

UNIVERSITY
OF OSLO

TEK4500 – Introduction to Cryptography

Problem Set 10

Fall 2026

Problem 1

Read Chapter 16 and Chapter 19 in [Ros] (16.2 + 19.3 can be skipped).

Problem 2

A simplified variant of the Schnorr signature scheme, which we call the *Simplified Schnorr scheme*, is shown in Fig. 1. It is defined over a cyclic group $(\mathbf{G}, \star) = \langle g \rangle$ having prime order p. Note that the message space of this simplified scheme is $\mathbf{Z}_p^*$, i,e, the integers 1 to $p-1$. As a convention we use uppercase letters for the elements in the group $\mathbf{G}$ (except for the generator element g) and lowercase letters for integers.

KeyGen:
1: $a \twoheadleftarrow \{1, \ldots, |\mathbf{G}|\}$
2: $A \leftarrow g^a$
3: **return** (a, A)

Sign$(d, m \in \mathbf{Z}_p^*)$:
1: $r \twoheadleftarrow \{1, \ldots, |\mathbf{G}|\}$
2: $R \leftarrow g^r$
3: $z \leftarrow a \cdot m + r \pmod{|\mathbf{G}|}$
4: **return** (R, z)

Vrfy(A, m, σ):
1: Parse σ as (R, z)
2: **return** $A^m \star R \stackrel{?}{=} g^z$

Figure 1: The Simplified Schnorr signature scheme. Based on a cyclic group $(\mathbf{G}, \star) = \langle g \rangle$ of prime order p.

a) Show that Simplified Schnorr is a correct signature scheme, i.e., for every key $(a, A) \twoheadleftarrow$ KeyGen and every message $m \in \mathbf{Z}_p^*$, show that $\mathsf{Vrfy}(A, m, \mathsf{Sign}(a, m)) = 1$.

b) List all the subgroups of $(\mathbf{Z}_{23}^*, \cdot)$.

Note: A *subgroup* of a group $(\mathbf{G}, \cdot)$ is a subset $\mathbf{H} \subseteq \mathbf{G}$, such that $(\mathbf{H}, \cdot)$ forms a group in its own right when "inheriting" the operation from the "parent" group $(\mathbf{G}, \cdot)$. Basically, $(\mathbf{H}, \cdot)$ is a group that lives "inside" $(\mathbf{G}, \cdot)$.

Note that not all subsets of $(\mathbf{Z}_{23}^*, \cdot)$ constitute a subgroup. For example $\{2, 3, 4\}$ is not a subgroup under $\cdot$ (it lacks an identity element), nor is $\{1, 2, 3, 4\}$ (the product $2 \cdot 3 = 6 \pmod{23}$ is not again an element in the subset).

Hint: Lagrange's Theorem.

In the following subproblems suppose we instantiate the Simplified Schnorr signature scheme with the subgroup $(\mathbf{H}, \cdot) = \langle 2 \rangle$.

c) What is the order of $(\mathbf{H}, \cdot)$?. That is, how many elements does it have?

d) Suppose we use $a = 5$ as our private signing key. Compute public verification key.

e) Suppose during signing of the message $m = 8$ we draw the random element $k = 7$. What is the corresponding signature on m?

f) Verify the signature computed in e).

g) Unfortunately, Simplified Schnorr is *not secure*! Show how you can break Simplified Schnorr by forging on an arbitrary message $m \in \mathbf{Z}_p^*$.

Another problem with Simplified Schnorr (beside it being completely insecure!) is that the message space is limited to the integers in $\mathbf{Z}_p^*$. In real life we want to sign arbitrary bit strings of any length, i.e., we want our message space to be $\{0,1\}^*$. As always, the solution is to use a hash function: the *actual* Schnorr scheme uses a hash function $H : \mathbf{G} \times \mathbf{G} \times \{0,1\}^* \to \mathbf{Z}_p^*$ to map a *triple* of elements (A, R, M) to an element in $\mathbf{Z}_p^*$, where A and R are group elements and M is the (bit string) message. The signing algorithm of actual Schnorr is shown below.

