Institut for Matematik og Datalogi
Syddansk Universitet

November 21, 2019
JFB

Cryptology – 2019 – Lecture 16

Announcements

1. I am trying to move the remaining lectures on Thursdays to Mondays and Tuesdays in December. In particular, the lecture on Thursday, November 21, is cancelled.

2. The next pizza meeting will be held starting at 16:00 on November 18.

3. The deadline for applications for TAships at IMADA for spring is November 21:

   https://www.sdu.dk/da/service/ledige_stillinger/1071043

Lecture, November 14

We continued with chapter 16 and section 3.1, covering the El Gamal cryptosystem, mostly with the slides on the course homepage. We also covered subsection 11.7.2, section 15.2 and subsections 18.4.2 and 18.4.3.

Lecture, November 19

We will cover sections 4.1, 4.2, 4.3, and 4.5 fairly quickly (from the slides), with emphasis on curves of characteristic $p > 3$. We will continue with chapter 16, covering up through section 16.5.3. We may begin on chapter 19.

Lecture, November 25

We will cover sections 19.1, 19.2.1, and 19.4 from chapter 19, plus enough from section 19.3 to understand why Shamir's threshold scheme works. We will begin on protocols and zero-knowledge from the slides.

Problem session November 26

We will finish the last problem from November 18. I may lecture some if we have covered enough to do some of these problems.

1. Draw a diagram for the game demonstrating the security/insecurity of secret sharing schemes.

2. Consider the Shamir secret sharing scheme with $p = 31$. Let the threshold be $t + 1 = 3$. Suppose the shares are:

   - (1, f(1)) = (1,16)
   - (2, f(2)) = (2,5)
   - (3, f(3)) = (3,5)

   which are distributed to the share recipients. Show how to compute the secret.

The following problems are from some notes by Ivan Damgård and Jesper Buus Nielsen, entitled "Commitment Schemes and Zero-Knowledge Protocols (2011)".

1. Call a function $f : \mathbf{N} \to \mathbf{R}$ *polynomial in l* if there exists a polynomial p and constant l_0 such that $f(l) \leq p(l)$ for all $l > l_0$. A function $\epsilon : \mathbf{N} \to \mathbf{R}$ is *negligible in l* if for all polynomials p there exists a constant l_p such that $\epsilon(l) \leq 1/p(l)$ for all $l > l_p$.

   (a) Prove that if ϵ and δ are negligible in l, then $\epsilon + \delta$ is negligible in l.

   (b) Prove that if ϵ is negligible in l and f is polynomial in l, then $f \cdot \epsilon$ is negligible in l.

2. The statistical distance, $SD(P, Q)$, between two distributions, P and Q, is defined in the textbook at the bottom of page 122. An equivalent definition is:

   $$SD(P, Q) = \frac{1}{2} \sum_{y} |P(y) - Q(y)|,$$

   where $P(y)$ (or $Q(y)$) is the probability P (or Q) assigns to y.

Given two families of distributions, U and V, indexed by strings x, U and V are *statistically indistinguishable*, written $U \sim^s V$, if $SD(U_x, V_x)$ is negligible in the length of the string x. (U and V could be distributions of valid transcripts versus simulations, to prove statistical zero-knowledge. For computational indistinguishability, any probabilistic polynomial time distinguisher run on the two distributions would produce distributions as output which were statistically indistinguishable.)

Show that if $U \sim^s V$ and $V \sim^s W$, then $U \sim^s W$.

3. Note that commitment schemes $C(x, r)$ depend on a random value r. Prove that a commitment scheme which is information theoretically binding, but does not depend on the random input r, cannot be even computationally concealing.

   Say that a commitment scheme is g-randomized if it only depends on the first g bits of r. If a commitment scheme is information theoretically binding, and $\log \ell$-randomized (where ℓ is the length of r and the security parameter), can it be computationally concealing? What if we replace $\log \ell$ by $c \log \ell$ for a constant c? or by $(\log \ell)^c$?

The following problems are from some notes by Ivan Damgård , entitled "CPT Notes, Graph Non Zero-Knowledge for NP and Exercises".

