SS2 - exercises

Everyone

August 10, 2022

Abstract

Contents

1 What is the identification scheme?

It is an interactive zero knowledge proof of knowledge of secret key corresponding to the public key (knowledge of a discrete logarithm of a public key). IS is a tuple of (*Init, KeyGen, P, V,* π), where

- *Init* - produces the space of computation, it takes X (security parameter) as input and outputs PAR
- *KeyGen* - key generation function, takes PAR and outputs a pair of (secret key, public key)
- *P* - prover, interactive Turing Machine that takes secret key and interacts with another ITM V in the protocol π
- *V* - verifier, ITM that takes public key and interacts with prover in the protocol π
- π - protocol, in which P interacts with V, the result is that V accepts P (it has secret key) or rejects P (it doesn't have secret key)

Functionalities:

- identification - system identifies its users
- correctness
- security

1.1 Correctness

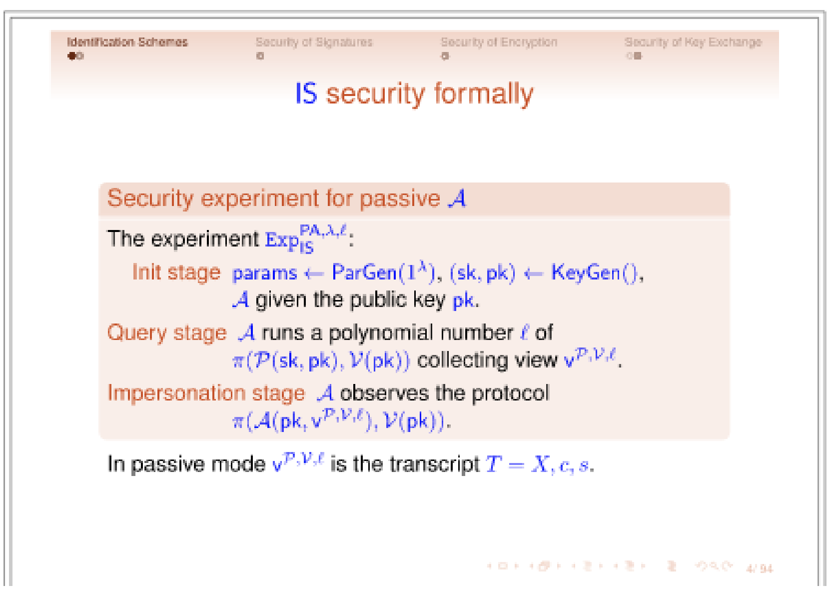
$Pr[PAR \leftarrow \lambda; (sk, pk) \leftarrow Gen(PAR); \pi(P(sk), V(pk)) \rightarrow 1] = 1$

1.1.1 Human readable

Probability of positive verification when prover uses secret key and verifier uses corresponding public key will always be 1.

1.2 Security

Security is defined via the experiment for a probabilistic polynomial time algorithm A

