Homework 2 Solutions

CS161 Computer Security, Fall 2008
Assigned 9/25/08
Due 10/02/08

1 Security Protocols

1. (3 points) *Three-pass protocol.* Suppose Alice and Bob decide to use the following "three-pass protocol" to setup a shared secret session key K. First, Alice chooses a random K. Alice also generates a random secret one-time pad key K_A and XORs it with K. She sends $M_1 = K_A \oplus K$ to Bob. Bob generates a random secret one-time pad key K_B, XORs what he receives with it to compute $M_2 = M_1 \oplus K_B$, and sends M_2 to Alice. Alice computes $M_3 = M_2 \oplus K_A$, and sends M_3 to Bob, who recovers K as $M_3 \oplus K_B$. Note that K_A is known only to Alice, K_B only to Bob.

   - (2 points) Show that $K = M_3 \oplus K_B$.
     Solution.
     $M_3 \oplus K_B = (M_2 \oplus K_A) \oplus K_B$
     $= ((M_1 \oplus K_B) \oplus K_A) \oplus K_B$
     $= (((K_A \oplus K) \oplus K_B) \oplus K_A) \oplus K_B$
     $= K \oplus (K_A \oplus K_A) \oplus (K_B \oplus K_B)$
     $= K \oplus 0 \oplus 0$
     $= K$
   - (1 points) Suppose Eve can intercept the communication. Is this protocol secure against an eavesdropper Eve? If not, what can Eve recover.
     Solution. No. Eve can essentially compute $M_1 \oplus M_2 \oplus M_3$ to get K.
     $M_1 \oplus M_2 \oplus M_3 = (K_A \oplus K) \oplus (M_2) \oplus (M_2 \oplus K_A)$.
     $= (K \oplus K_A) \oplus (M_2 \oplus M_2 \oplus K_A)$
     $= K \oplus (K_A \oplus K_A) \oplus (M_2 \oplus M_2)$
     $= K$

2. **Extra Credit** (5 points) *Diffie-Hellman key exchange (DHKE).*

   Recall from class that DHKE proceeds as follows.

   - **Step 1**. Alice selects a large prime number p and a multiplicative generator α ($mod\ p$). Both p and α are made public.
   - **Step 2**. Alice picks a secret random x, with $1 \leq x \leq (p-2)$. She sends $M_A = \alpha^x$ ($mod\ p$).

- **Step 3**. Bob picks a secret random y, with $1 \leq y \leq (p-2)$. She sends $M_B = \alpha^y \ (mod \ p)$.
- **Step 4**. Using the received messages, Bob and Alice compute the shared session key K. Alice calculates K with the knowledge of her secret and Bob's message, by computing ${M_B}^x \ (mod \ p)$. Similarly, Bob computes ${M_A}^y \ (mod \ p)$.

Here is man-in-the-middle attack on DHKE different from the one studied in class. This version of the man-in-the-middle attack differs from the one seen in class, as it has the "advantage" that Eve does not have to intercept and retransmit all the messages between Alice and Bob.

Suppose Eve discovers that $p = Mq + 1$, where q is an integer and M is small. Eve intercepts $\alpha^x \ (mod \ p)$ and $\alpha^y \ (mod \ p)$ sent by Alice and Bob in steps 2 and 3 respectively. Eve sends Bob $(\alpha^x)^q \ (mod \ p)$ as message M_A in step 2, and sends Alice $(\alpha^y)^q \ (mod \ p)$ in step 3. Step 4 proceeds as described, but using the modified values of M_A and M_B.

