Irving Reyes Bravo

05/10/2025

CS 4920 Spring 2025

HW 5

Problem 1.

a. Find all primitive roots of 17:
Since 17 is prime, the multiplicative group modulo 17 has order 16, and any primitive root must have order 16 – aka its powers generate all numbers from 1 to 16 (mod 17).
To find all primitive roots of modulo 17, I will construct a table like Table 2.7 in the textbook:

a	a^1	a^2	a^3	a^4	a^5	a^6	a^7	a^8	a^9	a^{10}	a^{11}	a^{12}	a^{13}	a^{14}	a^{15}	a^{16}
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
2	2	4	8	16	15	13	9	1	2	4	8	16	15	13	9	1
3	3	9	10	13	5	15	11	16	14	8	7	4	12	2	6	1
4	4	16	13	1	4	16	13	1	4	16	13	1	4	16	13	1
5	5	8	6	13	14	2	10	16	12	9	11	4	3	15	7	1
6	6	2	12	4	7	8	14	16	11	15	5	13	10	9	3	1
7	7	15	3	4	11	9	12	16	10	2	14	13	6	8	5	1
8	8	13	2	16	9	4	15	1	8	13	2	16	9	4	15	1
9	9	13	15	16	8	4	2	1	9	13	15	16	8	4	2	1
10	10	15	14	4	6	9	5	16	7	2	3	13	11	8	12	1
11	11	2	5	4	10	8	3	16	6	15	12	13	7	9	14	1
12	12	8	11	13	3	2	7	16	5	9	6	4	14	15	10	1
13	13	16	4	1	13	16	4	1	13	16	4	1	13	16	4	1
14	14	9	7	13	12	15	6	16	3	8	8	4	5	2	11	1
15	15	4	9	16	2	13	8	1	15	4	9	16	2	13	8	1
16	16	1	16	1	16	1	16	1	16	1	16	1	16	1	16	1

I need to look for rows where all 16 powers are distinct before repeating. Those rows are:

- a = 3
- a = 5
- a = 6
- a = 7
- a = 10
- a = 11
- a = 12
- a = 14

Those are the primitive roots of modulo 17.

b. Find discrete logarithms, modulo 17 while varying the base:
I know that the primitive roots modulo 17 are: 3, 5, 6, 7, 10, 11, 12, 14.
For this problem, I need to find $\log_g(a)$ such that:

$g^k \equiv a \pmod{17}$, for $k \in \{1, 2, .., 16\}$

1. Base g = 3: $3^n \bmod 17$ for {3,9,10,13,5,15,11,16,14,8,7,4,12,2,6,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_3(a)$	16	14	1	12	5	15	11	10	2	3	7	13	4	9	6	8

2. Base g = 5: $5^n \bmod 17$ for {5,8,6,13,14,2,10,16,12,9,11,4,3,15,7,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_5(a)$	16	6	13	12	1	3	15	2	10	7	11	9	4	5	14	8

3. Base g = 6: $6^n \bmod 17$ for {6,2,12,4,7,8,14,16,11,15,5,13,10,9,3,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_6(a)$	16	2	15	4	11	1	5	6	14	13	9	3	12	7	10	8

4. Base g = 7: $7^n \bmod 17$ for {7,15,3,4,11,9,12,16,10,2,14,13,6,8,5,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_7(a)$	16	10	3	4	15	13	1	14	6	9	5	7	12	11	2	8

5. Base g = 10: $10^n \bmod 17$ for {10,15,14,4,6,9,5,16,7,2,3,13,11,8,12,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_{10}(a)$	16	10	11	4	7	5	9	13	6	1	12	14	11	3	2	8

6. Base g = 11: $11^n \bmod 17$ for {11,2,5,4,10,8,3,16,6,15,12,13,7,9,14,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_{11}(a)$	16	2	7	4	3	9	13	6	14	5	1	11	12	15	10	8

7. Base g = 12: $12^n \bmod 17$ for {12,8,11,13,3,2,7,16,5,9,6,4,14,15,10,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_{12}(a)$	16	6	5	11	9	10	7	2	10	15	3	1	4	12	13	8

8. Base g = 14: $14^n \bmod 17$ for {14,9,7,13,12,15,6,16,3,8,10,4,5,2,11,1}

a	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**
$\log_{14}(a)$	16	14	9	12	13	7	3	10	2	11	15	5	4	1	6	8

Problem 2.

