Institut for Matematik og Datalogi
Syddansk Universitet

April 21, 2008
JFB

Cryptology – F08 – Week 11

Lecture, April 16

We covered sections 8.3 and 7.6 (actually from some notes).

Lecture, April 22

We will cover the Blum-Goldwasswer Public-key Cryptosystem from section 8.4 and begin on zero-knowledge (from the notes by Ivan Damgård and Jesper Buus Nielsen, available through the course's homepage).

Lecture, April 24

We will continue with zero-knowledge from the notes and slides.

Lecture, April 30

We will continue with zero-knowledge, from the notes and slides, and begin on chapter 9 in the textbook.

Problem session April 29. Note no class on May 1.

1. Do exercises 8 through 10 in the notes by Damgård and Nielsen.

2. Let p be a large prime and let $x, y \in Z_p^*$. Suppose that $x = g^k \pmod{p}$ and $y = h^k \pmod{p}$. Assume the Prover knows the value k and that both the Prover and the Verifier are given the values p, g, h, x, and y. To show that the discrete logarithm of x with respect to g is equal to the discrete logarithm of y with respect to h, one can execute the following protocol:

Prover		Verifier
Choose random $s \in Z_{p-1}$. Let $u \equiv g^s \pmod{p}$ $v \equiv h^s \pmod{p}$.	$\xrightarrow{u, v}$	
		Choose a random $c \in Z_{p-1}$.
	$\xleftarrow{c}$	
Let $e \equiv c \cdot k + s \pmod{p-1}$.		
	$\xrightarrow{e}$	
		Check that $g^e \equiv x^c \cdot u \pmod{p}$ and $h^e \equiv y^c \cdot v \pmod{p}$. If so, accept. Otherwise, reject.

a. Prove completeness for the above protocol, showing that (assuming that both the Prover and Verifier follow the protocol) the Verifier will accept if the discrete logarithm of x with respect to g is equal to the discrete logarithm of y with respect to h.

b. Prove soundness for the above protocol. Assume that the discrete logarithm of x with respect to g is not equal to the discrete logarithm of y with respect to h. (Hint: after assuming that the Prover can give acceptable answers for two different values of c, show how a transcript containing both executions could be used to find the discrete logarithm of x with respect to g and the discrete logarithm of y with respect to h.)

c. Prove that the above protocol is honest verifier zero-knowledge, i.e., show that one can efficiently generate conversations $((u, v), c, e)$ with the same distribution as produced by the honest Prover and Verifier, without knowing k.

3. The Subgroup Membership Problem is as follows: Given a positive integer n and two distinct elements $\alpha, \beta \in Z_n^*$, where the order of α is l and is publicly know, determine if β is in the subgroup generated by α.

   Suppose that α, β, l, and n are given as input to a Prover and Verifier, and that the Prover is also given k such that $\alpha^k = \beta \pmod{n}$. Consider the interactive protocol in which the following is repeated $\log_2 n$ times:

Prover		Verifier
Choose random $0 \leq j \leq l-1\}$. Compute $\gamma = \alpha^j \pmod{n}$.	$\xrightarrow{\gamma}$	
		Choose a random $c \in \{0, 1\}$.
	$\xleftarrow{c}$	
Compute $h = j + ck \pmod{l}$.	$\xrightarrow{h}$	
		Check that $\alpha^h \equiv \beta^c \gamma \pmod{n}$. If not, reject and halt.

(a) Prove that the above protocol is an interactive proof system for Subgroup Membership.

(b) Suppose that β is in the subgroup generated by α. Show that the number of triples (γ, c, h) which the Verifier would accept is $2l$ and that each such triple is generated with equal probability if both the Prover and Verifier follow the protocol.

(c) Suppose that β is in the subgroup generated by α. What is the distribution of the values γ, h sent by a Prover following the protocol?

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

(e) If n is a prime, what value can you use for l? If n is not prime, is it reasonable to make this value l known?

Assignment due Thursday, May 15, 12:15

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.

1. Consider bit commitments defined in the notes by Damgård and Nielsen which are secure under the RSA assumption (page 10). If a Prover has created two such bit commitments, B_1 and B_2, to the same bit b, she can prove that they are commitments to the same bit by opening $B_1 \cdot B_2^{-1} \pmod{n}$ as a zero. If they are commitments to different bits, she can prove that by opening $B_1 \cdot B_2$ as a one.

   (a) Show that if the Prover created the two bit commitments, that she will be able to do the computations require to do the appropriate proof. Show this for all cases.

   (b) Show that if she cannot find an x such that $f(x) = y$, then she cannot use this method to prove that two commitments are to different bits if they are commitments to the same bit, or to prove that they are commitments to the same bit if they are commitments to different bits.

   (c) Show that doing some number, s, of proofs of equality, gives no information as to whether all of the commitments were commitments to zeros or commitments to ones. (Hint: Recall that for any commitment, B, the Prover could have created it as either a commitment to a zero or a commitment to a one.)

2. Suppose a Prover has made bit commitments, $B_i = y^{b_i} x_i^q \pmod{n}$ for $i = 1, 2, ..., m = 2k + 1$, and $k \geq 2$, using the same scheme as above (page 10 in the notes by Damgård and Nielsen), and a majority (at least $k + 1$) are commitments to ones. The Prover can give a zero-knowledge proof of this to a Verifier by executing the following $\log_2 n$ times:

Prover		Verifier
Choose random $r_1, r_2, ..., r_m$. Set $B'_i = y^{b_i} r_i^q \pmod{n}$, $1 \leq i \leq m$. Choose a random permutation σ of $1..m$.	$B'_{\sigma(1)}, \underline{B'_{\sigma(2)}, ...,} B'_{\sigma(m)}$ →	Choose a random bit c.
	← c	
If $c = 0$, compute $O_i = x_i \cdot r_i^{-1}$, $1 \leq i \leq m$.	$\sigma, \underline{O_1, O_2, ...,} O_m$ →	
If $c = 1$, choose randomly $k+1$ of the B'_i which have $b_i = 1$: $B_{i_1}, B_{i_2}, ..., B_{i_{k+1}}$.	$r_{i_1}, \underline{r_{i_2}, ..., r_{i_{k+1}}}$ →	Check the conditions below. Accept if OK. Otherwise reject.

The check the Verifier does is:
If $c = 0$, use σ to find the B'_i, and check that $B_i \cdot (B'_i)^{-1} \equiv O_i^q \pmod{n}$, for $1 \leq i \leq m$.
If $c = 1$, check that $y \cdot r_{i_j}^q \equiv B'_{i_j} \pmod{n}$ for $1 \leq j \leq k+1$.

(a) Prove completeness for the above protocol, showing that if the Prover has created the m bits so that the majority are commitments to ones, and if both the Prover and the Verified follow the protocol, then the Verifier rejects with negligible probability.

(b) Prove soundness for the above protocol. Assume that the Prover cannot find an x such that $y \equiv x^q \pmod{n}$ and that at most k of the original $B_1, B_2, ..., B_m$ were created as commitments to ones. Show that the probability that the Verifier accepts if it follows its

protocol is negligible.

(c) Prove that the above protocol is perfect zero-knowledge.