University of Stuttgart
Institute of
Information Security

Security and Privacy
Summer Term 2018
Prof. Dr. Ralf Küsters

Homework 5

Submission into the *red SEC mailbox* in front of Room 2.041 until July 17, 2018, 14:00.

General Notes

- If you encounter difficulties, you SHOULD[1] ask the teaching assistants (see ILIAS for contact information).
- To solve the homework, you SHOULD form teams of 3 people.
- Your team size MUST NOT exceed 3 people.
- You MUST submit your homework on paper (one submission per team).
- You are free to choose whether you write your solutions in German or in English.
- If your submission contains multiple sheets, you MUST staple them.
- Each sheet of your submission MUST include all team member's names and matriculation numbers.
- If you do not adhere to these rules, you risk losing points.

Problem 1: Schnorr's protocol - special honest verifier zero-knowledge (4 points)

Recall Schnorr's protocol with parameter t for proving knowledge of the discrete logarithm (see also previous homework sheet):

$$R_{DL} = \{((\mathcal{G}, q, g, h), w) \mid q \textit{ prime}, \ \mathcal{G} \textit{ cyclic group}, \ |\mathcal{G}| = q, \ g, h \in \mathcal{G}, \ g \neq 1, \ g^w = h\}.$$

Common input: $(\mathcal{G}, q, g, h)$ where $q \geq 2^t$

P:

1. Compute $w \in \mathbb{Z}_q$ such that $g^w = h$
2. Choose $r \xleftarrow{\$} \mathbb{Z}_q$
3. Compute $a = g^r$
4. Send a
5. Receive e
6. Compute $z = r + e \cdot w \bmod q$
7. Send z

V:

1. Receive a
2. Choose $e \xleftarrow{\$} \{0,1\}^t$
3. Send e
4. Receive z
5. **if** $|\mathcal{G}| = q, q$ prime, $g, h \in \mathcal{G}, g \neq 1$, and $g^z = a \cdot h^e$ **then** output 1, **otherwise** output 0.

Show that Schnorr's protocol fulfills the "special honest verifier zero-knowledge" property of sigma protocols.

Hint: Given an input $x \in L_{R_{DL}}$ and a challenge e, you have to build a simulator that outputs an accepting transcript (a, e, z) with the correct distribution. Your simulator should start by choosing the response z in a suitable way and then compute a commitment a from z and e such that (a, e, z) fulfills all requirements. Do not forget to show that the distribution is correct!

[1] SHOULD, MUST, and MUST NOT are used as defined in RFC2119.

Problem 2: Homomorphic properties of algorithms (4 points)

In this task we want to explore homomorphic properties of algorithms. For this purpose we consider the Pedersen commitment scheme which we have introduced in the lecture. As we will show in this task, this algorithm is in fact homomorphic, i.e., it is possible to multiply two commitments on messages v_0 and v_1 to obtain a commitment on the message $v_0 + v_1$. More precisely, you have to show the following:

Let $p = (\mathcal{G}, q, g, h)$ be some fixed parameters of the Pedersen commitment scheme. Let $v_0, v_1 \in \mathbb{Z}_q$ be two input messages and let $r_0, r_1 \in \mathbb{Z}_q$ be randomness for the Pedersen commitment scheme. Then the following holds true:

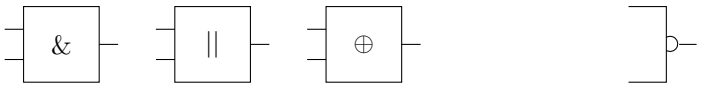
$$\mathsf{com}^{r_0+r_1}(p, v_0 + v_1) = \mathsf{com}^{r_0}(p, v_0) \cdot \mathsf{com}^{r_1}(p, v_1)$$

Problem 3: Building circuits for functions (4 points)

To compute arbitrary functions via a secure multi-party computation protocol, these functions usually have to be represented as a circuit (of either boolean or arithmetic operations). In the lecture, we introduced Yao's garbled circuits for computing arbitrary functions represented as a boolean circuit.

Consider the following function f: Let $a, b \in \{0,1\}^2$ be two bit strings of length two. Then $f(a, b)$ outputs 1 if $a > b$ and outputs 0 otherwise (where a and b are interpreted as binary numbers).

Give a circuit that computes f. Your solution must be in the form of a drawing of your circuit. You MUST use only the following gates and notation in your circuit:



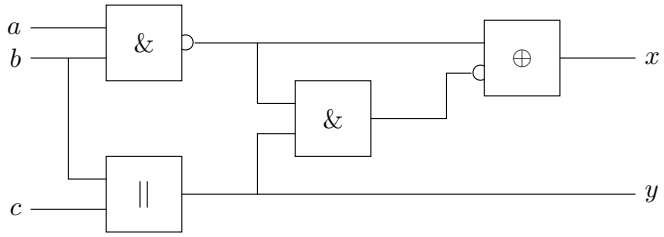
AND gate | OR gate | XOR gate | Negate an output (or input) of one of the gates by drawing a circle.

Each of the gates is defined in the usual way, where inputs and outputs are single bits. Clearly mark the input and output wire(s) of your circuit. Also mark which bit of the inputs a and b is assigned to which input wire.

Remark: Note that you have to enter and process the inputs a and b bitwise. Use a_0 to denote the first bit of a and a_1 for the second input bit. Analogous for b_0 and b_1. Use c to denote the output bit.

Problem 4: Garbled circuits (4 points)

Create a garbled circuit from the following circuit with input wires a, b, c and output wires x, y (see the previous problem for the notation used here):



Your solution must be a in the form of a drawing analogous to the one on Slide Set 09, Slide 11. That is, draw a version of the circuit where each gate contains a table of the garbled version. Also annotate all wires w with the keys k_w^0 and k_w^1 that correspond to the respective bits on that wire. The notation of the keys should be analogous to the lecture to make it directly obvious which key corresponds to which bit.

Problem 5: 51%-Attack on Bitcoin (4 points)

In the lecture we mentioned the 6-block rule, i.e., Bitcoin users should wait until 6 additional blocks have been generated after a transaction before they consider this transaction to be "unchangable". We want to evaluate whether/how much this rule protects against a 51%-attack.

(a) Assume that the hash-power of honest miners represents 49% of the overall hash-power in the Bitcoin network. Assume that the honest participants find a block on average every 20 min, each of them extending the same chain without generating any forks.

The attacker has control over the other 51% of the hash-power in the Bitcoin network. The attacker decides to run a 51% attack on the current longest chain to overwrite/change the transactions in the last 6 blocks of the chain. That is, he starts a new chain branching off the 7-th last block and tries to overtake the honest chain that is already 6 blocks longer.

How long does the adversary need (on average) until his chain has the same length as the "honest" chain?

Note: After this point in time, the attacker will be able to overwrite the honest chain.

(b) At the start of 2018, generating one Bitcoin block in Germany cost about 100000 € in terms of electricity. For each generated block, the miner earned 12.5₿ worth about 6000 € each at that point in time. How much money does the 51% attacker from the previous part of this problem have to earn via double-spending to make a profit?

(c) There are several ways to make a 51% attack more efficient. Come up with one strategy that accelerates a 51%-attack and describe it (briefly) on an intuitive level.