TWO PARTY CRYPTOGRAPHY II.

1.) Coin tossing

2.) Basic OT

3.) OT example

Coin tossing from slides (slide 9)

based on square roots

Elements: Alice knows two large primes p, q $(p, q \equiv 3 \bmod 4)$
Bob knows only $n = p \cdot q$

1) Bob chooses $x_B < \frac{n}{2}$ and sends $y = x^2 \bmod n$

2.) Alice calculates 4 solutions $x_i = \sqrt{y} \bmod n$

$\{x_1, x_2, x_3, x_4\}$, with $x_1 < x_2 < x_3 < x_4$

$x_1 < x_2 < \frac{n}{2}$

Bob chose one of these two in step 1.

$\left(\begin{array}{l} x_1 = n - x_4 \\ x_2 = n - x_3 \end{array} \right)$

3.) Alice tries to guess which one Bob chose. => sends x_A

□ → (e.g. by publishing the first bit in which x_1 and x_2 differ)

4.) Bob tells Alice if she guessed correctly (outcome of the coinflip is 1) ← Bob knows the outcome, Alice doesn't

not correctly (outcome of the coinflip is 0)

5.) They publish p, q, x_2

What would happen if Alice disclosed her guess fully?

1.) $x_A = x_B$ (x_1 x_1) then Bob can't cheat (this doesn't help Bob to calculate x_2. He can't lie about Alices guess

2.) $x_A \neq x_B$ (x_2 x_1) then Bob can cheat

$x_1 = x_{1n} x_{1(n-1)} \cdots \boxed{x_{14}} \cdots x_{10}$ (bit expression of x_1)

$x_2 = x_{2n} x_{2(n-1)} \cdots \boxed{x_{24}} \cdots x_{20}$ Same

$\downarrow$ $x_{12} \neq x_{c2}$

In this way Alice discloses her guess (x_1 or x_2), but she

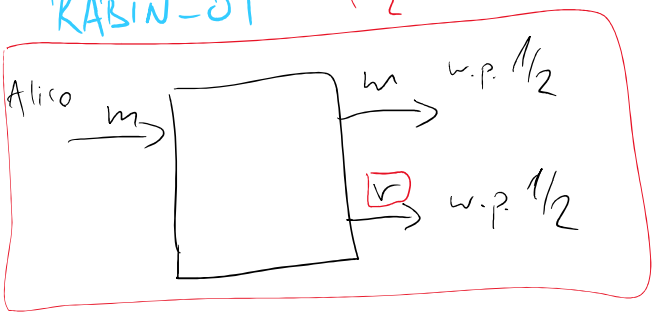
$\cup\ x_{i_1} \neq x_{i_2}$

In this way Alice discloses her guess (x_1 or x_2), but she doesn't disclose their values.

1.) Security against Alice is IT.

2.) Security against Bob is computational.

RABIN-OT ($1/2$-oblivious-transfer)



1.) Alice chooses two primes p, q and sends $n = p \cdot q$ to Bob

2.) Bob chooses x_B and sends $y = x^2 \bmod n$ to Alice

3.) Alice calculates $y_i = \sqrt{y} \bmod n$, chooses one solution at random and sends it to Bob. (x_A)

Information Alice is sending is p and q

Bob learns it if $(x_A \neq x_B \wedge x_A \neq n - x_B)$

this happens w.p. $1/2$

Rabin OT $\Rightarrow$ 1-out-of-2 OT $\Rightarrow$ 1-out-of-k OT

$\Downarrow$

SMC

RABIN OT => 1-out-of-2 OT

1.) Alice sends $3n$ bit randomly chosen bit messages $(x_{11} \ldots x_{3n})$

1.) Alice sends $3n$ bit randomly chosen bit messages $(x_{11}, \dots, x_{3n})$ to Bob via Rabin-OT

2.) Bob chooses a set of n indices of messages he learned I
a set of n indices of messages he did not learn J

3.) Bob sends (I, J) if he wants to learn m_1
Bob sends (J, I) if he wants to learn m_2

4.) Alice recieves (S_1, S_2) and sends

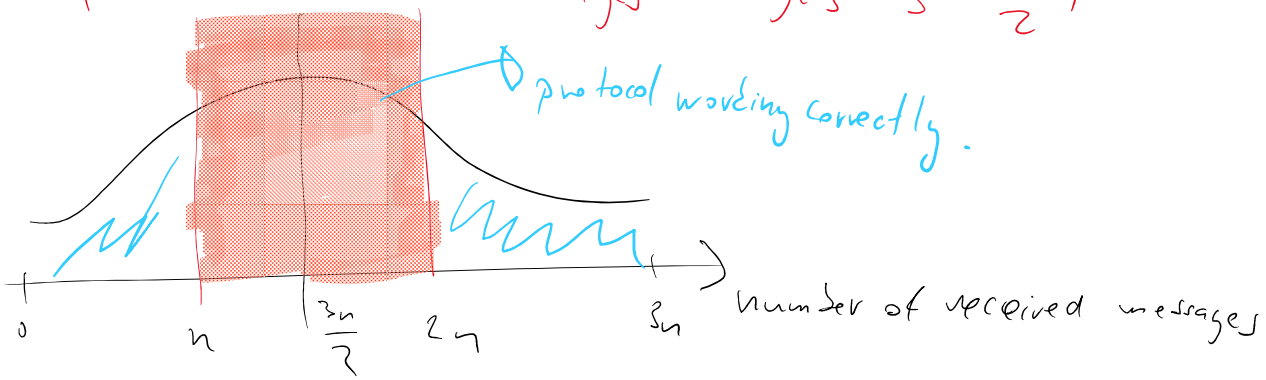
$m_1 \oplus_{i \in S_1} x_i$ and $m_2 \oplus_{i \in S_2} x_2$

n - security parameter.

Higher n means larger probability of security

Bob needs to learn $2n$ messages in order to cheat (then he knows all messages in both I and J)

Expected amount of messages Bob gets is $\frac{3}{2}n$



e.g. Chernoff tail inequalities $\Rightarrow$ Pr of good number of messages increases exponentially with n.

Example: 1-out-of-2 OT to implement online shopping vouchers

Scenario: Alice has an online shop and wants to sell vouchers to her clients, which they can use to pay later

Requirements: 1.) Vouchers are hard to forge

2.) Anonimity → Alice can't match vouchers to clients

1.) Alice creates a message

$x =$ "Voucher for 100 €"

and the voucher is a tuple (x, s) where s is Alice's signature

1.) Copying (x, s) can be used to create infinite vouchers

2.) This is anonymous. Since all vouchers are the same upon their use Alice doesn't know who is paying

2.) Alice has voucher id (counter)

$x_i =$ "Voucher for 100, id=i"

Voucher (x_i, s_i) where s_i is sig of x_i

Alice keeps a database of sold vouchers and after (x_i, s_i) is used to pay it gets removed from the database.

1.) Unforgeable

2.) Not anonymous - she can match vouchers to their buyers

3.) 1-out-of-n OT solution

1) Alice creates a database of valid vouchers

$(x_1, s_1), \ldots, (x_n, s_n)$

Upon Bob's payment she sends him a voucher via 1-out-of-n OT

Problem: She might sell the same voucher twice