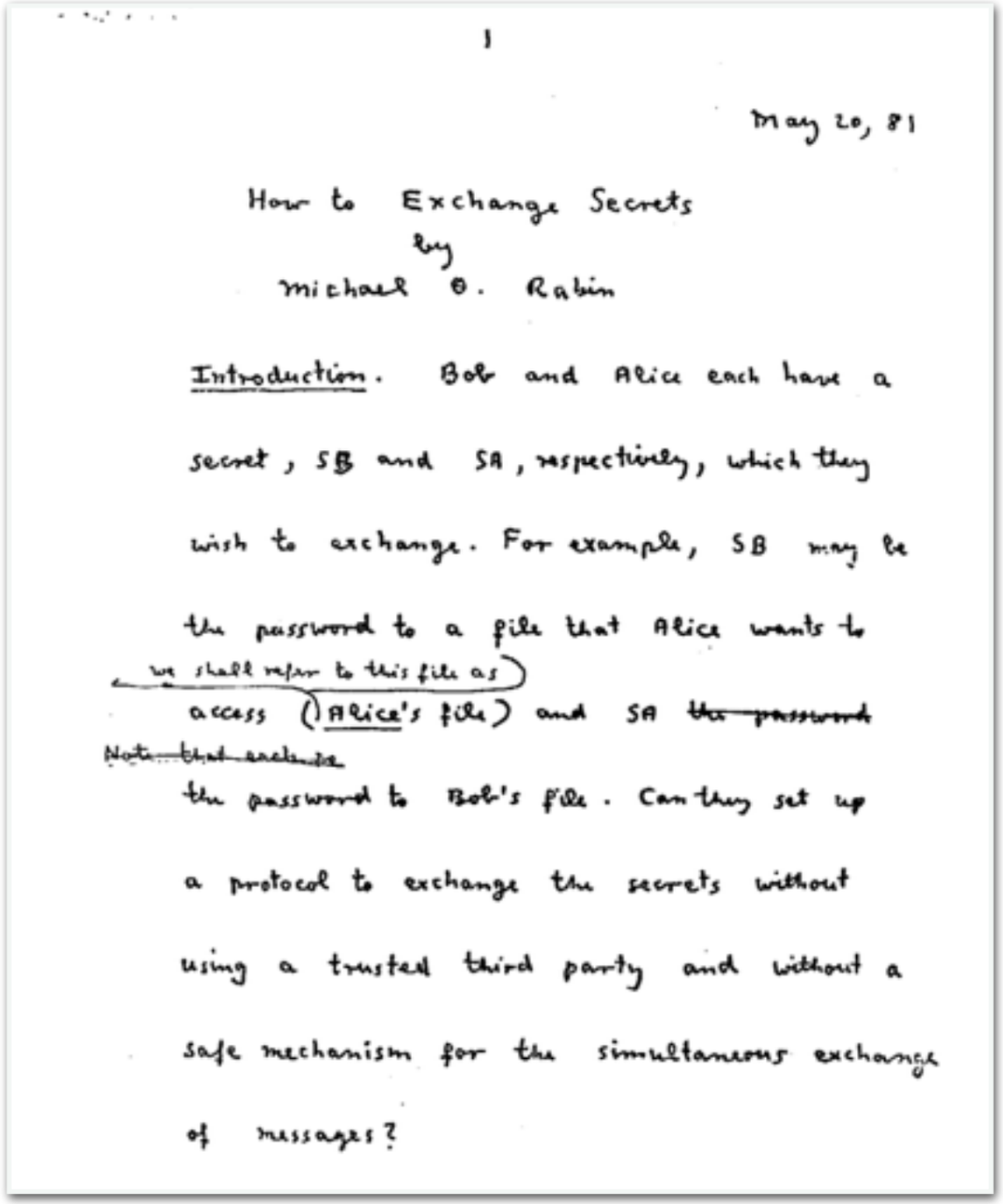
Oblivious Transfer from Noisy Channel

Longze Chen

First OT

Michael O. Rabin



1

May 20, 81

How to Exchange Secrets
by
Michael O. Rabin

Introduction. Bob and Alice each have a secret, S_B and S_A, respectively, which they wish to exchange. For example, S_B may be the password to a file that Alice wants to access (Alice's file) and S_A ~~the password~~ the password to Bob's file. ~~Note that each of~~ we shall refer to this file as 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?

Assumptions

- Intractibility: Rab81, Yao86
- Can we remove or weaken assumption?

OT from Noisy Channel

- Use Noisy Channel Transfer to Simulate a “Imperfect” Noiseless Channel Transfer

- Reduce from OT?

- Reduce from Weaker Form of OT?

In 1-2-OT: Alice (Sender) has two secret bits x_0 and x_1. Bob (Receiver) has the selection bit b. After running the protocol, Bob will learn x_b, Bob can correctly guess x_{1-b} with probability $\frac{1}{2}$, and Alice can correctly guess b with probability $\frac{1}{2}$.

R1

In α-1-2-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has two secret bits x_0 and x_1. Bob has the selection bit b. After running the protocol, Bob will learn x_b, Bob can correctly guess x_{1-b} with probability $\frac{1}{2}$, and Alice can correctly guess b with probability at most α.

R2

In α-A-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has a secret bit x, Bob has no inputs and he knows nothing about x. After running the protocol, either of the two events happens with probability p and $1-p$ respectively. 1. Bob learns the value of x. 2. Bob learns nothing. Furthermore, he knows which is the case, and Alice learns nothing. Alice can choose p where $1-\alpha \leq p \leq \alpha$.