Perform encryption and decryption using the RSA. If there's an error, explain why and describe its cause.

GIVEN:

- p and q are primes.
- Compute n = p * q
- Compute $\phi(n) = (p - 1) * (q - 1)$
- Choose e such that $1 < e < \phi(n)$ and $GCD(e, \phi(n)) = 1$
- Find d such that $e * d \equiv 1 \bmod \phi(n)$
- Encryption: $C = M^e \bmod n$
- Decryption: $M = C^d \bmod n$

PROOF:

a. $p = 3$; $q = 11$, $e = 7$; $M = 5$

1. Compute n and $\phi(n)$:

   n = 3 * 11 = 3

   $\phi(n)$ = (3 – 1) * (11 – 1) = (2) * (10) = 20

2. Check if GCD(7,20) = 1:

   Factors of 7: {1,7}

   Factors of 20: {1,2,4,5,10,20}

   Yes, 7 and 20 are coprime.

3. Find d:

   $7d \equiv 1 \bmod 20$

   d = 3 (since $7 * 3 = 21 \equiv 1 \bmod 20$)

4. Encrypt M = 5:

   $C = 5^7 \bmod 33$

= 78125 mod 33
= 14 *NO ERRORS*

5. Decrypt C = 14:
   M = 14^3 mod 33
   = 2744 mod 33
   = 5

b. $p = 7$; $q = 11$, $e = 17$; $M = 8$
1. Compute n and $\phi(n)$:
   n = 7 * 11 = 77
   $\phi(n)$= (7 – 1) * (11 – 1) = (6) * (10) = 60
2. Check if GCD(17,60) = 1:
   Factors of 17: {1,17}
   Factors of 60: {1,2,3,4,5,6,10,12,15,20,30,60}
   Yes, 17 and 60 are coprime.
3. Find d:
   17d ≡ 1 mod 60
   d = 53 (since 17 * 53 = 901 ≡ 1 mod 60)
4. Encrypt M = 8:
   C = 8^{17} mod 77
   = 2251799813685248 mod 77
   = 49 *NO ERRORS*
5. Decrypt C = 49:
   M = 49^{53} mod 77
   = 3.805329268808431529237064653868714e89 mod 77
   = 8

c. $p = 11$; $q = 15$, $e = 9$; $M = 7$
(q=15) is not prime. RSA requires p and q to be prime for $\phi(n)$ to be correctly computed.

d. $p = 17$; $q = 31$, $e = 7$; $M = 2$
1. Compute n and $\phi(n)$:
   n = 17 * 31 = 527
   $\phi(n)$= (17 – 1) * (31 – 1) = (16) * (30) = 480
2. Check if GCD(7,480) = 1:
   Factors of 7: {1,7}
   Factors of 480: {1,2,3,4,5,6,8,10,12,15,16,20,24,30,32,40,48,6080,96,120,160,240,480}
   Yes, 7 and 480 are coprime.
3. Find d:
   7d ≡ 1 mod 480
   d = 343 (since 7 * 343 = 2401 ≡ 1 mod 480)
4. Encrypt M = 2:
   C = 2^7 mod 527
   = 128 mod 527
   = 128 *NO ERRORS*
5. Decrypt C = 128:
   M = 128^{343} mod 527
   = 3.805329268808431529237064653868714e89 mod 77
   = 2

Problem 3.

a. In a public-key system using RSA for confidentiality, you intercept the ciphertext $C = 10$ sent to a user whose public key is $e = 5$, $n = 35$. What is the plaintext M?

In order to decrypt, I need the private key *d*, which satisfies:

First, I need to find $\phi(n)$:

n = 35 = p * q

So, p = 5 and q = 7

Therefore, $\phi(n) = (5 - 1) * (7 - 1)$

= (4) * (6) = 24

Now, I need to find *d* such that $5d \equiv 1 \bmod 24$:

I first try d = 5:

5 * 5 mod 24 = 1

25 mod 24 = 1

So, d = 5. Now, I decrypt $M = 10^5 \bmod 35$:

10^2

100 mod 35 = 30

$10^4 = (10^2)^2 = 30^2$

900 mod 35 = 20

$10^5 = (10^4)^1 = (20 * 10)$

200 mod 35 = 25

Thus, the plaintext M = 25.

b. In an RSA system, the public key of a given user is $e = 31$, $n = 3599$. What are p,q (where n is the product of p and q) and the private key of this user?

