Homework 2

Cryptography & Network Security
CS 406 : Spring 2023

Released: Sun Mar 12
Due: Mon Apr 3

PKE, Hash Functions, Signatures [Total 50 pts]

1. **CCA Secure PKE in the Random Oracle Model** [10 pts]

   Suppose $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is a CPA-secure PKE scheme. We shall write $\mathsf{Enc}_{PK}(m; r)$ to indicate encryption of the message m using randomness r; suppose Enc requires $r \leftarrow \{0,1\}^k$ (k, as always, being the security parameter). Also, suppose H is a hash function modeled as a random oracle with k-bit outputs.

   Consider a new encryption scheme with the encryption algorithm defined as follows: $\mathsf{Enc}^*_{PK}(m; r) = (\mathsf{Enc}_{PK}(m||r; H(r)), H(m||r))$, where $r \in \{0,1\}^k$.

   (a) What should the corresponding decryption algorithm Dec^* be so that $(\mathsf{KeyGen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ could be a CCA-secure encryption scheme?

   (b) Show that the scheme will not even be CPA secure if $H(m||r)$ is replaced by $H(m)$.

   (c) Show that, for some (possibly contrived) CPA-secure scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$, the modified scheme will not even be CPA secure if $H(m||r)$ is replaced by $H(r)$.

   (d) **Proof of CCA security** [Extra Credit]
   Prove that with Dec^* from part (a) above, $(\mathsf{KeyGen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ is indeed a CCA-secure encryption scheme in the random oracle model. Flesh out the details of the proof as much as you can, basing your arguments only on the CPA-security of the given scheme, and statistical properties.

   Hint: You should convert a CCA-adversary A^* for $(\mathsf{KeyGen}, \mathsf{Enc}^*, \mathsf{Dec}^*)$ into a CPA-adversary A for $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$. A will need to simulate the random oracle and the decryption oracle that A^* expects. As such, A gets to see all random oracle queries that A^* makes.

2. **2-Universal Hash Function.** [10 pts]

   For a prime number q and positive integers m, n, and $R := \mathbb{Z}_q^n$. Below, all probabilities refer to the uniformly random choice of $\mathbf{L} \leftarrow \mathbb{Z}_q^{n \times m}$, and all addition and multiplication of numbers are modulo q.

   (a) Suppose $D = \mathbb{Z}_q^m \setminus \{0^m\}$. Prove that $\forall \mathbf{x} \in D, \mathbf{a} \in R$, $\Pr_{\mathbf{L}}[\mathbf{L}\mathbf{x} = \mathbf{a}] = 1/|R|$.

   Hint: Fix an i s.t. $\mathbf{x}_i \neq 0$. Consider sampling $\mathbf{L}$ by picking the i^{th} column last.

   (b) Now suppose $D = \{0,1\}^m \setminus \{0^m\}$ (i.e., non-zero vectors with only 0 and 1 entries). Show that $\forall \mathbf{x}, \mathbf{y} \in D$ s.t. $\mathbf{x} \neq \mathbf{y}$, $\mathbf{a}, \mathbf{b} \in R$, $\Pr_{\mathbf{L}}[\mathbf{L}\mathbf{x} = \mathbf{a}, \mathbf{L}\mathbf{y} = \mathbf{b}] = 1/|R|^2$.

   Hint: Argue that if $\mathbf{x} \neq \mathbf{y}$ and $\mathbf{x}, \mathbf{y} \in \{0,1\}^m$ there are at least two coordinates i, j restricted to which $\mathbf{x}, \mathbf{y}$ are linearly independent. Consider sampling $\mathbf{L}$ by picking these two columns last.

