
Problem Set 2 Solutions

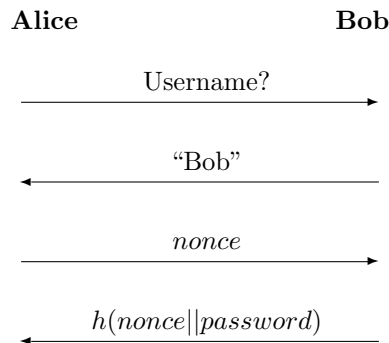
Problem 2-1. Revoke Your Fake Keys from PS1-1 [Individual]

Problem 2-2. Finding Hash Collisions (Without Wasting Memory) [Group]

The largest collision was 72 bits, found by David Chau, Will Stoltzman, and Will Stockwell. The values “ee82aecdcea51b2406” and “a52a82db156951524a” collided in the first 72 bits.

Problem 2-3. Hash Login Scheme [Individual]

Consider the following login scheme:



In this system, Alice keeps a local record of Bob’s password. For each login session, Alice will challenge Bob with a random “nonce”. Bob will hash this nonce concatenated with his password using the function h . Alice will allow Bob to login only if the value $h(\text{nonce}||\text{password})$ matches what she computes locally. Note that Bob’s password is never sent in the clear.

Consider an adversary, Eve, who monitors several valid login sessions by Bob, then tries to log in pretending to be him. Eve can compute h and perform any polynomial-time computations she wants. Eve can only talk to Alice and cannot send any messages to Bob in an attempt to play (wo)man-in-the-middle.

Suppose that h has both the one-way and weak collision resistance properties. In other words, h is a “one-way hash function” by Definition 9.3 from the Handbook of Applied Cryptography. Is h sufficient to protect this system from Eve?

Solution: A one-way hash function is not sufficient to protect this system from Eve. Suppose that h were constructed from a one-way hash function g as follows: $h(\text{nonce}||\text{password}) = \text{nonce}||g(\text{password})$. Anyone able to invert h would be able to invert g . Thus, h must also be one-way.

The first half of h ’s output is the identity function. Therefore any collisions on h for a particular $(\text{nonce}||\text{password})$ value must have the form $(\text{nonce}||\text{password}')$. Anyone who can find collisions for h given $(\text{nonce}||\text{password})$ can find a value $\text{password}'$ such that $g(\text{password}) = g(\text{password}')$. Therefore, h must be weak collision resistant.

However, if Eve monitors a single $(\text{nonce}, h(\text{nonce}||\text{password}))$ pair, she obtains the value $g(\text{password})$. Now she can respond to future challenges nonce' with $\text{nonce}'||g(\text{password})$, which Alice will accept as the correct $h(\text{nonce}'||\text{password})$ value.

So, h is a one-way hash function but is clearly not sufficient to protect the system from Eve, even if she monitors just a single login session.

Problem 2-4. One-way and Collision-resistant Hash Functions [Group]

Let h be a one-way collision-resistant hash function mapping some domain D to itself, i.e. $h : D \rightarrow D$.

Extend h to a mapping from sequences (a_1, a_2, \dots, a_n) to elements of D as follows. We call the resulting hash algorithm h^* :

$$1. h^*(()) = h(d_0), \text{ where } d_0 \text{ is some fixed element of } D.$$

$$2. h^*((a_1, a_2, \dots, a_n)) = h(a_1 || h^*(a_2, \dots, a_n))$$

Argue that h^* is both OW and CR.

Solution:

Suppose h^* is not OW. Then an adversary given some $d \in D$ is able to compute $a = (a_1, \dots, a_n)$ such that $h^*(a) = d$. However, such an adversary is effectively inverting the function h , since given d , they find a string y such that $h(y) = d$. In this case, the string y happens to have the form $y = a_1 || h^*(a_2, \dots, a_n)$. This is a contradiction since h is assumed to be OW. Therefore, h^* must be OW.

Now suppose that h^* is not CR. Then an adversary can find two strings a and a' such that $h^*(a) = h^*(a')$. Let a_i be the first block where the two strings differ, that is, where $a_i \neq a'_i$. If it were the case that $h^*(a_i, \dots, a_n) \neq h^*(a'_i, \dots, a'_n)$, then a and a' could not collide, since $a_j = a'_j$ for $j < i$. Therefore it must be the case that $h^*(a_i, \dots, a_n) = h^*(a'_i, \dots, a'_n)$.

Then the function h collides on two strings $h(a_i || h^*(a_{i+1}, \dots, a_n)) = h(a'_i || h^*(a'_{i+1}, \dots, a'_n))$. These two strings are necessarily different $a_i \neq a'_i$. Therefore, an adversary able to find two unique strings that collide for h^* has also found two unique strings that collide for h . This contradicts our assumptions that h is CR, therefore h^* must be CR.

