Introduction to Cryptography

TEK 4500 (Fall 2022)
Problem Set 10

Problem 1.

Read Chapter 12 in [BR] (Section 12.3.6 can be skipped) and Chapter 10 in [PP] (Section 10.3 can be skipped).

Problem 2.

Given an instance of the Textbook RSA signature scheme (Fig. 1) with public verification key $vk = (e, n) = (131, 9797)$, which of the following signatures are valid?

a) $(M, \sigma) = (123, 6292)$

b) $(M, \sigma) = (4333, 4768)$

c) $(M\sigma) = (4333, 1424)$

Problem 3.

Given the same Textbook RSA instance as in Problem 2, make a forgery on the message $M = 1234$. Suppose you're in the UF-CMA setting, i.e., you have access to a signing oracle that returns signatures on messages of your choice.

Problem 4. [Katz & Lindell]

Consider a padded RSA signature scheme where the public key is (n, e) as usual, and a signature on a message $m \in \{0,1\}^\ell$ is computed by choosing uniform $r \in \{0,1\}^{2n-\ell-1}$ and outputting $(r\|m)^d \pmod n$.

a) How is verification done in this scheme?

b) Show that this padded RSA variant is not secure.

RSA.KeyGen:

$p, q \xleftarrow{\$}$ two large prime numbers
$n \leftarrow p \cdot q$
$\phi(n) = (p-1) \cdot (q-1)$
choose $e \in \mathbf{Z}^*_{\phi(n)}$
$d \leftarrow e^{-1} \pmod{\phi(n)}$
$sk \leftarrow (d, n)$
$vk \leftarrow (e, n)$
return (sk, vk)

RSA.Sign(sk, M):

Parse sk as (d, n)
$\sigma \leftarrow M^d \pmod{n}$
return σ

RSA.Vrfy(vk, M, σ):

Parse vk as (e, n)
if $\sigma^e = M \pmod{n}$**:**
  return 1
else
  return 0

Figure 1: The Textbook RSA signature scheme.

Problem 5.

The Schnorr signature scheme, like ElGamal encryption and the Diffie-Hellman protocol, is based on the discrete log problem. A simplified variant of the Schnorr signature scheme is shown in Fig. 2. It is defined over a cyclic group $(G, \star) = \langle g \rangle$ having prime order q. Note that the message space of this simplified scheme is $\mathbf{Z}_q$, i,e, the integers 0 to $q-1$. As a convention we use uppercase letters for the elements in the group G (except for the generator element g) and lowercase letters for integers.

a) Show that Simplified Schnorr is a correct signature scheme, i.e., for every key $(d, D) \xleftarrow{\$}$ KeyGen and every message $m \in \mathbf{Z}_q$, show that Vrfy$(D, m, $Sign$(d, m)) = 1$.

Let $p = 2 \cdot 11 + 1 = 23$ and consider the group $(\mathbf{Z}^*_{23}, \cdot)$

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

Let $G < (\mathbf{Z}^*_{23}, \cdot)$ be the subgroup of $(\mathbf{Z}^*_{23}, \cdot)$ having order $q = 11$ and assume we use 2 as the generator of G. In the following subproblems suppose we instantiate the Simplified Schnorr signature scheme with the group $G = \langle 2 \rangle$.

KeyGen:

$d \xleftarrow{\$} \{1, \ldots, q\}$
$D \leftarrow g^d$
return (d, D)

Sign$(d, m \in \mathbf{Z}_q)$:

$k \xleftarrow{\$} \{1, \ldots, q\}$
$R \leftarrow g^k$
$s \leftarrow dm + k \pmod{q}$
return (R, s)

Vrfy(D, m, σ):

Parse σ as (R, s)
return $D^m \star R \stackrel{?}{=} g^s$

Figure 2: The Simplified Schnorr signature scheme.

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

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

e) Verify the signature computed in d).

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

Another problem with Simplified Schnorr (beside it being completely insecure!) is that the message space is limited to the integers in $\mathbf{Z}_q$. 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 hash functions. Concretely, the *actual* Schnorr scheme also uses a hash function $H : G \times \{0,1\}^* \rightarrow \mathbf{Z}_q$, i.e. the hash function takes in a *pair* of elements (X, M) where $X \in G$ is a group element and $M \in \{0,1\}^*$ is the message. The signing algorithm of actual Schnorr is shown below.

Sign$(d, M \in \{0,1\}^*)$:

1: $k \xleftarrow{\$} \{1, \ldots, q\}$
2: $r \leftarrow H(g^k, M)$
3: $s \leftarrow dr + k \pmod q$
4: **return** (r, s)

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

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

Problem 6.

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 d! Show this.

Hint: Suppose you are given two signatures $\sigma = (r, s)$ and $\sigma' = (r', s')$ that both used the same value k 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: $k \leftarrow H(sk, M)$
2: ...

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

References

[BR] Mihir Bellare and Phillip Rogaway. *Introduction to Modern Cryptography*. https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf.

[PP] Christof Paar and Jan Pelzl. *Understanding Cryptography - A Textbook for Students and Practitioners*. Springer, 2010.