Foundations of Cryptography: Course Projects

Deadline: Dec. 30, 2016

Project: Secure Two-Party Computation (40 points)

Problem: *Secure two-party computation* protocols allow two parties, Alice and Bob, to jointly compute a function $f(x, y)$ of their inputs x and y such that at the end of the protocol execution:

(a) both party learn the output $f(x, y)$; and

(b) each party learns no more information about the other party's input than what can be deduced from its own input and the output, that is,

- Alice learns no more information about y than what can be deduced from its own input x and the output $f(x, y)$, and
- Bob learns no more information about x than what can be deduced from its own input y and the output $f(x, y)$.

In an *honest-but-curious* model of secure two-party computation, the parties Alice and Bob always follow the instructions of the protocol. In lecture 20, we presented a construction of secure two-party computation in the honest-but-curious model, based on *oblivious transfer* and *garbled circuits*.

Let $x = x_1x_2 \cdots x_n, y = y_1y_2 \cdots y_n \in \{0,1\}^n$ be two n-bit strings (numbers). Let $f(x, y)$ be a function defined as below:

$$f(x, y) = \begin{cases} 1, & x \geq y; \\ 0, & x < y. \end{cases}$$

In this project, you are asked to design and implement a secure protocol for Alice (with input x) and Bob (with input y) to jointly compute the function $f(x, y)$ in the honest-but-curious model.

Guidelines:

- You can design and implement a protocol for a specific integer $n \geq 4$ or a protocol that takes $n \geq 4$ as parameter (i.e., n is not fixed).
- The protocol you design and implement should satisfy the requirements (a) and (b).
- Submit to zhanglf@shanghaitech.edu.cn
  - (15 points) a short report (.docx or .pdf, prefer English) that clearly explains your design (e.g., how to realize the oblivious transfer, how to realize the private-key encryption, how to represent the f as a circuit, how to construct a garbled circuit, how to evaluate the garbled circuit) and implementation, and
  - (25 points) a set of programs that implement your protocol.
- The evaluation is based on the functionality, security and efficiency of your design and implementation. You have the opportunity to demonstrate your implementation if it does not work on our test platform.

- Recommended reading material (m1.pdf):
  - Yehuda Lindell and Benny Pinkas: A Proof of Security of Yao's Protocol for Two-Party Computation, 2008.

  You can also consult any other materials in the process of designing and implementing your protocol.

Bonus Project: A Dedicated Attack of MD4 (10 points)

Problem: The MD4 Message-Digest Algorithm (https://tools.ietf.org/html/rfc1320) is a cryptographic hash function that maps any finite bit string to a digest of 128 bits. This algorithm has influenced many later hash designs such as MD5 and SHA-1. However, it has been shown not collision-resistant since 1995.

In lecture 13 we presented a generic *birthday attack*, when applied to any hash function with l-bit digests, that can find a collision with probability $\geq 1/2$ using $O(2^{l/2})$ hash computations. In this project, you are asked to implement a dedicated birthday attack of simplified MD4.

Guidelines:

- The description of the attack can be found from the following material (m2.pdf):
  - Serge Vaudenay, "A Dedicated Attack on MD4", page 74-77, from the book "A Classical Introduction to Cryptography: Applications for Communications Security", 2006.
- Submit to zhanglf@shanghaitech.edu.cn
  - (10 points) a set of programs that implement the attack.
- The evaluation is based on your understanding of the implementation. Getting some of the 10 bonus points will be difficult. You need to provide a completely correct implementation. You will be asked to explain the attack idea, in order to show that you really understand the attack and implemented it by yourself.