$\underline{\mathsf{Sign}(a, A \in \{0,1\}^*)}$:
1: $r \twoheadleftarrow \{1, \ldots, |\mathbf{G}|\}$
2: $R \leftarrow g^r$
3: $b \leftarrow H(A, R, M)$
4: $z \leftarrow a \cdot b + r \pmod{p}$
5: **return** (R, z)

h) Describe the corresponding verification algorithm of the actual Schnorr signature scheme and show that the scheme is correct.

i) What happens if you try to run your attack from g) on the actual Schnorr scheme?

Problem 3

a) The Schnorr signature scheme (both simplified and actual) has a very sharp edge: if the same random value k is ever used to sign two different messages then an attacker can obtain the private signing key a! Show this.

Hint: Suppose you are given two signatures $\sigma = (R, z)$ and $\sigma' = (R', z')$ that both used the same value r during signing. What is $s - s'$?

b) Given the catastrophic failure mode of Schnorr on k reuse it would be good if we didn't have to rely on any randomness at all. And this turns out to be possible! To do this, on Line 1 of the (actual) Schnorr Sign algorithm, instead of picking k at random, we instead derive it as

1: $r \leftarrow G(sk, M)$
2: ...

where $sk = a$ is the long-term private signing key of Schnorr, M is the message to be signed, and G is a hash function. Explain why this solves the problem of r reuse.

c) Unfortunately, it turns out that making Schnorr deterministic also makes it more vulnerable to certain side-channel attacks that are able to measure the power drawn while signing. Suggest a way of bringing non-determinism back to deterministic Schnorr, but without re-introducing the r-reuse problem.

Problem 4

Let $\Sigma = (\mathsf{KeyGen}, \mathsf{Sign}, \mathsf{Vrfy})$ be a UF-CMA secure signature scheme. Suppose we generate two key pairs (sk_1, vk_1) and (sk_2, vk_2) using the key generation algorithm KeyGen. In the following sub-problems we consider three new signature schemes Σ_1, Σ_2, Σ_3 that use these two keys in different ways.

a) Let Σ_1 be the signatures scheme defined in Fig. 2a. This scheme signs the message with each signing key, and verification returns true if *at least one* of the signatures is valid for the message (under the corresponding verification key). Is Σ_1 UF-CMA secure? If yes, give a high-level justification; if no, give a concrete attack and calculate the advantage.

b) Let Σ_2 be the signatures scheme defined in Fig. 2b. This scheme splits the message into two halves and signs the left part with sk_1 and the right part with sk_2. Verification returns true if *both* signatures are valid. (for each respective halves). Is Σ_2 UF-CMA secure? If yes, give a high-level justification; if no, give a concrete attack and calculate the advantage.

$\underline{\mathsf{Sign}_1(sk = (sk_1, sk_2), M)}$:
1: $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, M)$
2: $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, M)$
3: **return** (σ_1, σ_2)

$\underline{\mathsf{Vrfy}_1(vk = (vk_1, vk_2), M, \sigma)}$:
1: Parse σ as (σ_1, σ_2)
2: **return**
$\mathsf{Vrfy}(vk_1, M, \sigma_1)$ **or**
$\mathsf{Vrfy}(vk_2, M, \sigma_2)$

(a)

$\underline{\mathsf{Sign}_2(sk = (sk_1, sk_2), M)}$:
1: Parse M as (M_1, M_2)
2: $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, M_1)$
3: $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, M_2)$
4: **return** (σ_1, σ_2)

$\underline{\mathsf{Vrfy}_2(vk = (vk_1, vk_2), M, \sigma)}$:
1: Parse M as (M_1, M_2)
2: Parse σ as (σ_1, σ_2)
3: **return**
$\mathsf{Vrfy}(vk_1, M_1, \sigma_1)$ **and**
$\mathsf{Vrfy}(vk_2, M_2, \sigma_2)$

