CS 161 – Computer Security
Instructor: Tygar
23 September 2015

Homework 2 Answer Set

Notes

- Homework 2 is due on 22 September 2015 at 3PM.
- Please work on this homework individually – no collaboration allowed.
- Please submit this homework in PDF format.
- It is possible to answer all questions relatively briefly. Please limit your answer to each question to a page at most.
- Submit this homework using Gradescope.

Please start the answer to each question on a new page

1.

a. Compute 500^{-1} mod 10007 using EGCD. Show your work. Note 10007 is prime.

$0 \times 500 + 1 \times 10007 = 10007 \qquad \left(\text{quotient} \left\lfloor \frac{10007}{500} \right\rfloor = 20\right)$

$a = 0 - (20 \times 1) = -20; \quad b = 1 - (20 \times 0) = 1$

$-20 \times 500 + 1 \times 10007 = 7 \qquad \left(\text{quotient} \left\lfloor \frac{500}{7} \right\rfloor = 71\right)$

$a = 1 - (71 \times -20) = 1421; \quad b = 0 - (71 \times 1) = -71$

$1421 \times 500 - 71 \times 10007 = 3 \quad (\text{quotient} \lfloor 7/3 \rfloor = 2)$

$a = -20 - (2 \times 1421) = -2862; \quad b = 1 - (2 \times -71) = 143$

$-2862 \times 500 + 143 \times 10007 = 1$

So $-2862 = 7145$ is the inverse of 500 mod 10007

b. Compute 500^{-1} mod 10007 using the Euler-Fermat theorem. Show your work. Note 10007 is prime. You may use at most 30 multiplication operations; you may not use a computer to compute exponentials.

10007 is prime; so $\varphi(10007) - 1 = 10005$.
$500^2 = 9832$; $500^4 = 604$; $500^8 = 4564$; $500^{16} = 5529$; $500^{32} = 8463$;
$500^{64} = 2270$; $500^{128} = 9302$; $500^{256} = 6682$; $500^{512} = 7897$; $500^{1024} = 8992$
$500^{2048} = 9511$; $500^{4096} = 5848$; $500^{8192} = 5185$

10005 in binary is 10011100010101_2 so we multiply $500^{8192} \times 500^{1024} \times 500^{512} \times 500^{256} \times 500^{16} \times 500^4 \times 500 = 5185 \times 8992 \times 7897 \times 6682 \times 5529 \times 604 \times 500 = 7145$.

2. Consider the following protocol. Alice and Bob choose a common prime p. Alice picks a random $r \in \mathbb{Z}_p$ and sets s such that $rs = 1 \pmod{p-1}$. Bob similarly picks a random $t \in \mathbb{Z}_p$ and sets u such that $tu = 1 \pmod{p-1}$. They then exchange messages as follows:

   $A \to B$: $m^r \bmod p\ (= m')$

   $B \to A$: $(m')^t \bmod p\ (= m'')$

   $A \to B$: $(m'')^s \bmod p\ (= m''')$

   B computes $(m''')^u \bmod p$ and recovers m

   a. Why does this protocol work?

It uses the Fermat-Euler theorem twice.

   b. Show the protocol is vulnerable to a man in the middle attack

MITM computes a pair r', s' satisfying $r's' = 1 \pmod{p-1}$ and a pair t', u' satisfying $t'u' = 1 \pmod{p-1}$. The MITM then pretends to be Bob to Alice and pretends to be Alice to Bob.

   c. Show that if an eavesdropper can compute discrete logarithms, it can break this protocol.

The eavesdropper fixes a generator $g \bmod p$. Let "log" denote discrete logarithm with respect to $g \bmod p$. The eavesdropper intercepts the three messages m^r, m^{rt}, m^{rts} and computes $\frac{\log m^{rt}}{\log m^r} = \frac{rt \log m}{r \log m} = t$ and uses EGCD to calculate u. The eavesdropper calculates $(m^{rts})^u = m \bmod p$.

3. Let $h()$ be a collision-resistant, pre-image resistant, and second pre-image resistant hash function that outputs n bits. Let $expand(x)$ output the n-bit binary string representing x left-padded by zeros when $0 \le x < 2^n$ and otherwise be undefined. Let $\|$ be the string concatenation operator. We construct a new function $h'()$:

$$h'(x) = \begin{cases} 0 \parallel expand(x) & 0 \le x < 2^n \\ 1 \parallel h(x) & 2^n \le x \end{cases}$$

   a. Is $h'()$ pre-image resistant?

No. It is trivial to invert any h' hash value beginning with a zero.

b. Is $h'()$ second pre-image resistant?

Yes. All h' hash values beginning with a zero have only a single pre-image, so they are pre-image resistant. If h' hash values beginning with a one were not second pre-image resistant, it would contradict the assumption that h is second pre-image resistant.

c. Is $h'()$ collision resistant?

Yes. All h' hash values beginning with a one have only a single pre-image, so they are collision resistant. If h' hash values beginning with a one were not collision resistant, it would contradict the assumption that h is collision resistant.

d. Does second pre-image resistance imply pre-image resistance? Why or why not?

No. $h'()$ is second pre-image resistant but not pre-image resistant.

e. Does collision resistance imply pre-image resistance?

No. $h'()$ is collision resistant but not pre-image resistant.