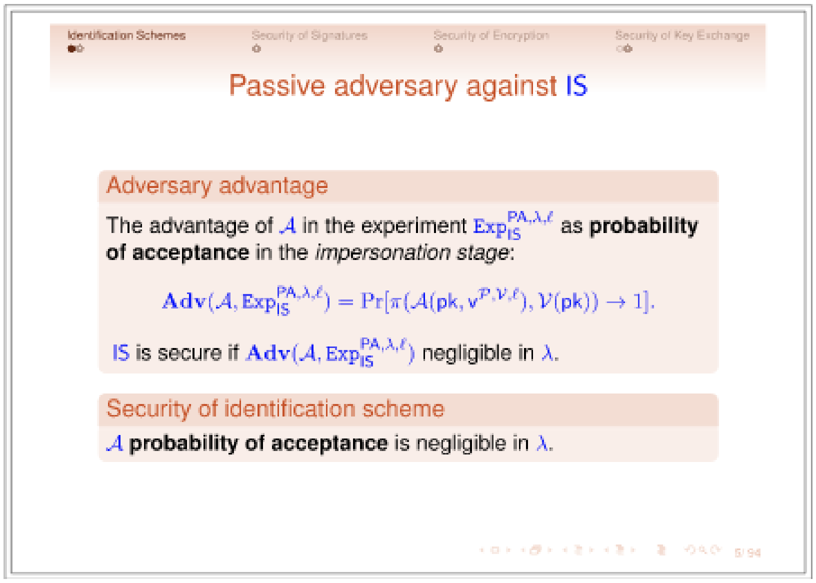


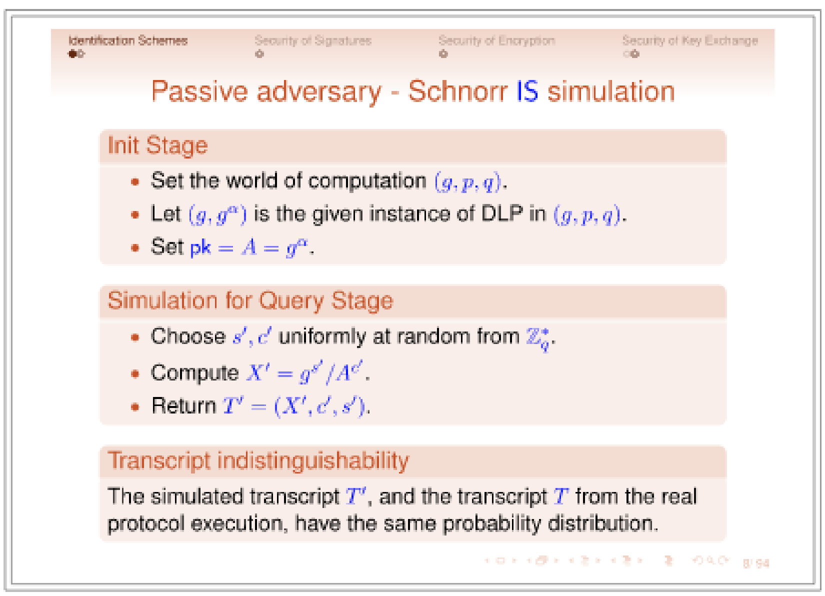
Advantage of adversary $Adv(\mathcal{A}, EXP_\lambda^P) = Pr[\pi(\mathcal{A}(PAR, v = \{T_i\}_{i=1}^l, pk), \mathcal{V}(pk)]$ We say that the IS is secure if $Adv(\mathcal{A}, EXP_\lambda^P)$ is negligible

