On Concurrent Identification Protocols
(Extended Abstract)

Oded Goldreich
Laboratory for Computer Science
MIT,room NE43-836,Cambridge,Ma 02139

Abstract

We consider communication networks in which it is not possible to identify the source of a message which is broadcasted through the network. A natural question is whether it is possible for two users to identify each other concurrently, through a secure two-party protocol. We show that more than the existence of a secure Public Key Cryptosystem should be assumed in order to present a secure protocol for concurrent identification. We present two concurrent identification protocols: The first one relies on the existence of a center who has distributed "identification tags" to the users; while the second protocol relies on the distribution of "experimental sequences" by instances of a pre-protocol which have taken place between every two users.

This research was carried out at the Computer Science Department
Technion - Israel Institute of Technology

T. Beth, N. Cot, and I. Ingemarsson (Eds.): Advances in Cryptology - EUROCRYPT '84, LNCS 209, pp. 387-396, 1985.

1. Introduction

Let N be a set of users in a communication network in which it is not possible to identify the source of a message broadcasted on the network . Thus, identification of the source of a message can only rely on the content of the message. Clearly, this would require some sort of a secure authentication scheme as well as a secure protocol which makes use of it.

The task of reaching concurrent identification is somewhat more involved. It requires not only that identification takes place but also that it takes place concurrently; i.e that through this process there would be no situation in which one party had a "substantial" advantage in guessing and/or computing his counterpart's identity. Methods for reaching concurrent identification may be of value in certain business environments in which transactions are carried out in two stages: first reaching an anonymous agreement and only then yielding the identities of the parties to the agreement, as quickly as possible.(An example of such an environment is a future stock exchange without brokers[dealers] or even a present stock exchange controlled by an agency that wishes to prevent biased deals.)

Clearly, if one allows the participation of trusted third parties in the concurrent identification process, trivial solutions exist. However,we are interested in the existence of two-party protocols through which concurrent identification takes place (hereafter referred to as *Concurrent Identification Protocols* or as *cips*).

In Sec. 2 we show that the mere existence of a PKCS (Public Key Cryptosystem [DH]) and a public file of all public keys does not suffice for the existence of a secure cip in the net (i.e. there exists no secure cip in such a net).

In Sec.3 we present a cip which relies on a trusted center which has prepared and distributed "identification tags" to the users at the time the net has been established. (**This center does not participate in the cip!**) The number of transmissions needed to distribute these tags is linear in the number of users; thus the complexity of establishing a net in which this cip can be used securely is still linear in the number of its users. This fact combined with the simplicity of the cip itself makes its implementation reasonably practical.

In Sec. 4 we present a secure cip which does not rely on the honesty of some center nor even on its mere existence. Instead this cip relies on information which has been passed between every pair of users , via instances of a pre-protocol which have taken place at the time the net was established. The fact that the pre-protocol is fairly complicated combined with the fact that $O(|N|^2)$ instances must take place, cause this concurrent identification scheme to be impractical, especially for large networks. However it demonstrates that concurrent identification can take place even if no center exist (at the time the net has been established as well as later).

In both Sec. 3 and 4 we assume the existence of secure cryptosystems, in particular the existence of a secure public key cryptosystem (PKCS)[DH].

A natural problem which arises when designing identification protocols is the *replay problem*, which is hereafter described. User A may try to impersonate user B by using information B has revealed to him in previous instances of the identification protocol. Note that this information has been used to authenticate B and can be used by A to cheat C, unless the protocol has features which prevent such an attempt to cheat. In case of simple identification it is enough to ask for a signature to some time dependent message. (Note that this can not be done trivially in a cip since a signature to any message will immediately reveal the identity of the signer.)

To solve the replay problem in the concurrent identification protocols presented in this paper we use an *Oblivious Transfer* (OT) subprotocol. The notion of OT was first introduced and implemented by Rabin [R]. Another definition of OT, which we believe to be more natural, was suggested by Even,Goldreich and Lempel [EGL] (and implemented using any PKCS). By their definition an OT of a recognizable message ,M, is a protocol by which a *sender* ,S, transfers to a *receiver* ,R, the message M so that R gets M with probability one half while for S the a-posteriori probability that R got M remains one half. In this work, we use a modification of the above definition; for details see the Appendix.