- (2 points) Show that the Alice and Bob calculate the same key K'.
  Solution. Alice computes K' as $(M_B)^x \ (mod \ p)$. Eve tampers sending $(\alpha^y)^q \ (mod \ p)$ as M_B. So Alice computes $((\alpha^y)^q)^x \ (mod \ p)$ as K', which is $\alpha^{xyq} \ (mod \ p)$. Similary, Bob computes K' as $(M_A)^y \ (mod \ p)$. Eve tampers sending $(\alpha^x)^q \ (mod \ p)$ as M_A. Similary, Bob computes K' as $((\alpha^x)^q)^y \ (mod \ p)$, which is same K' as what Alice computes ($\alpha^{xyq} \ (mod \ p)$).
- (3 points) Show that there are only M possible values for K', so Eve may find K' by exhaustive search.
  Hint. Recall that the generator α is specially chosen. It generates all elements between 1 and p under the exponentiation operation without repeats. Therefore, $\alpha^{(p-1)} = 1 \ (mod \ p)$.
  Solution. The exponent of α that results in computation of K' is (xyq) – let us term this value as t. We know that $(p-1)$ is Mq. There must exist some n,r such that $xy = nM + r$, where $r = (xy) \ (mod \ M)$. So, we get :
  $K' = \alpha^{xyq} \ (mod \ p)$
  $= \alpha^{(nM+r)q} \ (mod \ p)$
  $= \alpha^{(nMq+rq)} \ (mod \ p)$
  $= (\alpha^{nMq}) \ (\alpha^{rq}) \ (mod \ p)$
  $= ((\alpha^{Mq})^n) \ (\alpha^{rq}) \ (mod \ p)$
  By Fermat's Theorem (also restated as hint) α^{Mq} is 1 $(mod \ p)$,
  $= (1) \ (\alpha^{rq}) \ (mod \ p)$
  Since r can only take values between 0 and $(M-1)$, the exponent has to be one of M multiples of q. Eve knows q and M is small, so she can exhaustively search for K'.

2 Secret Sharing

In this section we study secret sharing schemes.

Definition of (n,t) threshold secret sharing scheme. A (n, t) threshold secret sharing scheme is one where the secret can be efficiently computed given t of the n shares, but any $(t-1)$ shares reveal no information about the secret.

1. *Shamir polynomial scheme.* (4 points) Suppose we using the scheme using polynomials module a prime p, as described in class and the lecture notes. This scheme is called Shamir polynomial scheme.

   - (3 points) You have to setup a $(30, 2)$ Shamir scheme, working mod prime $p = 101$. Two of the shares are $(1, 13)$ and $(3, 12)$. Another person received the share $(2, *)$, but the part denoted by $*$ is unreadable. What is the correct value of $*$?

     Solution. Let the polynomial be $M + S_1 x \ (mod\ 101)$. The polynomial has degree 1 because we need 2 shares to be sufficient to reconstruct the secret M.

     Substituting for the two given shares

     $M + 1 \times S_1 = 13 \ (mod\ 101)$

     $M + 3 \times S_1 = 12 \ (mod\ 101)$

     Writing it in matrix form,

     $\begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix} \begin{pmatrix} M \\ S1 \end{pmatrix} = \begin{pmatrix} 13 \\ 12 \end{pmatrix} \ (mod\ 101)$

     Solving the set of equations, we get,

     $\begin{pmatrix} M \\ S1 \end{pmatrix} = \begin{pmatrix} 27/2 \\ -1/2 \end{pmatrix} \ (mod\ 101)$

     The result is :

     $\begin{pmatrix} M \\ S1 \end{pmatrix} = \begin{pmatrix} 27/2 \\ -1/2 \end{pmatrix} \ (mod\ 101)$

     We are working $(mod\ 101)$, and we need the coefficients $mod\ 101$. Since we know that $1/2 \equiv 51 \ (mod\ 101)$, we can replace $1/2 \ (mod\ 101)$ with 51. Therefore,

     $M \equiv 27 \ (1/2) \ (mod\ 101) \equiv 27 \ (51) \ (mod\ 101) \equiv 64 \ (mod\ 101)$.

     Similarly,

     $S_1 \equiv -51 \ (mod\ 101)$

     Polynomial is thus,

     $64 - 51x \ (mod\ 101)$

     The third share is simply an evaluation of this polynomial at $x = 2$, which is 63.

   - *Extensibility.* (1 points) It is easy to extend Shamir's polynomial $mod\ p$ scheme, to add new users. Show how could you extend a (n, t) Shamir scheme to a $(n + 1, t)$ scheme that includes an extra user, without changing the shares for existing n users.

     Solution. Simply evaluate the polynomial at another point x_{n+1}, and give $(x_{n+1}, f(x_{n+1}))$ as the new share. No changes to existing shares are needed for this.

   - *Military office* (4 points) A certain military office consists of 1 general, 2 colonels and 5 desk clerks. They have control of a powerful missile. They don't want the missile launched unless the general decides to launch it, or the 2 colonels decide to launch it, or the 5 desk clerks decide to launch it, or 1 colonel and 3 desk clerks decide to launch it. Describe how would you realize this policy with a secret sharing scheme.