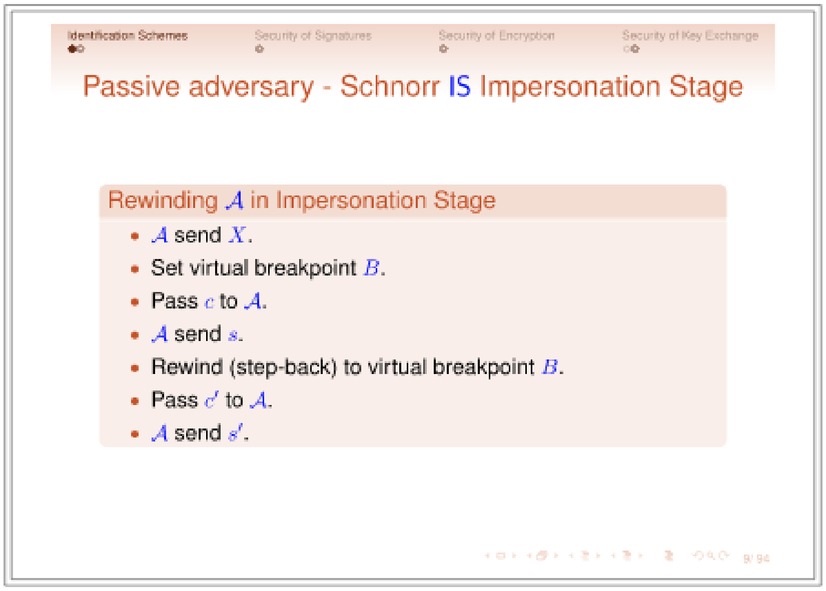
1.2.1 Passive adversary

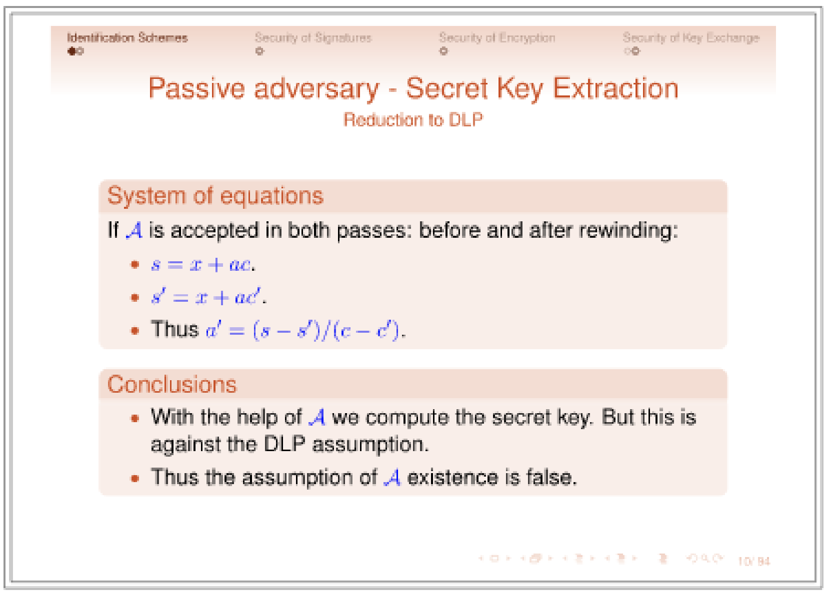
Experiment for passive adversary:

1. $\mathcal{A}$ obtains public key
2. $\mathcal{A}$ observes the protocol n times and obtains it's transcript
3. $\mathcal{A}$ runs the protocol as the prover without possession of secret key









Human readable

1.2.2 Active adversary

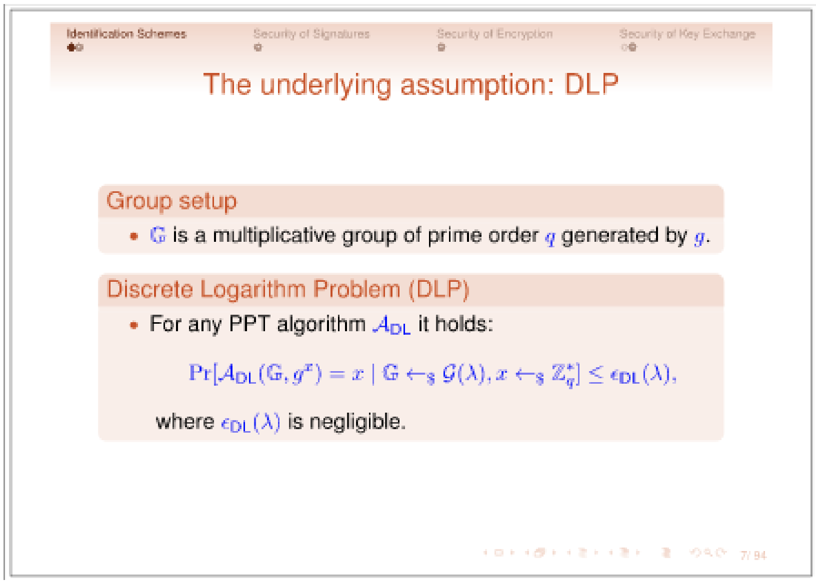
2 What is a Discrete Logarithm Problem

2.1 In human words

Discrete Logarithm Problem is a case when we have a group G. Assuming that G is a multiplicative group of prime order q generated by the generator g and assuming that there exists a random x from Z_q (but we don't know this x, the probability of finding x while having value g^x is negligible.

