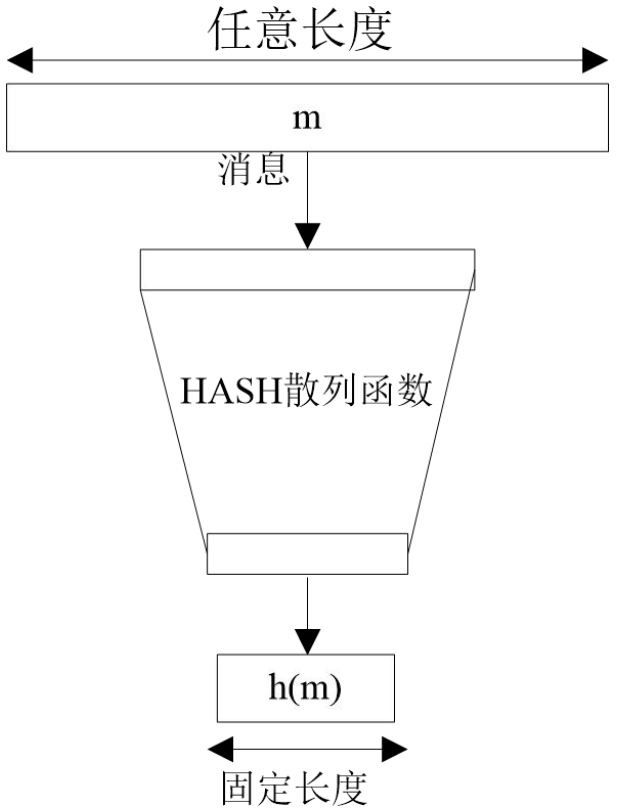
密碼學

現代密碼學

現代密碼學

- 哈希函數——MD5、Sha-1等
- 對稱加密演算法——DES、3DES、AES
- 非對稱加密演算法——RSA

單向散列函數——HASH



- 什麼是單向散列函數

單向散列函數（one-way hash function）也稱為消息摘要函數、哈希函數。

單向散列函數有一個輸入和輸出，其中輸入稱為消息，輸出稱為散列值。單向散列函數可以根據消息計算出散列值，而散列值可以被用來檢查消息的完整性。

- 單向散列函數的性質

1. 根據任意長度的消息計算出固定長度的散列值
2. 能夠快速計算出散列值
3. 消息不同散列值也不同
4. 具備單向性

單向性顯然表明了哈希函數和加密演算法最主要的區別

單向散列函數的具體例子

- MD4、MD5

MD5（Message Digest Algorithm MD5，消息摘要演算法第五版）是Rivest於1991年設計的單向散列函數，能夠產生128比特的散列值。

MD5的強抗碰撞性已經被攻破，也就是說，現在已經能夠產生具備相同散列值的兩條不同消息，因此它已經不安全了。

- SHA-1、SHA-256、SHA-384、SHA-512

SHA-1是由美國國家標準技術研究所設計的一種能夠產生160比特的單向散列函數，SHA-1的消息長度存在上限，和MD5一樣，也已經不安全了。

從安全開發角度，目前可以信任的用來存儲密碼的函數為

Bcrypt 和 CRYPT_BLOWFISH

使用任何存儲密碼的哈希函數都需要加鹽

DES

- DES（Data Encryption Standard）是1977年美國聯邦資訊處理標準中所採用的一種對稱密碼。DES一直以來被美國以及其他國家的政府和銀行等廣泛使用。然而隨著電腦的進步，DES的密文可以在短時間內被破譯，因此除了用它來解密以前的密文外，現在我們不應該再使用DES了。
- 解密和加密

DES是一種將64比特的明文加密成64比特密文的對稱加密演算法，它的密鑰長度是56比特。盡管從規格上來說，DES的密鑰長度是64比特，但由於每隔7比特會設置一個用於錯誤檢查的比特，因此實質上其密鑰長度是56比特。

