

## CAS CS 538. Solutions to Problem Set 5

**Due electronically via gradescope, Tuesday February 24, 2026 11:59pm**  
**Om Khadka, U51801771**

**Problem 1.** (50 points)

In Discussion 5 we introduced the Signal symmetric ratchet and we showed that even if a subsequent key is leaked to the adversary, it is hard to recover the previous key. In this problem, we will prove something stronger: even if a subsequent key is leaked (and the previous key has been deleted), past encryptions are still semantically secure.

Specifically, let  $\mathcal{E} = (\text{Enc}, \text{Dec})$  be a semantically secure cipher over  $(\mathcal{K}, \mathcal{M}, \mathcal{C})$ . Let  $G : \mathcal{S} \rightarrow \mathcal{S} \times \mathcal{K}$  be a secure PRG such that  $G : \{0, 1\}^\ell \rightarrow \{0, 1\}^{2\ell}$ . Define the symmetric ratchet scheme as  $\text{Enc}^1(s, m) = \text{Enc}(G(s)_2, m)$  and  $\text{Dec}^1(s, c) = \text{Dec}(G(s)_2, c)$ . Here  $G(s)_2$  denotes the second part of the output of  $G(s)$ , i.e. the key part. We refer you to Figure 1 of Discussion 5 for a diagram of the symmetric ratchet.

At time step  $n$ , participants will advance the state of the ratchet by computing  $(s_n, k_n) \leftarrow G(s_{n-1})$ . Then, the past is erased by deleting the old seed  $s_{n-1}$  from memory. The new ciphertext is computed as  $c_n = \text{Enc}(k_n, m_n)$ .

Prove that  $\mathcal{E}^1$  is *semantically secure* for the previous encryption  $c_{n-1}$  even if a subsequent seed and key  $(s_n, k_n)$  is leaked to the adversary.

**Getting started.** To get started, you should consider an adversary  $\mathcal{A}$  playing a new game called the SSKeyLeak game against  $\mathcal{E}^1$ . The SSKeyLeak is the same as the Semantic Security Game (Boneh + Shoup Attack Game 2.1) except that in addition to a ciphertext  $c_{n-1}$ , the adversary *also* receives a leaked seed and key  $(s_n, k_n)$  where  $s_n, k_n$  are pseudorandom values produced by  $G^n$  with an initial random seed.

In particular, say  $\mathcal{A}$  outputs two messages  $m_0, m_1$ . Let Game 0 be when  $c_{n-1} = \text{Enc}^1(k_{n-1}, m_0)$  and Game 1 be when  $c_{n-1} = \text{Enc}(k_{n-1}, m_1)$ .  $\mathcal{A}$  receives the tuple  $(c_{n-1}, s_n, k_n)$  and outputs a bit  $\hat{b}$ . Let  $W_b$  be the event that  $\hat{b} = 1$  in Game  $b$  of the SSKeyLeak game. Then we define the key-leakage advantage as  $\text{SSKeyLeakadv}[\mathcal{A}, \mathcal{E}^1] = |\Pr[W_0] - \Pr[W_1]|$ .

**Building a hybrid argument.** Use a hybrid argument to prove that  $\mathcal{E}^1$  is semantically secure against leaked keys by showing that  $\text{SSKeyLeakadv}[\mathcal{A}, \mathcal{E}^1]$  is negligible.

*Hint: you will have to introduce two hybrid games in between Game 0 and Game 1 of SSKeyLeak game. Then you will give three reductions with wrapper adversaries  $\mathcal{B}_1, \mathcal{B}_2$ , and  $\mathcal{B}_3$ . In no particular order, two of these reductions yield a PRG distinguisher for  $G^n$  and one of them yields an SS adversary for  $\mathcal{E}$ .*

**Solution.** The whole idea behind how I'll prove this by essentially splitting up the whole process into a bunch of hybrid arguments, and showing how the sum of all of the reductions end up being negligible.

