Tutorial X

21 November 2019 09:59

TWO PARTY CRYPTOGRAPHY

Alice and Bob do not trust each other and try to cheat each other. There is no external adversary.

→ Bit commitment

→ Oblivious transfer

→ Zero-knowledge proofs (graph isomorphism)

Bit commitment

there are protocols for which the commited value is not a bit but a larger value

1.) Commitment → Alice commits to a bit $b \in \{0,1\}$

2.) Reveal → Alice reveals b to Bob (later in time)

1.) Alice writes b on a paper, locks the paper into a box sends the box to Bob but keeps the key

2.) She sends the key

Binding – Alice can't change the value of b after Commitment

Hiding – Bob can't find the value of b before reveal phase.

Slides: Protocol I

Elements: $n = p \cdot q$ p and q are large primes.

Elements: $n = p \cdot q$ p and q are large primes.

$m \in QNR(\mathbb{Z}_n)$

Calculating $\sqrt{x}$ mod n is computationally hard (without the knowledge of p and q).

Deciding whether $x \in QR(\mathbb{Z}_n)$ is computationally hard (- " -)

1.) Commitment: Alice chooses a random number $x \in \mathbb{Z}_n$ and calculates $C = m^b x^2 \bmod n$
(b is the commited bit)

She sends C to Bob

2.) Reveal phase: Alice reveals b and x, Bob checks whether $C = m^b x^2 \bmod n$

Hiding: if $b=0$ then $C \in QR(\mathbb{Z}_n)$
is if $b=1$ then $C \notin QR(\mathbb{Z}_n)$
Computational

Binding: How can Alice cheat?
is information theoretic (IT) She needs to find two numbers x, y, such that both $(0,x)$, $(1,y)$ lead to a correct reveal phase.

$$m^0 x^2 = m \cdot y^2 = C \bmod n$$

QR ~ QNR

⇓

there are no such numbers x and y

it is impossible to have IT security for both security properties in two party cryptography. The best we can hope for is IT security for the hiding one and computational for the second one.

PEDERSEN SCHEME

p - large prime $(\mathbb{Z}_p^*)$

q - a large prime dividing $(p-1)$ ← exponent algebra is mod q (q is a prime)

g of order q in $\mathbb{Z}_p^*$ $(g^q = 1 \bmod p)$

$h = g^k \bmod p$ $0 < k < q$ (k is not known to any party)

1.) Commitment. commit (b, r), b is a commited bit, r is a random number $0 < r < q$

$$\text{commit}(b, r) = (A, B)$$

$$A = g^r \bmod p$$

$$B = h^{(r+b)} \bmod p$$

2.) reveal Alice reveals b and r, Bob checks whether $A = g^r \bmod p$ and $B = h^{(r+b)} \bmod p$

BINDING is IT

Alice needs to find two numbers x and y such that $(0,x)$ and $(1,y)$ can both open her commitment

$$A = g^x = g^y \bmod p \Rightarrow x = y$$

$$B = h^x = h^{x+1} \bmod p \rightarrow \text{not possible}$$

HIDING is computational

How can Bob calculate b from A and B?

