care and provided many stimulating discussions. I would also like to thank D. Gusfield, B. Mont-Reynaud, A. Moose, and S. Wecker for their comments and encouragement. The referees have been very helpful.

Received 2/79; accepted 4/80; revised 10/80

References

1. Baran, P. On distributed communications: IX security secrecy and tamper-free considerations. Memo RM-3765-PR, Rand Corp., Santa Monica, CA, Aug. 1964.
2. Diffie,W. and Hellman, M.E. New directions in cryptography. *IEEE Trans. Information Theory IT-22*, 6 (Nov. 1976), 644–654.
3. Kahn, D. *The Code Breakers, The Story of Secret Writing.* Macmillan, New York, 1967.
4. Merkle, R.C. Secure communications over insecure channels. *Comm. ACM 21*, 4 (Apr. 1978), 294–299.
5. Rivest, R.L., Shamir, A., and Adleman, L. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM 21*, 2 (Feb. 1977), 120–126.

Corrigendum. Technical Note, Graphics and Image Processing

M.L.V. Pitteway and D.J. Watkinson, "Bresenham's Algorithm with Grey Scale," *Comm. ACM 23*, 11 (Nov. 1980), 625–626.

The figure on p. 626 has been printed erroneously. The correct figure should have the black portions on top and be reversed as per the description in the figure caption.

Technical Note
Operations and
Management

H. Morgan
Editor

On Uniformly Inserting One Data Structure into Another

Arnold L. Rosenberg
IBM Thomas J. Watson Research Center

Two recent papers ([3] and [1]) define the operation of *uniform insertion* of one data structure in another, as a step toward a structured methodology for defining data structures. This note repairs a flaw in the definition of this operation that occurs in both of the cited papers.

Key Words and Phrases: data structures, uniform insertion, uniform substitution

CR Category: 4.34

Shneiderman and Scheuermann [3] have defined an operation *uniform insertion* on a pair of data structures whereby an instance of one of the structures is appended from each data node of the other. Hollander [1] has noted a potential inconsistency in the Shneiderman–Scheuermann definition of uniform insertion and has proposed an addendum to the definition which precludes the inconsistency. However, all three authors seem to have missed a fundamental flaw in the original definition of uniform insertion, a flaw which persists in Hollander's modified definition. Before exposing and repairing the flawed definition, the notion of a *structured data structure* from [1, 3] is paraphrased.

Definition 1. A *structured data structure* (*sds*, for short) is a system

$\Sigma = (e, D, L, \Gamma)$

which specifies a connected edge-labelled directed graph in the following way:

(a) $\{e\} \cup D$ is the set of nodes of the graph;
(b) L is the set of edge labels of the graph;
(c) $\Gamma : (\{e\} \cup D) \times L \rightarrow D$ is the (not necessarily total) edge-specification function. Note in particular that the *entry node e* has indegree 0.

Author's present address: Arnold L. Rosenberg, Mathematical Sciences Department, IBM Thomas J. Watson Research Center, Yorktown Heights, New York 10598.

ACM wishes to extend an apology to the author for the extraordinarily long delay in the publication of this note, which was due to no fault of his own.