1. Consider the Pedersen commitment scheme $B_a(x) = h^x \cdot g^a$ where the commitments are only to bits, so $x \in \{0, 1\}$. Suppose a prover P has committed to bits b_1, b_2 using commitments c_1, c_2, where $b_1 \neq b_2$. Now P wants to convince the verifier V that the bits are different. We claim he can do this by sending to V a number $s \in \mathbb{Z}_{p-1}$ such that $c_1 c_2 = h g^s$.

   - Show how an honest P can compute the required s, and argue that the distribution of s is the same when $(b_1, b_2) = (0, 1)$ as when $(b_1, b_2) = (1, 0)$. This means that V learns nothing except that $b_1 \neq b_2$.
   - Argue that if P has in fact committed in c_1, c_2 to $(0, 0)$ or $(1, 1)$, he cannot efficiently find s as above unless he can compute the discrete logarithm of h.
   - Argue in a similar way that P can convince V that she has committed to two bits that are equal by revealing s such that $c_1 c_2^{-1} = g^s$.

2. Assume P commits to two string $b_1, ..., b_t, b'_1, ..., b'_t$ using commitments $c_1, ..., c_t, c'_1, ..., c'_t$ as in the previous exercise. She claims that the strings are different and wants to convince V that this is the case while revealing no extra information. Note that he cannot point to an index j where $b_j \neq b'_j$ and use the above method on c_j, c'_j. This would reveal where the strings are different. Instead consider the following protocol:

   (a) P chooses a random permutation π on the set of indices $\{1, ..., t\}$. She computes, for $i = 1, ..., t$ a commitment $d_i = C(b_{\pi(i)}, r_i)$ and $d_i = C(b'_{\pi(i)}, r'_i)$. In other words, permute both strings randomly with the same permutation and commit bit by bit to the resulting strings. Send $d_1, ..., d_t, d'_1, ..., d'_t$ to V.

   (b) V chooses a random bit b and sends it to P.

   (c) If $b = 1$, P reveals π and uses the above method to convince V for all i that $c_{\pi(i)}$ contains the same bit as d_i. Similarly for $c'_{\pi(i)}$ and d'_i. If $b = 1$, P finds a position i where $b_\pi(i) \neq b'_{\pi(i)}$ and uses the above method to convince V that d_i, d'_i contain different bits.

   - Completeness: Argue that an honest prover always convinces the verifier.
   - Soundness: Show that if P can, for some set of commitments $d_1, ..., d_t, d'_1, ..., d'_r$ answer V correctly for both $b = 0$ and $b = 1$, then there is at least one j, where P can open c_j, c'_j to reveal different bits. Note that we assume she knows how to open the commitments $c_1, ..., c_t, c'_1, ..., c'_t$. The protocol in this exercise does not verify that P knows this — if one wants to check this, there are other protocols one can use.
   - Zero-knowledge: Sketch a simulator for this protocol. Hint: given commitment c, if you set $d = cg^{-s} \pmod p$, then $cd^{-1} = g^s \pmod p$. This means that even if the simulator does not know how to open c, it can create d and fake a proof that d contains the same bit as c. You do not have to formally prove that your simulator works.

3. Let n be the product of two large primes, p and q, where $p \equiv 1 \pmod 3$, and let $y \in \mathbb{Z}_n^*$. Suppose the Prover knows x such that $x^3 \equiv y \pmod n$. The Prover convinces the Verifier that there exists an x satisfying $x^3 \equiv y \pmod n$ by repeating the following protocol $\lceil \log_2 n \rceil$ times:

Prover		Verifier
Choose random $u \in \mathbb{Z}_n^*$. Let $v \equiv u^3 \pmod{n}$.		
	$\xrightarrow{v}$	
		Choose a random $c \in \{0, 1\}$.
	$\xleftarrow{c}$	
Let $z = u \cdot x^c \pmod{n}$.		
	$\xrightarrow{z}$	
		Check that $z^3 \equiv v \cdot y^c \pmod{n}$. If so, continue. Otherwise, reject.

The Verifier accepts if it has not rejected in any round.

a. Prove that the above protocol is an interactive proof system.

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