Homework 8

Due: November 15, 2024

CS 1510: Intro. to Cryptography and Computer Security

1 Digital Signatures

Let $\mathcal{S} = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a secure digital signature scheme for k-bit messages.

Consider the following three signature schemes created using $\mathcal{S}$. Each will use the original Gen algorithm, but provide modified algorithms Sign' and Vrfy' that sign and verify variable-length messages that can be larger than k-bits.

1. **Scheme 1:** $\mathcal{S}^1 = (\mathsf{Gen}, \mathsf{Sign}^1, \mathsf{Vrfy}^1)$. For a given message, M, let $M = m_1 \| m_2 \| \dots \| m_n$, such that each m_i is of length k. Note that if M is not a multiple of k, then we will pad the end of M with extra 0s.

   Let $\mathsf{Sign}^1(sk, M) = (\mathsf{Sign}(sk, m_1),\ \mathsf{Sign}(sk, m_2),\ \dots,\ \mathsf{Sign}(sk, m_n))$. Thus, the output of S^1 is a vector of signatures, $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_n\}$.

   Define Vrfy^1 canonically:

   $\mathsf{Vrfy}^1(pk, M, \sigma) = (\mathsf{Vrfy}(pk, m_1, \sigma_1),\ \mathsf{Vrfy}(pk, m_2, \sigma_2),\ \dots,\ \mathsf{Vrfy}(pk, m_n, \sigma_n)) \stackrel{?}{=} 1^n$

   This means checking each σ with the corresponding message and outputting 1 only if all sub-verifications output 1.

2. **Scheme 2:** $\mathcal{S}^2 = (\mathsf{Gen}, \mathsf{Sign}^2, \mathsf{Vrfy}^2)$. For a given message, M, choose the smallest n such that $\lceil \log_2(n+1) \rceil + \lceil \frac{|M|}{n} \rceil \leq k$. (Assume that $|M|$ is small enough and k is large enough to make this possible.) Then break M up into $M = m_1 \| \dots \| m_n$, where each m_i is such that $|m_i| = k - \lceil \log_2(n+1) \rceil$, and m_n is padded with 0s as necessary.

   Let $\mathsf{Sign}^2(sk, M) = (\mathsf{Sign}(sk, 1 \| m_1),\ \mathsf{Sign}(sk, 2 \| m_2),\ \dots,\ \mathsf{Sign}(sk, n \| m_n))$, where each index i is represented using $\lceil \log_2(n+1) \rceil$ bits.

   Again, define Vrfy^2 canonically:

   $\mathsf{Vrfy}^2(pk, M, \sigma) = (\mathsf{Vrfy}(pk, 1 \| m_1, \sigma_1),\ \mathsf{Vrfy}(pk, 2 \| m_2, \sigma_2),\ \dots,\ \mathsf{Vrfy}(pk, n \| m_n, \sigma_n)) \stackrel{?}{=} 1^n$.

3. **Scheme 3:** For a given message, M, choose the smallest n such that $2\lceil \log_2(n+1) \rceil + \lceil \frac{|M|}{n} \rceil \leq k$. (Assume that $|M|$ is small enough and k is large enough to make this possible.) Then break M up into $M = m_1 \| \dots \| m_n$, where each m_i is such that $|m_i| = k - 2\lceil \log_2(n+1) \rceil$, and m_n is padded with 0s as necessary.

   Let $S^3(sk, M) = (\mathsf{Sign}(sk, n \| 1 \| m_1),\ \mathsf{Sign}(sk, n \| 2 \| m_2),\ \dots,\ \mathsf{Sign}(sk, n \| n \| m_n))$, where each index is represented using $\lceil \log_2(n+1) \rceil$ bits.

Let Vrfy^3 be defined canonically:

$$\mathsf{Vrfy}^3(pk, M, \sigma) =$$
$$(\mathsf{Vrfy}(pk, n\|1\|m_1, \sigma_1),\ \mathsf{Vrfy}(pk, n\|2\|m_2, \sigma_2),\ \ldots,\ \mathsf{Vrfy}(pk, n\|n\|m_n, \sigma_n)) \stackrel{?}{=} 1^n$$

a. Show that, by issuing just one query to the signer in **Scheme 1**, the adversary can succeed in forging a signature on a message of its choosing; find an attack that breaks **Scheme 1** but not **Scheme 2**.

b. Show that **Scheme 2** is still broken: by issuing just a single signing query to the signer in **Scheme 2**, the adversary can succeed in forging a signature on a message of its choosing; find an attack that breaks **Scheme 2** but not **Scheme 3**.

c. Show that **Scheme 3** is still broken: by issuing two signing queries to the signer in **Scheme 3**, the adversary can still succeed in forging a signature on a message of its choosing

d. Give a scheme that is based on **Scheme 3**, but is in fact secure (The only building block you're given is the signature scheme for k-bit messages. You may assume that k is sufficiently large—at least in the order of the security parameter.) Explain why your scheme fixes the vulnerability that is exhibited by **Scheme 3**, and prove it secure.

