J. Cryptology (1996) 9: 191–195

Journal of
CRYPTOLOGY

A Secure Protocol for the Oblivious Transfer (Extended Abstract)

M. J. Fischer
Yale University

S. Micali
Massachusetts Institute of Technology

C. Rackoff
University of Toronto

Communicated by Claude Crépeau

Written 14 February 1984 and presented 11 April 1984

1. Introduction

There has been a great deal of interest lately in cryptographic protocols. Vaguely, a cryptographic protocol is a mean whereby two or more people can interact in such a manner as to exchange a certain amount of information, while keeping other information secret (either from one of the participants or from another party).

One interesting task for which one would like a protocol is the "oblivious transfer" introduced by Rabin in [4]. This involves two parties A and B; A would like to send a message of some sort to B, with the constraint that B has a 50% chance of receiving the message, and the other half of the time B receives no information at all about the message. The additional constraint is that A has no idea whether or not B received the message. Oblivious transfer can be viewed as a special type of coin tossing. Although it is not useful in and of itself, it appears to be very useful as a means towards other ends, and in fact has been used in a number of other protocols by a number of different researchers [1]–[3].

In [4], Rabin proposed a protocol for the oblivious transfer. It was intended that the protocol be correct, assuming only that it is hard to factor certain large composite numbers. However, as described below, there is a potential flaw in his protocol; it is possible that B can cheat and obtain extra information from A, even if the assumption about the difficulty of factoring is true. Although we cannot prove that B can cheat in this way, no one has yet been able to prove that B cannot. In Section 4 we present a new protocol for the oblivious transfer. It is similar to Rabin's, but we fix the potential flaw so that it is possible to *prove* that our protocol works, subject only to the assumption about the difficulty of factoring.

2. An Informal Discussion of the Oblivious Transfer, and Rabin's Protocol

Although it could be defined more generally, our definition of the oblivious transfer will assume the message to be sent is the factorization of a number m.

Let m be an (at most) n bit number which is the product of two (at most) $n/2$ bit odd primes. Say that A knows the factorization of m but B only knows the number m. A and B do not trust each other, but wish to interact in such a way that with probability $1/2$ B learns the factorization of m, and with probability $1/2$ B is not able to factor m. In addition, A should have no idea which of the two cases hold. Before expressing this more rigorously, we describe Rabin's proposed protocol.

Rabin's proposed protocol for the oblivious transfer

We assume that A and B both know the number m, and that A knows the factorization.

step 1. B chooses a random $x \in \{x | 1 \leq x \leq m$, and x and m are relatively prime$\} = Z_m^*$. B then computes $y = x^2 \bmod m$ and sends y to A.

step 2. A computes a random square root (mod m) z of y, and sends z to B. (If no square root exists, then A does nothing.)

step 3. B checks that $z^2 = y \bmod m$, and if not output "c" for cheating. (See the formal definition in Section 3.) Let us assume that $z^2 = y \bmod m$. Now it is well known that y has 4 square roots mod m, which can be written as $\{x, -x, w, -w\}$, where B knows x. With probability $1/2$, z will be x or $-x$, and hence B receives no information. (In this case, B outputs "?" in order to conform with the formal definition in Section 3.) With probability $1/2$, however, z will be w or $-w$, and in this case $\gcd(m, x - z)$ will be a factor of m, allowing B to output the factorization of m.

Discussion of Rabin's protocol

A cannot cheat by sending back some cleverly chosen square root z of y: no matter what A does, $z \in \{x, -x\}$ with probability $1/2$, and $z \in \{w, -w\}$ with probability $1/2$, and A can have no idea which is the case.

Is it clear, however that B cannot cheat? We wish it to be the case that B cannot factor m with probability (much) bigger than $1/2$, even if B cheats, and we wish to *prove* this assuming *only* that factoring is difficult. What if B chooses to send a y which is not a square residue mod m? In this case A will not respond, and B only learns the *one* bit of information that y is not a square residue, which can not help B to factor m. What if instead of sending A the square y of a *randomly* chosen x, B sends the square of a *particular* cleverly chosen x? This cannot help B, since with probability $1/2$, B will still receive $z \in \{x, -x\}$.

However, what if B doesn't square any x at all, but instead picks a particular cleverly chosen square residue y to send? Perhaps knowing *any* square root mod m of y will allow B to factor m. That is, perhaps there is a polynomial time algorithm which given m produces (with high probability) a square residue y, and another algorithm which given m, y, and any square root of y mod m *factors* m (with high probability). The point is not that we have such algorithms, but that *no one has shown that they do not exist.*

Hence, the proof that Rabin's protocol is correct relys not only on an assumption about the difficulty of factoring, but on an *additional* complicated and unnatural assumption (saying, essentially, that the above algorithms do not exist).

In Section 4 we show how to *fix* Rabin's protocol so that it can be *proven* correct, assuming *only* that factoring is difficult.

3. A Rigorous Statement of the Problem

We first formalize what we mean for factoring to be hard. Let

$$H_n = \{m | m \text{ has } \leq n \text{ bits, and is the product of two distinct odd primes, each with } \leq n/2 \text{ bits.}\}$$

The *factoring assumption* is that any polynomial time, probabilistic algorithm has the property that for each k and sufficiently large n, if the algorithm is given a random member m of H_n, then the probability it outputs the factors of m is $\leq 1/n^k$.