$$b = \log_h B - \log_g A \bmod q$$

$$= r+b - r \bmod q$$

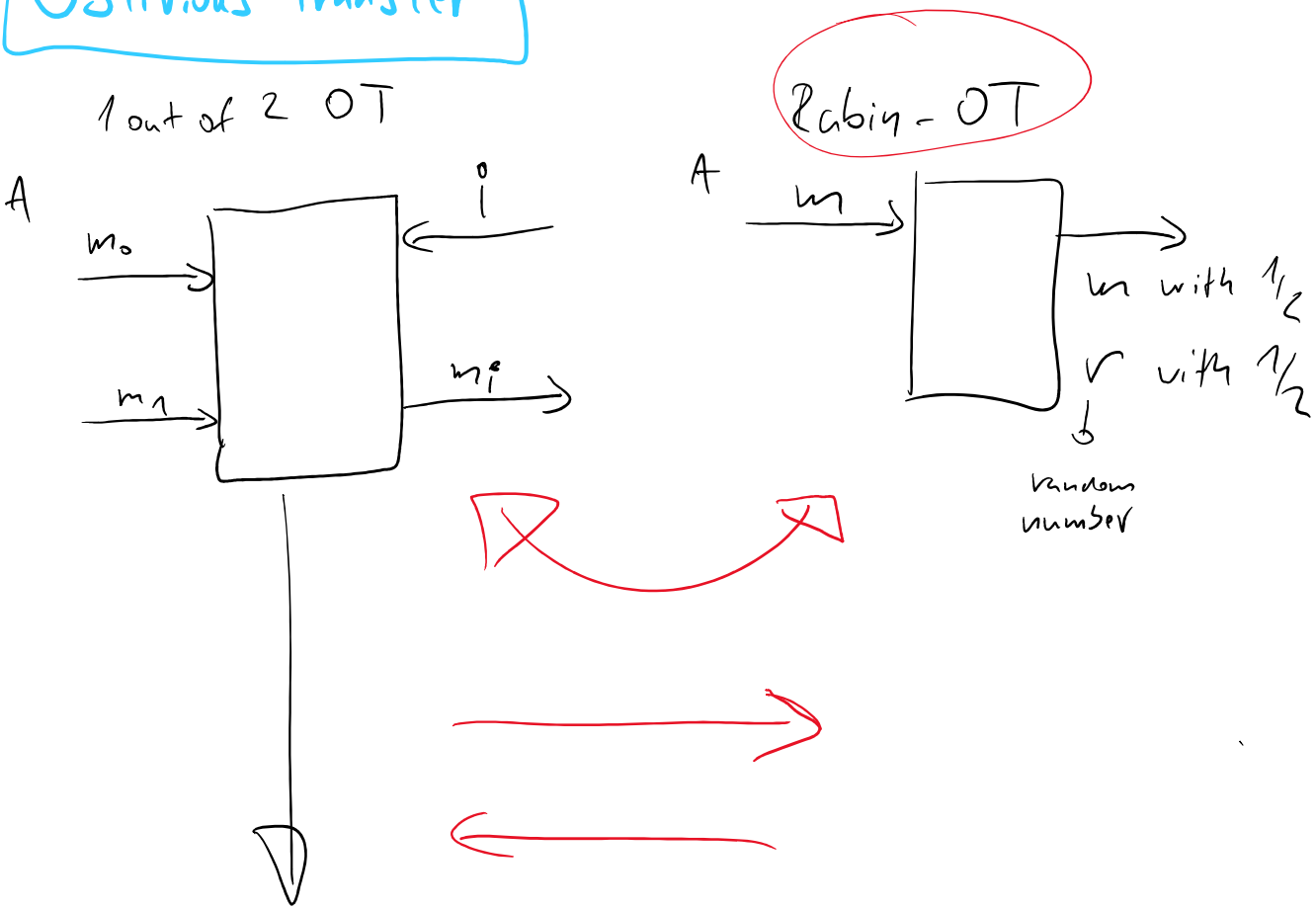
Computationally hard

Why is bit commitment important / interesting?

→ you can build more difficult protocols from it

→ Coin tossing (Slide 19)

Oblivious transfer



Security properties

1.) After the protocol Alice doesn't know i

2.) Bob learns only m_i and knows nothing about $m_{i \oplus 1}$

Protocol using PKE (public key encryption)

1.) Alice generates two PKE keys. She sends public keys p_0 and p_1 to Bob

2.) Bob encrypts a secret key k with a key of his choice (p_0 if he wants to learn m_0 and p_1 if he wants to

choice (p_0 if he wants to learn m_0 and p_1 if he wants to learn m_1) (B is the message he sends)

3.) Alice calculates $A_0 = d_{p_0}(B)$ and $A_1 = d_{p_1}(B)$

then she encrypts m_0 with A_0 ($m_0 \oplus A_0$) and m_1 with A_1 ($m_1 \oplus A_1$)

and sends them to Bob (using one time pad)

4.) Bob can now decrypt m of his choice, the other message is not available.

Security:

Can Alice find Bob's choice?

IT → she can't decide which key was used in step 2

Can Bob find both messages?

COMPUTATIONAL

→ Bob can calculate private keys and then both A_0 and A_1

Why is OT interesting?

OT is a crypto-primitive which can be used to build larger protocols.

→ More general protocols: 1-out of 2, n out of m ...

↝ SMC (Secure multiparty computation)
SFE (Secure function evaluation)
→ holy grail of crypto

↳ Multiple parties (n) parties are trying to calculate $f(x_1, \dots, x_n)$. Each party i holds x_i, which they want to keep secret.

HW 4

Zero-knowledge proofs

Graph isomorphism

What does a graph permutation mean

G_1 and G_2 are isomorphic $\Leftrightarrow$ there is a permutation σ

$G_1 = \sigma \cdot G_2$

$G = [g_{ij}]^n$

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \end{pmatrix} \end{array} \rightsquigarrow \begin{array}{c} \\ 1 \\ 3 \\ 2 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 3 & 2 \end{array} \\ \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ \end{pmatrix} \end{array}$$

$$G_1 = [g_{ij}]_{i,j=1}^{n} \qquad \begin{matrix} 1 \\ 2 \\ 3 \end{matrix}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \rightsquigarrow \begin{matrix} 1 \\ 3 \\ 2 \end{matrix}\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

reorder

$$\begin{matrix} & 1 & 2 & 3 \\ 1 \\ 2 \\ 3 \end{matrix}\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} = G_2$$

$$\sigma = (23) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix} = \begin{pmatrix} a \\ c \\ b \end{pmatrix}$$

$$\sigma \, G_1 \cdot \sigma^{-1} = G_2$$

$$G_1 = \sigma \circ G_2$$

ZK - proof of isomorphism of G_1 and G_2

Alice knows isomorphism σ $G_1 = \sigma G_2$. She wants to convince Bob G_1 and G_2 are isomorphic without revealing anything about σ.

→ 1.) Alice chooses a random permutation ρ and calculates $H = \rho \circ G_1$ and sens H to Bob

→ 2.) Bob randomly chooses G_1 or G_2 and tells Alice (G_i)

3.) if Bob chose G_1 Alice replies with $\boxed{\rho}$

if Bob chose $\boxed{G_2}$ Alice replies with $\boxed{\rho \circ \sigma^{-1}}$

4.) Bob checks whether isomorphism of H and G_i is defined by Alice's reply.

Transcripts: (H, i, π) $\quad H = \pi \circ G_j \rightarrow$ correct transcript

it is hard to find $(H, 1, \pi_1)$

$(H, 2, \pi_2)$