How to Exchange Secrets with Oblivious Transfer

Michael O. Rabin

May 20, 1981

This paper originally appeared as

> *Michael O. Rabin. How to exchange secrets with oblivious transfer. Technical Report TR-81, Aiken Computation Lab, Harvard University, 1981.*

Although the original manuscript has a different title, the paper is most commonly (if not only) cited with the title given here. Thus, it should continue to be cited in this manner with reference to the original technical report.

This file consists of two parts that preserve the original work, namely a scanned version and a typeset version in LaTeX.

A scanned version of the original hand-written manuscript (pp. 2–21)

This paper appeared in print as a Harvard University Technical Report, but at some point the university ran out of copies. At that time copies of the hand written version started to circulate, and were the only ones available. As access to these copies has become difficult I have scanned my copy of the paper and I'm posting it on the web for others to read.

—Tal Rabin
`talr@us.ibm.com`

A typeset version (pp. 22–25)

As this paper puts forward the notion of "Oblivious Transfers" and is a well-known and frequently cited paper, I felt I should typeset the manuscript, and here is the result.

While typesetting, I tried to stick to the original manuscript as much as possible. However, there were some cases, such as a few typos or punctuation marks, which were changed.

As in many papers on cryptography, Alice and Bob play the role of participants of the given cryptographic protocols. For the sake of readability, Alice's and Bob's messages were printed in red and blue ink, respectively.

This work was carefully proofread by my colleague Y. Sobhdel (`sobhdel@ce.sharif.edu`). Thanks also to H. M. Moghaddam for mentioning a minor mistake in an earlier version. That said, I will be thankful if you inform me of any possible mistakes.

—Mohammad Sadeq Dousti
`dousti@ce.sharif.edu`

May 20, 81

How to Exchange Secrets

by

Michael O. Rabin

<u>Introduction</u>. Bob and Alice each have a secret, SB and SA, respectively, which they wish to exchange. For example, SB may be the password to a file that Alice wants to access (we shall refer to this file as <u>Alice</u>'s file) and SA ~~the password~~ ~~Note that each to~~ the password to Bob's file. Can they set up a protocol to exchange the secrets without using a trusted third party and without a safe mechanism for the simultaneous exchange of messages?

To exclude the possibility of randomizing on the possible digits of the password, we assume that if an incorrect password is used then the file is erased, and that Bob and Alice want to guarantee that this will not happen to their respective files.

Because of this assumption we can take, ~~S_A and S_B~~ without loss of generality, S_A and S_B to be single Bits.

As stated, there is nothing to prevent Bob from giving Alice a wrong password S, possibly even in exchange for the correct secret S_A. Now Bob will read his file while Alice, using $S + S_B$, will destroy her file.

~~Thus~~ We assume that the correct passwords SB and SA are indelibly transcribed as prefixes to Alice's and Bob's file. Furthermore, ~~that~~ Alice and Bob have a procedure to ~~each~~ give each other signed messages (contracts), and can resort to ~~adj~~ subsequent adjudication to prove fraud.

Under these conditions Bob can, for example, give Alice a message "My secret is S, signed Bob". If Alice now uses the password S and $S \neq SB$ then her file, with the exception of the prefix containing SB, is destroyed and Alice can resort to adjudication with a provable case against Bob.

The above mentioned message, however, does not provide a solution to the EOS problem. Alice can recieve the signed message ~~it~~, and read her file without giving SA to Bob. When Bob goes to court, Alice can say: "I gave Bob the password SA and he has not used it"; I am willing to reveal ~~give~~ it again right now". Even if Bob obtains SA at the time of ~~[illegible]~~ adjudication, Alice has gained an advantage ~~over him~~ by having read her file well ~~[illegible]~~ ahead of him.

With all the above assumptions the problem still seems to be unsolvable. Any EOS protocol must have the form: Alice gives to Bob some information I_1, Bob gives to Alice J_1, Alice gives to Bob I_2, etc. There must exist a first k such that, say, Bob can determine SA from $I_1, \ldots, I_k$, while Alice still cannot determine SB from $J_1, \ldots, J_{k-1}$. Bob can ~~stop~~ withhold J_k

from Alice and thus obtain SA without revealing SB.

The way out of this difficulty is to construct an EOS protocol such that ~~when~~ ~~Alice knows that~~ from the fact that Bob knows SA, Alice can deduce SB.

