SECURE INFORMATION STORAGE AND RETRIEVAL USING NEW RESULTS IN CRYPTOGRAPHY*

by

K. Culik II[1]

and

H.A. Maurer[2]

Research Report CS-78-40

Department of Computer Science

University of Waterloo
Waterloo, Ontario, Canada

Revised January 1979

* Research in this paper was supported by the National Sciences and Engineering Research Council of Canada, Grant No. A 7403, and by the Austrian Federal Ministry of Science and Research.

[1] Department of Computer Science
University of Waterloo
Waterloo, Ontario, Canada
N2L 3G1

[2] Institute of Information Processing
Technical University of Graz
Steyrerg. 17
Graz, Austria

ABSTRACT

Consider an information system in which each user is authorized to access only certain information, different users being permitted to share none, some or all such information. Encouraged by recent advances in cryptography, we propose an approach to designing information systems which prevent unauthorized access, store information (including shared information) only once in securely encrypted form and provide for each user a single (secret) uniform retrieval (= decryption) algorithm.

Keywords

public key cryptosystems, privacy, security, information systems, cryptography

1. Introduction

In this paper we argue that by generalizing recently developed encryption methods information systems can be designed to be secure and simple. By "secure" we mean that information can only be retrieved by authorized users. By "simple" we mean that each user is able to retrieve information based on a single (parameter-free) retrieval algorithm, independent of the fact that information may reside in the information system encrypted in many different ways for authorization reasons.

We start by precisely specifying the type of information system we want to deal with.

Definition

Let S be an information system with users $\mathcal{U} = \{U_1, U_2, \ldots, U_k\}$ and a collection $A = \{A_1, A_2, \ldots, A_m\}$ of subsets of $\mathcal{U}$ called authorization classes. Information system S is called secure if a user can deposit information for any of the authorization classes and if a user U in $\mathcal{U}$ has access to some information M iff M was designated for some authorization class A which contains U. (Observe that a user may belong to many authorization classes).

Secure information systems can be obtained by encrypting information so that the decrypting algorithm required is only known by users of the appropriate authorization class. The design of such secure information systems using encryption methods rests on the fact that for each authorization class A a pair of algorithms (E_A,D_A) can be constructed such that for each message M we have $D_A(E_A(M)) = M$ and that the knowledge of the (encryption) algorithm E_A does not give away an algorithm for the decryption D_A. (See [1,4] and Section 2.) Providing each user of an authorization class A with the algorithm D_A (to be kept secret) and making all algorithms E_A public, information M can be made available to some authorization class A by depositing $E_A(M)$. Such information can only be retrieved by users of the authorization class A as they are the only ones to have D_A available.

The method explained suffers from the fact that any user belonging to a number of authorization classes has to know (and keep secret) the decryption algorithms (keys) of all those classes. This can be overcome by a realization of a secure information system suggested by Rivest [5]. We will show that if suitable extensions of the methods in [1 and 4] can be obtained even simpler realization is possible. Under these assumptions we will obtain what we call a secure information system that is simple, where "simple" refers to the fact that each user needs only a single fixed (secret) decryption algorithm, all other information being public.

In section 2 we briefly review the encryption techniques proposed in [1]. In our central section 3 we propose a model of a secure and simple information system.

In section 4 we present a partial realization of secure and simple information systems by extending results in [4]. Unfortunately, the approach does not yeild a full implementation. We pose the development of such a full realization as an important open problem. For further information on encrypting techniques, we refer the reader to the excellent papers [1] and [4] and the popular survey [2].

We would like to mention that all that follows could also be formulated in terms of a communication network with users $\mathcal{U} = \{U_1, U_2, \ldots, U_k\}$ and authorization classes $A = \{A_1, A_2, \ldots, A_m\}$. We believe that the aspect of having a single decryption algorithm to keep secret is particularly attractive in situations with many and heavily overlapping authorization classes, a situation more likely to occur in large information systems. We would further like to state that the conceivable criticism of our proposal on the basis of the cost of encryption and decryption does not seem to be valid to us. If methods akin to what we propose do come into widespread use all encrypting and decrypting will certainly not be done by some general-purpose computer but some extra piece of hardware, e.g. incorporated into an I/O channel. This would make additional costs negligible.

2. A recently developed cryptographic method

In this section we briefly summarize the most important points of the pioneering papers [1] and [4], using the terminology of information systems. For further details, the references quoted should be consulted.