This shows that the family of functions $\mathcal{H} = \{h_{\mathbf{L}} \mid \mathbf{L} \in \mathbb{Z}_q^{n\times m}\}$, where $h_{\mathbf{L}} : D \to R$ is defined as $h_{\mathbf{L}}(\mathbf{x}) = \mathbf{L}\mathbf{x}$ is a 2-universal hash function when $D = \{0,1\}^m \setminus \{0^m\}$. We can upgrade this to a 2-universal hash function family for $D = \{0,1\}^m$ (i.e., including the all-zero vector) by considering $h_{\mathbf{L},\mathbf{u}}(\mathbf{x}) = \mathbf{L}\mathbf{x} + \mathbf{u}$ over all $(\mathbf{L}, \mathbf{u}) \in \mathbb{Z}_q^{n\times m} \times \mathbb{Z}_q^m$.

3. **Computational Hash Functions.**

   In this problem we consider hash functions on a finite domain (from $\{0,1\}^{n(k)}$ to $\{0,1\}^{m(k)}$).

   (a) **Preimage collision resistance $\not\Rightarrow$ Second-preimage collision resistance.** Suppose $\mathcal{H}$ is preimage collision resistant. Modify $\mathcal{H}$ to $\mathcal{H}'$ (possibly with a different domain), so that the latter remains preimage collision resistant, but is not second-preimage collision resistant. (You must prove that $\mathcal{H}'$ has both these properties.) [4 pts]

   (b) **Second-preimage collision resistance $\not\Rightarrow$ Preimage collision resistance.** Given a CRHF $\mathcal{H}$ which compresses by two bits (say from n bits to $n-2$ bits), construct a CRHF $\mathcal{H}'$ that compresses by one bit (say from $n+1$ bits to n bits), such that the function f, defined by $f(h', x) = (h', h'(x))$ (where $h' \in \mathcal{H}'$), is **not** a OWF. (In both $\mathcal{H}$ and $\mathcal{H}'$, collision-resistance holds when the hash function is drawn uniformly at random from the family.) [6 pts]

   Hint: Can you define h' so that it includes (disjoint) copies of h and a copy of an easy to invert one-to-one function? Why would this retain second-preimage collision resistance? Why would this destroy preimage collision resistance?

   (c) **(Sufficiently Shrinking) CRHF implies OWF.** [Extra Credit]
   Show that if $\mathcal{H}$ is a CRHF from n bits to $n/2$ bits, then the function $f(h, x) = (h, h(x))$ is a OWF.

   Hint: You may use the following intermediate steps. Below we say that "x has a collision under f" if there exists an $x' \neq x$ such that $f(x) = f(x')$.

   i. Let $\mathcal{H}$ be a CRHF and suppose that for every $h \in \mathcal{H}$ and every x, x has a collision under h. Show that the function $f(h, x) = (h, h(x))$ is a OWF.

   ii. Now, suppose that for each $h \in \mathcal{H}$, all but a negligible fraction of x's have a collision under h. Show that the function $f(h, x) = (h, h(x))$ is a OWF.

   iii. Finally, apply the above to the case of $f : \{0,1\}^n \to \{0,1\}^{n/2}$.

4. **Composition: UOWHF vs. CRHF**

   In this problem we consider hash functions which take arbitrarily long strings as inputs, and produces an output of a fixed length.

   (a) Suppose $\mathcal{H}$ is a CRHF family. Then show that the hash function family $\mathcal{H}' = \{h^2 | h \in \mathcal{H}\}$ is also a CRHF, where h^2 is defined by $h^2(x) = h(h(x))$. [3 pts]

   (b) In this problem you will argue that when a UOWHF is composed with itself in the above manner, it need not remain a UOWHF. For this we need to show *some* UOWHF with this behaviour. However, since UOWHF may not exist at all (in one-way functions do not exist), we shall be able to do this only assuming UOWHFs exist.