To ~~make~~ render this feasable, we make a final assumption that if Bob uses SA to read his file then Alice knows about this and vice-versa.

The ~~pro~~ general problem of the exchange of secrets, ~~that~~ without the particular setting and assumptions discussed above, was suggested to me by Richard De Millo.

The EOS Protocol. We assume that Alice has a public key K_A and Bob has a public key K_B which they can use for encription and for digital signatures. Every message sent by Alice to Bob will be signed by her, using K_A, and ~~vice versa~~. similarly for Bob.

~~Bob~~ Alice chooses two large primes p, q and creates a one-time key $n_A = p \cdot q$.

She then ~~send~~ gives Bob a message: "The one-time key is n_A, signed Alice". Bob chooses primes p_1, q_1 and ~~send~~ gives $\overset{= p_1 \cdot q_1}{n_B}$ to Alice in a signed message.

Bob now chooses randomly an $x \le n_A$, computes $c = x^2 \bmod n_A$, and ~~send~~ gives Alice the message "$E_{K_B}(x)$ is the encoding by my public key K_B of my chosen number and ~~x~~ c is the square mod n_A of that number, signed Bob".

Alice who knows the factors p, q of n_A calculates an x_1 such that

$x_1^2 = c \bmod n_A$. (See [1] for the square-root extraction algorithm and for the facts used in the next paragraphs.) Alice now gives Bob the message: "x_1 is a square-root mod n_A of c", signed Alice".

Bob calculates the g.c.d $(x - x_1, n_A) = d$. With probability 1/2 we have [$d = p$ or $d = q$], so that with probability 1/2 Bob now has the factorization $n_A = p \cdot q$. However, since Alice does not know Bob's x, she does not know whether Bob has the factorization of n_A.

We refer to this mode of transfering information, where the sender does not know whether the recipient actually received the information, as an ~~blindfolded~~ oblivious transfer.

Next Bob effects an ~~blindfolded~~ oblivious transfer of n_B to Alice.

Define

$$\nu_B = \begin{cases} 0 & \text{if } (x-x_1, n_A) = p \text{ or } q \\ 1 & \text{otherwise} \end{cases}$$

Thus $\nu_B = 0$ iff ~~in~~ after the above ~~blindfolded~~ oblivious transfer of the factorization of n_A from Alice to Bob, he knows the factors. ~~The bit~~ ~~ν_A for Alice~~ Alice's bit ν_A is defined in a similar way.

¶ Recall that SA and SB are each a single bit.

Bob forms the exclusive-or $\varepsilon_B = S_B \oplus \nu_B$ (Reader: $S_B = SB$!) and gives it to Alice in a signed message

"ε_B is the exclusive-or of my secret with

> my state of knowledge of the factors of n_A,

signen, Bob." Knowledge of ε_B does not contribute anything to Alice's ability to access her file,

Similarly Alice forms $\varepsilon_A = S_A \oplus \nu_A$ and gives it to Bob in a signed message.

We come to the final round of the EOS protocol. Alice places her secret S_A as the center bit in an otherwise random message m_A. She then encodes m_A as $E_{n_A}(m_a) = ($ using any of the public-key systems which require the factors p, q of n_A for decoding. (We may, for example use the encoding $E_{n_A}(m_A) = m_A^2 \bmod n_A$ of [1], provided that we have a fixed small prefix of m_A to distinguish m_A among the 4 square roots mod n_A of $E_{n_A}(m_A)$.) Alice sends $d_A = E_{n_A}(m_A)$ to Bob in a signed message.

Bob follows the same steps using S_B and n_B and sends the encoded result to Alice

Theorem. The above protocol gives, under the assumptions in the Introduction, a solution of the Exchange of Secrets Problem. The probability that neither side will ~~gain~~ obtain the other's secret is 1/4. ~~access to his file is 1/4.~~

Proof. We omit the proof that the signed messages exchanged between Alice and Bob, and the indelible incorporation of S_A and S_B in the files, ~~suffice for either side to~~ provide ~~to~~ each participant ~~[illegible]~~ with a ~~[illegible]~~ provable case against the other, ~~[illegible]~~ if the other one cheated.

It is clear that if either Alice or Bob stop participation in the EOS protocol (,in which case the other one will also stop) before the final phase), then neither can know the other's secret.