For convenience, "information" and "message" will always be assumed to be (encoded as) a natural number in what follows. Further, whenever we talk about functions, we will mean one-to-one functions defined on and yielding natural numbers.

We now define the term trap-door function somewhat analogously to the way it is used in the pioneering paper [1].

Definition

A total (one-to-one) function f is called a trap-door function if

(i) there exists an algorithm for easily computing f and

(ii) knowledge of an algorithm for easily computing f does not provide a simple method for obtaining an algorithm for computing f^{-1}.

Observe that condition (ii) does not imply that f^{-1} is hard to compute. It may be hard to compute; or it may be easily computable by some algorithm B: what (ii) asserts is that such an algorithm B cannot be obtained easily from an algorithm computing f.

The term "trap-door" seems to be justified in the sense that a trap-door f can easily be passed in one direction (f can easily be computed) but a passage in the other direction (computing f^{-1}) is very hard to find. Indeed, such a passage back may be easy (depending on the trap-door) if the right "trick" (the algorithm B for computing f^{-1}) is known, in which case discovering the right "trick" (the algorithm B for computing f^{-1}) should be hard, or else passage back may be outright hard (no algorithm for easily computing f^{-1} exists).

Diffie and Hellman [1] use the notion of what they call a one-way trap-door function to obtain new cryptographical techniques. We present their ideas by first introducing what we would like to call a Diffie-Hellman pair, D H pair, for short.

Definition:

Let f be a trap-door function. If f^{-1} is easily computable, then (f,f^{-1}) is called a D H pair. If f^{-1} is also a total function, then (f,f^{-1}) is called a total D H pair.

Consider now an information system with users $\mathcal{U} = \{U_1,U_2,\ldots,U_k\}$ and a collection $\mathcal{A}$ of subsets of $\mathcal{U}$, $\mathcal{A} = \{A_1,A_2,\ldots,A_m\}$ ($A_i \subseteq \mathcal{U}$ for all i). Each $A \in \mathcal{A}$ represents an authorization class, i.e. a group of users authorized to have access to certain types of information.

By using for each authorization class A a D H pair (E_A,D_A), where E_A is made public, but D_A only known to members of of A , information M can be deposited in the system in such a way that it can be read ("decrypted") only by authorized users, i.e. by members of A. M is simply stored ("encrypted") as $E_A(M)$ and can be read by any user of A, but by nobody else, using the secret decryption algorithm D_A, $D_A(E_A(M)) = M$.

If each user is assigned a total D H pair (E_U, D_U) - again with E_U public but D_U only known to the user U - then user U can deposit even "signed" information, i.e. information of which U can conclusively be proven to be the originator.

Suppose U wants to store such a "signed" information M for members of authorization class A. Then U deposits M as $E_A(D_U(M))$. Applying D_A, every member of A (and nobody else) can obtain $D_U(M) = D_A(E_A(D_U(M)))$. Using the publicly known E_U, $M = E_U(D_U(M))$ can be obtained. Note that the pair $(D_U(M),M)$ establishes that M has been originated by U: nobody but U can obtain $D_U(M)$ from M.

3. A model for a secure and simple information system

In the previous section we have mentioned that a secure information system can be obtained by choosing one distinct DH pair (E_A, D_A) for each authorization class A. This has the disadvantage that a user who is a member of many authorization classes must use and keep secret many decryption algorithms (keys). This difficulty can be overcome as suggested by Rivest [5] as follows: Each user chooses also a DH pair (E_U, D_U). For every pair (A,U), $U \in A$, $E_U(\text{"}D_A\text{"})$ - the encrypted form under E_U of the decryption key D_A is published. A message M for authorization class A is stored in the encrypted form $E_A(M)$. To decrypt it, the user U first uses D_U to calculate $D_U(E_U(\text{"}D_A\text{"})$ to get D_A. Then he computes $D_A(E_A(M)) = M$.

In the above realization each user needs to keep secret one decryption key only, but cannot use it directly as defining one fixed decryption algorithm. So such realization is not simple in the sense defined below.

Definition

An information system S with users $\mathcal{U} = \{U_1, U_2, \ldots, U_k\}$ and authorization classes $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ is called simple if a user U belonging to a number of authorization $A_{i_1}, A_{i_2}, \ldots, A_{i_{k_U}}$ can obtain information designated for all these authorization classes with a single (parameter-free) decryption algorithm.