Let  $G^n : \{0, 1\}^\ell \rightarrow \{0, 1\}^{(n+1)\ell}$  be the  $n$ -wise sequential composition of  $G$ , outputting  $(k_1, s_1, \dots, k_n, s_n)$ . Also assume that  $G$  is a secure PRG, thereby also meaning that  $G^n$  is also secure.

### The Hybrid Games

**Game 0:** The initial real SSKeyLeak game w/ bit  $b = 0$ .

1. Challenger samples  $s_0 \leftarrow \{0, 1\}^\ell$ .
2. Calculates  $(k_1, s_1, \dots, k_n, s_n) \leftarrow G^n(s_0)$ .
3.  $\mathcal{A}$  then outputs  $m_0, m_1$ .
4. Challenger computes  $c_{n-1} = \text{Enc}(k_{n-1}, m_0)$ .
5. Challenger then gives  $(c_{n-1}, k_n, s_n)$  to  $\mathcal{A}$ .
6.  $\mathcal{A}$  returns  $\hat{b}$ .

Let  $\Pr[W_0]$  denote the probability of  $\mathcal{A}$  outputting 1.

**Hybrid 1:** Same as Game 0, but now the keys and seeds are replaced w/ rnd uniform values.

1. Challenger samples  $(k_1, \dots, k_n, s_n) \leftarrow \{0, 1\}^{(n+1)\ell}$  uniformly at random.
2.  $\mathcal{A}$  outputs  $m_0, m_1$ .
3. Challenger then Computes  $c_{n-1} = \text{Enc}(k_{n-1}, m_0)$ .
4. Challenger will then give  $(c_{n-1}, k_n, s_n)$  to  $\mathcal{A}$ .
5.  $\mathcal{A}$  finally returns  $\hat{b}$ .

$\Pr[H_1]$  will be the probability of  $\mathcal{A}$  outputting 1.

**Hybrid 2:** Hybrid 1, but now the msg is  $m_1$ .

1. Challenger samples  $(k_1, \dots, k_n, s_n) \leftarrow \{0, 1\}^{(n+1)\ell}$  uniformly at random.
2.  $\mathcal{A}$  outputs  $m_0, m_1$ .
3. Challenger calcualtes  $c_{n-1} = \text{Enc}(k_{n-1}, m_1)$ .
4. Challenger then gives  $(c_{n-1}, k_n, s_n)$  to  $\mathcal{A}$ .
5.  $\mathcal{A}$  returns  $\hat{b}$ .

Let  $\Pr[H_2]$  be the chance of  $\mathcal{A}$  outputting 1.

**Game 1:** Finally circling back to the SSKeyLeak game, but now w/ bit  $b = 1$ .

1. Challenger samples  $s_0 \leftarrow \{0, 1\}^\ell$ .
2. Compute  $(k_1, s_1, \dots, k_n, s_n) \leftarrow G^n(s_0)$ .
3.  $\mathcal{A}$  outputs  $m_0, m_1$ .
4. Challenger then calculates  $c_{n-1} = \text{Enc}(k_{n-1}, m_1)$ .
5. Give that value,  $(c_{n-1}, k_n, s_n)$ , to  $\mathcal{A}$ .
6.  $\mathcal{A}$  returns  $\hat{b}$ .

Let  $\Pr[W_1]$  be the probability of  $\mathcal{A}$  outputting 1.

