ICS 180: Introduction to Cryptography 4/23/2004

Homework 3

Due **Tuesday**, *5/04/2004* [[**you get more than a week!**]]

1 Authentication Scheme from One-Way Permutations

Let PPT algorithms $(Gen, Sample, Eval)$ define a OWF (or OWP) $\{f_i\}_{i\in\mathcal{I}}$. Suppose that players U and B use the following authentication scheme. For example, say that B is a bank's web portal and C is a web applet run by the bank's client. The scheme is designed to last for one year, and needs to be reinitialized after that:

- **Initialization Protocol:** Let $n = 365$. B runs $Gen(1^\tau)$ to pick a one-way function f_i with security parameter τ and runs $Sample(i)$ to pick a random element $x^{(n)}$ in the domain D_i of f_i. Then B computes, for k going from n down to 1, values $x^{(k-1)} = f_i(x^{(k)}) = Eval(i, x^{(k)})$. (You'll see in a second why we are computing them backward rather than forward.) B keeps for himself $x^{(0)}$ as the "verification value" for C, and gives to C (over some secure channel) the "root authentication secret" $x^{(365)}$. C then re-generates all the $x^{(k)}$ values for $k = 0, ..., 364$ by consecutive applications of f_i. Let's denote k-times repeated application of f_i as a function $(f_i)^{(k)} : D_i \to \{0,1\}^*$. With this notation we have $x^{(n-k)} = (f_i)^{(k)}(x^{(n)})$ for every k.
- **Authentication Protocol:** To authenticate himself to B on day t, C sends to B value $x = x^{(t)}$ and announces that he is "C". B then picks the yesterday's verification value $x^{(t-1)}$ for that client, and authenticates this client as indeed "C" if $f_i(x) = x^{(t-1)}$. If the equation holds B stores x as $x^{(t)}$. (It's easy to generalize this to the case when C contacted B last on any day $t' < t$: Just compute $(f_i)^{(t-t')}$ on $x^{(t)}$ and compare with $x^{(t')}$.)

Assume that the adversary E, who tries to authenticate himself as "C" to B too, can *eavesdrop* on all instances of the (C, B) authentication protocol but cannot interrupt any such instance. On the other hand E can *initiate* an instance of the authentication protocol with B himself and try to make B authenticate him as "C".

1.1 [25 points]

Prove that if the function collection $\{f_i\}$ defined by $(Gen, Sample, Eval)$ is a One Way *Permutation* collection then the above authentication protocol is secure against the eavesdropping adversary E in the following sense: Show that if there exists a PPT E which, after listening to some number $k \in [1, n]$ of authentication sessions (C, B), has a non-negligible chance of being authenticated by B as "C" on a session that E initializes, then you can use such adversary E to create a PPT algorithm A which has a non negligible advantage in an attack against one-wayness of the OWP collection f_i. In other words, algorithm A should succeed with non negligible probability in inverting permutation f_i on value $y = f_i(x)$ for a random $x \in D_i$. This will show that if the above authentication protocol is insecure against eavesdroppers (i.e. there exists a PPT attacker E which cheats the scheme with a significant

enough probability) then the function collection $\{f_i\}$ cannot be a OWP collection. Therefore, by counterpositive, if $\{f_i\}$ *is* a OWP collection then this is a secure authentication scheme. Note that in this way you showed that one can build a provably secure authentication scheme (but against eavesdroppers only!) based on either the DL assumption or the RSA assumption (because either of these assumptions implies a OWP family $\{f_i\}$).

1.2 [10 points]

What goes wrong with this argument if $\{f_i\}$ is only a family of one-way *functions*, i.e. not necessarily permutations? In particular, assume that each for every i, the range of f_i, $R_i = \{f(x) \mid x \in D_i\}$ is twice smaller than its domain, i.e. $|R_i|/|D_i| \leq 1/2$. A one-way function can indeed have this property.[1] What if f_i is "domain-halving" not just on its domain but also on all consecutive iterations too, so that $|R_i^{(k)}|/|D_i| \leq (1/2)^k$ where $R_i^{(k)} = \{(f_i^{(k)})(x) \mid x \in D_i\}$. Assume that n, the number of days you want the scheme to be good for, is polynomial in τ. Show an efficient attack E that breaks this authentication scheme even if $\{f_i\}$ is a OWF collection.