Assume that Alice has given Bob, in the final phase, the encoded secret $d_A = E_{n_A}(m_A)$ If Bob infact knows the factorization $n_A = p \cdot q$, in which case $\nu_B = 0$, he can decode d_A, finding m_A and S_A. If Bob now uses the password (bit) s_A to read his file then, by assumption, Alice will know this. Again by assumption Bob would attempt reading his file only if he knows S_A with certainty (a mistake will destroy the file). Thus Alice knows that $\nu_B = 0$ and hence that $E_B = S_B \oplus \nu_B = S_B$ so that she knows S_B .

If Bob gave Alice $d_B = E_{n_B}(m_B)$ in the final phase, then the above argument applies to ~~the case~~ yield that if Alice reads her file before Bob, then Bob will know S_A .

Thus if either Alice or Bob reads her or his file, the other one will know the password for his or her file.

The probability, when the protocol was completed, that neither one knows the other's secret is $(1/2)^2 = 1/4$. ∎

<u>Remarks</u>. In the case that the exchange of secrets has not been effected, it is not possible to iterate the procedure. ~~Because~~ One participant, say Alice, may actually know S_B after the first round but deliberately not access her file until after the second round. Bob ~~this~~ may not know whether v_A was 0 in the first or second round

and then will not be able to read his file.

Remark 2. The probability of success of the EOS protocol can be enhanced by ~~b~~ modifying the oblivious transfer of information subprotocol. After ~~recie~~ receiving n_A from Alice, Bob chooses two numbers $x, y \leq n_A$ ~~so~~ and gives ~~Al~~ Alice the squares $x^2, y^2 \bmod n_A$. Alice gives Bob two ~~a~~ square roots $x_1, y_1 \bmod n_A$ of x^2 and~~y~~ y^2 respectively. Now Bob has a probability 3/4 of knowing the factorization $n_A = p \cdot q$.

When Bob gives Alice $\varepsilon_B = S_B \oplus \nu_B$, she knows that with probability 3/4, $\varepsilon_B = S_B$. Since we assumed that ~~her~~ Alice is determined to guarantee that her file will not be erased, it still follows that she will not use ε_B as the password. Rather, as before, she will wait until she either can read S_B by deciphering $E_{n_B}(m_B)$, or can infer $\nu_B = 0$ from the fact that Bob has acessed his file.

The above double iteration of the oblivious transfer of information is also effected from Bob to Alice. The rest of the EOS protocol is as before.

Each participant has now just a 1/4 probability of not knowing the factorization of the other's one-time key. Thus the probability of non-termination of the EOS protocol is $(1/4)^2 = 1/16$.

There is a limit beyond which the above enhancement cannot be carried. If, for example, the oblivious transfer subprotocol is modified so that $\Pr(\vartheta_B = 0) \sim 1/32{,}000$ then $\Pr(\varepsilon_B = \mathcal{S}_B) = 1 - 1/32{,}000$. Now there is a real temptation for Alice to halt the protocol after receiving ε_B, and ~~using~~ use

$\mathcal{E}_B$ as the password to her file.

Conclusion. Let us mention some problems for further research.

The oblivious transfer of information subprotocol is valid even without any of the assumptions we made in order to make EOS feasable. What other applications can one find for this sub-protocol.

Can any of ~~Which of~~ the assumption we made ~~can~~ be relaxed or eliminated ~~and still~~ without losing the possibility of EOS.

Is it possible to construct an EOS protocol which will always terminate, or can one prove that the non-zero probability of non-termination is essential.

Bibliography

1. Rabin, M.O., Digital Signatures and Public Key Systems as Intractable as Factorization MIT LCS TM 1979.

How to Exchange Secrets with Oblivious Transfer

Michael O. Rabin

May 20, 1981

1 Introduction

Bob and Alice each have a secret, SB and SA, respectively, which they wish to exchange. For example, SB may be the password to a file that Alice wants to access (we shall refer to this file as Alice's file), and SA the password to Bob's file. Can they set up a protocol to exchange the secrets without using a trusted third party and without a safe mechanism for the simultaneous exchange of messages?

To exclude the possibility of randomizing on the possible digits of the password, we assume that if an incorrect password is used then the file is erased, and that Bob and Alice want to guarantee that this will not happen to their respective files. Because of this assumption, we can take, without loss of generality, SA and SB to be single bits.

As stated, there is nothing to prevent Bob from giving Alice a wrong password S, possibly even in exchange for the correct secret SA. Now Bob will read his file, while Alice, using $S \neq SB$, will destroy her file.

