Institut for Matematik og Datalogi
Syddansk Universitet

April 27, 2005
JFB

Cryptology – F05 – Lecture 11

Lecture, April 15

We finished chapter 4, skipping section 4.3.1 and the first four sections of chapter 7.

Lecture, April 29

We will cover undeniable signatures following that handout given in class and begin on zero-knowledge (from the handout).

Lecture, May 6

Problem session May 4

Bring your second assignment. We will use time at the end to go over the last problem from that assignment.

We will also discuss the programming assignment.

1. Problem 6.21d in the textbook.
2. In the verification protocol for undeniable signatures (in the textbook), the verifier chooses randomly two values e_1 and e_2. Why are there two values? Why not just let $e_2 = 0$ always?
3. Give a protocol for digital signatures in which the verification (which can be shown to the judge) does not reveal to the judge the contents of the document which was signed.

4. Some applications are sensitive to *replay attacks*, where an adversary takes a copy of an original signed message and sends it again later. (For example, it should not be possible to repeat a request to transfer money from one bank account to another.) Design a protocol (using signatures) to prevent replay attacks.

5. According to Ivan Damgård, the essence of SSL (authentication between a server S and a client C) is as follows:

   (a) C sends a hello message containing a nonce (a random challenge) n_C.

   (b) S sends a nonce n_S and its certificate *Cert(S)* (issued by a certification authority and containing the public key K_S of S.)

   (c) C verifies *Cert(S)* and chooses a pre-master secret pms at random. C sends $E(K_S, pms)$, its certificate *Cert(C)* to S, and its signature sig_C on the concatenation of n_C, n_S, and $E(K_S, pms)$.

   (d) S sends C a MAC on all messages sent so far in this protocol, using pms as the secret key.

   (e) C verifies the MAC. IF OK, it send S a MAC on all messages sent so far in this protocol.

   (f) Use a shared function to compute keys for authentication and encryption from n_S, n_C, and pms.

   In this protocol, how does S authenticate itself? How does C authenticate itself. Why do the keys depend on n_S and n_C, instead of just pms? Is it important that C actually send a MAC at the end, or would OK be enough?

Assignment due Friday, May 20, 10:15 AM

Note that this is part of your exam project, so it must be approved in order for you to take the exam in June, and you may not work with others not in your group. If it is late, it will not be accepted (though it could become the assignment you redo). You may work in groups of two or three.

Let n be the product of two large primes, and let $y_i \equiv x_i^2 \pmod{n}$ for $1 \leq i \leq k$ be quadratic residues in the group $\mathbb{Z}_n^*$. Assume the Prover knows the values $x_1, x_2, ..., x_k$ and that both the Prover and the Verifier are given the values n and $y_1, y_2, ..., y_k$. To show that $y_1, y_2, ..., y_k$ are all quadratic residues, one can execute the following protocol $\lceil \log_2 n \rceil$ times.

Prover		Verifier
		Choose random subset R of $\{1, ..., k\}$.
	$\xleftarrow{R}$	
Choose random $r \in \mathbb{Z}_n^*$. Let $z \equiv r^2 \cdot \prod_{i \in R} y_i \pmod{n}$.		
	$\xrightarrow{z}$	
		Choose a random $b \in \{0, 1\}$.
	$\xleftarrow{b}$	
Let $s \equiv r \cdot (\prod_{i \in R} x_i \pmod{n})^b$.		
	$\xrightarrow{s}$	
		Check that $s^2 \equiv z \cdot (\prod_{i \in R} y_i \pmod{n})^{b-1}$ and $s \in \mathbb{Z}_n^*$. If not, reject and halt.

You may use the following fact:

Fact: Let $S = A \cup B$, where A and B are disjoint sets. Suppose R is a randomly chosen subset of S, i.e. each element of S is chosen with probability $\frac{1}{2}$, independently of all other choices. Then, if $A \neq \emptyset$, the probability that an odd number of elements from A is chosen is $\frac{1}{2}$.

a. Suppose that at least one of the y_i is a quadratic nonresidue. What distribution do the values for z have when the Prover follows the protocol?

b. Prove that the above protocol is an interactive proof system showing that all of the y_i are quadratic residues.

c. Suppose that all of the y_i are quadratic residues. What distribution do the values for z have when the Prover follows the protocol?

d. Prove that the above protocol is perfect zero-knowledge.