First, I ty small primes to factor 3599:

I know that $\sqrt{3600} = 60$

I know that $\sqrt{3481} = 59$

So, $\sqrt{3599} \approx 60$

I now check the divisibility:

59 * 61 = 3599 *CHECK*

Thus, p = 59, q = 61.

I will now find $\phi(n) = (59 - 1) * (61 - 1)$

= (58) * (60) = 3480

Now, I will find *d* such that $31d \equiv 1 \bmod 3480$:

3480 = 31 * 122 + 8

31 = 8 * 3 + 7

8 = 7 * 1 + 1

7 = 1 * 7 + 0

So, GCD is 1 – which is good. Now, I back-substitute:

1 = 8 – 7 * 1

7 = 31 – 8 * 3

1 = 8 – (31 – 8 * 3) = (4 * 8) – 31

8 = 3480 – 31 * 112

1 = 4 * (3480 – 31 * 112) – 31 = (4 * 3480) – (31 * 449)

Thus: (-449 * 31) + (4 * 3480) = 1

Which means *d* = -449 mod 3480 = 3480 – 449 = 3031

c. Use the fast exponentiation algorithm of Figure 9.8 to determine $5^{596} \bmod 1234$.

First, I need to express 596 in binary: $596 = (1001010100)_2$

***i* (Bit Index)**	***b_i* (Bit Value)**
9	1
8	0

7	0
6	1
5	0
4	1
3	0
2	1
1	0
0	0

Given the initial values ($c = 0, f = 1$), I follow these steps:

- In each iteration:
  - $c \leftarrow 2c$
  - $f \leftarrow (f * f) \bmod n$
  - If $b_i = 1$:
    - $c \leftarrow c + 1$
    - $f \leftarrow (f * a) \bmod n$

Where $a = 5$ and $n = 1234$. Applying that to each iteration:

***i* (Bit Index)**	***b*$_i$ (Bit Value)**	***c* (before/after)**	***f* (before/after)**
9	1	0 → 2 → 3	1 → (1*1) mod 1234 = 1 → (1*5) mod 1234 = 5
8	0	3 → 6	5 → (5*5) mod 1234 = 25
7	0	6 → 12	25 → (25*25) mod 1234 = 625
6	1	12 → 24 → 25	625 → (625*625) mod 1234 = 721 → (721*5) mod 1234 = 1177
5	0	25 → 50	1177 → (1177*1177) mod 1234 = 1057
4	1	50 → 100 → 101	1057 → (1057*1057) mod 1234 = 321 → (321*5) mod 1234 = 605
3	0	101 → 202	605 → (605*605) mod 1234 = 719
2	1	202 → 404 → 405	719 → (719*719) mod 1234 = 782 → (782*5) mod 1234 = 213
1	0	405 → 810	213 → (213*213) mod 1234 = 782
0	0	810 → 1620	782 → (782*782) mod 1234 = 1037

So, $5^{596} \bmod 1234 = 1037$.

Problem 4.

Users A and B use the Diffie-Hellman scheme with a common prime $q = 71$ and a primitive root $a = 7$.

a. If user A has private key $X_A = 4$, what is A's public key Y_A?

I know that:

$$Y_A = 7^4 \bmod 71$$

Solving for 7^4:

$7^4 = 49^2 = 2401$

Solving for mod 71:

$2401 \bmod 71$

$2401 \div 71 \approx 33$

$33 * 71 = 2343$

$2401 - 2342 = 58$

So, A's public key $Y_A = 58$.

b. If user B has private key $X_B = 11$, what is B's public key Y_B?

I know that:

$Y_B = 7^{11} \bmod 71$

Solving for 7^{11} by breaking it down:

- $7^2 = 49 \bmod 71 = 49$
- $7^4 = (7^2)^2 = 49^2 = 2401 \bmod 31 = 58$
- $7^8 = (7^4)^2 = 58^2 = 3364 \bmod 71 = 27$

Now, $7^{11} = 7^8 * 7^2 * 7$.

$(7^8 * 7^2) \bmod 71$

$(27 * 49) \bmod 71$

$1323 \bmod 71$

$1323 \div 71 \approx 18 * (71) = 1278$

$1323 - 1278 = 45$

$(45 * 7) \bmod 71$

$315 \bmod 71$

$315 \div 71 = 4 * (71) = 284$

$315 - 284 = 31$

So, B's private key $Y_B = 31$

c. What is the shared secret key?

