ICS 268: Cryptography and Communication Security 10/28/2004

Homework 4

Due **Tuesday**, *11/09/2004*

1 Hash Function Properties

1.1

Explain why the modular exponentiation function, $Exp : \{0,1\}^* \rightarrow \mathbb{Z}_p^*$ where $Exp(x) = g^x \bmod p$ for a large prime p and g generator of $\mathbb{Z}_p^*$, is a *preimage resistant hash function* under the discrete logarithm assumption.

1.2

Assuming g is a generator of $\mathbb{Z}_p^*$, under the same discrete logarithm assumption, is function $Exp' : \{0,1\}^* \rightarrow \{0,1\}^n$ where $Exp'(x) = (Exp(x)) \bmod 2^n$ where $n = |p| - 1$, still a preimage resistant hash function?

1.3

Show that modular exponentiation function Exp defined as above is *not second preimage resistant* and also *not collision resistant.*

2 Authentication Scheme From A Hash Function

Recall from lecture that a symmetric authentication scheme consists of an PPT algorithm KeyGen, which given a security parameter τ generates a symmetric key k which is then given to both the Client and the Server, and an interactive protocol Auth in which the Server authenticates the Client as long as both run this protocol on input k. Recall that we call a (symmetric) authentication scheme "secure" if for every efficient adversary algorithms A, after engaging in any number n of instances of the Auth protocol with the Client (who runs on input k), the adversary has still only a negligible probability that when he turns around and engages with the Server (who runs on the same input k) in a fresh instance of the Auth protocol, the Server is going to accept that instance of the protocol.

(The first authentication scheme we did in class did not meet the above requirement because the players needed to update their state between instances and the scheme was good for a priori limited number of times. The second scheme we covered was actually stronger than the above model, because it generated a (public,private) key pair (PK, SK), where the public key could be given to everybody, including the adversary.)

2.1

Assuming $H : \{0,1\}^* \rightarrow \{0,1\}^n$ is a hash function *which can be modeled as a random oracle* (which is the strongest security requirement posed on a hash function), construct a secure symmetric authentication scheme along the following lines: The KeyGen algorithm will pick the symmetric key k as a random binary number of length l_1, and the Auth protocol should

look as follows: The Server will send to the Client a random challenge (a.k.a. a "nonce") c, which is a random binary string of length l_2, and the client will respond with $r = H(k, c)$.

Assuming that H can be modeled as a random oracle, prove that this authentication scheme is secure by doing the following calculation:

Express the advantage of the adversary in breaking this authentication scheme (i.e. the probability that the adversary succeeds in making the Server authenticate him as the Client) as a function of n, l_1, and l_2.

Given the above calculation, what key length do you recommend to achieve realistic security?

2.2

Show that the above scheme become *insecure* if the hash function H is implemented as the modular exponentiation function (as in the first problem above). Describe this attack explicitely, and argue why it breaks the authentication scheme.

2.3

You should conclude that the one-wayness of a function (which we believe that the exponentiation function does posses) is a *weaker* property than "effectively looking like a random function", which we believe that some hash functions like SHA1 or MD5 do possess.

3 Insecure Variant of the Schnorr Signature Scheme

Consider the following variant of the Schnorr signature scheme (i.e. the discrete-log based signature scheme we discussed in class): The private key is $x \in \mathbb{Z}_q$, public key is $y = g^x \bmod p$, and the signature on m is a pair (r, z) s.t. $r = g^k \bmod p$ for a random $k \in \mathbb{Z}_q$ and $z = k + x(m * r) \bmod q$.

(This is a simplification of the Schnorr scheme where hash $H(m, r)$ is replaced with just a product $m * r \bmod q$.)

3.1

Show that this signature is not secure against an existential forgery after the adversary sees only one valid (message,signature) pair.

3.2

Extend your attack to show an efficient adversary which after seeing a single (message,signature) pair can actually create (message,signature) pairs where the message is a uniformly distributed element in $\mathbb{Z}_q$.

3.3

Give an example of a system (i.e. some computer/communication application) which uses signatures and the system is secure if the signature scheme is secure against an existential forgery under the chosen message attack, but it is *insecure* if it is instantited with the above signature scheme.