San Diego State University
CS574 Computer Security
Homework Assignment #2
Due: Mar 07, 2020 11:59 PM

- Please type the solutions using a word processor such as MS Word, Latex, or write by hand neatly and upload the scanned copy of it.
- I, ___________________ (sign your name here), guarantee that this homework is my independent work and I have never copied any part from other resources. Also, I acknowledge and agree with the plagiarism penalty specified in the course syllabus.
- Turn in your assignment through the blackboard before the deadline. Penalty will be applied to late submission.

1. (10 points) Explain the relationship between Pre-image resistance and Strong Collision resistance. Give examples to support your answers.

2. (10 points) When encrypting and signing a message m, does the order of encryption and signature operations matter? Explain.

3. (10 points) Which one of the following Hash function requirements is used to prevent forgery when an encrypted hash code is used? Justify your answer

a. Preimage resistant
b. Second preimage resistant
c. Collision resistant
d. Pseudo-randomness

4. (20 points) In this question,

a. Consider the following hash function. Messages are in the form of a sequence of decimal numbers, $\mathrm{M} = (\mathrm{a}_1, \mathrm{a}_2, \dots, \mathrm{a}_\mathrm{t})$. The hash value h is calculated as: $(\sum_{i=1}^{t} a_i) \ mod \ n$, for some predefined value n. Does this hash function satisfy the basic properties of a secure hash function? Explain your answer.
b. Repeat part (a) for the hash function: $\mathrm{h}' = (\sum_{i=1}^{t} a_i^2) \ mod \ n$
c. Calculate the hash function of part (b) for $\mathrm{M} = (189, 632,900,722,349)$ and $n = 989$.

5. (10 points) State the value of the padding field in SHA-512 if the length of the message is
a. 1919 bits
b. 1921 bits

6. (20 points) In this problem, we will compare the security services that are provided by digital signatures (DS) and message authentication codes (MAC). We assume that Oscar can observe all messages sent from Alice to Bob and vice versa. Oscar has no knowledge of any keys but the public one in case of DS. State whether and how (i) DS and (ii) MAC protect

against each attack. The value auth(x) is computed with a DS or a MAC algorithm, respectively.

a. (Message integrity) Alice sends a message x = “Transfer $1000 to Mark” in the clear and sends auth(x) to Bob. Oscar intercepts the message and replaces “Mark” with “Oscar.” Will Bob detect this?
b. (Replay) Alice sends a message x = “Transfer $1000 to Oscar” in the clear and sends auth(x) to Bob. Oscar observes the message and signature and sends them 100 times to Bob. Will Bob detect this?
c. (Sender authentication with cheating third party) Oscar claims that he sent some message x with a valid auth(x) to Bob but Alice claims the same. Can Bob clear the question in either case?
d. (Authentication with Bob cheating) Bob claims that he received a message x with a valid signature auth(x) from Alice (e.g., “Transfer $1000 from Alice to Bob”) but Alice claims she has never sent it. Can Alice clear this question in either case?

7. (20 points) RSA cryptographic system is the most widely-used public key algorithm around the world. For example, given the plaintext “security is important”, the first thing we need to do is encoding the text since characters are a non-numerical concept. There are various encoding options including but not limited to the ASCII or Unicode Table. But in this problem, we will keep it simple by encoding “a → z” to “01 → 26” and space to 27. After the encoding, we need to segment it into multiple blocks with the same length. The security principle here is that the encoding value of a message block should be less than the modulus n. Say that we set the length to be 2, then n should be larger than the maximum value of a message block, which is 2727. Afterward, we can apply the RSA encryption on each message block. Symmetrically, after the decryption, we need to apply to decoding to achieve the correct plaintext.

Write a program to implement the above RSA algorithm based on the public key.

a. Your program will allow the user to provide p and q during runtime. To avoid improper inputs, a prime check needs to be conducted. Also, if the inputs are not compatible with your block length setting, your program shall warn the user to re-input larger values.
b. Your program will allow the user to provide the plaintext. The plaintext only contains characters “a → z” and space.
c. Your program shall select an appropriate value of e based on the respective calculations. Remember that e needs to be relatively prime to φ.
d. Your program needs to consider the padding if the input message is not exactly the multiplication of the block size.
e. Your program needs to print out the public key, the private key, the ciphertext, decrypted plaintext respectively.

Template:

```
def encryptmessage():
        #your code here

def decryptmessage():
	#your code here

encryptmessage ()
decryptmessage ()
```

Note: You should write separate functions for primary calculations and for value checks. You can write the program in Python. No crypto lib will be allowed.