2. Necessary Conditions for the Existence of a CIP

It was already mentioned that no cip (as well as no identification protocol) can exist in a net if it is not assumed that the users are provided with some secure cryptographic identification scheme. We will assume the existence of both a secure conventional cryptosystem (e.g. the DES[NBS]) and a secure PKCS. However, we shall show that this assumption does not suffice to allow the existence of a secure cip, namely:

Theorem 1: A cip, which relies only on the existence of secure cryptosystems (the instances of which are free of any relation other than the cancellation of encryption by the corresponding decryption and vice versa) and a public file of all public keys , can not be secure.

The proof appears in the full version of this paper.

To conclude this section we point out that the "replay problem" is trivially solvable only under irreasonable assumptions, namely:

(i) Each user eavesdrops on all the instances of the cip and records the information he reads.

or

(ii) Each user notifies all the other users about every instance of the cip he participates in.

3. A CIP which Relies on Preparations by a Trusted Center

In this section we show how identification tags distributed, to the users, by a trusted center can grant the existence of a cip. The center can distribute these tags at the time the network is established. The center must be trusted not to collaborate with any user, in the process of distributing the tags as well as during the time the cip is run. It is preferred that the center would seize to exist after distributing the tags.The tags will bear the center's signature and thus be unforgable. Every user can protect himself against the replay of his tags (by other users), by using a tag only once. Thus, the center should provide each user with enough tags.

We assume the existence of a secure PKCS (e.g. the RSA[RSA]) and of a conventional cryptosystem (e.g. the DES[NBS]).We also assume that all users have equel computing power.

3.1. The Identification Tag

Before describing the structure of the identification tag let us introduce some notation:

(i) F denotes a conventional cryptosystem and $F_K(M)[F_K^{-1}(M)]$ denotes the encryption[decryption] of M by F using the key K.

(ii) E_X , D_X will denote the encryption and decryption algorithms of user X (i.e. the PKCS's instance generated by X). Note that $D_X(M)$ can serve as X's signature to M.

(iii) C denotes the center.

(iv) N_X denotes the binary representation of X's name.

An *Identification Tag* (IT) of user X consists of three parts:

(1) The *header* , which contains an (unforgeable) encryption of X's name : $D_C(z, F_y(S), F_y(N_X))$,where y is a randomly chosen key (of length k) to F and z is a random "serial" number.

(2) The *anti-replay* part , which consist of n pairs of recognizable (and unforgeable) messages. The i-th pair denoted AR_i is $(D_C(z, L_i), D_C(z, R_i))$.

(3) The *certified key-bits* part , which consists of the bits of the key , which was used for the encryption of X's name, certified by the center: the certification of y_i (the i-th bit of y) is $D_C(z, i, y_i)$.

Note that all parts of a IT bear the same serial number and that they are signed by the center. User X is called the *legitimate holder* (or just the holder) of the above identification tag. (Note the although other users can have parts of X's tag only X can have all of it if he follows the cip described below properly.)

Remark: S , the L_i's and the R_i's are arbitrary , fixed messages (i.e. invariant of X ,y and z).

We remind the reader that these IT's will be distributed to the users by C at the time the network is established. Note that at that time only X has X's ITs. In the next subsection we will present a cip in which X uses one of his ITs to identify himself without yielding the entire IT. It will be shown that this prevents the replay of this IT by another user.

3.2. The Protocol

The cip described below uses an OT subprotocol which allows a user to send two recognizable messages such that : (1) his counterpart receives exactly one of them; (2) with probability one half the receiver receives the first message; (3) for the sender the a-posteriori probability that the first message was received remains one half; (4) if the sender tries to cheat the receiver will detect it with probability at least one half.

(An implementation of this OT is described in the Appendix and is based on ideas which first appeared in Even,Goldreich and Lempel [EGL].)