   Suppose $\mathcal{H}_0$ is a UOWHF family which produces k-bit hash values. We shall use this to define a new hash function family $\mathcal{H}$ with $2k$-bit hash values, as follows: $\mathcal{H} = \{f_{h,a} | h \in \mathcal{H}_0, a \in \{0,1\}^k\}$ where (using $\|$ to denote concatenation),

$$f_{h,a}(x) = \begin{cases} a\|0^k & \text{if } x \text{ has prefix } a \\ a\|h(x) & \text{otherwise.} \end{cases}$$

   i. Show that $\mathcal{H}$ is a UOWHF, but not a CRHF. [6 pts]

ii. Argue that the hash function family $\mathcal{H}' = \{h^2 | h \in \mathcal{H}\}$ is not a UOWHF. [1 pt]

5. **Attacking a Signature Scheme** [10 pts]

   In this problem, we consider a seemingly minor modification of the Schnorr signature scheme, and show that it results in a broken scheme.

   Recall that, in the original scheme, the verification key is $(\mathbb{G}, g, Y)$, where $\mathbb{G}$ is a prime-order group with a generator g and $Y = g^y$ is a random group element, with $y \leftarrow \mathbb{Z}_{|\mathbb{G}|}$ being the signing key; the signature on a message M is produced as $\mathsf{Sign}_y(M) = (e, s)$, where $e = H(M||g^r)$ and $s = r - ye$, for a random $r \leftarrow \mathbb{Z}_{|\mathbb{G}|}$.

   In the modfied scheme the messages belong to $\mathbb{G}$, and $e = H(M||g^r)$ is replaced by $e = H(M \cdot g^r)$, where $\cdot$ denotes the group operation.

   Give an existential forgery attack on this modified scheme (in the random oracle model).

6. **Needham-Schroeder Protocol.** [Extra Credit]

   The Needham-Schroeder Public Key protocol was an early protocol (proposed in 1978) for "authenticated key exchange," using a public-key "encryption" scheme. (This was well before Goldwasser and Micali had developed the CPA security notion for encryption.)

   The protocol uses a trusted server, S, to help two parties exchange secret keys with each other. A priori, there are no secrecy or authentication guarantees on the communication network, and the parties know only each other's identities and a public key of the server S. The server, S, knows public keys of all the users. The goal of the protocol is that at the end A and B should agree on random nonces N_A and N_B (chosen by A and B respectively).

   The protocol is shown in Figure 1. It is described in terms of a public key "encryption" algorithm Enc. It is a *deterministic* encryption scheme with the property that $\mathsf{Enc}_{PK}(\mathsf{Enc}^{-1}_{SK}(M)) = M$. If M is sufficiently random, $\mathsf{Enc}^{-1}_{SK}(M)$ is assumed to behave like a (very weak) signature on M: it is infeasible for an adversary who is *given a random* M to create the signature on M (note that this is weaker than the notion of existential unforgeability, which is not satisfied by this scheme). PA, PB are Alice and Bob's public keys and SA, SB are their secret keys, respectively. Likewise, the server's public and secret keys are PS, SS.

$A \to S$:	A, B	(This is A requesting S to send B's public-key)
$S \to A$:	$\mathsf{Enc}^{-1}_{SS}(PB, B)$	(A will use Enc_{PS} to recover B's public key)
$A \to B$:	$\mathsf{Enc}_{PB}(N_A, A)$	(where N_A is a fresh nonce, picked by A)
$B \to S$:	B, A	(Now B requests S to send A's public-key)
$S \to B$:	$\mathsf{Enc}^{-1}_{SS}(PA, A)$	(B will use Enc_{PS} to recover A's public key)
$B \to A$:	$\mathsf{Enc}_{PA}(N_B, N_A)$	(where N_B is a fresh nonce picked by B)
$A \to B$:	$\mathsf{Enc}_{PB}(N_B)$	(A and B agree on N_A, N_B at this point)

Figure 1: The Needham-Schroeder public-key protocol.

   (a) There is a (famous) man-in-the-middle attack on this protocol, whereby a party E in the system can set up a shared key with B, such that B thinks that she has shared that key with A. Describe such an attack (without looking it up!). [Extra Credit]

   Hint: The adversary can run a concurrent session with A.

   (b) Suggest a (small) fix for the attack. [Extra Credit]

   (c) If you were designing this protocol today, using public-key encryption and signatures, how would you do it? [Extra Credit]