CS2107 Live Class Lec3

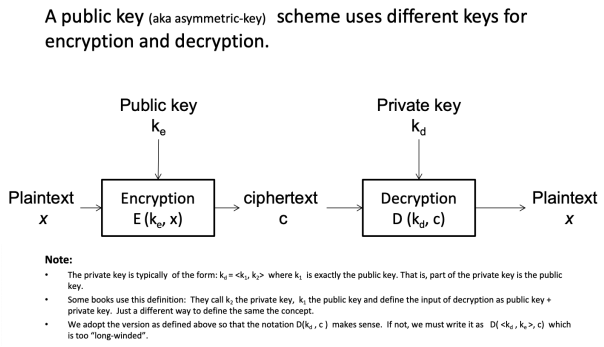
$_/|/|U_Ch@NgRu!$

May 11, 2021

1 Symmetric-key encryption and Public/asymmetric-key encryption

A symmetric key applies the same key for both encryption and decryption while a asymmetric/public key scheme uses different keys for encryption and decryption

2 Public scheme



IMPORTANT: the private key is typically of the form k_d=k_1, k_2 $where$k$_1$ is exactly the public key; some other books call k_1 the public key and define the input of decryption as public key + private key

2.1 Security requirement for the public key scheme

1. given the public key and the ciphertext(not the private key), it is difficult to determine the plaintext(The"difficult to determine" is the indistinguishability of the ciphertext from a random source)
2. It must be difficult to get the private key from the public key

2.2 Advantage of public key scheme

1. less key needed: if there are N entity should guarantee encryption communication with each other; Symmetric key scheme needs N*(N-1)/2 keys while public key scheme needs only N keys

3 Popular PKC schemes

1. RSA: key size $\approx$ 2048 bits
2. EIGamal: EIGamal can exploit techniques in Ellipitic Curve Cryptography(ECC), reducing the key size to around 300 bits
3. Paillier: Partial homomorphic with respect to addition

4 Classroom RSA

The reason that the RSA is called is because the variant used in practive is different, with padding and special considerations in choosing the primes; In practice, padded RSA with strong primes. In some cases, secure implementation to guard against side-channel attack
Note: The total number of 2048-bit integer is 2^{2048}. So, an algorithm that exhausively searches 2048-bit number is infeasible

4.1 "Classroom RSA" setup

1. Owner randomly chooses 2 large prime number p, q and compute n=p*q
2. Owner randomly choose an encryption exponent e s.t. gcd(e,n)=1(i.e. e<n and e is not multiple of p or q)
3. Owner finds the decryption exponent d where de mod (p-1)(q-1) =1
   There is an algorithm that finds d, when given e,p and q. (here the (p-1)(q-1)=ϕ(n), where ϕ(n) is Euler's totient function, which is is the number of co-primes n)
4. Owner publishes(n,e) as public key, and safe-keep (n,d) as the private key**(note that owner shouldn't keep the p and q)**
   The e here is always 65537 in practice, there is no potential vulnerability found yet

Encryption:

1. Given plain-text
2. the cipher-text c=m^e mod n

Decryption:

1. Given the cipher text c
2. The plain-text m=c^d mod n

4.1.1 The different between decrypt without private key and Discrete log problem

c=m^e mod n
The decryption: given c,n,e, finding the e-th root of c
The discrete log problem: given c,n,m, finding e, the discrete log of m
IMPORTANT: the RSA only holds when mn, so the n should be very big

4.2 Simple proof of RSA

1. given m,n,k where n=pq
2. m^k mod n=$m^{kmod\phi(n)}modn$ where $\phi(n) = (p-1)(q-1)whenpandqaretwoprimenumbers$
3. because de mod (p-1)(q-1)=1, hence $m^{demod\phi(n)}modn$=$m^{demod(p-1)(q-1)}modn$=m mod n=m