The cip proceeds as follows:

```
(The parties to the protocol are denoted $A$ and $B$)
step 1: (linking identity with a secret serial number)
   $A$ chooses one of his unused ITs (hereafter denoted $t_A$)
   marks $t_A$ as "used"
   and transmits $t_A$'s header to $B$.
   $B$ acts symmetrically transmitting $t_B$'s header to $A$.
   (Each checks whether the center's signature
    to the header is authentic.)
step 2: (protection against replay attempts.)
   for $i = 1$ to $n$ do begin
      $A$ sends to $B$ one element out of $t_A$'s $AR_i$ , via OT.
      $B$ acts symmetrically w.r.t. $t_B$ .
      (Each uses the cheat detection mechanism of the OT.)
   end
step 3: (decreasing the time of computing the identity.)
   for $i = 1$ to $k$ do begin
      $A$ transmits to $B$ the $i$-th certified key-bit of $t_A$ .
      $B$ acts symmetrically w.r.t. $t_B$ .
      (Each checks the signature certifying the bit received)
   end
```

3.3. Analysis of the Protocol and the Structure of the IT

Remarks (for $X \in \{A, B\}$)

(R1) The header of t_X establishes a linkage among X's name (although encrypted) the key y (which is used for the encryption of both N_X and the standard message S) and z (which is used as a serial number). It also provides information for the computation of y although this computation becomes feasible only during step(3).

(R2) The anti-replay of t_X allows X to protect himself against the replay of t_X. Note that if X uses t_X only in one instance of the protocol and execute this instance properly then he is (still) the only user in the net who knows both elements of each AR_i in t_X. (Note that his counterpart to the cip instance only got one element out of each AR_i.) User Y ,$Y \neq X$, will succeed in replaying t_X only if he is asked in the OT of each AR_i (which occurs in step (2) of the protocol) to disclose the element of AR_i which is known to him. Note that for Y, both the element he is asked to disclose and the element known to him are randomly chosen out of an AR_i of t_X (this is due to the use of the OT in step(2)). Thus, the probability that Y will succeed in replaying t_X is bounded from above by 2^{-n}. Thus, a proper execution of step (2) of the protocol (only) assures the parties that the identification tags are in the hands of their legitimate holders.

(R3) The third part of t_X (which is exchanged in step (3) of the protocol) allows the gradual decrease in the time of computation which is required to extract N_X from the header of t_X. N_X is extracted by first finding the key y which transforms the message S into the cryptogram $F_y(S)$. Note that this computation becomes feasible (during step(3) of the protocol) only after the tag holder has proven himself to be the legitimate one (by succeeding in an unfaulty execution of step(2) of the protocol).

(R4) If the rate ,in which the time which is required to compute N_X given the header of t_X decreases, is considered to be too fast one may slow it down by using simple "exchange of half bit" schemes (e.g. Tedricks' schemes[T]).

(R5) The interleaving in step(2) of the protocol is not material.

(R6) One can use the "conventional OT" instead of the "one-out-of-two OT" for an oblivious tranfer of each element of the anti-replay. However, the analysis of such a protocol will be more involved.

(R7) There is some similarity between the ideas used in the above anti-replay, and the ideas of Bennett et al. ([BBBW]). However, Bennett et al. consider a specific physical device which stores 2 messages such that only one of them can be read; while we consider a protocol through which one out of two messages is randomly transferred.

We claim that this cip is secure provided the following assumptions hold:

(A1) A trusted center has distributed the identification tags described in sec. 3.1 to the legitimate holders.(The center is trusted not to convey any information about the tags he has provided user X to any other user.He is also trusted not to yield his signature algorithm.)

(A2) All parties have equal computing power.

(A3) Both the conventional cryptosystem and the PKCS used by the protocol are secure. (No one can forge C's signature. Extracting M from $F_K(M)$ given S,$F_K(S)$ and some of K's bits requires exhaustive search on all keys which match the known bits of K ; when no bit of K is known this computation is infeasible.)

Theorem 2: If the above assumptions hold and a user ,U, plays the protocol properly then the following hold:

(1) In any phase during the execution of the protocol,if U's counterpart can find out U's identity using expected time t then U can find out what is claimed to be his counterpart's identity in about the same expected time.
(2) If U's counterparts is honest U will find out his identity.
(3) If U's counterpart is impersonating then with high probability $(1 - 2^{-n})$ U will find this out before reaching a stage in which the cómputation of his identity is feasible.