R1

Alice has x_0 and x_1 and Bob has b. They run α-1-2-OT n times to simulate 1-2-OT.

1. Alice chooses $2n$ bits, where:

$$r_i^0 = \begin{cases} \text{randomly sampled from } \{0,1\} & \text{if } 1 \le i \le n-1 \\ x_0 \oplus (\oplus_{i=1}^{n-1} r_i^0) & \text{if } i = n \end{cases} \tag{1}$$

$$r_i^1 = r_i^0 \oplus x_0 \oplus x_1, \text{ for all } i \in [1,n] \tag{2}$$

2. Bob chooses n bits c_i, where $\oplus_{i=1}^{n} c_i = b$.
3. Alice and Bob run the α-1-2-OT n times, each time Alice inputs (r_i^0, r_i^1) and Bod inputs c_i.
4. Bob computes $x_b = \oplus_{i=1}^{n} r_i^{c_i}$.

It is easy to know that Alice can correctly guess b with at most $\frac{1}{2} + \frac{(2\alpha-1)^n}{2}$, which is exponentially close to 1-2-OT.

R2

Lets assume both Alice and Bob follows the protocol. Alice has x_0 and x_1 and Bob has b. They run α-A-OT k times to simulate α-1-2-OT.

1. Alice randomly chooses k bits $C_1, C_2, \cdots, C_k$. She send them to Bob through the α-A-OT channel by run the protocol k times.
2. Bob randomly choose two indexes (i_0, i_1), where $i_0 \in \{i|\text{Bob receives } C_i\}$ and $i_1 \in \{i|\text{Bob does not receive } C_i\}$. He sends (i_b, i_{1-b}) to Alice through a clean channel.
3. Alice returns $x_0 \oplus C_{i_b}$ and $x_1 \oplus C_{i_{1-b}}$. Bob will reconstruct x_b but has no information about x_{1-b}.

Since Alice never knows which bit Bob receives according to the protocol, as long as k is large Bob will be able to find proper (i_b, i_{1-b}), this is a perfect simulation of α-1-2-OT.

OT from Noisy Channel

- Use Noisy Channel Transfer to Simulate a "Imperfect" Noiseless Channel Transfer

- A variant of $\boldsymbol{\alpha}$-A-OT which can be reduced to from $\boldsymbol{\alpha}$-1-2-OT similar to R2

> In α-A-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has a secret bit x, Bob has no inputs and he knows nothing about x. After running the protocol, either of the two events happens with probability p and $1 - p$ respectively. 1. Bob learns the value of x. 2. Bob learns nothing. Furthermore, he knows which is the case, and Alice learns nothing. Alice can choose p where $1 - \alpha \leq p \leq \alpha$.

In communication using a noisy channel: when Alice sends a bit x to Bob, x is flipped with probability ρ. Noisy Channel Transfer is a simulation of such noisy communication: Alice has a secret bit x, and Bob knows nothing about x. After running the protocol, Bob receives $x^{'}$, with probability ρ ($x^{'} = x$) he learns x ; with $1 - \rho$, he learns nothing (wrong value). We expect $\frac{1}{2} < \rho < 1$ for the channel to be useful.

In order for Bob to be able to tell whether he received the correct bit, Alice send each bit twice through the NCT. This can be viewed as an variant of α-A-OT. Alice tries to send bit x. If Bob receives 00 or 11, we say that he receives the correct x; otherwise, we say he fails to receive the bit. Obviously, Bob knows which event happens at the end of the protocol. The only difference compared with α-A-OT is that in this protocol, Bob is only ρ^2 certain of the value he received. We denote this transfer as α-A-OT-dirty.

In α-A-OT ($\frac{1}{2} \leq \alpha < 1$): Alice has a secret bit x, Bob has no inputs and he knows nothing about x. After running the protocol, either of the two events happens with probability p and $1 - p$ respectively. 1. Bob learns the value of x. 2. Bob learns nothing. Furthermore, he knows which is the case, and Alice learns nothing. Alice can choose p where $1 - \alpha \leq p \leq \alpha$.

R3

Suppose both Alice and Bob follow the protocol. Alice has x_0, x_1 and Bob has b. They run α-A-OT-dirty k^c times to simulate α-1-2-OT, where c is a constant.

1. Alice randomly chooses k^c bits $C_1, C_2, \cdots, C_{k^c}$. She send them to Bob through the α-A-OT-dirty channel by running the protocol k^c times.
2. Bob receives $C_1^{'}, C_2^{'}, \cdots, C_{k^c}^{'}$. (If B receives 00, let $C_i^{'} = 0$; if B receives 11, let $C_i^{'} = 1$; and if B receives 01 or 10, let $C_i^{'} = \perp$.) Instead of picking two indexes as before, Bob randomly chooses two sets of indexes:

$$I_s : |I_s| = k, \text{and for all } i \in I_s, C_i^{'} \in \{0, 1\} \tag{3}$$

$$I_{1-s} : |I_{1-s}| = k, \text{and for all } i \in I_s, C_i^{'} = \perp \tag{4}$$

, and sends I_s, I_{1-s} to Alice.

3. Alice returns two sets: $W_0 : \{w_0 | w_0 = x_0 \oplus C_{i_0^j}, j \in [1, k]\}$ and $W_1 : \{w_1 | w_1 = x_1 \oplus C_{i_1^j}, j \in [1, k]\}$. Bob computes $X_s = \{x_s | x_s = C_{i_s^j}^{'} \oplus w_s^j, j \in [1, k]\}$ and guesses x_s to be the majority of X_s.

Discussion