$$s_k = DL_g p_k$$

2.2 Mathematically



3 What is Signature Scheme?

It is a non interactive zero knowledge proof of knowledge of discrete logarithm from the public key. Signature Scheme is a tuple of (*Init, KeyGen, S, V*), where

- *Init* - produces the space of computation, it takes X (security parameter) as input and outputs PAR
- *KeyGen* - key generation function, takes PAR and outputs a pair of (secret key, public key)
- *S* - signer, non interactive Turing Machine that signs some message using his secret key and produces the signature
- *V* - verifier, NITM that takes the message and signature over this message and, by using the private key, verifies whether it was really signed by the signer

3.1 Correctness

The signature scheme is correct when probability of

- $PAR \leftarrow Init(\lambda)$
- $(sk, pk) \leftarrow KeyGen(PAR)$
- $\sigma \leftarrow Sign(sk, m)$
- $1 \leftarrow Verify(pk, \sigma, m)$

is equal to 1.

3.1.1 Human readable

The signature scheme is correct when having the secret key and signing the message with this secret key, verification of the signature with the public key corresponding to this secret key, will always hold

3.2 Security

The security of signature scheme is defined in such a way that the probability of

- $PAR \leftarrow Init(\lambda)$
- $(sk, pk) \leftarrow KeyGen(PAR)$
- $\mathcal{F}^{O_{sign}}(pk, PAR) \rightarrow (m^*, \sigma^*)$
- $Verify(pk, \sigma^*, m^*) \rightarrow 1$
- $m^* \neq m_0, m_1, ..., m_l = M^*$

is negligible. $(\neq \epsilon(\lambda, l))$

3.2.1 Human readable

We assume that the adversary can initially query a finite number of messages and their signatures. The scheme is secure if the probability that adversary creates a new, different message and it's signature (m^*, σ^*) without possession of a secret key and the message is correctly certified is negligible.

4 What is security model?

The security model is the tuple of algorithms

5 What is the proof of security?

Proofs that there is no better algorithm than brute-force to solve the problem, so the probability of the adversary advantage is negligible.

6 What is hash function?

Hash function is any function that takes data of any size as an input and outputs the data of fixed size. The output value is computed randomly and then the pair (in, out) is saved in so called hash table. For the same input, hash function returns the same output all the time. Good hash function is very fast and minimizes the collisions. Every hash value is computed with the same probability. For two similar input, hash function should return two roughly different outputs.

6.1 Hash function requirements

- mathematical function - results will always be equal if inputs are equal
- fixed output, inputs and outputs strictly defined (compress function)

6.2 Secure hash requirements

- negligible (but non-zero) probability of collision
- small change in input results in large changes of output
- one way function - should not be reversable
- finding input which results in a given hash should not be more efficient than brute-force