The proof appears in the full version of this paper.

4. A CIP which Relies on Preparations by Instances of a Pre-Protocol

In this section we (only) assume the existence of a secure PKCS. We show how a pre-protocol, played between every pair of users,can grant the existence of a cip in the net. Note that we do not assume that there exists some (trusted) center and that we do not assume that all parties have equal computing power. (It should be stressed that we do not refer to the public file of the users' encryption keys as a center.) Since instances of the pre-protocol must take place between every pair of users, the result of this section ,although being of theoretical interest, is practical only for "small" networks. The purpose of the pre-protocol is to distribute *secure experimental sequences* which will be used in the identification process. These sequences will be unforgeable and will yield the identity of their legitimate holder[1] if some parts of them are read completely. However it will be possible to give away only small (still unforgeable) fragments of the sequence yielding only a "small amount of information" about their legitimate holder.

The idea behind the implementation of these experimental sequences (hereafter referred to as SES's) is to allow a user to conduct experiments on the bits of another user's name. The experiment is gauranteed to give a result equal to the tested bit with some fixed probability greater than one half. Thus conducting enough experiments on a bit gives certainty of knowing its right value ; whereas on the other hand a single experiment does not give much information about the corresponding bit. The cip consist of letting each user experiment on each of his counterpart's name bits by just sending one entry in the experimental sequence. The implementation of a process which constructs secure experimental sequences is discussed in the full version of this paper ([G]). (Its essence is that the SES will be built anonymously by the user who will later experiment on it. The sequence will be built by flipping a biased coin so that its builder will only know the expected value of an entry in it and not the concrete value. This will be achieved by using an OT.)

Remark: The idea of using a biased coin as a tool for exchanging a bit of information was suggested ,independently, by Lubi,Micali and Rackoff in their MiRackoLus paper [LMR]. It should be stressed that the problem they were facing was much more difficult

[1] As in Sec.3 it will happen that other users know part of the sequence but only one user (its holder) knows all of it, provided he follows the cip which reveals parts of it properly.

and their solution (a coin the bias of which is determined by the secrets of both parties and without yielding these secrets) much more inspirating. However , the author does not know of any reduction between the biased coin used here and the symmetric biased coin suggested in [LMR]; there are too many differences in the setting, conception and implementation!

4.1. Sketch of the Concurrent Identification Protocol

```
(The parties to the cip will be denoted $A$ and $B$)
(0) $A$ notifies $B$ which of $B$'s SESs he would like to examine.
    $B$ acts symmetrically w.r.t $A$'s SESs.
(1) $A$ checks whether he is communicating with the legitimate holder
    of the SES (i.e. $B$).
    $B$ acts symmetrically.
    (This is done by testing the anti-replay part of the SES
     similarly to the way it was done in the cip of Sec. 3.)
(2) for $i = 1$ to $q$ (the number of entries in a SES) do begin
        $A$ transmits the $i$-th entry of his SES to $B$.
        $B$ acts symmetrically.
    end
```

4.2. Analysis of the Protocol

Under the assumption that there exist SESs in the network it is straightforward to prove that the cip presented above is secure,namely:

Theorem 3: If a user ,U plays the above cip properly then the following hold:

(1) In any phase during the execution of the protocol,if for
U's counterpart the entropy of U's name is e then for U
the entropy of what is claimed to be his counterparts name
is very close to e.

(2) If U's counterparts is honest U will find out his identity.

(3) If U's counterpart is impersonating then with high probability
$(1 - 2^{-n})$ U will find this out before reaching a stage
in which he has revealed any information about his identity .

The proof appears in the full version of this paper.

5. Acknowledgements

I would like to thank Prof. Shimon Even for suggesting the problem to me, for many helpful discussions and especially for a briliant insight that was of much help in the analysis of the cip presented in Sec. 4.

I would also like to thank Prof. Adi Shamir for the discussion which led to the cip presented in Sec. 3.

I am very grateful to Tom Tedrick for carefully reading the manuscript, finding many errors and giving me feedback.