(b)

$\underline{\mathsf{Sign}_3(sk = (sk_1, sk_2), M)}$:
1: $R \twoheadleftarrow \{0,1\}^{|M|}$
2: $\sigma_1 \leftarrow \mathsf{Sign}(sk_1, R)$
3: $\sigma_2 \leftarrow \mathsf{Sign}(sk_2, M \oplus R)$
4: **return** (R, σ_1, σ_2)

$\underline{\mathsf{Vrfy}_3(vk = (vk_1, vk_2), M, \sigma)}$:
1: Parse σ as (R, σ_1, σ_2)
2: **return**
$\mathsf{Vrfy}(vk_1, R, \sigma_1)$ **and**
$\mathsf{Vrfy}(vk_2, M \oplus R, \sigma_2)$

(c)

Figure 2: Signature schemes for Problem 4.

c) Let Σ_3 be the signatures scheme defined in Fig. 2c. This scheme creates a random string of equal length to the message, signs the random string with sk_1, and signs the XOR of the message and the random string with sk_2. Verification returns true if *both* signatures are valid (for their respective verification keys). Is Σ_3 UF-CMA secure? If yes, give a high-level justification; if no, give a concrete attack and calculate the advantage.

Problem 5 [Problem 16.12 in [Ros]]

This exercise explores whether we can convert a IND-CPA-secure PKE into an IND-CCA-secure one, by incorporating signatures similar to how MACs promote IND-CPA security to IND-CCA/AE security for symmetric encryption schemes.

a) Let Σ be an IND-CPA secure PKE with $\Sigma.\mathcal{M} = \{0,1\}^n$. Prove that the following PKE is correct and also IND-CPA secure.

$\underline{\Sigma'.\mathsf{KeyGen}}$:
1: $(sk, pk) \twoheadleftarrow \Sigma.\mathsf{KeyGen}$
2: **return** (sk, pk)

$\underline{\Sigma'.\mathsf{Enc}(pk, M)}$:
1: $C \leftarrow \Sigma.\mathsf{Enc}(pk, M)$
2: **return** $(C, 1^n, 0^n)$

$\underline{\Sigma'.\mathsf{Dec}(sk, (C, R, S))}$:
1: $M \leftarrow \Sigma.\mathsf{Dec}(sk, C)$
2: // $\wedge$ denotes bitwise-and
3: **return** $(M \wedge R) \oplus S$

b) Now consider the following natural approach to authenticate ciphertexts using a signature: Encrypt the plaintext along with a verification key, and then sign the resulting ciphertext. More formally, let Σ be a PKE scheme and Π be a signature scheme, and define the following PKE:

$\underline{\Sigma''.\mathsf{KeyGen}}$:
1: $(sk, pk) \twoheadleftarrow \Sigma.\mathsf{KeyGen}$
2: **return** (sk, pk)

$\underline{\Sigma''.\mathsf{Enc}(pk, M)}$:
1: $(sk, vk) \twoheadleftarrow \Pi.\mathsf{KeyGen}$
2: $C \leftarrow \Sigma.\mathsf{Enc}(pk, M \| vk)$
3: $\sigma \leftarrow \Pi.\mathsf{Sign}(sk, C)$
4: **return** (C, σ)

$\underline{\Sigma''.\mathsf{Dec}(sk, (C, \sigma))}$:
1: $M \| vk \leftarrow \Sigma.\mathsf{Dec}(sk, C)$
2: **if** $\Pi.\mathsf{Vrfy}(vk, C, \sigma) = 1$ **:**
3: **return** M
4: **else**
5: **return** $\perp$

Show that Σ'' is not an IND-CCA secure PKE when Σ is the PKE Σ' from a) and Π is any UF-CMA secure signature scheme.

References

[Ros] Mike Rosulek. *The Joy of Cryptography*. https://joyofcryptography.com.