I first solve for the secret value from A's perspective:

Shared Key = $Y_B^{\{X_A\}} \bmod q$

$= 31^4 \bmod 71$

I know that $31^2 \equiv 961 \bmod 71$

$961 \div 71 = 13 * (71) = 923$

$961 - 923 = 38$

So now $31^2 \equiv 38 \bmod 71 \rightarrow 38^2 \bmod 71$

$1444 \div 71 = 20 * (71) = 1420$

$1444 - 1420 = 24$

Thus, $31^4 \equiv 24 \bmod 71$

I will now double-check the value from B's perspective:

Shared Key = $Y_A^{\{X_B\}} \bmod q$

$= 58^{11} \bmod 71$

I know that $58^2 = 3364 \bmod 71$

$= 27$ *From Earlier Steps*

I know that $58^8 = (58^4)^2 = 19^2 = 361 \bmod 71$

$361 \div 71 = 5 * (71) = 355$

$361 - 355 = 6$

Thus, $58^8 = 6 \bmod 71$

Now, $58^{11} = 58^8 * 58^2 * 58$.

$(58^8 * 58^2) \bmod 71$

$(6 * 27) \bmod 71$

162 mod 71 = 20
(20 * 58) mod 71
1160 mod 71
1160 / 71 = 16 * (71) = 1136
1160 – 1136 = 24
Once again, the shared key = 24

Now consider a Diffie-Hellman scheme with a common prime q = 11 and a primitive root a = 2.

d. Show that 2 is a primitive root of 11.

$\mathbf{2^1}$	= 2 mod 11	= 2
$\mathbf{2^2}$	= 4 mod 11	= 4
$\mathbf{2^3}$	= 8 mod 11	= 8
$\mathbf{2^4}$	= 16 mod 11	= 5
$\mathbf{2^5}$	= 32 mod 11	= 10
$\mathbf{2^6}$	= 64 mod 11	= 9
$\mathbf{2^7}$	= 128 mod 11	= 7
$\mathbf{2^8}$	= 256 mod 11	= 3
$\mathbf{2^9}$	= 512 mod 11	= 6
$\mathbf{2^{10}}$	= 1024 mod 11	= 1

Since all the values from 1 to 10 (mod 11) are present, 2 is definitely a primitive root mod 11.

e. If user A has public key Y_A = 9, what is A's private key X_A?
I want to find X_A such that $Y_A = a^{\{X_A\}}$ mod q
$= 2^{\{X_A\}}$ mod 11 = 9
From the table, I know that $X_A = 6$

f. If user B has public key Y_B = 3, what is the secret key K shared with A?
I want to find $K = Y_B^{\{X_A\}}$ mod q
$= 3^6$ mod 11
I know that:
3^2 = 9
3^4 = 81 mod 11 = 4
3^6 = $3^2 * 3^4 = 9 * 4$ = 36 mod 11 = 3
So, the shared key K = 3

g. If attacker E has both public keys Y_A and Y_B, can it also know K? If so, find K.
It is possible for the attacker E to compute K because q is small (only 11). The attacker E can perform a brute-force search for both private keys like so:
I know that $Y_A = 9 \rightarrow X_A = 6$
Now I must find X_B such that $2^{\{X_B\}}$ mod 11 = 3
I try 2^8: 256 mod 11 = 3 $\rightarrow X_B = 8$
Now an attacker can compute $K = 9^8$ mod 11
9^8 mod 11 = $(9^4)^2$ mod 11 = $(5)^2$ mod 11 = 3

Now use a large common prime q and a primitive root a of q. Assume brute-force is difficult.

g. If attacker E has both public keys Y_A and Y_B, can it also know K? Explain why not.
If a sufficiently large prime q and secure primitive root a are used, attacker E cannot compute the shared secret key K from the public keys $Y_A = a^{\{X_A\}}$ mod q and $Y_B = a^{\{X_B\}}$ mod q. This is because doing so would require solving the Discrete Logarithm Problem (DLP) – aka finding X_A from Y_A, or X_B from Y_B. For large primes (like 2048 bits), this is computationally infeasible with current technology and algorithms. The security of the Diffie-Hellman scheme relies on this mathematical hardness assumption.

h. If attacker E has both public keys Y_A and Y_B, can it launch man-in-the-middle attack? Explain why.
An attacker E can launch a MITM attack if users A and B do not authenticate each other's public keys. During a MITM attack, E may intercept the public key Y_A sent from A to B and replace it with a fake public key Y_E, pretending to be A. Similarly, E may send another fake public key to A, pretending to be B. Both A and B then unknowingly compute secret keys with E instead of each other. E ends up with two shared secrets – one with A and another with B – and can decrypt, alter, and re-encrypt messages between them, fully impersonating both users. To prevent this, Diffie-Hellman must be combined with digital signatures or certificates to verify public keys.

