**COMP 537: Cryptography** **Fall 2024**

# Problem Set 3

**Instructions:** You **must** typeset your solution in LaTeX using the provided template. Please submit your problem set via Gradescope. Include your name and the names of any collaborators at the top of your submission.

**Acknowledgment:** Several of the problems in this problem set come from the Boneh-Shoup textbook.

**Problem 1: DDH PRG [15 points].** Let $\mathbb{G}$ be a cyclic group of prime order $q$ generated by $g \in \mathbb{G}$. Consider the following PRG defined over $(\mathbb{Z}_q^2, \mathbb{G}^3)$:

$$G(\alpha, \beta) := (g^\alpha, g^\beta, g^{\alpha\beta}).$$

(a) Prove that $G$ is a secure PRG assuming that the DDH assumption holds in $\mathbb{G}$.

(b) Let $\mathbb{G}$ be the prime order order subgroup of $\mathbb{Z}_{23}^*$ generated by $g = 2$. What are the elements of the group $\mathbb{G}$, and what is the order $q$ of group $\mathbb{G}$? In addition to stating the answer, please explain why this is the answer.

(c) Let $\mathbb{G}$ be the prime order order subgroup of $\mathbb{Z}_{23}^*$ generated by $g = 2$. What is the output of $G(5,6)$?

**Note:** you should be able to do this without a calculator, but it's ok to use one if you want.

**Problem 2: Non-Binding Signatures [15 points].** It turns out that secure signatures are not necessarily *binding*. That is, suppose the signer generates a signature $\sigma$ on a message $m$. The definition of a secure signature does not preclude the signer from producing another message $m' \neq m$ for which $\sigma$ is a valid signature. It turns out binding isn't needed for many applications, so it's left out of the definition. That said, many signature schemes we have seen are in fact binding.

(a) Please give an example of a signature scheme that is *not* binding: for a given $(\mathsf{pk}, \mathsf{sk})$, the signer can find two distinct messages $m_0$ and $m_1$ where the same signature $\sigma$ is valid for both messages under $\mathsf{pk}$. You may use an underlying secure signature scheme as part of your construction, and you can pick the message space for your scheme.

**Hint:** Consider using a hash and sign approach to building the signature scheme but with the discrete log-based hash function we discussed in class.

(b) Give the intuition for why the scheme is secure. You should state the relevant assumptions and why they are needed, but you don't need to give a proof.

(c) Describe how the signer can produce a second message for the same signature.

**Problem 3: El-Gamal and CCA Security [10 points].** In class we saw an attack on the CCA security of El-Gamal encryption. Consider the following (failed) attempt to strengthen El-Gamal encryption to defend against CCA attacks.

We saw in class that El-Gamal ciphertexts are malleable: given a ciphertext $(c_0, c_1)$, an attacker can multiply some value $v$ into $c_1$ and send this "fresh" ciphertext to the CCA decryption oracle. Then the attacker will distinguish if it gets back a decryption of $m_0 \cdot v$ or $m_1 \cdot v$. To defend against this, we introduce a variant of El-Gamal encryption that authenticates the message inside the ciphertext, using a MAC scheme where $\mathcal{K} = \mathcal{M} = \mathcal{T} = \mathbb{G}$. Our scheme will effectively consist of three El-Gamal encryptions: one is a standard encryption of the message $m$, one is a random value that acts as a symmetric key $k$, and one is a MAC on $m$ using the key $k$. This way, an attacker who tries to change part of the ciphertext will cause MAC verification to fail, resulting in the decryption oracle always returning $\perp$. Concretely, our scheme works as follows:

```
$\underline{\mathsf{Enc}(\mathsf{pk}, m)}$:
$k \xleftarrow{R} \mathbb{G}$
$t \leftarrow \mathsf{MAC.Sign}(k, m)$
$r_1, r_2, r_3 \xleftarrow{R} \mathbb{Z}_p^3$
output $(g^{r_1}, g^{r_2}, g^{r_3}, \mathsf{pk}^{r_1} \cdot m, \mathsf{pk}^{r_2} \cdot k, \mathsf{pk}^{r_3} \cdot t)$
```

```
$\underline{\mathsf{Dec}(\mathsf{sk}, c)}$:
$(c_0, c_1, c_2, c_3, c_4, c_5) \leftarrow c$
$m \leftarrow c_3 / c_0^{\mathsf{sk}}$
$k \leftarrow c_4 / c_1^{\mathsf{sk}}$
$t \leftarrow c_5 / c_2^{\mathsf{sk}}$
if $\mathsf{MAC.Verify}(k, m, t) = 0$:
  return $\perp$
else: return $m$
```

Unfortunately, the reasoning above is flawed, and this scheme is completely broken.

(a) Show an attack on the CCA security of this scheme.

(b) What is the flaw in the reasoning given for why this scheme has CCA security?

**Problem 4: RSA Signatures with Same Modulus [15 points].** This problem explores why every party has to be assigned a different modulus $N = pq$ in the RSA trapdoor permutation. Suppose we try to use the same modulus $N = pq$ for everyone. Every party is assigned a public exponent $e_i \in \mathbb{Z}$ and a private exponent $d_i \in \mathbb{Z}$ such that $e_i \cdot d_i = 1 \bmod \varphi(N)$. At first, this appears to work fine. To sign a message, $m \in \mathcal{M}$, Alice would publish the signature $\sigma_a \leftarrow H(m)^{d_a} \in \mathbb{Z}_N$ where $H : \mathcal{M} \to \mathbb{Z}_N^*$ is a hash function. Similarly, Bob would publish the signature $\sigma_b \leftarrow H(m)^{d_b} \in \mathbb{Z}_N$. Since Alice is the only one who knows $d_a$ and Bob is the only one who knows $d_b$, this seems fine.
Unfortunately, this scheme is completely insecure. Bob can use his secret key $d_b$ to sign messages on behalf of Alice.

(a) Show that Bob can use his public-private key pair $(e_b, d_b)$ to obtain a multiple of $\varphi(N)$. Denote this integer by $V$.

(b) Suppose Bob knows Alice's public key $e_a$, and assume for now that $e_a$ is relatively prime to $V$. Show that for any message $m \in \mathcal{M}$, Bob can compute $\sigma \leftarrow H(m)^{1/e_a}$. In other words, Bob can invert Alice's trapdoor permutation and obtain her signature on $m$.

**Hint.** Recall since $e_a$ and $V$ are relatively prime, Bob can find an integer $d$ such that $d \cdot e_a = 1 \bmod V$, i.e., Bob can compute the inverse of $e_a$ mod $V$.

(c) Show how to make your solution in part (b) work even if $e_a$ is not relatively prime to $V$.

**Optional Feedback [5 points].** Please answer the following questions to help design future problem sets. You are not required to answer these questions (the points are free), and if you would prefer to answer anonymously, please use the anonymous feedback form. However, we do encourage you to provide feedback on how to improve the course experience.

(a) Roughly how long did you spend on this problem set?

(b) What was your favorite problem on this problem set?

(c) What was your least favorite problem on this problem set?

(d) Any other feedback for this problem set? Was it too easy/difficult?

(e) Any other feedback on the course so far?