COMP547A Homework set #4

Due Tuesday December 3rd, 2020, 23:59:59

Exercises (from Katz and Lindell's book)

4.7 Let F be a pseudorandom function. Show that each of the following MACs is insecure, even if used to authenticate fixed-length messages. (In each case Gen outputs a uniform $k \in \{0,1\}^n$. Let $\langle i \rangle$ denote an $n/2$-bit encoding of the integer i.)

[5%] (b) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, compute $t := F_k(\langle 1 \rangle \| m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle \| m_\ell)$.

[5%] (c) To authenticate a message $m = m_1, \ldots, m_\ell$, where $m_i \in \{0,1\}^{n/2}$, choose uniform $r \leftarrow \{0,1\}^n$, compute

$$t := F_k(r) \oplus F_k(\langle 1 \rangle \| m_1) \oplus \cdots \oplus F_k(\langle \ell \rangle \| m_\ell),$$

and let the tag be $\langle r, t \rangle$.

4.13 We explore what happens when the basic CBC-MAC construction is used with messages of different lengths.

[5%] (a) Say the sender and receiver do not agree on the message length in advance (and so $\mathsf{Vrfy}_k(m,t) = 1$ iff $t \stackrel{?}{=} \mathsf{Mac}_k(m)$, regardless of the length of m), but the sender is careful to only authenticate messages of length $2n$. Show that an adversary can forge a valid tag on a message of length $4n$.

[5%] (b) Say the receiver only accepts 3-block messages (so $\mathsf{Vrfy}_k(m,t) = 1$ only if m has length $3n$ and $t \stackrel{?}{=} \mathsf{Mac}_k(m)$), but the sender authenticates messages of any length a multiple of n. Show that an adversary can forge a valid tag on a new message.

4.14 Prove that the following modifications of basic CBC-MAC do not yield a secure MAC (even for fixed-length messages):

[5%] (b) A random initial block is used each time a message is authenticated. That is, change Construction 4.11 by choosing uniform $t_0 \in \{0,1\}^n$, computing t_ℓ as before, and then outputting the tag $\langle t_0, t_\ell \rangle$; verification is done in the natural way.

[5%] 10.3 Describe a man-in-the-middle attack on the Diffie–Hellman protocol where the adversary shares a key k_A with Alice and a (different) key k_B with Bob, and Alice and Bob cannot detect that anything is wrong.

More on back…

[5%]

10.4 Consider the following key-exchange protocol:

(a) Alice chooses uniform $k, r \in \{0,1\}^n$, and sends $s := k \oplus r$ to Bob.
(b) Bob chooses uniform $t \in \{0,1\}^n$, and sends $u := s \oplus t$ to Alice.
(c) Alice computes $w := u \oplus r$ and sends w to Bob.
(d) Alice outputs k and Bob outputs $w \oplus t$.

Show that Alice and Bob output the same key. Analyze the security of the scheme (i.e., either prove its security or show a concrete attack).

11.6 Consider the following public-key encryption scheme. The public key is $(\mathbb{G}, q, g, h)$ and the private key is x, generated exactly as in the El Gamal encryption scheme. In order to encrypt a bit b, the sender does the following:

(a) If $b = 0$ then choose a uniform $y \in \mathbb{Z}_q$ and compute $c_1 := g^y$ and $c_2 := h^y$. The ciphertext is $\langle c_1, c_2 \rangle$.
(b) If $b = 1$ then choose independent uniform $y, z \in \mathbb{Z}_q$, compute $c_1 := g^y$ and $c_2 := g^z$, and set the ciphertext equal to $\langle c_1, c_2 \rangle$.

[1%] Show that it is possible to decrypt efficiently given knowledge of x.
[4%] Prove that this encryption scheme is CPA-secure if the decisional Diffie–Hellman problem is hard relative to $\mathcal{G}$.

Hint: Prove that if "not CPA-secure" then "DDH problem is efficiently solved ».

[5%]

11.7 Consider the following variant of El Gamal encryption. Let $p = 2q + 1$, let $\mathbb{G}$ be the group of squares modulo p (so $\mathbb{G}$ is a subgroup of $\mathbb{Z}_p^*$ of order q), and let g be a generator of $\mathbb{G}$. The private key is $(\mathbb{G}, g, q, x)$ and the public key is $(\mathbb{G}, g, q, h)$, where $h = g^x$ and $x \in \mathbb{Z}_q$ is chosen uniformly. To encrypt a message $m \in \mathbb{Z}_q$, choose a uniform $r \in \mathbb{Z}_q$, compute $c_1 := g^r \bmod p$ and $c_2 := h^r + m \bmod p$, and let the ciphertext be $\langle c_1, c_2 \rangle$. Is this scheme CPA-secure? Prove your answer.

