Cryptology I (spring 2020) Dominique Unruh

Exercise Sheet 8

Out: 2020-05-14 Due: 2020-05-22

Problem 1: Yao's Garbled Circuits

(a) One application of secure function evaluation is the so-called "dating problem". Two parties A and B are wondering whether they should date, but none of them wishes to admit their interest unless they know that the other side is interested, too. The solution is to perform a two-party computation on their inputs a and b (where a and b are a bit corresponding to whether A or B wishes to date) that returns $f(a, b) := a \wedge b$. (We ignore the fact that this is silly: by suggesting to run this SFE, one already expresses interest. But we could consider a case where some app is doing this automatically with all potential matches – a privacy preserving dating app.)

A and B want to use Yao's Garbled Circuits for this. (We ignore the fact that that protocol only has security against passive adversaries.) That is, A will have to pick some circuit C, and B some input x for that circuit. What should C and x be in this concrete case (i.e., how to convert a and b into C and x) so that B learns $f(a, b)$?

(b) **(Bonus problem)** Implement part of Yao's protocol. That is, implement a function `make_gate` that garbles a single gate. (Given four input keys, and four messages.) And a function `eval_gate` that recovers the message m_{ij} given the corresponding keys.

Use the template in `yao-gate.py`.

Problem 2: Zero-knowledge proofs

Below are several variations of the graph isomorphism zero-knowledge proof. Each of them is broken: either it is not a proof (lacks soundness[1]), or it is not complete (does not succeed even if P and V are honest), or it is not zero-knowledge (leaks something about the witness). In each case, say which property is missing, and why (e.g., how to attack, how to compute the witness, in which case the protocol does not give the right output, etc.)

Original graph isomorphism protocol (the one that has completeness, 1/2-soundness, and zero-knowledge):

- Input of P and V: Two graphs G_1, G_2, supposedly isomorphic.
- Input of P: An isomorphism $\phi : G_1 \to G_2$.
- P picks uniformly random permutation ψ and computes $H := \psi(G_1)$.

[1] We count 1/2-soundness as OK. But something like 1-soundness would not be.

- P sends H to V.
- V picks $i \in \{1, 2\}$ uniformly and sends i to P.
- P sends ψ_i where $\psi_1 := \psi$ and $\psi_2 := \psi \circ \phi^{-1}$.
- V checks whether $\psi_i(G_i) = H$.

Changes with respect to this protocols are boldface below.

(a) Protocol:

- P picks uniformly random permutation ψ and computes $H := \psi(G_1)$.
- P sends H to V.
- V **sets** $\boldsymbol{i := 1}$ and sends i to P.
- P sends ψ_i where $\psi_1 := \psi$ and $\psi_2 := \psi \circ \phi^{-1}$.
- V checks whether $\psi_i(G_i) = H$.

(b) Protocol:

- P picks uniformly random permutation ψ and computes $H := \psi(G_1)$.
- P sends H to V.
- V picks $i \in \{1, 2\}$ uniformly and sends i to P.
- P sends ψ_i where $\psi_1 := \psi$ and $\boldsymbol{\psi_2 := \psi}$.
- V checks whether $\psi_i(G_i) = H$.

(c) Protocol:

- P picks uniformly random permutation ψ and computes $H := \psi(G_1)$.
- P sends $\boldsymbol{\psi}$ **and** H to V.
- V picks $i \in \{1, 2\}$ uniformly and sends i to P.
- P sends ψ_i where $\psi_1 := \psi$ and $\psi_2 := \psi \circ \phi^{-1}$.
- V checks whether $\psi_i(G_i) = H$.