CS-457
Assignment 2 Tutorial

Christos K. Panorios, csdp1318

Assignment 2

- In this assignment you are asked to implement 3 ciphers and one decryptor for part A, one program to crack MD5 hashes for part B, and one simple RSA implementation for part C.
- For part A (ciphers), you should create 2 files and a test file (demo):
  - cs457_crypto.h, containing function declarations and
  - cs457_crypto.c, containing the implementation of the functions
- For parts B and C you should create separate files containing your implementation.

Part A
Cryptography Algorithms

One-time pad

- It is a cryptographic cipher
- It uses a predetermined random shared key that is at least as the size of the plaintext
- The algorithm XORs each byte of the plaintext with the corresponding key byte
- Use /dev/urandom (Linux based system) to generate a random key
- Encryption is done by XORing the plaintext with the key and decryption by XORing the ciphertext with the key
- Store the random generated key to use it for the decryption process
- Assume that plaintext consists only of letters or numbers
- Implement the functions:
    - one_time_pad_encr
    - one_time_pad_decr
- These functions take as arguments the plaintext or ciphertext, its size, and the random generated key, and return the result of the operation

One-time pad encryption

Plaintext	ThisIsACat
Key	randombyte
Output	(T⊕r)(h⊕a)(i⊕n)(s⊕d)(I⊕o)(s⊕m)(A⊕b)(C⊕y)(a⊕t)(t⊕e)=
Hex	26 09 07 17 26 1E 23 3A 15 11

Affine Cipher

- It is a cryptographic cipher that uses mathematical functions for encryption and decryption to map letters to their equivalent counterparts.
- Encryption: $(3x + 8) \bmod 26$
- Decryption: $9(y - 8) \bmod 26$
- Assume that the plaintext consists only of letters and/or spaces, and the program should handle letters in both upper and lower cases.
- Implement the functions:
  - affine_encr
  - affine_decr
- The functions take as arguments the plaintext / ciphertext and return the result accordingly.

Affine Cipher encryption

Map each letter of the alphabet to its corresponding numeric value.

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

Affine Cipher encryption

PLAINTEXT	A	F	F	I	N	E	C	I	P	H	E	R
x	0	5	5	8	13	4	2	8	15	7	4	17
(3x+8)	8	23	23	32	47	20	14	32	53	28	20	59
(3x+8)mod26	8	23	23	6	21	20	14	6	1	2	20	7
ciphertext	I	X	X	G	V	U	O	G	B	C	U	H

Substitution algorithm decryptor

- Write a decryptor for the simple substitution algorithm, that decrypts a ciphertext without knowing the key.
    - Usage of the frequencies of characters in the ciphertext and the English Dictionary (https://github.com/dwyl/english-words) to detect word patterns (small recurring words such as “in”, “the” etc.)
    - Each iteration:
        - Takes as input a mapping (cipher alphabet -> alphabet) and prints the current plaintext
        - Takes as input a partially decrypted word and prints the matching words
- The case (upper/lower) of each letter of the ciphertext remains the same in the plaintext.
    - To make the process easier, you can convert the ciphertext to uppercase/lowercase and restore the case in the generated plaintext.

Substitution algorithm decryptor

An Example

Ciphertext: Zrwu wu i uwqflc ctiqflc hap zrc zezapwil za urao zrc euimc ah zrwu ilmapwzrq.

Detect small words that may be common from the frequency of the English Dictionary and the ciphertext:

- “wu” has multiple occurrences
- can be the word “an”
- replace w with a and u with n
- repeat this process until the original message can be retrieved

Substitution algorithm decryptor

- a → w

a* a* * a* ******* *** *** *****a** ** **** *** ***** ** **a* *****a***.

- n → u

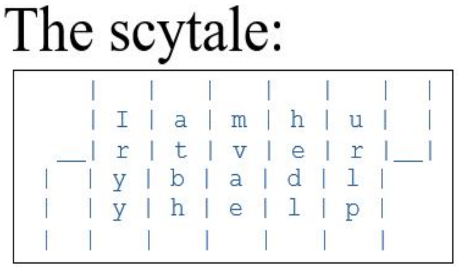
an an * na* ******* *** *** *****a** ** n*** *** *n*** ** **an *****a***.