As a basis for a realization of a secure and simple information system we now develop the notion of a safe system of encryption triples.

Notation

Let $f : X \to Y$, $g : Y \to Z$ be functions. The composition of f and g is denoted by $f \circ g$ and defined as $f \circ g(x) = g(f(x))$ for all $x \in X$.

Definition

Let E, R, D be bijections on X . The triple (E, R, D) is called an encryption triple, if

(i) $E \circ R \circ D$ is an identity function, i.e.
$D(R(E(M))) = M$ for all $M \in X$.

(ii) There exist algorithms for easily computing E, R and D.

(iii) The knowledge of simple algorithms for E and R does not provide a simple method for obtaining an algorithm for computing either D or $R \circ D$.

Lemma If (E, R, D) is an encryption triple, then $(E \circ R, D)$ and $(E, R \circ D)$ are total DH pairs.

Proof Obvious.

We now come to the central notion of this paper.

Definition

Let S be an information system with users $\mathcal{U} = \{U_1,U_2,\ldots,U_k\}$ and authorization classes $\mathcal{A} = \{A_1,A_2,\ldots,A_m\}$. Let F be a set of functions containing for each authorization class A and each user $U \in A$ three functions which will be denoted by E_A, $R_{A,U}$ and D_U. F is called a safe system of encryption triples for S if

(i) $(E_A,R_{A,U},D_U)$ is an encryption triple for each $A \in \mathcal{A}$ and $U \in A$ and

(ii) the knowledge of all algorithms E_A and $R_{A,U}$ for all $A \in \mathcal{A}$ and $U \in A$ and of one specific $D_{U'}$, provides an algorithm for easily retrieving M from $E_A(M)$ or $R_{A,U}(E_A(M))$ if and only if $U' \in A$.

The intuitive idea behind above definition is this. Information M is stored for a particular authorization class A as $E_A(M)$, i.e. in an encryption specific to A. Retrieval of M by a member U of the authorization class A is done in a two-step process, the first a non-secret recryption algorithm $R_{A,U}$ (possibly carried out by the facility providing information storage and transfer), the second a secret decryption algorithm D_U known only to U. It is this idea of "splitting" the decryption process into two processes, one of them public, which reduces the necessary amount of secrecy as required in a simple information system. Condition (i) of the above definition will assure that the information remains safe throughout recryption while condition (ii) safeguards against the possibility that while each encryption triple on its own could conceivably be "safe", the combination of such triples could conceivably offer a way of retrieving information without authorization.

Theorem

Let S be an information system with users $\mathcal{U} = \{U_1,U_2,\dots,U_k\}$ and authorization classes $\mathcal{A} = \{A_1,A_2,\dots,A_m\}$. If we can find a safe system F of encryption triples for S, then S is a safe and simple information system.

Proof

To deposit some message M for an authorization class A, M is stored as $E_A(M)$ in the information system. A user $U \in A$ can retrieve M by first applying the publicly known recrypting algorithm $R_{A,U}$ yielding $R_{A,U}(E_A(M))$ and then applying his secret decrypting algorithm D_U yielding $D_U(R_{A,U}(E_A(M))) = M$. Note that a user U belonging to arbitrarily many authorization classes requires only a single secret decryption algorithm, i.e. S is simple.

To understand that information is secure despite public knowledge of the recryption algorithms $R_{A,U}$ let us analyze the significance of an encryption triple $(E_A, R_{A,U}, D_U)$ more closely. Since $E_A, R_{A,U}$ and D_U are bijections, each of them has an inverse.

Consider the defining equation (1):

$$(1) \quad D_U(R_{A,U}(E_A(M))) = M.$$

Applying D_U^{-1} to both sides we obtain:

$$(2) \quad R_{A,U}(E_A(M)) = D_U^{-1}(M), \quad \text{i.e.} \quad D_U^{-1} = E_A \circ R_{A,U} \quad .$$

Putting $M = E_A^{-1}(Y)$ we have

$$(3) \quad R_{A,U}(Y) = D_U^{-1}(E_A^{-1}(Y)), \text{ i.e. } \quad R_{A,U} = E_A^{-1} \circ D_U^{-1} \; .$$