2 Signatures: From Weak to Strong

A signature scheme is **weakly secure** if the probability that a PPT adversary $\mathcal{A}$ wins the following game is negligible:

Signing query: On input 1^k, the adversary chooses the messages $M_1, \ldots, M_n$ to be signed.

Response: The challenger runs the key generation and the signing algorithms and sends to the adversary the public key pk and the signatures $\{\sigma_j\}_{j=1}^n$ on the adversary's messages.

Forgery: The adversary outputs a message M^* and a signature σ^* and wins the game if M^* was not included in its signing query, and yet the verification algorithm accepts the signature σ^*.

The key difference for a weakly secure scheme is that the adversary must submit their messages all at once rather than adaptively asking for messages to be signed (i.e. they

submit messages to be signed one-by-one and can thus use responses from previous queries to inform their next message query).

A signature scheme is **one-time secure** if the probability that a PPT adversary $\mathcal{A}$ wins the following game is negligible:

Key generation: The challenger runs the key generation algorithm and generates (pk, sk).

Signing query: On input pk, the adversary chooses a message m to be signed.

Response: The challenger computes a signature σ on m using the signing algorithm, and returns it to the adversary.

Forgery: The adversary outputs a message m^* and a signature σ^* and wins the game if $m^* \neq m$, and yet the verification algorithm accepts the signature σ^*.

Given a weakly secure signature scheme $(\mathsf{Gen}_{\mathsf{weak}}, \mathsf{Sign}_{\mathsf{weak}}, \mathsf{Vrfy}_{\mathsf{weak}})$ and a one-time secure signature scheme $(\mathsf{Gen}_{\mathsf{one\text{-}time}}, \mathsf{Sign}_{\mathsf{one\text{-}time}}, \mathsf{Vrfy}_{\mathsf{one\text{-}time}})$, construct a secure signature scheme. Let the message space and key space for all the signature schemes here be binary strings of length k, the security parameter, and let the weakly secure scheme sign $n = p(k)$ messages for any polynomial p. Don't forget to prove that your construction is correct and secure.

3 Deterministic Digital Signature

Assume the existence of a digital signature scheme $\Pi = (\mathsf{Gen}, \mathsf{Sign}, \mathsf{Verify})$ for which Sign is a probabilistic algorithm. Construct a digital signature scheme $\Pi' = (\mathsf{Gen}', \mathsf{Sign}', \mathsf{Verify}')$ where Sign' is deterministic, and prove it is correct and secure. You may use one-way functions (or any other symmetric-key primitive implied by one-way functions) in your construction.

4 Random Oracles and RSA-FDH Discussion

The following are reflection questions that you may answer informally, with your understanding and intuition rather than mathematical arguments and proofs.

a. What is the fundamental modeling difference between a hash function and a random oracle? Specifically, where in the security reduction would the hash function exist, vs. where does the random oracle exist?

b. Why is it crucial to replace the hash function in RSA-FDH with a random oracle in order for the security reduction to go through?

c. What security properties would we need from the hash function in RSA-FDH to avoid the attacks discussed on Slide 17 in Lecture 17 (or Slide 4 in Lecture 18)? How are these properties guaranteed by the random oracle?

d. In practice, "oracles" do not exist—participants run cryptographic protocols internally. Random oracles, for example, are instantiated with hash functions. What does this mean for the practical security of the cryptographic protocols like RSA-FDH and Schnorr signatures, which are widely used in practice?

5 Summary Question

Summarize the most important insights from this week's material, including from the lectures, notes, textbooks, homework problems, and other resources you find helpful, into a one-page resource. We expect that these summary pages will help you with the take-home midterm and final. Please note this question is graded based on completion—we will not be checking it for correctness. However, we will be checking it for relevance to the week's content and length.