Scytale cipher

- It is a transposition cipher that involves a cylinder with a strip of parchment wound around it, containing the written message.
- The recipient utilizes a rod of the same diameter on which the parchment is wrapped, to read the message.
- It is essential to store the number of rods in memory for both encryption and decryption processes.
- Implement the functions:
    - scytale_encr
    - scytale_decr
- The functions take as arguments the plaintext / ciphertext, the diameter of the rod and return the result of the operation.

Scytale cipher Encryption

- Suppose we have 5 rods (number of columns).
- **Initial text**: “I am hurt very badly help”
- Plaintext after omitting the spaces and punctuation:
  - “Iamhurtverybadlyhelp”
- Ciphertext after unwinding across the rows:
  - “Iryyatbhmvaehedlurlp”

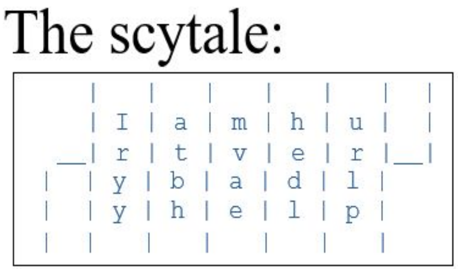


The scytale:

I	a	m	h	u
r	t	v	e	r
y	b	a	d	l
y	h	e	l	p

Scytale cipher Decryption

- Suppose we have 5 rods (number of columns).
- Ciphertext after unwinding across the rows:
    - “Iryyatbhmvaehedlurlp
- Every fourth letter will appear on the same line
- Plaintext after re-insertion of spaces:
    - “I am hurt very badly help”



The scytale:

```
      |   |   |   |   |   |   |
      | I | a | m | h | u |   |
    __| r | t | v | e | r |___|
   |  | y | b | a | d | l |
   |  | y | h | e | l | p |
   |  |   |   |   |   |   |
```

Part B
MD5 Hashing

MD5 Hashing

- MD5 stands for Message Digest 5.
- It produces a 128-bit (32-character hexadecimal) hash.
- Properties:
  - Deterministic: same input → same output.
  - Fast to compute.
  - Irreversible: it’s hard to find the original input from the hash.
- Used for:
  - Password storage (historically, now considered insecure)
  - Integrity checking (e.g., file downloads)
- Small Visual Idea:
  - Input "hello" ➔ MD5 ➔ 5d41402abc4b2a76b9719d911017c592

MD5 Hashing
How to solve the exercise

- You have three unknown passwords stored as MD5 hashes.
- Your task:
  - Dictionary attack: Try common passwords from rockyou.txt (click here to download).
  - Brute-force attack: Generate all possible passwords (a-z, 0-9, up to 8 characters).
  - Compare Execution Times
    - (Hint) For large passwords, if you think that brute-forcing takes forever, it’s normal.
    - You can stop it manually and write it in the report
- How fast is dictionary attack vs. brute-force?
- Tips:
  - Use OpenSSL library to compute MD5 hashes.
- Make functions clean and reuse hashing code.

Part C
RSA Implementation

RSA Implementation

- Asymmetric Encryption:
  - Two keys: Public key (encrypt) and Private key (decrypt).
- Main Idea:
  - Pick two large prime numbers p and q.
  - Compute $n = p \times q$ and
    $\varphi(n) = (p-1)(q-1)$.
  - Choose e such that $\gcd(e, \varphi) = 1$.
    - Common choices for e include **3,17,65537**
  - Compute d: the modular inverse of $e \bmod \varphi(n)$, such that:
    $(d \text{ x } e) \bmod \varphi(n) == 1$
- Encryption: $c = m^e \bmod n$
- Decryption: $m = c^d \bmod n$

Notes

Notes

- This year assignment 1 is **15%** of final grade
- Your final implementation should be one executable file per part
- Follow the execution instructions
  - e.g. CLI arguments, arguments order
- Allowed to use mentioned libraries
  - To use the openssl library you have to use the flag -lcrypto when compiling!

Turnin

- What to submit:
  a. Source files
  b. Test programs
  c. Makefile
  d. README

- turnin assignment_1@hy457 directory_name

That's all Folks!