     Solution. Distribute the shares of a $(30, 10)$ threshold scheme as follows : The general gets 10 shares, each colonel gets 5 shares, and desk clerks gets 2 shares each. Atleast 10 shares are needed to get the secret.

   - *XOR created shares.* (3 points) Consider the following secret sharing scheme. To share a secret $a \in \{0, 1\}^\ell$ among a group of n people, we choose $n - 1$ values $a_1, a_2 \ldots a_{n-1}$,

independently at random, where each $a_i \in \{0,1\}^\ell$. We select $a_n = a \oplus a_1 \oplus a_2 \oplus \ldots a_{n-1}$. We then distribute a_i to the i^{th} person in the group.

(a) (1 point) Is this a secure (n, n) threshold secret sharing scheme? **Solution.** Yes.

(b) (2 points) If your answer in previous sub-part is 'yes', specifically :
- Show that a can be recovered from the n shares.
- Argue that any $(n-1)$ shares do not reveal any information about a.

If your answer in previous sub-part is 'no', then either show that a can not be recovered from all the n shares, or, that less than n shares are sufficient to recover some information about a.

Solution. If all 'n' shares are present, then a can be computed as $a_1 \oplus a_2 \ldots \oplus a_n$. Suppose we have 1 share missing, say a_i. Then at best we can compute $(a \oplus a_i) = a_1 \oplus a_2 \ldots \oplus a_n$. Since a_i is chosen at random, the quantity $(a \oplus a_i)$ is akin to a one-time pad encryption, which reveals no information about a.

(c) *Visual Secret Sharing.* (7 points) It turns out that you don't need computers or sophisticated mathematics to realize secret sharing - you can implement even some of the more complex secret sharing scenarios using nothing but transparency sheets that can be overlaid on one another.

Suppose we decide to store a secret key text as a black-and-white image, say I. We would ideally like to break the image I into two image shares, say I_1 and I_2, such that neither I_1 nor I_2 provides any information about I individually, but if the two are superimposed one on top of the other than I "pops out".

To do this, the following idea is proposed: each pixel in I will be converted into a 2×2 square of sub-pixels in I_1 and I_2. In I_1, we choose the shape of that sub-pixel square to be one out of the patterns shown below (in Figure 1) at random.

X	
	X

	X
X	

Figure 1: Two possible patters for a sub-pixel block in I_1 and I_2. (**X** denotes a black sub-pixel).

In I_2 , if the corresponding pixel in I is white, we choose the 2×2 square to have exactly the same pattern as used in I_1 for this pixel, and if it is black, we choose it to have the opposite pattern.

- (2 points) Show how can we reconstruct each pixel in I, given the shares I_1 and I_2.

  Solution. When we superimpose I_1 on I_2, for any given subpixel 2×2 block, if exactly 2 of subpixels are black – then we know that the corresponding original pixel is white. Similarly, in the superimposed image if all subpixels in a 2×2 are black, then we know that the original pixel is black. This way, we can reconstruct the original image – the superimposition of I_1 and I_2 would have "grey" pixels (with only 2 subpixels black) and "black" pixels (with all 4 subpixels black).
- (2 points) Show that this is a secure (2,2) scheme, i.e individual shares do not reveal anything by the original image.

  Solution. Yes, this is a secure $(2, 2)$ threshold scheme. In any share, each subpixel pattern has two white and two black subpixels. Furthermore, each pattern is chosen independently at random for a given pixel in the original

pixel. Therefore, the presence of any of the two patterns reveals no information about the corresponding pixel in the original image.

– (3 points) Consider a slightly revised scheme where we split image I into three image shares I'_1, I'_2 and I'_3. In I'_1, we randomly choose the shape of that sub-pixel square to be one of the 3 patterns shown in Figure 2.

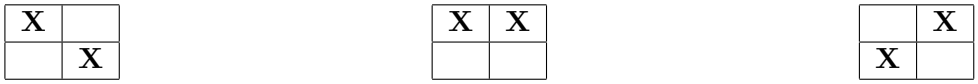

Figure 2: Revised scheme – Three possible patterns for a sub-pixel block in I'_1, I'_2 and I'_3 . (**X** denotes a black sub-pixel).

