Lab 2 Theory

The material for problem

This assignment is due **at 10:00pm ET** on **Thursday, October 14, 2021**.

Please make note of the following instructions:

- Remember that your solutions must be submitted on Gradescope. Please sign-up for 6.S060 Fall 2021 on Gradescope, with the entry code `KYRBPK`, using your MIT email.
- We require that the solution to the problems is submitted as a PDF file, **typeset on LaTeX**, using the template available on the course website (`https://6s060.csail.mit.edu/2021/`). Each submitted solution should start with your name, the course number, the problem number, the date, and the names of any students with whom you collaborated.

Problem 2-1. Message Signing [30 points]

Let $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ be a signature scheme with message space $\{0,1\}^k$ (where k is the security parameter), and let H be a seeded hash function with domain $\{0,1\}^*$ and range $\{0,1\}^k$. Consider the new signature scheme $(\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$, with message space $\{0,1\}^*$, defined via the following "hash-then-sign" paradigm:

- Gen' runs Gen to generate a pair (sk, vk) and samples a seed s for H. It outputs $sk' = (sk, s)$ and $vk' = (vk, s)$.
- Sig' takes as input a secret key $sk' = (sk, s)$ and a message M, and outputs a $\mathsf{Sig}(sk, H_s(M))$, i.e., it signs the hashed message $H_s(M)$.
- Ver', given the verification key $vk' = (vk, s)$, a message M, and a signature σ, outputs 1 if and only if $\mathsf{Ver}(vk, H_s(M), \sigma) = 1$.

(a) Suppose that $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ is secure (existentially against adaptive chosen message attack) then which of the following properties of H do we need to ensure that $(\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ is also secure?

1. One-wayness.
2. Target collision resistance
3. Collision resistance.

[15 points]

(b) Suppose that H is modeled as a random oracle. What is the minimal security notion we need of $(\mathsf{Gen}, \mathsf{Sig}, \mathsf{Ver})$ to ensure that $(\mathsf{Gen}', \mathsf{Sig}', \mathsf{Ver}')$ is secure existentially against adaptive chosen message attacks:

1. Security for any message (existential security) against adaptive chosen message attacks.
2. Security for random messages against adaptive chosen message attacks.
3. Security for any message (existential security) against random message attacks.
4. Security for random messages against random message attacks.

We encourage the students to refer to the lecture notes for the definitions of these security notions. [15 points]

Problem 2-2. Pseudo-Random Functions [40 points]

Let F be a pseudorandom function (PRF) that takes messages in $\{0,1\}^n$ to messages in $\{0,1\}^n$.

(a) Propose a way to use F to construct a PRF that takes messages in $\{0,1\}^n$ to messages in $\{0,1\}^{2n}$. [10 points]

(b) We wish to use F to construct a PRF that takes messages in $\{0,1\}^{2n}$ to messages in $\{0,1\}^{2n}$. Suppose the key to F is also in $\{0,1\}^n$.

Below are four proposals of such a PRF, where $x_0, x_1, K, K_0, K_1 \in \{0,1\}^n$ and where we use $||$ to denote concatenation. Notice that the constructions 1, 3, and 4 use a key in $\{0,1\}^n$ whereas the second construction uses a key in $\{0,1\}^{2n}$.

1. $G_1(K, x_0||x_1) = F(K, x_0)||F(K, x_1)$.
2. $G_2(K_0||K_1, x_0||x_1) = F(K_0, x_0)||F(K_1, x_1)$.
3. $G_3(K, x_0||x_1) = F(K', 0^n)||F(K', 1^n)$, where $K' = F(F(K, x_0), x_1)$
4. $G_4(K, x_0||x_1) = F(F(K, x_0), x_1)||F(F(K, x_0), x_1 \oplus 1^n)$.