Problem 2-5. One-Time Pads [Group]

The one-time pad works as follows:

- The message M is divided into a sequence of “elements” (in the usual case, these elements are bits). Let S denote the set of possible elements. (For example, $S = \{0, 1\}$ if the elements are bits.) Suppose that $M = m_1 m_2 \dots m_n$.
- A “pad” $P = p_1 p_2 \dots p_n$ of the same length is chosen by choosing each element p_i uniformly at random from S .
- The ciphertext $C = c_1 c_2 \dots c_n$ is formed by combining each message element with the corresponding element of the pad, using some encryption function e :

$$c_i = e(p_i, m_i)$$

- Of course, it must be decryptable to someone who knows the pad, so there is a corresponding decryption function d that operates element by element, using each element of pad and each element of the ciphertext to produce the corresponding element of the message:

$$m_i = d(p_i, c_i)$$

- In the case where the elements are bits, the operations e and d are usually taken as addition modulo 2 (i.e. xor or \oplus). What is another choice for e and d that provides unconditional security? (Proof not needed.)

(a) The key to the proof of unconditional security is that

$$\forall c, m \Pr[(c_i = c)|(m_i = m)] = \frac{1}{|S|} \quad (*)$$

That is, for each element of the ciphertext, any of the $|S|$ possible message elements could have produced it with equal probability. This will hold whenever:

1. The elements of C are also elements of the set S .
2. There is some function $r(.,.)$ that permits recovering the pad element p_i from the corresponding elements m_i and c_i :

$$r(m_i, c_i) = p_i$$

Argue that (*) follows from (1) and (2).

Solution:

We know by (1) that that $C \subseteq S$. Fix a particular $m \in S$ and consider the functions $e_m : S \rightarrow C$ and $r_m : C \rightarrow S$. Suppose $C \subsetneq S$. Then there must exist at least two pads p_1 and p_2 such that $e_m(p_1) = e_m(p_2) = c$. But by (2), $r_m(c)$ should recover the specific pad used to produce c . This is impossible since either p_1 or p_2 could have produced c . Therefore, only one p exists such that $e_m(p) = c$ and $(e_m(p) = c) \iff (r_m(c) = p)$.

$$\begin{aligned} \Pr[c_i = c | m_i = m] &= \frac{\Pr[c_i = c \wedge m_i = m]}{\Pr[m_i = m]} \\ \Pr[c_i = c | m_i = m] \Pr[m_i = m] &= \Pr[c_i = c \wedge m_i = m] \\ &= \Pr[c_i = e(m_i, p_i) = e(m, p) = c \wedge m_i = m] \text{ (By definition of } e) \\ &= \Pr[p_i = r(m_i, c_i) = r(m, c) = p \wedge m_i = m] \text{ (Since } (e_m(p) = c) \iff (r_m(c) = p)) \\ &= \Pr[p_i = p] \Pr[m_i = m] \text{ (} p \text{ is independent of } m.) \\ &= \frac{\Pr[m_i = m]}{|S|} \text{ (} p_i \text{ is chosen uniformly at random.)} \\ \Pr[c_i = c | m_i = m] &= \frac{1}{|S|} \end{aligned}$$

(b) A finite *group* consists of a finite nonempty set S together with a binary operator \circ (“composition”) mapping S^2 into S , such that:

1. \circ is associative (i.e. $\forall x, y, z \in S : (x \circ y) \circ z = x \circ (y \circ z)$)
2. S contains an *identity element* denoted I such that for all $x \in S$: $x = I \circ x = x \circ I$
3. For every element $x \in S$ there is another unique element in S , denoted x^{-1} , such that $x \circ x^{-1} = x^{-1} \circ x = I$

(It is not necessary that $x \circ y = y \circ x$; if this holds we have a special kind of group known as a commutative group or an abelian group.)

Show that an arbitrary finite group can be used as a basis for a one-time pad, by defining how the encryption operator e and the decryption operator d should work, and by defining how the recovery operator r works, all in terms of the group operations of composition \circ and inverse $^{-1}$.

Solution:

Let $e(p, m) = p \circ m$, $d(p, c) = p^{-1} \circ c$, and $r(m, c) = c \circ m^{-1}$. The decryption operator computes the correct message: $d(p, e(p, m)) = p^{-1} \circ (p \circ m) = m$. The recovery operator computes the correct p : $r(m, c) = (p \circ m) \circ m^{-1} = p$.

- (c) Suppose that your message was naturally composed of elements that are the 27 symbols A, B, ..., Z and "space". What group could you use to make a one-time pad for this set of symbols?

Solution: Addition modulo 27: $(+, \mathbb{Z}_{27})$.