11.15 Say three users have RSA public keys $\langle N_1, 3 \rangle$, $\langle N_2, 3 \rangle$, and $\langle N_3, 3 \rangle$ (i.e., they all use $e = 3$), with $N_1 < N_2 < N_3$. Consider the following method for sending the same message $m \in \{0,1\}^\ell$ to each of these parties: choose uniform $r \leftarrow \mathbb{Z}_{N_1}^*$, and send to everyone the same ciphertext

$$\left\langle [r^3 \bmod N_1], [r^3 \bmod N_2], [r^3 \bmod N_3], H(r) \oplus m \right\rangle,$$

where $H : \mathbb{Z}_{N_1}^* \to \{0,1\}^\ell$. Assume $\ell \gg n$.

[5%] (a) Show that this is not CPA-secure, and an adversary can recover m from the ciphertext even when H is modeled as a random oracle.
Hint: See Section 11.5.1.

[5%] (b) Show a simple way to fix this and get a CPA-secure method that transmits a ciphertext of length $3\ell + \mathcal{O}(n)$.

[5%] (c) Show a better approach that is still CPA-secure but with a ciphertext of length $\ell + \mathcal{O}(n)$.

More on back…

[5%] 12.1 Show that Construction 4.7 for constructing a variable-length MAC from any fixed-length MAC can also be used (with appropriate modifications) to construct a signature scheme for arbitrary-length messages from any signature scheme for messages of fixed length $\ell(n) \geq n$.

12.5 Another approach (besides hashing) that has been tried to construct secure RSA-based signatures is to *encode* the message before applying the RSA permutation. Here the signer fixes a public encoding function $\mathsf{enc} : \{0,1\}^\ell \to \mathbb{Z}_N^*$ as part of its public key, and the signature on a message m is $\sigma := [\mathsf{enc}(m)^d \bmod N]$.

[5%] (a) How is verification performed in encoded RSA?

[5%] (b) Discuss why appropriate choice of encoding function for $\ell \ll \|N\|$ prevents the "no-message attack" described in Section 12.4.1.

[5%] (c) Show that encoded RSA is insecure if $\mathsf{enc}(m) = \mathtt{0x00}\|m\|0^{\kappa/10}$ (where $\kappa \stackrel{\text{def}}{=} \|N\|$, $\ell = |m| \stackrel{\text{def}}{=} 4\kappa/5$, and m is not the all-0 message). Assume $e = 3$.

[5%] (d) Show that encoded RSA is insecure for $\mathsf{enc}(m) = 0\|m\|0\|m$ (where $\ell = |m| \stackrel{\text{def}}{=} (\|N\| - 1)/2$ and m is not the all-0 message). Assume $e = 3$.

HOMEMADE Question: Defeating Rivest

Alice and Bob are a bit confused. They are going to use a Digital Signature scheme as a **Mac**. Let $\Pi = (\text{Gen}, \text{Sign}, \text{Vrfy})$ be a deterministic digital signature scheme (such as hashed RSA for instance). They run $\text{Gen}(1^n)$ to obtain (p_k, s_k) but only share and use s_k as the private-key of a **Mac**.

[5%] (A) Let $\Pi' = (\text{Gen}', \text{Mac}', \text{Vrfy}')$ be the **Mac** resulting from this idea. Used as a **Mac** they simply set $t := \text{Mac}'_{sk}(m) := \text{Sign}_{sk}(m)$. However, since they only use s_k, how will the receiver verify the message-tag pair (m, t) ? In other words, what is $\text{Vrfy}'_{sk}(m, t)$? Why did I underlined the word *deterministic* above ?

[5%] (B) Show that if Π is a digital signature scheme existentially unforgeable under an adaptive chosen-message attack then Π' is a Mac existentially unforgeable under an adaptive chosen-message attack (whether p_k is made public or not).

[5%] (C) Image that Alice and Bob use Π' as above, and that p_k is disclosed publicly. Explain how this defeats Rivest's argument seen in class that private-key authentication implies private-key encryption.

More on back...