We assume that the correct passwords SB and SA are indelibly transcribed as prefixes to Alice's and Bob's files. Furthermore, Alice and Bob have a procedure to give each other signed messages (contracts), and can resort to subsequent adjudication to prove fraud.

Under these conditions, Bob can, for example, give Alice a message "My secret is S, signed Bob". If Alice now uses the password S and $S \neq SB$, then her file, with the exception of the prefix containing SB, is destroyed and Alice can resort to adjudication with a provable case against Bob.

The above mentioned message, however, does not provide a solution to the EOS[1] problem. Alice can receive the signed message, and read her file without giving SA to Bob. When Bob goes to court, Alice can say: "I gave Bob the password SA and he has not used it; I am willing to reveal it again right now." Even if Bob obtains SA at the time of adjudication, Alice has gained an advantage by having read the file well ahead of him.

With all the above assumptions the problem still seems to be unsolvable. Any EOS protocol must have the form: Alice gives to Bob some information I_1, Bob gives to Alice J_1, Alice gives to Bob I_2, etc. There must exist a first k such that, say, Bob can determine SA from $I_1, \ldots, I_k$, while Alice cannot determine SB from $J_1, \ldots, J_{k-1}$. Bob can withhold J_k from Alice and thus obtain SA without revealing SB.

The way out of this difficulty is to construct an EOS protocol such that, from the fact that Bob knows SA, Alice can deduce SB.

[1] Footnote added during typesetting: Exchange of Secrets

To render this feasible, we make a final assumption that if Bob uses SA to read his file then Alice knows about this and vice versa.

The general problem of exchange of secrets, without the particular setting and assumptions discussed above, was suggested to me by Richard DeMillo.

2 The EOS Protocol

We assume that Alice has a public key K_A and Bob has a public key K_B which they can use for encryption and for digital signatures. Every message sent by Alice to Bob will be signed by her, using K_A, and similarly for Bob.

Alice chooses two large primes p, q and creates a one-time key $n_A = p \cdot q$. She then gives Bob a message: "The one-time key is n_A, signed Alice". Bob chooses primes p_1, q_1 and gives $n_B = p_1 \cdot q_1$ to Alice in a signed message.

Bob now chooses randomly an $x \leq n_A$, computes $c = x^2 \bmod n_A$, and gives Alice the message "$E_{K_B}(x)$ is the encoding by my public key K_B of my chosen number, and c is the square $\bmod n_A$ of that number, signed Bob".[2]

Alice who knows the factors p, q of n_A calculates an x_1 such that ${x_1}^2 = c \bmod n_A$. (See [1] for the square-root extraction algorithm and for the facts used in the next paragraphs.) Alice now gives Bob the message: "x_1 is a square-root $\bmod n_A$ of c, signed Alice".

Bob calculates the g.c.d $(x - x_1, n_A) = d$. With probability $1/2$ we have $[d = p \text{ or } d = q]$, so that with probability $1/2$ Bob now has the factorization $n_A = p \cdot q$. However, since Alice does not know Bob's x, she does not know whether Bob has the factorization of n_A.

We refer to this mode of transferring information, where the sender does not know whether the recipient actually received the information, as an oblivious transfer.

Next Bob effects an oblivious transfer of n_B to Alice.

Define

$$\nu_B = \begin{cases} 0 & \text{if } (x - x_1, n_A) = p \text{ or } q, \\ 1 & \text{otherwise.} \end{cases}$$

Thus $\nu_B = 0$ iff after the above oblivious transfer of the factorization of n_A from Alice to Bob, he knows the factors. Alice's bit ν_A is defined in a similar way.

Recall that SA and SB are each a single bit. Bob forms the exclusive-or $\varepsilon_B = S_B \oplus \nu_B$ (Reader: $S_B = SB$!), and gives it to Alice in a signed message "ε_B is the exclusive-or of my secret with my state of knowledge of the factors of n_A, signer, Bob." Knowledge of ε_B does not contribute anything to Alice's ability to access her file.

Similarly, Alice forms $\varepsilon_A = S_A \oplus \nu_A$ and gives it to Bob in a signed message.