4.3 Interchangeable role of encryption and decryption key of classroom RSA

In the RSA, it can be found that the public key and the private key is basically the same. So the private key can also be used to encrypt and the public key used to decrypt
This property is special in DSA.Other PKC may not hold(e.g EIGamal PKC)
ps: some material introduce the DSA without talking about the difference between the public key and private key, which can be confusing

4.4 The way to find the prime number

Randomly choose a number and test whether the chosen number applies

4.5 Padding of RSA

In industry, some forms of IV is required so that encryption of the same plaintext at different time would give different ciphertext.
Classromm RSA also has interesting perperties such as homomorphic property, which can be applied in applications(Blind signature, encrypted domain processing). However, this property also lead to vulnerability to attacks and information leakages. Padding can be applied to destroy such

property(e.g The standard (Public-key cryptography standards) PKC#1, add ”optimal padding” to achieve this)

4.6 Efficiency of RSA comparing with AES

According to NIST recommendation, 128 bits AES has the same key strength with 3072-bits RSA.

1. Large key may lead to difficult management of key.
2. RSA encryotion.decryption is significantly slower than AES

4.7 Security of RSA compares with AES

RSA is not necessarily ”more secure ” than AES

1. While it can be proven that getting the private key form the public key is as difficult as factorization, but isis still not known whether the problem of getting the plaintext from the cipher text and public key is as difficult as factorization. So same as AES, there is no rigorous proof that RSA is ”secure” in protecting the ciphertext
2. The ”classromm” RSA neerds to be modifies to prevent some attacks(the homomorphic property);
3. It turns out that the classroom RSA leaks one-bit of information about the plaintext. If adversary knows the module n, and knows the ciphertext c, the adversary can derive one ”bit” of information regarding theplaintext. This is because RSA encryoption preserveds the ”Jacobi symbol”. i.e. the jacobi symbol of the plaintext and ciphertext is the same
4. Facorization can be efficiently done by quantum computer, but it’s no known whether the quantum computer cna be used to broken AES(quantum computer cna speed up exhaustive search by square root, that is , to exhausitively search 128 bits keys, it only takes 2^{64} under quantum computer. So, to be secure against quantum computer, one can adopt 256-bit AES(exhaustive search would be 2^{128}. This is not true for RSA under quantum computer. Double the length of RSA prime number would help))

4.8 Cooperation of AES and RSA when encrypting large file

1. There is a large file F
2. choose a random AES 128-bit key k, encrypt k using PKC
3. encrypt F using AES with k as the key

4.9 Other PKC

1. **ElGamal Encryption**: ElGamal is a "Discrete Log-based" encryption, whereas RSA is "factorization-based" . There are many choices of Algebraic groups for discrete log-based encryption, e.g. Elliptic Curve. Those using Elliptic Cure are often called Elliptic Curve Cryptography (ECC). Certain choices of ECC reduce the key size. E.g. 300 bit for equivalent of 2048-bit RSA.
2. **Paillier encryption**:Paillier Encryption is also discrete-log based. ElGamal can be easily modified so that it is homomorphic w.r.t. multiplication, whereas Paillier is homomorphic w.r.t. addition.

5 Hash

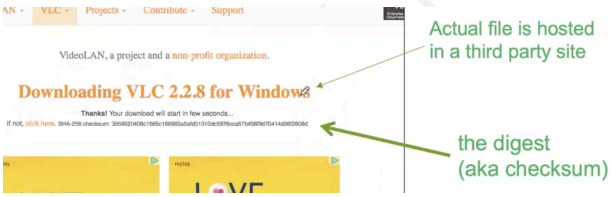
5.1 Hash function

arbitary long message → hash → fixed size digest(hash value/hash/checksum)

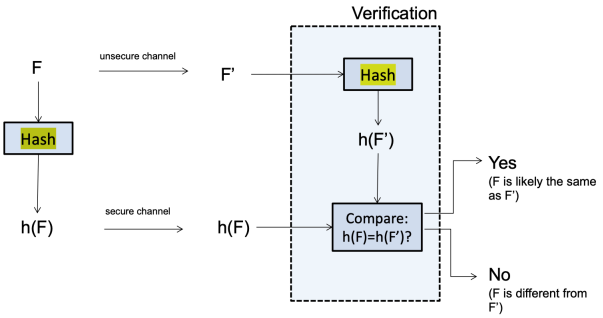
5.2 Hash function security requirement

1. **difficult** (can does not means easy) for an attacker to find two different messages m1,m2, that "hash" to the same digest. That is h(m1) = h(m2) (**collision-resistant**)
2. a collision-resistant hash function is **one-way**