By the triangle inequality, the advantage is then bounded by the following expression:

$$|\Pr[W_0] - \Pr[W_1]| \leq |\Pr[W_0] - \Pr[H_1]| + |\Pr[H_1] - \Pr[H_2]| + |\Pr[H_2] - \Pr[W_1]|. \quad (1)$$

## Reducing and Analyzing

**Game 0 → Game  $H_1$**  Let  $\mathcal{B}_1$  be a new adversary against  $G^n$ .

- $\mathcal{B}_1$  will receive a challenge string  $r \in \{0,1\}^{(n+1)\ell}$  from  $G^n$ .
- $\mathcal{B}_1$  will then parse  $r$  as  $(k_1, \dots, k_n, s_n)$ .
- $\mathcal{B}_1$  will run  $\mathcal{A}$  to get  $m_0, m_1$ .
- $\mathcal{B}_1$  then computes  $c_{n-1} = \text{Enc}(k_{n-1}, m_0)$ .
- Finally,  $\mathcal{B}_1$  gives  $(c_{n-1}, k_n, s_n)$  to  $\mathcal{A}$  and its response.

Analyzing this:

- If  $r$  is outputted by  $G^n(s_0)$ , then this is just **Game 0**. So then  $\Pr[\mathcal{B}_1 \rightarrow 1 \mid \text{Real}] = \Pr[W_0]$ .
- In the same manner, if  $r$  is uniform random, then this is just **Hybrid 1**. Thus,  $\Pr[\mathcal{B}_1 \rightarrow 1 \mid \text{Random}] = \Pr[H_1]$ .

$|\Pr[W_0] - \Pr[H_1]| = \text{PRGadv}[\mathcal{B}_1, G^n]$ , which is negligible.

**Game  $H_1 \rightarrow \text{Game } H_2$**   $\mathcal{B}_2$  will now be the adversary against the semantic security of  $\mathcal{E}$ .

- $\mathcal{B}_2$  will run  $\mathcal{A}$  to get  $m_0, m_1$ .
- $\mathcal{B}_2$  then sends  $m_0, m_1$  to the SS challenger for  $\mathcal{E}$ .
- $\mathcal{B}_2$  will then get a ciphertext,  $c^*$ , from the challenger (which is just  $\text{Enc}(k^*, m_b)$  for a rnd key  $k^*$ ).
- $\mathcal{B}_2$  then samples  $(k_n, s_n) \leftarrow \{0,1\}^{2\ell}$  uniformly at random.
- And then  $\mathcal{B}_2$  will give  $(c^*, k_n, s_n)$  to  $\mathcal{A}$ , and output its response.

Looking at this:

- The key for the SS game,  $c^*$ , is rnd and independant of the leak. This is also the same distribution as  $k_{n-1}$  back in **Hybrid 1** and **Hybrid 2**.
- So then, if the SS challenger chooses  $m_0$ , then this is just **Hybrid 1**.
- If instead the SS challenger chooses  $m_1$ , then it is just **Hybrid 2**.

$|\Pr[H_1] - \Pr[H_2]| = \text{SSadv}[\mathcal{B}_2, \mathcal{E}]$ , which is negligible.

**Game  $H_2 \rightarrow \text{Game 1}$**   $\mathcal{B}_3$  will now be challenging the PRG security of  $G^n$ .

- $\mathcal{B}_3$  gets a challenge string  $r \in \{0, 1\}^{(n+1)\ell}$  from  $G^n$ .
- $\mathcal{B}_3$  will parse  $r$  as  $(k_1, \dots, k_n, s_n)$ .
- $\mathcal{B}_3$  then runs  $\mathcal{A}$  to get  $m_0, m_1$ .
- $\mathcal{B}_3$  computes  $c_{n-1} = \text{Enc}(k_{n-1}, m_1)$ .
- And finally,  $\mathcal{B}_3$  gives  $(c_{n-1}, k_n, s_n)$  to  $\mathcal{A}$  and outputs its response.

Looking at this game:

- If  $r$  is outputted by  $G^n(s_0)$ , then this is just **Game 1**. This basically means that  $\Pr[\mathcal{B}_3 \rightarrow 1 \mid \text{Real}] = \Pr[W_1]$ .
- If instead  $r$  is uniform random, then this is just **Hybrid 2**, meaning that  $\Pr[\mathcal{B}_3 \rightarrow 1 \mid \text{Random}] = \Pr[H_2]$ .