We came to the final round of the EOS protocol. Alice places her secret S_A as the center bit in an otherwise random message m_A. She then encodes m_A as $E_{n_A}(m_A) = C$ using any of the public-key systems which require the factors p, q of n_A for decoding. (We may, for example use the encoding $E_{n_A}(m_A) = {m_A}^2 \bmod n_A$ of [1], provided that we have a fixed small prefix of m_A

[2] Footnote added during typesetting: Of course, Bob should use a semantically secure public-key cryptosystem for encrypting x under K_B.

to distinguish m_A among the 4 square roots $\bmod n_A$ of $E_{n_A}(m_A)$.) Alice sends $d_A = E_{n_A}(m_A)$ to Bob in a signed message.

Bob follows the same steps using S_B and n_B and sends the encoded result to Alice.

Theorem 1. *The above protocol gives, under the assumptions in the Introduction, a solution of the Exchange of Secrets Problem. The probability that neither side will obtain the other's secret is* $1/4$.

Proof. We omit the proof that the signed messages exchanged between Alice and Bob, and the indelible incorporation of S_A and S_B in the files, provide each participant with a provable case against the other, if the other one cheated.

It is clear that if either Alice or Bob stop participation in the EOS protocol before the final phase, in which case the other one will also stop, then neither can know the other's secret.

Assume that Alice has given Bob, in the final phase, the encoded secret $d_A = E_{n_A}(m_A)$. If Bob in fact knows the factorization $n_A = p \cdot q$, in which case $\nu_B = 0$, he can decode d_A, finding m_A and S_A. If Bob now uses the password (bit) S_A to read his file, then, by assumption, Alice will know this. Again, by assumption, Bob would attempt reading his file only if he knows S_A with certainty (a mistake will destroy the file). Thus Alice knows that $\nu_B = 0$ and hence that $\varepsilon_B = S_B \oplus \nu_B = S_B$ so that she knows S_B.

If Bob gave Alice $d_B = E_{n_B}(m_B)$ in the final phase, then the above argument applies to yield that if Alice reads her file before Bob, then Bob will know S_A.

Thus, if either Alice or Bob reads her or his file, the other one will know the password for his or her file.

The probability, when the protocol was completed, that neither one knows the other's secret is $(1/2)^2 = 1/4$. ■

Remark 1. *In the case that the exchange of secrets has not been effected, it is not possible to iterate the procedure. One participant, say Alice, may actually know S_B after the first round but deliberately not access her file until after the second round. Bob may not know whether ν_A was 0 in the first or second round and then will not be able to read his file.*

Remark 2. *The probability of success of the EOS protocol can be enhanced by modifying the oblivious transfer of information subprotocol. After receiving n_A from Alice, Bob chooses two numbers $x, y \leq n_A$ and gives Alice the squares $x^2, y^2 \bmod n_A$. Alice gives Bob two square roots $x_1, y_1 \bmod n_A$ of x^2 and y^2 respectively. Now Bob has a probability $3/4$ of knowing the factorization $n_A = p \cdot q$.*

When Bob gives Alice $\varepsilon_B = S_B \oplus \nu_B$, she knows that with probability $3/4$, $\varepsilon_B = S_B$. Since we assume that Alice is determined to guarantee that her file will not be erase, it still follows that she will not use ε_B as the password. Rather, as before, she will wait until she either can read S_B by deciphering $E_{n_B}(m_B)$, or can infer $\nu_B = 0$ from the fact that Bob has accessed his file.

The above double iteration of the oblivious transfer of information is also effected from Bob to Alice. The rest of the EOS protocol is as before.

Each participant has now just a $1/4$ probability of not knowing the factorization of the other's one-time key. Thus the probability of non-termination of the EOS protocol is $(1/4)^2 = 1/16$.

There is a limit beyond which the above enhancement cannot be carried. If, for example, the oblivious transfer subprotocol is modified so that $\Pr[\nu_B = 0] \sim 1/32{,}000$ then $\Pr[\varepsilon_B = S_B] =$

$1 - 1/32{,}000$. Now there is a real temptation for Alice to halt the protocol after receiving ε_B, and use ε_B as the password to her file.

3 Conclusion

Let us mention some problems for further research.

The oblivious transfer of information subprotocol is valid even without any of the assumptions we made in order to make EOS feasible. What other applications can one find for this subprotocol?

Can any of the assumptions we made be relaxed or eliminated without losing the possibility of EOS?

Is it possible to construct an EOS protocol which will always terminate, or can one prove that the non-zero probability of non-termination is essential?

Bibliography

[1] Michael O. Rabin. *Digitalized Signatures and Public-Key Functions as Intractable as Factorization*, MIT/LCS/TR-212, 1979.