6. References

[A] Abramson,N., *Information Theory and Coding*, *M_c*Graw-Hill,1963, pp. 100-105.

[BBBW] Bennett,C.H., Brassard,G., Breidbart,S., and Wiesner,S., "Quantum Cryptography or Unforgeable Subway Tokens", in *Advances in Cryptology:Proceedings of Crypto82*, (Chaum,D. et al. editors), Plenum Press, 1983, pp. 267-275.

[DH] Diffie,W., and Hellman,M.E., "New Directions in Cryptography", *IEEE Trans. on Inform. Theory*,Vol. IT-22,No. 6,November 1976, pp. 644-654

[EGL] Even,S., Goldreich,O., and Lempel,A., "A Randomized Protocol for Signing Contracts", in *Advances in Cryptology:Proceedings of Crypto82*, (Chaum,D. et al. editors), Plenum Press, 1983, pp. 205-210

[*EGL'*] Even,S., Goldreich,O., and Lempel,A., "A Randomized Protocol for Signing Contracts", TR No. 233, Computer Science Dept., Technion, Haifa, Israel, February 1982

[G] Goldreich,O., "On Concurrent Identification Protocols", MIT/LCS/TM-250, December 1983

[LMR] Lubi,M., Micali,S., and Rackoff,C., "How to Simultaneously Exchange a Secret Bit by Flipping a Symmetrically-Biased Coin", *proceedings of the 24th IEEE Symp. on Foundation Of Computer Science*, 1983, pp. 11-21

[NBS] National Bureau of Standards, Data Encryption Standard, *Federal Information Processing Standards*, Publ. 46, 1977

[R] Rabin,M.O., "How to Exchange Secrets by Oblivious Transfer", Technical memo TR-81, Harvard Center for Research in Computing, (1981).

[RSA] Rivest,R.L., Shamir,A., and Adleman,L., "A Method for Obtaining Digital Signature and Public Key Cryptosystems", *Comm. of the ACM* ,Vol.21, February 1978, pp. 120-126

[S] Shannon,C.E., "Communication Theory of Secrecy Systems",*Bell Syst. Jour.* 28, October 1949, pp. 656-715

[T] Tedrick,T., "How to Exchange Half a Bit", to appear in the proceedings of *Crypto83*

7. Appendix: An Implementation of OT

Assume S wants to transfer to R exactly one of the messages M_1 and M_2,such that:

(1) R can recognize both M_1 and M_2
(e.g. they are signatures to known messages).
(2) If S is honest then R gets M_1 with probability one half.
For S the a-posteriori probability that R got M_1 remains one half.
(3) If S tries to cheat, R will detect it with probability at least one half.

An implementation of this transfer proceeds as follows:

(0) S chooses ,randomly, two pairs (E_1, D_1) and (E_2, D_2) of
encryption-decryption algorithms of the PKCS.
R chooses ,randomly, a key K
for the conventional cryptosystem F.
(1) S transmits E_1 and E_2 to R.
(2) R chooses ,randomly, $r \in \{1,2\}$
and transmits $E_r(K)$ to S.
(3) S computes $K'_i = D_i(E_r(K))$,for $i \in \{1,2\}$.
S chooses ,randomly, $s \in \{1,2\}$ and transmits
$(F_{K'_1}(M'_1), F_{K'_2}(M'_2), s)$
to R, where $M'_s = M_1$ and $M'_{3-s} = M_2$.

Remarks:

(1) Assuming that K looks like random noise and that E_1,E_2 have the same range, S can not guess with probability of success greater than one half which of the K'_i's, computed by him is the K choosen by R.

(2) Assume that the instances of the PKCS are free of any relation other than the cancellation of encryption by the corresponding decryption and that K'_i must be known in order to read M'_i.

(3) By (1) and (2) if S is not cheating then R can read M'_i iff $i = r$. Thus, he can detect cheating by S with probability one half.

(4) In the RSA[RSA] scheme, distinct E_i's may have different ranges. However, this difficulty can be overcome (see [EGL']).

(5) One can use a one-time pad instead of the conventional cryptosystem F.