Thus, $R_{A,U}$ is the composition of a function E_A^{-1} which allows the decryption of messages encrypted by E_A and of a function D_U^{-1} which encrypts messages decryptable by U's secret algorithm D_U. As a matter of fact, D_U^{-1} is easily computable by the public because of (2). However, since $(E_A, R_{A,U}, D_U)$ is an encryption triple this does not give a clue how to compute functions D_U or $R_{A,U} \circ D_U = E_A^{-1}$. Hence the original information cannot be retrieved based on E_A and $R_{A,U}$. Because of condition (ii) other encrypting triples do not help to unauthorizedly retrieve information,either,establishing the Theorem. □

Observe that in an information system as described above information can be deposited for an authorization class A by someone pretending to be a user U. However,similarly to the method of "signing" information M as described in Section 2, information can also be signed in our proposed system.

To store some information M for some authorization class A signed by U, M is deposited by U as $E_A(D_U(M))$. Note that U is the only one knowing D_U and hence able to process information that way. A user U' belonging to authorization class A applies $R_{A,U'}$ followed by $D_{U'}$ to obtain $D_U(M)$. U' now applies E_A followed by $R_{A,U}$ obtaining $R_{A,U}(E_A(D_U(M))) = D_U^{-1}(D_U(M)) = M$ due to equation (2). The pair $(D_U(M),M)$ obtained this way is conclusive proof for U' that U is the originator of M.

Summarizing, we have shown in this section that the use of safe systems of encryption triples allows the design of secure and simple information systems. In the next section we extend the approach of [4] in an attempt to obtain such safe systems of encryption triples. Although we establish the existence of encryption triples we are only partially successful since combinations of the triples without added restrictions do not yeild safe systems.

4. An implementation proposal

Consider an in information system S with users $\mathcal{U} = \{U_1, U_2, \ldots, U_k\}$ and authorization classes $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$. Based on results in [4] we establish the existence of encryption triples. However, the combination of such triples does not lead to a safe system; and modifications assuring safety create other drawbacks. The design of safe systems of encryption triples as described in Section 3 remains an important open problem.

As is stated in [4], it is impossible to prove that encryption methods of the type considered here are unbreakable. Hence it is not feasible to mathematically prove that our encryption triples are safe. The ultimate safety of such a triple can only be established by the futility of repeated attempts to break it.

Choose some large primes p,q such that lcf(p-1,q-1) - lcf standing for "largest common factor" - is small and both p-1 and q-1 have large prime factors. Let $n = pq$ and $\varphi(n) = (p-1)(q-1)$ where $\varphi(n)$ is the Euler totient function (see[4]). Choose some large primes p' and q' such that $\text{lcf}(p', \varphi(n)) = 1$ and $\text{lcf}(q', \varphi(n)) = 1$. For each $A \in \mathcal{A}$ choose some number d_A which is a large multiple of p', $d_A = p' \cdot g_A$, such that $\text{lcf}(d_A, \varphi(n)) = 1$, and such that the smallest nonnegative integer e_A satisfying the congruence $d_A \cdot e_A \equiv 1 \pmod{\varphi(n)}$ is at least as large as $\ell g(n)$. This can be done by [4]).

For every user $U \in A$ choose some number e_U which is a large multiple of q', $e_U = q' \cdot h_U$, such that $\text{lcf}(e_U, \varphi(n)) = 1$. Choose a solution d_U of the congruence $e_U \cdot d_U \equiv 1 \pmod{\varphi(n)}$ which is relative prime to $\varphi(n)$, i.e. $\mathrm{lcf}(d_U,\varphi(n)) = 1$. Observe that such a choice is possible: the congruence $e_U y \equiv 1 \pmod{\varphi(n)}$ is known by elementary number theory to have infinitely many solutions since $\mathrm{lcf}(e_U,\varphi(n)) = 1$. The solutions are known to be of the form $y = ai + b$ for $i = 0,1,2,\ldots$ where $\mathrm{lcf}(a,b) = 1$. By Dirichlet's theorem on primes in arithmetic progressions, cf. [3, p.13], $y = ai + b$ is a prime number for infinitely many values of i.