特點：

加密解密使用相同演算法，只需要將秘鑰進行逆序處理，有興趣的同學可以試著操作一下

三重DES

- 什麼是三重DES

三重DES是為了增加DES的強度，將DES重複三次所得到的一種密碼演算法，通常縮寫為3DES。三重DES並不是進行三次DES加密，而是加密→解密→加密的過程。

三重DES的現狀

儘管三重DES目前還被部分銀行等機構使用，但其處理速度不高，而且在安全方面也逐漸顯現出了一些問題。

AES與DES

- 什麼是AES

AES（Advanced Encryption Standard）是取代其前任標準（DES）而成為一種新標準的一種對稱密碼演算法。

DES 加密強度不夠，現有電腦算力可以輕鬆爆破標準的64bit秘鑰（提高秘鑰長度後解密速度指數級增長）相比之下AES原生支持128-256位秘鑰

同時，DES受差分和線性密碼分析威脅，但目前AES能夠解決

目前來說，AES性能良好，抗爆破能力強，抗差分密碼分析和線性密碼分析能力強，演算法層面上目前無懈可擊，但使用不當仍然危險。

非對稱加密名詞

- 明文P（Plain text）：指沒有經過加密的普通文本
- 密文C（Cipher text）：指加密後的文本
- 加密（Encryption/Encipherment）：將明文轉化為密文的過程
- 解密（Decryption/Decipherment）：將密文還原為明文的過程
- 加密鑰匙Ek（Encryption Key）：加密時使用的鑰匙（配合加密演算法的數據）
- 解密鑰匙Dk（Decryption Key）：解密時使用的鑰匙（配合解密演算法的數據）
- 公鑰Pk（Public Key），RSA中同Ek
- 私鑰Sk（Secret Key）， RSA中同Dk

RSA部分建議大家多記一下筆記，不然容易忘

RSA

RSA是目前最有影響力的公鑰加密演算法，它能夠抵抗到目前為止已知的絕大多數密碼攻擊，已被ISO推薦為公鑰數據加密標準。

RSA公開密鑰密碼體制。所謂的公開密鑰密碼體制就是使用不同的加密密鑰與解密密鑰，是一種“由已知加密密鑰推導出解密密鑰在計算上是不可行的”密碼體制。

基本理論：在公開密鑰密碼體制中，加密密鑰（即公開密鑰）PK是公開信息，而解密密鑰（即秘密密鑰）SK是需要保密的。加密演算法E和解密演算法D也都是公開的。雖然解密密鑰SK是由公開密鑰PK決定的，但卻不能根據PK計算出SK。

總結如下

- 除了原文和秘鑰均可公開
- 公鑰決定私鑰，但無法直接推導出私鑰
- 公鑰加密數據，私鑰解密數據

破解方法之利用最大公約數的拓展歐幾裏得演算法

RSA題目考點不止這一種，但這種比較簡單且常考

原理——互質關係：

如果兩個正整數，除了1以外，沒有其他公因數，我們就稱這兩個數是互質關係（coprime）。比如，15和32沒有公因數，所以它們是互質關係。這說明，不是質數也可以構成互質關係

現代密碼學-RSA

原理——歐拉函數:

定義：任意給定正整數n，請問在小於等於n的正整數之中，有多少個與n構成互質關係？（比如，在1到8之中，有多少個數與8構成互質關係？），計算這個值的方法就叫做歐拉函數，以φ(n)表示。

歐拉函數求法及性質：

- 對於素數p, φ(p)=p-1
- 除了n=2,φ(n)都是偶數
- 如果n可以分解成兩個互質的整數之積，$n = p_1 \times p_2$，則$\varphi(n) = \varphi(p_1 p_2) = \varphi(p_1) \times \varphi(p_2)$
- 對於素數冪分解：$n = p_1^{q_1} \times p_2^{q_2} \times \cdots \times p_n^{q_n}$，存在 $\varphi(n) = n \times \frac{p_1-1}{p_1} \times \frac{p_2-1}{p_2} \times \cdots \times \frac{p_n-1}{p_n}$

現代密碼學-RSA

原理——RSA如何加密？

首先需要隨機選擇兩個不相等的質數p和q，假設我們選定 13，5

所以此時p=13 q=5

第二步，計算p和q的乘積n

n=p*q=65

n的長度就是密鑰長度,將n轉化為二進位1000001,一共有7位，所以這個密鑰就是7位。

實際應用中，RSA密鑰一般是1024位，重要場合則為2048位。

第三步，計算n的歐拉函数φ(n)