We now define what we mean by a protocol for the oblivious transfer. Note that in our definition B has 3 possible outputs: either B outputs the factors of m, or B outputs "?" indicating it doesn't know the factors, or B outputs "c" (for cheating) indicating that it has detected that A hasn't followed its protocol correctly. When reading this definition for the first time, the reader can ignore the possibility that B outputs "c", and just assume that A will never do anything so blatantly dishonest that it will be caught (since in practice, in the known protocols, this is a reasonable assumption).

We define an *oblivious transfer protocol* to be a pair of probabilistic, interacting Turing machines A and B; A and B will both see the same composite number m, and A will also be given the factorization of m; A and B will both halt in time polynomial in the *length* of m, and B will either output "c", "?", or the factorization of m. In addition, if m is a randomly chosen member of H_n, then the following hold

1) If A and B properly follow the protocol, then for each k and sufficiently large n, B outputs the factorization of m with probability $\geq (1/2) - (1/n^k)$, "?" with probability $\geq (1/2) - (1/n^k)$, and "c" with probability $\leq 1/n^k$.
2) Say that A is replaced by any other machine A', and B still follows its protocol correctly. A' has an output representing its guess at the output of B. Then the probability that A' outputs the same thing as B, given that B doesn't output "c", is $\leq (1/2) + (1/n^k)$ for all k and sufficiently large n. (This implies that the probability that B outputs the factorization is within $1/n^k$ of the probability that B outputs "?".)
3) Say that B is replaced by any other machine B', and A still follows its protocol correctly. Then the probability that B' outputs the factorization of m is $\leq (1/2) + (1/n^k)$ for each k and sufficiently large n.

4. Our Protocol

Our goal is to fix Rabin's protocol so that it can be *proven* correct, assuming only the factoring assumption. Recall that the problem is that in step 1, when B sends y to A, it

is possible that B does not know a square root of $y \bmod m$. We therefore add step 1.5 to the protocol:

step 1.5. *B proves* to A that *B knows* a square root of $y \bmod m$, without giving A any ideas *which* square root B knows.

Step 1.5 requires a *(sub)protocol* to be performed which is very interesting in its own right. We now describe that protocol. The properties possessed by the subprotocol will become more clear in the discussion that follows it.

(Sub)protocol for B to prove to A that B knows a square root of $y \bmod m$, *without giving A any idea which square root B knows*

(Assume B knows x, a square root of $y \bmod m$.)

step I. B chooses n random numbers $r_1, r_2, \ldots, r_n \in Z_m^*$ and computes $y_1 = yr_1^2$, $y_2 = yr_2^2, \ldots, y_n = yr_n^2$, and sends $S = \{y_1, y_2, \ldots, y_n\}$ to A.

step II. A chooses a random subset $S_1 \subseteq S$ of size $n/2$ and sends it to B.

step III. B checks that $S_1 \subseteq S$ and is of size $n/2$ (and otherwise outputs "c"). For each $y_i \in S_1$, B sends r_i to A. For each $y_i \in S - S_1$, B sends $z_i = xr_i$ (a square root of y_i) to A.

step IV. A checks that for each $y_i \in S_1$, $y_i = yr_i^2$, and for each $y_i \in S - S_1$, $y_i = z_i^2$. (Intuitive remark: in this case, it is very likely that for some i, B knows r_i and z_i such that $y_i = yr_i^2$ and $y_i = z_i^2$; in this case, B "knows" a square root of y, namely $z_i/r_i \bmod m$.) If so, A is "convinced"; if not, A is "not convinced".

Discussion of the (sub)protocol

A gets no information about which square root of y B knows. The reason for this is that if r_i is randomly chosen, then xr_i is a random square root of yr_i^2.

It remains to show that B cannot, even by cheating, convince A that B knows a square root of y if B does not. More precisely, let B' be any machine replacing B in the protocol. Say that B' has flipped some coins in order to determine all its random choices, including the values of $y_1, y_2, \ldots, y_n$. For some of the i, B' will be able to give a square root z_i of y_i and for some of the i, B' will be able to give r_i such that $y_i = yr_i^2$. If for some i, B' can do both, then B can find a square root of y, namely z_i/r_i. If for no i can B' do both, then there is at most *one* choice for S_1 which will allow B' to convince A; in this case the probability A will be convinced is $\leq 1/\binom{n}{n/2} \leq 1/n^k$ for all k.

A more complete definition of this protocol, and a proof of correctness, will appear in the final paper.

Discussion of the new protocol for oblivious transfer

The complete formal proof that the new protocol is correct assuming the factoring assumption, will not be given in this extended abstract, but all the necessary ideas have been presented here. The proof is constructive in the following sense: there is no A' which violates condition 2 in the definition of oblivious transfer, and if there is any B' which violates condition 3, then this B' can be used as a subroutine in an algorithm which factors numbers so well as to violate the factoring assumption.

References

[1] Blum, M. "Three applications of the oblivious transfer", manuscript, Univ. of Calif. at Berkeley, 1981.
[2] Blum, M. "How to exchange (secret) keys", 15 ACM Symp. on Theory of Computing, 1983, 440–447.
[3] Even, S., Goldreich, O., Lempel, A. "A randomized protocol for signing contracts", *Advances in Cryptology*; *Proceedings of Crypto* 82, Plenum Press, 1983, 205–210.
[4] Rabin, M. O. "How to exchange secrets by oblivious transfer", manuscript, Harvard Center for Research in Computer Technology, 1981.