Let $r_{A,U} = d_A \cdot e_U$ and define three functions $E_A, R_{A,U}$ and D_U based on $e_A, r_{A,U}$ and d_U as follows:

$$E_A(M) = M^{e_A} \pmod{n};$$

$$R_{A,U}(M) = M^{r_{A,U}} \pmod{n};$$

$$D_U(M) = M^{d_U} \pmod{n};$$

After having established $E_A, R_{A,U}$ and D_U for all $A \in A$ and $U \in A$, all information used in the calculation is destroyed. The algorithms E_A (i.e. the numbers n and e_A) are made public for all $A \in A$; the algorithm D_U (i.e. the number d_U) is only made known to the user U. (It is possible to perform all necessary calculations within a "black box" and passing the secret numbers d_U to the users U in an encrypted form so that security is guaranteed in this crucial phase of calculating the keys required for the various algorithms).

We will now show that each triple $(E_A, R_{A,U}, D_U)$ is an encryption triple.

By [4], (E_A, D_A) for $D_A(M) = M^{d_A}$ (mod n), and (E_U, D_U) for $E_U(M) = M^{e_U}$ (mod n) are total D H pairs. We finally have:

$$D_U(R_{A,U}(E_A(M))) = M^{e_A \cdot r_{A,U} \cdot d_U} \pmod{n}$$

$$= M^{e_A \cdot d_A \cdot e_U \cdot d_U} \pmod{n}$$

$$= D_U(E_U(D_A(E_A(M)))) = M$$

establishing the validity of (1).

D_U and E_A are trap door functions. Their inverses can only be computed easily if the numbers e_U and d_A can be computed on the basis of the known numbers e_A, d_U, $r_{A,U}$, n and their known relations:

$$e_A \cdot d_A \equiv 1 \quad (\text{mod } \varphi(n))$$

$$e_U \cdot d_U \equiv 1 \quad (\text{mod } \varphi(n))$$

$$d_A \cdot e_U = r_{A,U} .$$

The computation of d_A and e_U on the basis of the first two equations would require knowledge of $\varphi(n)$ and hence factoring of n. It is argued at length that factoring is hard in [4]. Since $d_A \cdot e_U$ contains the product $c = p'q'$ of two large primes, a complete factoring of $r_{A,U}$ is equally hard.

Observe further that $E_U = D_U^{-1}$ can be computed easily using (2) and that (2) indeed implies $M^{r_{A,U} \cdot e_A}$ (mod n) = M^{e_U} (mod n).

This, however does not allow one to compute e_U. Thus we also conclude that $R_{A,U}(m) = E_U(D_A(M))$ is indeed a trap door composition: D_A can only be computed easily if d_A is known. Thus, $(E_A, R_{A,U} D_U)$ is an encryption triple.

It now seems natural to obtain a system of encryption triples as required by making all $R_{A,U}$ (i.e. the numbers $r_{A,U}$) public. Unfortunately, this does not lead to a safe system of encryption triples: because of $e_A \cdot r_{A,U} \cdot d_U \equiv 1 \pmod{\varphi(n)}$, each user can calculate a multiple of $\varphi(n)$. By [5], this enables U to factor n, and the system breaks down.

To avoid this problem, $R_{A,U}$ must be kept secret by the information system. Although an information system built on such premise may be viable, it has serious drawbacks when compared with the use of safe systems of encryption triples: in particular, there is the additional security risk that $R_{A,U}$ is divulged and the use of signatures becomes impossible.

We do not know how to generalize [4] to obtain more than above partial implementation. The full implementation of a safe system of encryption triples, either as an extension of [4] or by some other method, remains an important open problem.

5. Conclusion

We have proposed the notion of safe systems of encryption triples suitable for the design of safe and simple information systems. We believe that this novel notion of insuring information security could prove to be of great significance, provided that easy-to-use safe systems of encryption triples can be constructed. A partial solution is presented in Section 4.

Acknowledgement

The authors are grateful to R.L. Rivest for pointing out that the system of encryption triples designed in Section 4 is not safe as was claimed in a preliminary version of this work. He also suggested the "almost simple" realization of a secure information system mentioned at the beginning of Section 3.

References

[1] W. Diffie, M. Hellman: New directions in cryptography, IEEE Transactions on Information Theory (Nov. 1976), 644-654.

[2] M. Gardner: A new kind of cipher that would take millions of years to break, Scientific American 233 (Aug. 1977), 120-124.

[3] G.H. Hardy, E.M. Wright: An introduction to the theory of numbers, Oxford, Clarendon Press (1971).

[4] R. Rivest, A. Shamir, L. Adleman: A method for obtaining digital signatures and public-key cryptosystems, MIT Report, MIT/LCS/TM-82 (1977).

[5] R. Rivest: Private communication (1978).