2 One-Way Encryption Implies One-Way Function [15 points]

Show that if one-way encryption scheme $\Sigma = (KGen, Enc, Dec)$ exists then there exists a OWF collection. Construct such collection $\{f_i\}_{i \in \mathcal{I}}$. (Hint: unlike in the DL and RSA examples of OWF collections, the index set can be very easy here, e.g. $\mathcal{I} = \{1^\tau\}_{\tau=1,2,\ldots}$.) Rememember that each function f_i must be deterministic, so all the randomness used by the encryption scheme will probably need to be an input to f_i.

Side Note: Remember that in the previous homework you showed that indistinguishable encryption implies one-way encryption? Taking the two together means that indistinguishable encryption also implies OWFs. In fact, you can show a similar implication for most cryptographic notions, like secure signature schemes, secure authentication schemes, etc. In other words you can show that if any of them exist (i.e. are secure according to their respective definitions) then one-way functions must exist. This is why we call one-way functions a fundamental notion: If they did not exist then none of this other stuff could be ever achieved!

3 Prime Modular Arithmetics [25 points]

Let p be a prime. Recall groups $\mathbb{Z}_p$ and $\mathbb{Z}_p^*$ from lecture and the handout on modular arithmetics. Recall the fact that $\mathbb{Z}_p^*$ is cyclic and has a generator (in fact it has many of them). Recall set QR_p of squares modulo p.

3.1

Write out the elements of group $\mathbb{Z}_{11}$. How many elements are there? Pick an element g which is a generator of this group. For every element $a \in \mathbb{Z}_{11}$, write down $DL_{(g,11)}(a)$.

[1] You can actually easily construct such function from any assumed one-way functions like RSA or Exp and show that it is both "domain-halving" and still one-way if the original one is.

3.2

Let g be a generator of $\mathbb{Z}_p^*$. Prove that if $g^x = g^y \bmod p$ then $x = y \bmod p-1$.

[*Side note:* In lecture we showed that for any p (even composite), if $x = y \bmod \phi(p)$ then $g^x = g^y \bmod p$ for *any* $g \in \mathbb{Z}_p^*$, which of course includes g which is a generator of $\mathbb{Z}_p^*$.]

3.3

Let g be a generator of $\mathbb{Z}_p^*$. Prove that $y \in QR_p$ if and only if $x = DL_{(g,p)}(y)$ is even. (Note that since g is a generator then for every $y \in \mathbb{Z}_p^*$ there is $x = DL_{(g,p)}(y)$.) Which of the elements of $\mathbb{Z}_{11}$ are squares?

3.4

Prove that $y \in QR_p$ if and only if $y^{(p-1)/2} = 1 \bmod p$.

3.5

Prove that if y, z are quadratic residues modulo p then so is $yz \bmod p$ and $y^{-1} \bmod p$. (In fact this holds for any p, not necessarily prime one.)

3.6

Recall that an *order* $ord_p(a)$ of an element $a \in \mathbb{Z}_p^*$ is defined as the smallest i s.t. $a^i = 1 \bmod p$. The order of the group is the maximum order of any of the group's elements. For each element in $\mathbb{Z}_{11}$ tell its order. Can you see a relation between orders of elements of $\mathbb{Z}_{11}$ and number $\phi(11) = 10$? Can you formulate a general hypothesis about orders of group elements and $\phi(p)$ Can you prove it?

4 Composite Modular Arithmetics [25 points]

Let $n = pq$ for prime p, q. Recall that by the Chinese Reminder Theorem [CRT], for every $r_1 \in \mathbb{Z}_p$ and $r_2 \in \mathbb{Z}_q$ there is a unique $s \in \mathbb{Z}_n$ s.t. $s = r_1 \bmod p$ and $s = r_2 \bmod p$, and vice versa.

4.1

Show how to reconstruct s from (r_1, r_2) if you know elements a, b s.t. $ap + bq = 1$. (Note that $ap = 0 \bmod p$, $ap = 1 \bmod q$, $bq = 0 \bmod q$ and $bq = 1 \bmod p$...)

4.2

Consider group $\mathbb{Z}_{35}^*$. List its elements. How many of them are there? What's $\phi(35)$?

4.3

Consider the CRT theorem for $n = 35$, $p = 5$, $q = 7$. Find a, b s.t. $ap + bq = 1$ (you can find them by hand or by a Euclidean algorithm). Compute $205^{140} \bmod 35$ without a computer. Show your work. (Hint: CRT helps here.)