6.3 ROM - Random Oracle Model

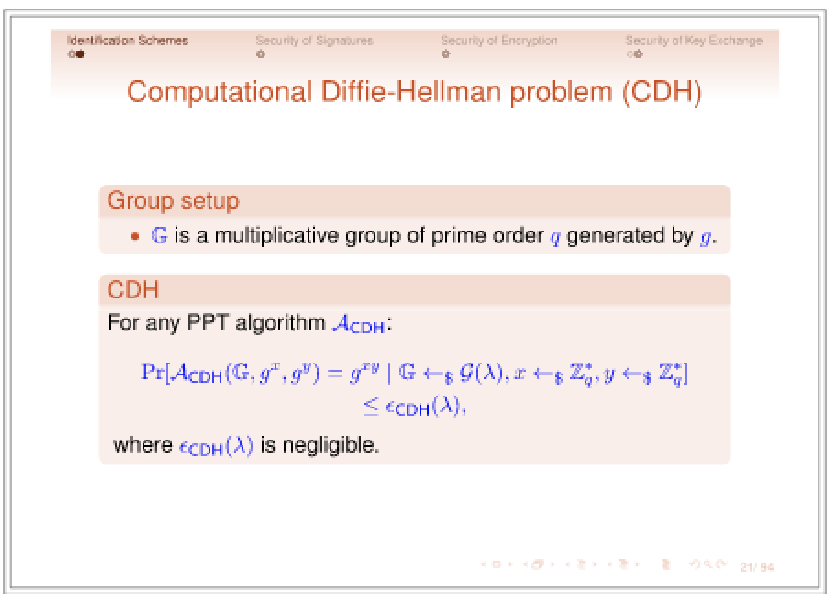
Describes a perfect hash function

- table with mappings (Input, Output) is initially empty
- for each hash query H(I):
- if I exists in table return corresponding hash value
- if I does not exist:
  - generate O = random value from the output group
  - store the (I, O) pair in ROM table
  - return O

7 Diffie-Hellmann

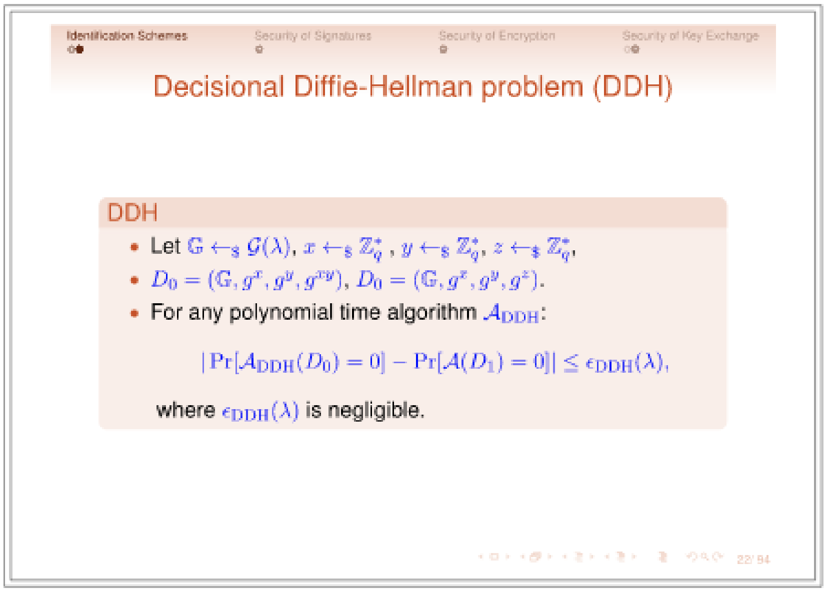
7.1 What is CDH problem?

CDH - Computational Diffie-Hellman
$A = g^{\alpha}$
$B = g^{\beta}$
$C = g^{\alpha\beta}$
$CDH(A, B) = C$



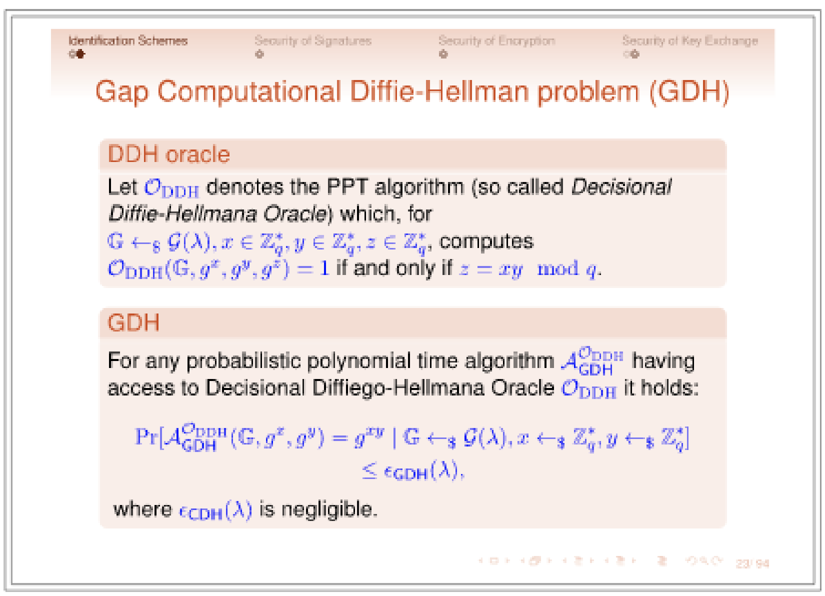
7.2 What is DDH problem?