Problem 5.

a. Suppose H(m) is a collision-resistant hash function that maps a message of arbitrary bit length into an n-bit hash value. Is it true that, for all messages x, x' with x[1] x', we have H(x)[1] H(x')? Explain.
No, it is not true that for all distinct messages $x \neq x'$, the hash vales H(x) and H(x') must be different, even if H is a collision-resistant hash function. A hash function maps a potentially infinite set of input messages to a finite set of n-bit outputs. By the pigeonhole principle, there must exist some different inputs x and x′ that produce the same hash output – called a collision. However, a collision-resistant hash function is not required to prevent collisions from existing; instead, it must be computationally infeasible to find such a pair $x \neq x'$ such that H(x) = H(x′). In practice, this means that although collisions exist, no efficient algorithm can discover them in a reasonable amount of time.

b. Define weak collision resistance and strong collision resistance. Explain how they are different.
According to *Computer & Internet Security*, hash functions are evaluated under two main security properties: weak collision resistance (also called second preimage resistance) and strong collision resistance. The property Weak Collision Resistance means that given a specific message x, it is computationally infeasible to find another message $x' \neq x$ such that H(x) = H(x'). This means once a user is given one input and its hash, it should be difficult to find any other input that hashes to the same value. This forms the foundation for digital signatures and file integrity, where a user might try to substitute a benign file with a malicious one that has the same hash. The property Strong Collision Resistance means that it is computationally infeasible to find any two messages $x \neq x'$ such that H(x) = H(x'), without being given one of them in advance. That is, even without prior knowledge of any input, a user can't find two distinct messages that collide. Strong collision resistance is important for broader security applications like certificate signing and blockchain technologies. The key difference is Weak Resistance protects against attacks given a specific input while Strong resistance protects against attacks where no input is known ahead of time. Strong collision resistance implies weak collision resistance, but not vice versa.

c. Let's use an encryption algorithm to construct a one-way hash function. Consider using RSA with a known key. Then process a message consisting of a sequence of blocks as follows: encrypt the first block, XOR the result with the second block and encrypt again, and so on.

Show that this scheme is not secure by solving the following problem. You are given a two-block message B1, B2, and its hash:

$$\text{RSAH}(B_1, B_2) = \text{RSA}(\text{RSA}(B_1) \oplus B_2)$$

i) First, given an arbitrary block D_1, choose D_2 so that RSAH (D_1, D_2) = RSAH (B_1, B_2).
I denote: $H = RSAH(B_1, B_2) = RSA(RSA(B_1) \oplus (B_2))$
My goal is to find D_2 such that $RSAH(D_1, D_2) = H$
I choose an arbitrary D_1

1. I compute $RSA(D_1)$
2. I know that:
   $$D_2 = RSA(D_1) \oplus RSA(B_1) \oplus B_2$$
3. Then:
   $$RSAH(D_1, D_2) = RSA(RSA(D_1) \oplus D_2)$$
   Substituting D_2 I get:
   $$= RSA(RSA(D_1) \oplus (RSA(D_1) \oplus RSA(B_1) \oplus B_2))$$
   Since $x \oplus x = 0$:
   $$= RSA(RSA(B_1) \oplus B_2) = H$$

Thus, for any arbitrary D_1, I can find a matching D_2 that produces the same hash output.

ii) Second, what is the issue of using this hash function for cryptographic applications?

The proposed hash constriction's core issues are that it fails to ensure collision requirement, which is a fundamental requirement for cryptographic hash functions. As I showed before, for any arbitrary first block D_1, one can constrict a corresponding D_2 such that the hash of (D_1, D_2) equals the hash of the original (B_1, B_2). This implies that collisions can be found trivially.

This vulnerability arises because the RSA encryption used is deterministic and reversible, and the XOR operation is linear and symmetric. When these are combined like so, they fail to provide the avalanche effect and non-linearity essentials in hash functions. As a result, the construction allows an attacker to generate multiple distinct message pairs with identical hash values, making it useless for digital signatures, data integrity, and authentication.