Problem sheet 9 for Course 02231, 2025

These practice problems have the purpose of helping you understand the material better and learning the skills that are necessary to analyze cryptographic constructions, and sometimes to prepare you for the next class. All answers should be supported by a written justification. To gauge whether a justification is sufficient, ask yourself if your peers would be convinced by it without additional explanations.

Exercise 1. (Secure digital signatures for messages of arbitrary length)

In the lecture you saw a proof sketch how RSA signatures for arbitrary-length messages are constructed. Because plain RSA signatures are not secure, not even for fixed-length messages, we had to use the Random Oracle Model (ROM). As it turns out, the ROM is not necessary when using a hash function and a secure digital signature scheme (DSS) to construct a DSS for arbitrary-length messages. Assume that $\Sigma = (\Sigma.\mathsf{KeyGen}, \Sigma.\mathsf{Sign}, \Sigma.\mathsf{Ver})$ is a secure DSS for messages of length n bits. Furthermore, assume that H is a collision-resistant hash function mapping $\{0,1\}^*$ to $\{0,1\}^n$.

1. Construct a signature scheme which can sign messages from $\{0,1\}^*$, using the given signature scheme and H, following the idea of RSA-FDH.
2. Show that your constructed signature scheme is EUF-CMA secure, given that Σ is secure and H is collision-resistant.

Exercise 2. (Non-Repudiation)

In the lecture, it was mentioned that secure DSS have the non-repudiation property. Here, we explore this property a bit more explicitly.

1. Show that the following holds: Assume that a signer uses a digital signature scheme Σ, has generated a public key vk for it but never releases the signing key sk to anyone. Furthermore, assume that there exists an efficient algorithm $\mathcal{A}$ which generates a valid signature verifiable under vk that was never generated by the signer (i.e. the signature scheme does not provide non-repudiation). Then $\mathcal{A}$ can be used to efficiently distinguish $L_{sig-real}$ and $L_{sig-fake}$ for the digital signature scheme.
2. Why does the same argument not apply for a single message sender in a MAC scheme, even though it's security is essentially defined in the same way?

Exercise 3. (**Why we need hashes for RSA signatures**)

It might be tempting to design RSA signatures without using a cryptographic hash function H. The idea could be the following, alternative signing algorithm:

1. The input is a message $m \in \mathbb{Z}_N^*$ such that $m < \lfloor N/2^{16} \rfloor$ (i.e. the first two bytes of m are 00).
2. We then sign this m by computing $\sigma = m^d \bmod N$.

For verification, we now do the following:

1. Check that the message $m \in \mathbb{Z}_N^*$ is of the correct form, namely that it is smaller than $\lfloor N/2^{16} \rfloor$.
2. Check that $\sigma^e = m \bmod N$.

Let us assume that raising to the eth power modulo N is a perfect permutation, i.e. on input $x \in \mathbb{Z}_N^*$ the value $x^e \bmod N$ is uniformly random in $\mathbb{Z}_N^*$. Show that an attacker, by only using the public key pk, can break the EUF-CMA property for the aforementioned "signature" scheme by making around 2^{16} exponentiations modulo N.

Exercise 4. (**Hashing to $\mathbb{Z}_N^*$**)

In the lecture, we assumed that the hash function H used in the RSA-FDH construction outputs a value in $\mathbb{Z}_N^*$ for every message m. It might appear unrealistic to construct such a hash function, because this is a very specific requirement on the output set.

Show that using a hash function $\hat{H}$ which outputs random values from $\mathbb{Z}_N$ instead of $\mathbb{Z}_N^*$ is sufficient for the task. To do so, assume that p, q are both of size around 2^{1000} (i.e. 1000 bits long), and calculate how many outputs of $\hat{H}$ will lie outside of $\mathbb{Z}_N^*$ by not being coprime to N. To do so, you can count the total number of integers in $[0, N-1]$ that are not coprime to N in a clever way.

Bonus: Assume that $\hat{H}$ is outputting values in $\mathbb{Z}_N \setminus (\mathbb{Z}_N^* \cup \{0\})$ with a high probability over its possible inputs. Show that you can use it to factor N efficiently with essentially the same probability.