$|\Pr[H_2] - \Pr[W_1]| = \text{PRGadv}[\mathcal{B}_3, G^n]$ , which is negligible.

## End

Combining the bounds, we get:

$$\text{SSKeyLeakadv}[\mathcal{A}, \mathcal{E}^1] \leq \text{PRGadv}[\mathcal{B}_1, G^n] + \text{SSadv}[\mathcal{B}_2, \mathcal{E}] + \text{PRGadv}[\mathcal{B}_3, G^n]. \quad (2)$$

I've already assumed, in the beginning, that  $G^n$  was secure, making the PRG advantages negligible. Also, since  $\mathcal{E}$  is semantically secure, then that also means that SS advantage is negligible. And since the sum of negligibles is negligible too, then  $\therefore \text{SSKeyLeakadv}[\mathcal{A}, \mathcal{E}^1]$  is negligible.

**Problem 2.** (50 points at 10 each) Let  $p > 2$  be an odd prime. For this problem, suppose the order of  $g$  is  $p - 1$  (which is even). Such a  $g$  is called a *generator* of  $\mathbb{Z}_p^*$ . Such a  $g$  exists for every  $p$  (we won't prove this fact; see, for example, Section 7.5 of Victor Shoup's book at <http://www.shoup.net/ntb/ntb-v2.pdf> for the existence proof and Section 11.1 for how to sample it efficiently).

(a) Let  $a = g^x$  for some  $x \in \mathbb{Z}_{p-1}$  (the exponent works modulo  $p - 1$  due to Fermat's little theorem). We can talk about the exponent  $x$  being even since  $p - 1$  is even, and therefore  $x + k(p - 1)$  has the same evenness as  $x$  for any  $k \in \mathbb{Z}$ .

Clearly if  $x$  is even, then  $a$  has a square root  $g^{x/2}$  modulo  $p$ . Now show the converse: if  $a$  has a square root modulo  $p$ , then  $x$  is even. (*Hint: you can rewrite any element  $b \in \mathbb{Z}_p^*$  as  $b = g^y$  for some  $y \in \mathbb{Z}_{p-1}$ .*)

**Solution.** I can't really explain the way I'll solve this proof without just restating the proof, but I will say that this solution basically involves just doing a bunch of substitution, "technically".

Firstly, assume that  $a$  has a sqrt modulo  $p$ . Now let  $r \in \mathbb{Z}_p^*$  be such that  $r^2 \equiv_p a$ .

Since  $g$  is a generator of  $\mathbb{Z}_p^*$ , then every element in  $\mathbb{Z}_p^*$  can be represented as a power of  $g$ . Therefore, there exists some  $y \in \mathbb{Z}_{p-1}$  such that  $r = g^y$ .

Substituting this into the square root equation:

$$a \equiv_p r^2 \equiv_p (g^y)^2 \equiv_p g^{2y}. \quad (3)$$

Also, since we know that  $a = g^x$ , then that also means:

$$g^x \equiv_p g^{2y}. \quad (4)$$

Since  $g$  has multiplicative order  $p - 1$  in  $\mathbb{Z}_p^*$ , then we have:

$$x \equiv 2y \pmod{p-1}. \quad (5)$$

This ultimately means that there exists some integer  $k$  such that:

$$x = 2y + k(p-1). \quad (6)$$

Now observe:

- $2y$  is even (it is a multiple of 2).
- $p - 1$  is even (since  $p > 2$  is an odd prime).
- Therefore  $k(p - 1)$  is even for any int  $k$ .

And since  $x$  is the sum of two even numbers,  $x$  must be even. QED.

It'd be nice to check if a value is a square modulo  $p$  without having to explicitly know what power of  $g$  it is. Luckily we have the following test:  $a$  is a square iff  $a^{(p-1)/2} \equiv_p 1$ .

**(b)** Prove the forward direction: if  $a$  is square then  $a^{(p-1)/2} \equiv_p 1$ .

**Solution.** Like again, these are one of those convoluted math-substitution problems, where I just gotta go back to rules to solve it. But I think this particular question does rely on Fermat's little theorem.