In I'_2 , if the corresponding pixel in I is white, we choose the 2×2 square to have exactly the same pattern as used in I'_1 for this pixel. If it is black, we choose it to have a pattern randomly selected from the remaining two (other than the one picked in I'_1).
In I'_3, if the corresponding pixel in I is white, we choose the 2×2 square to have exactly the same pattern as used in I'_1 and I'_2 . If it is black, we choose it to have the remaining pattern (the one not picked in I'_1 and I'_2).
We can now distribute I'_1, I'_2 and I'_3. We claim that this revised scheme is a secure $(3, 3)$ threshold secret sharing scheme. Is our claim sound? If not, how it violates the definition of $(3, 3)$ secret sharing scheme. If yes, explain why it satisfies the definition.
Solution. The claim that the revised scheme is a $(3, 3)$ secure threshold scheme is *not sound*. Any two of the patterns when superimposed, have more than 2 subpixels black if the original pixel was black. Thus, 2 shares reveal enough information to reconstruct the image. Hence, it is not a secure $(3, 3)$ scheme. But is it a secure $(3, 2)$ scheme.

3 Zero Knowledge Proofs

1. *Square Roots.* (9 points) Let $n = pq$ be the product of two large primes. Let y be a square mod n with $gcd(y, n) = 1$, y and n are public. Recall that finding square roots *mod n* is hard. However, Peggy claims to know a square root s of y. Victor wants to verify this, but Peggy does not want to reveal s. Given that, here is a method they use:

   - **Step 1**. Peggy chooses a random number r_1 and lets $r_2 = sr_1^{-1}$ $(mod\ n)$, such that $r_1 r_2 = s(mod\ n)$.
     She computes $x_1 = r_1^2(mod\ n)$ and $x_2 = r_2^2(mod\ n)$, and sends x_1 and x_2 to Victor.
   - **Step 2**. Victor checks that $x_1 x_2 = y$ $(mod\ n)$, then chooses either x_1 or x_2 at random and asks Peggy to supply a square root of it. He checks that it is a correct square root.

   The first two steps are repeated in several rounds, until Victor is convinced. It should be clear that of course, if Peggy knows s, the procedure works without problems.

   - (3 points) Suppose Peggy does not know s. Can she construct two numbers x_1, x_2 for each of which she knows the square roots, and such that $x_1 x_2 = y$ $(mod\ n)$? Why or

why not ? How does this fact help Victor find out that she does not know s?

Solution. No, she can not know such x_1 and x_2. We can prove it by contradiction. Suppose she does not s but knows such x_1 and x_2. Let their square roots be a_1 and a_2 respectively which Peggy knows. If so, then a_1a_2 is the sqaure root of y (as $(a_1a_2)^2 = x_1x_2 \ (mod\ N)$), which can be computed by Peggy. This violates the assumption that Peggy does not know s.

- (3 points) Suppose, however, that Peggy predicts correctly that Victor will always ask for square root of x_2. How can she compute x_1 and x_2 such that the method always falsely convinces Victor that Peggy knows s at Step 2 of each round? (The trivial solution is to send $x_2 = 1$; this is easily detected and Victor is smart enough to check for this case.)

  Solution. Peggy picks a random r $(mod\ N)$ such that $gcd\ (r, N) = 1$. At step 1 she sends $x_2 = r^2\ (mod\ N)$ and $x_1 = x_2^{-1}y\ (mod\ N)$. She sends r as the square root of x_2 in step 2.

- (3 points) Suppose that Victor chooses to ask Peggy for square root of either x_1 or x_2 randomly in Step 2. Peggy does not know s, and tries her luck in each round by guessing whether Victor will ask for x_2 or not in step 2. She constructs x_1,x_2 using her guess, in the way you've devised in the previous sub-part, and sends them to Victor. What is the probability that Victor is falsely convinced that Peggy knows s after $t = 10$ rounds of the method.

  Solution. She could compute either x_1 or x_2 using a random r as shown above. With a probabolity p, Victor could ask her for the same x_i which is computed from $r^2\ (mod\ N)$ only and requires no use of s. The probability that Victor always asks for this x_i in any step is 0.5. Thus, after t rounds, $p = (0.5)^t$. Substituting $t = 10$, $p = (0.5)^{10}$.