φ(n) = (p-1)(q-1)=(13-1)*(5-1)=48

原理——RSA如何加密？

φ(n)=48

第四步，隨機選擇一個整數e，條件是1< e < φ(n)，且e與φ(n) 互質。我們隨機選一個 11（實際應用中，常常選擇65537）

第五步，計算e對於φ(n)的模反元素d

所謂"模反元素"就是指有一個整數d，可以使得e*d被φ(n)除的餘數為1。這裏的d就是我們的私鑰 不能公開！

$ed \equiv 1 \pmod{\varphi(n)}$ <-（這個是同餘公式）

這個式子等價於

$ed - 1 = k\varphi(n)$ （k為任意整數，對應到微積分演算法中常見的變數叫法就是C）

於是，找到模反元素d，實質上就是對下麵這個二元一次方程求解

$ed + k \cdot \varphi(n) = 1$

原理——RSA如何加密？

φ(n)=48 e=11

求解

$$ed \equiv 1 \ (mod \ \varphi(n))$$

$$ed + k \cdot \varphi(n) = 1$$

原理——RSA如何加密？

φ(n)=48 e=11 11d+48k=1

```
p=5
q=13
n=p*q
e=11
fin=(p-1)*(q-1)
d=modinv(e,fin)
print d
```

```
└─[$] <> python rsa_test.py
35
```

得到d=35 這裏的d就是我們的私鑰 不可以隨便分享

第六步，將n和e封裝成公鑰，n和d封裝成私鑰

所以公鑰為(65,11) 私鑰為(65,35)

現代密碼學-RSA

原理——RSA如何加密？

加密的pq一定要選擇相對大的數值

(如1024位秘鑰，n就是在2^{1023}和$2^{1024}-1$之間選擇)

將需要加密的密文例如 test 轉為16進制為0x74657374加密方式（公鑰加密 私鑰解密 ）

密文c=明文的e次方模n =>c = pow(m, e, n) 可以得到密文c

如何解密？

明文m=密文的d次方模n =>m= pow(c, d, n) 可以得到明文m

可以看出加解密方式相同，而rsa題目中通常會給一個公鑰 n e 和密文 c

那我們需要通過分解n得到p與q 接著利用計算φ(n) =>歐幾裏得擴展式求d 接著pow(c,d,n)解出m 也就是flag

現代密碼學-RSA

總結：

1. 隨機選擇兩個不同大質數 p 和 q，計算 $N=p\times q$

2. 根據歐拉函数，求得 $\phi(N)=\phi(p)\phi(q)=(p-1)(q-1)$

3. 選擇一個小於 $\phi(N)$ 的整數 e，使 e 和 $\phi(N)$ 互質。並求得 e 關於 $\phi(N)$ 的模反元素，命名為 d，有 $ed\equiv1(mod\phi(N))$

4. 將p 和 q 的記錄銷毀，此時，(N,e) 是公鑰，(N,d)是私鑰。

加密：首先需要將消息 以一個雙方約定好的格式轉化為一個小於 N，且與 N 互質的整數 m。利用如下公式加密： $m^e\equiv c(modN)$

解密：利用密鑰d 進行解密$c^d\equiv m(modN)$

現代密碼學-RSA

工具和庫：

RSAtool

- 安装

```
git clone https://github.com/ius/rsatool.git
cd rsatool
python rsatool.py -h
```

- 生成私钥

```
python rsatool.py -f PEM -o private.pem -p 1234567 -q 7654321
```

分解整數工具

- 網站分解，factor.db
- 命令行分解，factordb-pycli，借用 factordb 資料庫。
- yafu

現代密碼學-RSA

工具和庫：

python 庫

- primefac

  整數分解庫，包含了很多整數分解的演算法。

- gmpy

  gmpy.root(a, b)，返回一個元組 (x, y)，其中 x 為 a 開 b 次方的值，y 是判斷 x 是否為整數的布爾型變數

- gmpy2

  安裝時，可能會需要自己另行安裝 mpfr 與 mpc 庫。gmpy2.iroot(a, b)，類似於 gmpy.root(a,b)

- Pycrypto

  sudo pip install pycrypto/sudo pip install pycryptodemo

現代密碼學-RSA

工具和庫：

python 庫

- gmpy

  gmpy.root(a, b)，返回一個元組 (x, y)，其中 x 為 a 開 b 次方的值，y 是判斷x 是否為整數的布爾型變數

- gmpy2

  安裝時，可能會需要自己另行安裝 mpfr 與 mpc 庫。gmpy2.iroot(a, b)，類似於 gmpy.root(a,b)

- Pycrypto

  sudo pip install pycrypto/sudo pip install pycryptodemo