Being by assuming that  $a$  is a square modulo  $p$ . Then by its definition, there exists some  $r \in \mathbb{Z}_p^*$  such that  $r^2 \equiv_p a$ .

Like I said before, every element can be written as a power of  $g$ . Thus,  $\exists y \in \mathbb{Z}_{p-1}, r = g^y$ .

Now by subbing this into the equation for  $a$ , we have:

$$a \equiv_p r^2 \equiv_p (g^y)^2 \equiv_p g^{2y}. \quad (7)$$

Calcuating  $a^{(p-1)/2}$ :

$$a^{(p-1)/2} \equiv_p (g^{2y})^{(p-1)/2} \equiv_p g^{2y \cdot \frac{p-1}{2}} \equiv_p g^{y(p-1)}. \quad (8)$$

And now, by Fermat's Little Theorem, since  $g \in \mathbb{Z}_p^*$ , then it must be the case that  $g^{p-1} \equiv_p 1$ . Therefore:

$$g^{y(p-1)} \equiv_p (g^{p-1})^y \equiv_p 1^y \equiv_p 1. \quad (9)$$

Thus, I have shwon that if  $a$  is a square, then  $a^{(p-1)/2} \equiv_p 1$ .

**(c)** Now prove the converse: if  $a^{(p-1)/2} \equiv_p 1$ , then  $a$  is square. (*Hint: Assume not and find a contradiction. Begin by writing  $a$  as  $g^x$ .*)

**Solution.** Your solution goes here

We have thus shown that exactly half the values in  $\mathbb{Z}_p^*$  have square roots, and we know how to identify them: by raising to  $(p-1)/2$ . Note also that values that do have square roots have at least two of them: if  $r \in \mathbb{Z}_p^*$  is a square root of  $a \in \mathbb{Z}_p^*$ , then so is  $-r$ , because  $(-r)^2 = (-1) \cdot r \cdot (-1) \cdot r = (-1)(-1)r^2 = 1 \cdot r^2 = r^2$ . The two square roots are different, because  $p - r \neq r$  as  $p$  is odd. Therefore, each square cannot have more than two square roots by a simple counting argument: if some square had more than two square roots, there wouldn't be enough square roots for all the  $(p-1)/2$  squares, because just two square roots per square already takes up all the  $p-1$  possible square root values in  $\mathbb{Z}_p^*$ . Thus, each square has exactly two square roots.

**(d)** Show that if  $(g^x)^2 \equiv_p a$ , then  $(g^{x+(p-1)/2})^2 \equiv_p a$ , as well. Show that these two square roots are distinct: that is, show that  $g^x \not\equiv_p g^{x+(p-1)/2}$ .

**Solution.** Your solution goes here

We know from the paragraph above that  $a$  has only two square roots. But we have three values that all when squared give us  $a$ :  $g^x$ ,  $g^{x+(p-1)/2}$ , and  $-g^x$ . Thus, it must be that  $g^{x+(p-1)/2} \equiv_p -g^x$ .

(e) Given that  $g^{x+(p-1)/2} \equiv_p -g^x$ , show that that  $g^{(p-1)/2} \equiv_p -1$ . Now show that for any  $b \in \mathbb{Z}_p^*$  that is a non-square,  $b^{(p-1)/2} \equiv_p -1$ .

This refines our previous test for squares from parts (b) and (c): to test if something is a square, you raise it to  $(p-1)/2$  modulo  $p$  and check if the result is 1 or  $-1$ . By the way, the value of  $a^{(p-1)/2} \% p$  is called the Legendre symbol of  $a$  and is often written as  $\left(\frac{a}{p}\right)$ . The Legendre symbol can be generalized for composite  $p$  and this generalization is called the Jacobi symbol (we will not cover it here).

**Solution.** Your solution goes here