DDH - Decisional Diffie-Hellman
$A = g^{\alpha}$
$B = g^{\beta}$
$C = g^{\alpha\beta}$
$DDH(A, B, C) = 1$ if $C = g^{\alpha\beta}$ else 0



7.3 What is GDH problem?

GDH - Gap Diffie-Hellman



8 Knowledge proofs

Knowledge proof is a method of proving by one party to another party that one party have knowledge about the certain information without revealing this information and any additional.

8.1 What is IZKP?

Interactive zero knowledge proof requires the interaction between two parties, which means they have to communicate at the same time.

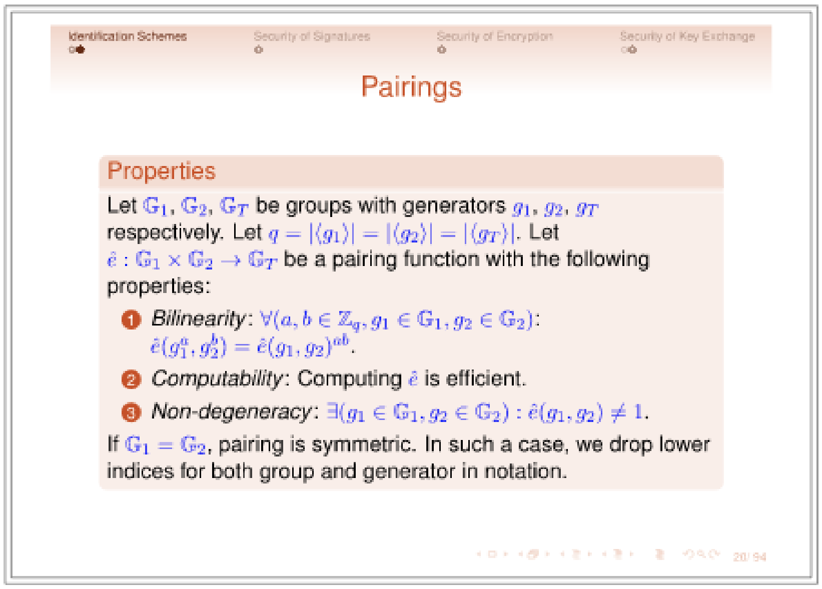
8.2 What is NIZKP?

Non interactive zero knowledge proof does not require any interaction between both parties. It works in such a way that one party performs its computations at first, then, at any time, the second party can verify the incoming data.

9 Goh-Jarecki

TBD

10 Pairing



11 Modified Schnorr

The idea behind the modification is to address the threat that an adversary with the knowledge of c, $s = x + ac$ and a leaked x can compute a static secret a. Therefore, instead of sending s in plain, a prover calculates and sens $S = g^s$, hiding s in the exponent. A new generator g is computed by hashing X and c using function H, that is, $g = H(X|c)$. Even if the ephemeral value x is leaked, the adversary has to solve the instance of Discrete Logarithm problem to obtain value a from S. On the verifier's side, bilinear pairing e is used to check the linear equation $s = x + ac$ in the exponent. The equation $e(S, g) = e(H(X|c), XA^c)$ holds because $e(H(X|c)^{x+ac}, g) = e(H(X|c), XA^c)$.

Why the equation holds:

$$e(H(X|c)^{x+ac}, g) = e(H(X|c), XA^c) \tag{1}$$

$$e(H(X|c)^{x+ac}, g) = e(H(X|c), g^x g^{ac}) \tag{2}$$

$$e(H(X|c)^{x+ac}, g) = e(H(X|c), g^{x+ac}) \tag{3}$$

$$e(H(X|c), g)^{x+ac} = e(H(X|c), g)^{x+ac} \tag{4}$$