# IT University Of Copenhagen

October 24, 2021

Security

Mandatory exercise set II

Name: Vivian Luisa Machindano Seerup

Email: vise@itu.dk

**Implementation details**

The program is written in Java

Contain 2 programs: Alice and Bob

**Folder Content:**

Report of the protocol

code folder: The implementation of protocol

# SECURITY PROTOCOL DESIGN

In this report I am presenting a protocol for a secure communication between Alice and Bob for playing an online dice game in a insecure network.

The next image represents an overview of the technologies and tools I will be using in my implementation.

| GOALS | TOOLS |
|---|---|
| Confidentiality | Asymetric Encryption El Gamal |
| Integrity and Authenticity | Digital Signatures |
| Biding and Hiding | Pederson Commitment Scheme |
| Programming Language | Java, read Readme.txt file |

## Alice:

**Step 1 (Pederson Commitment):** Based on discrete logarithm problem, get h=g^e and compute a commitment "C = ($g^a h^r$)" where "a" is Alice's message and "r" is a random number.

**Step2 (Digital Signatures):** Lets assume that both parts has a valid digital certificate issued by a legitimate CA.

The sender (Alice) with her pair of keys (ska, vka) sign the commitment from step1 using her ska.

**Step3 (Share key and Encryption):** Run El Gamal algorithm to share keys between the two parts and get a "sharedKey = (g^x)^y". Using Elgamal Encryption scheme compute cypherText1 by Encrypting the signed commitment from step2 with the sharedKey, and encrypt a non-signed commitment with the sharedKey let it be cypherText2.

**Step 4 (Send the message):** The sender (Alice) sends the cypherText1 (with encrypted signed commitment), cypherText2 (with encrypted non-signed commitment) and her PK.

**Bob:**

**Step 5 (Decryption of the message):** The receiver (Bob) gets the cypherText1 , cypherText2. After that with the shared key he decryption both “cypherTexts”. Finally he is able to reveal the signed commitment and the non-signed commitment.

**Step 6 (Digital signature correctness):** Bob applies Alice’s vka to the signed commitment and compares with the non-signed commitment this should return valid or invalid confirming the authenticity and integrity of the message.

**Step 7( Bob dice roll answer):** Bob finds “b”. After that he signs “b” and encrypts. Let it be cypherTextB1 . Finally he encrypts a non-signed “b” let it be cypherTextB2 as well using his skb from the digital signatures pair of keys (skb, vkb) .

**Step 7( Bob send message):** Bob sends cypherTextB1 and cypherTextB2 to Alice.

**Alice:**

**Step 8 (Decryption and digital signatures correctness):** Alice gets the “cypherTexts” from Bob decrypts using the shared key, and with Bob’s vsk she runs the digital signature correctness similar to step 6. Alice computes the dice **a_xor_b mod t.** In the next steps she will help Bob open the commitment to reveal her “a”.

**Step 9 (Send the Key and claimed answer):** At this point Alice sends to Bob the key (r) to open the commitment and the claimed “a”. Lets assume that each time anything is sent on the network it must be first signed and than encrypted , after that the receiver should decryption and run the digital signature correctness algorithm to ensure it’s Integrity and authenticity as seen at the above steps.

**Bob:**

**Step 10 (Opening commitment):** After receiving (a, r) from Alice Bob computes $g^{a}h^{r}$ .
next he checks if the computation is equal to the commitment sent by Alice earlier.

If the result returns true he reads the message otherwise discard. Finally he computes the dice **a_xor_b mod t.**

## CONCLUSION

Although making assumption is a potential vulnerability in terms of security, I strongly believe that this protocol is safe and answers the concerns listed at the problem definition.

The protocol makes possible for Bob and Alice play the Dice game in a insecure network and with both part not trusting each other. It also presents solutions for confidentiality, Integrity and authenticity.

To ensure **confidentiality** I use Asymmetric Encryption Elgamal, which is very efficient when constructed over any group where DDH assumption holds. Although for my implementation I used very small number lets assume that in practice should be a bigger number.

In order to ensure **Integrity** and **Authenticity** I used digital signatures, because it helps eliminate for example, “the man in the middle problem”. If the message is corrupted the receiver will know by running the correctness algorithm. Any change made to the signed data invalidates the whole signature. Is possible to confirm the identity by checking if the sender has a valid digital certificate and by than we a sure that the secret key must belong to the sender and only the sender vsk (public key) could unsigned the message.

Finally to solve the problem of both **players not trusting each other** I use Pederson Commitment scheme, which has two powerful properties of Biding and Hiding making sure that the receiver can check whether is opening the same message that was sent earlier and it guarantees that the receiver will learn nothing of the message until the opening phase.

So at this point we have a secure network knowing that the communication will be confidential, we are able to check the identity of the sender and will know if the message gets corrupted in the way.

Finally in terms of the game by the time Alice sends her commitment to Bob he will not be able to see the content. After getting Bob’s b at this point Alice can compute the dice. However she can’t change

her answer because Bob will compare the new “a” with the “a” in the commitment and realize that Alice is not being fair. The diagram below shows this interaction.