5.3 One application of hash in web



In a website with https starting(authenticated), there are some button can out source to **third-party** website, which may be modified or without control of the website(there may be malicious file inside)
So some website paste the checksum on the http website



There is no secret in the step flow

5.3.1 Attack

1. trick Alice to download the file from the side channel
2. secure channel cannot touch
3. the attack knows the file F, so the attacker can change the F to F' so that h(F')=h(F_{real}), collision
4. So the Alice cannot differentiate the modified F'

5.3.2 collision

6 Hash Classification

6.1 Un-secure Hash

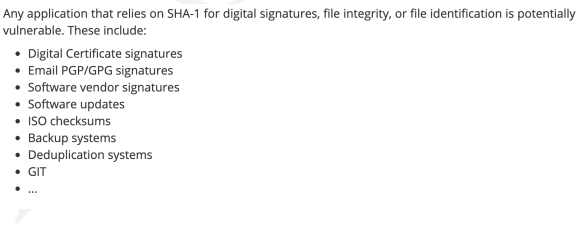
1. Taking selected bits from the data
2. CRC checksum(Cyclic redundancy check)
   the way the CRC can be attached
   (a) There is no authentication, hence an attacker can edit a message and recompute the CRC without the substitution being detected. Note: this may also applies for cryptographic hash(When stored alongside the data, CRCs and cryptographic hash function by themselves do not protect against intentional modification of data. Cryptographic auithetication mechanisms(such as authentication codes and digital signature) are needed to protect against intentional modification of data)
   (b) CRC is an easily reversible function(not one-way)

(c) CRC is a linear function with property that $crc(x \oplus y \oplus z) = crc(x) \oplus crc(y) \oplus crc(z)$ as a result, even if the CRC is encrypted with a stream cipher that uses XOR as its combining operation (or mode of block cipher which effectively turns it into a stream cipher, such as OFB or CFB), both the message and the associated CRC can be manipulated without knowledge of the encryption key;

6.2 Popular secure hash function

1. SHA-0: by NIST; 1993; 160-bits digest, replaced shortly by SHA-1; was attacked with straight forward birthday attack
2. SHA-1: 160-bits message digest, was employed in SSL and SSH
   (a) SSL stands for Secure Sockets Layer and, in short, it's the standard technology for keeping an internet connection secure and safeguarding any sensitive data that is being sent between two systems, preventing criminals from reading and modifying any information transferred, including potential personal details.
   (b) SSH(Secure Shell) is a software package that enables secure system administration and file transfers over insecure networks. It is used in nearly every data center and in every large enterprise.

   A method was found by Huang Xiaoping that can find collision of SHA-1 using 2^{63} operations, which was realised with 110 GPU years
   Google succeed collide SHA-1 and the following aspect may be affected



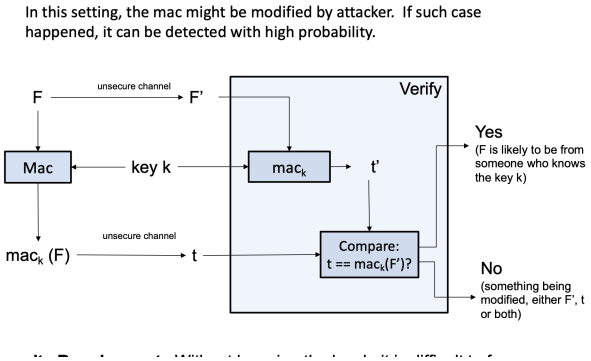
   Any Certification Authority abiding by the CA/Browser Forum regulations is not allowed to issue SHA-1 certificates anymore. Furthermore, it is required that certificate authorities insert at least 64 bits of randomness inside the serial number field. If properly implemented this helps preventing a practical exploitation. Since 2017, any website protected with a SHA-1 certificate was considered as insecure; But git is insecure
3. SHA-2, 2001there are variasion: SHA-224, SHA-256, SHA-384, SHA-512;The number in the name indicates the digest length; No known attack on full SHA-2 but there are known attacks on "partial" SHA-2, for e.g. attack on a 41-rounds SHA-256 (the full SHA-256

takes **64 rounds**)