Only one of the above four proposals is a secure PRF. Which one is the secure one? For each of the three others, show an attack that distinguishes it from a truly random function (recall the definition of a PRF given in the lecture). [30 points]

Problem 2-3. Certificate Authorities [30 points]

In lecture, we saw the role that certificate authorities (CAs) play in the public-key infrastructure. For this problem:

- let $(\mathsf{Gen}, \mathsf{Sign}, \mathsf{Ver})$ be a secure signature scheme whose public keys are 256 bits long
- let $H\colon \{0,1\}^* \to \{0,1\}^n$ be a collision-resistant hash function (where $n \approx 256$).

SecureCo offers the following service: any client can send SecureCo a pair of an email address and a public key $(\mathsf{addr}, \mathsf{pk})$. For $5, SecureCo will produce a signature $\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{CA}}, \mathsf{addr} \| \mathsf{pk})$ using SecureCo's secret key $\mathsf{sk}_{\mathsf{CA}}$. SecureCo will send the resulting signature σ to the email address addr. Here, addr is an ASCII string, the 256 bits of pk are represented as 32 raw bytes, and "$\|$" denotes string concatenation.

The tuple $(\mathsf{addr}, \mathsf{pk}, \sigma)$ is a simple form of certificate: anyone with the SecureCo's public key $\mathsf{pk}_{\mathsf{CA}}$ can check that $\mathsf{Ver}(\mathsf{pk}_{\mathsf{CA}}, \mathsf{addr} \| \mathsf{pk}, \sigma) = 1$ to confirm that SecureCo believes public key pk to be associated with email address addr.

(a) SecureCo's hardware signing device is slow, so the company wants to minimize the number of signatures it must make per day. One employee proposes the following "batch signing" strategy: the company will take a batch of B signing requests $(\mathsf{addr}_1, \mathsf{pk}_1), \ldots, (\mathsf{addr}_B, \mathsf{pk}_B)$, will compute:

$$m \leftarrow (\mathsf{addr}_1 \| \mathsf{pk}_1 \| \cdots \| \mathsf{addr}_B \| \mathsf{pk}_B)$$
$$\sigma \leftarrow \mathsf{Sign}(\mathsf{sk}_{\mathsf{CA}}, m)$$

and will send the pair (m, σ) to email addresses $(\mathsf{addr}_1, \ldots, \mathsf{addr}_B)$.

Now, for any $i \in \{1, \ldots, B\}$, SecureCo's client i can use the tuple $(\mathsf{addr}_i, \mathsf{pk}_i, m, \sigma)$ as a certificate. Anyone can verify the certificate by checking that $(\mathsf{addr}_i \| \mathsf{pk}_i)$ appears as a substring of m and that $\mathsf{Ver}(\mathsf{pk}_{\mathsf{CA}}, m, \sigma) = 1$. As in practice, the email addresses $(\mathsf{addr}_1, \ldots, \mathsf{addr}_B)$ are strings that are of variable length.

Unfortunately, this simple batch scheme is broken. Explain how a client who does not control `joe.biden@gmail.com` can obtain a SecureCo certificate for this email address with her own public key $\mathsf{pk}_{\mathsf{evil}}$. [10 points]

(b) Explain how to fix the batch-signing scheme to prevent this attack. Your scheme should take as input B pairs $(\mathsf{addr}_1, \mathsf{pk}_1), \ldots, (\mathsf{addr}_B, \mathsf{pk}_B)$, for any batch size B, should make a single invocation of the Sign algorithm, and should output B certificates—one per input pair. Explain why your solution is secure (i.e., no attacker can obtain a valid certificate for an address that it does not control). [10 points]

(c) The SecureCo certificates in this batch-signing scheme grows linearly with the batch size B. SecureCo's customers complain that the certificates are too long. Explain how SecureCo can produce much shorter batch certificates—of size $O(\log B)$ while still only requiring one signature per batch.

You should explain how SecureCo's signing process works, what the new type of SecureCo certificate contains, and how a client verifies it.
Hint: Use a collision-resistant hash function. [10 points]