4. SHA-3, 2007, Keccak (pronounced "catch-ack"). reference:https://www.nist.gov/news-events/news/2012/10/nist-selects-winner-secure-hash-algorithm-sha-3-competition

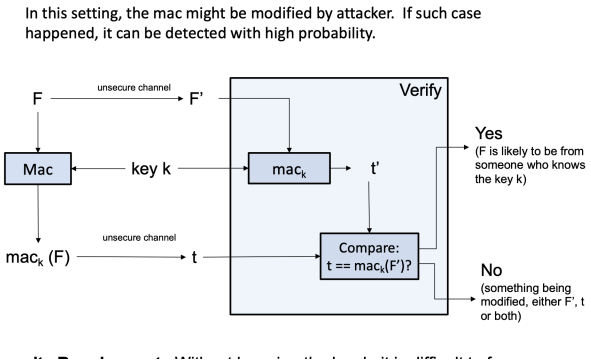
7 Data origin authenticity

7.1 Scenario



If no secure channel exists at all there is one key built to protect the digit, there are two vision

1. In the symmetric key setting, it is called the mac(in lower case means message authentication code).
   Key-ed hash(MAC,in capital) is a function that takes an arbitrary large message and secret key as input, and output a fixed size mac(message authentication code)
   Security requirement(forgery): after seeing multiple valid pairs of messages and their corresponding mac,it's still hard for attacker to forge the mac of a message not seen(i.e. it can resist known-plaintext attack)
2. In the public key setting, it is called the digital signature.



7.1.1 The step flow

suppose key is known by both sender and receiver Sender(There is a key established long before the real connection)

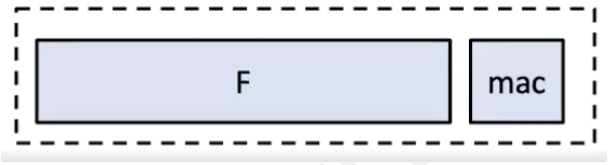
1. Sender use Mac and compute a mac_k of F
2. Sender send F and $mac_k(F)$ both through unsecure channel

Receiver

1. The receiver receive the F and t to, and use F,k get mac_k, use mac_k to compute t'=$mac_k(F')$
2. compare t and t'

7.1.2 Attack

don't know the key
Goal:get a F' and t' such that F'!=F and Alice will accept F'
i.e. forge a valid pair of (message,mac)
IMPORTANT: there is no issue on confidentiality, because the data F can be sent in clear(i.e every one can see)
Typically, the mac is appended to F, they are then stored as a single file, or transmitted through the communication channel together(so mac is also called authentication tag or authentication code)

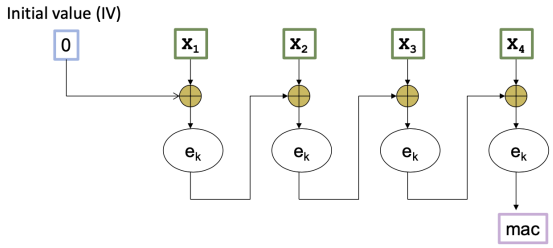


Sign/Mac the file means taking the file and computing the mac, and put the mac behind
As an attacker, there are two choices

1. Modify F
2. Modify t

7.2 The design of mac

7.2.1 CBC-mac



In CBC-mode, we put the initial value as zero(!the initial value has to be zero); **Cannot let the user choose**, because there is a security vulnerability to let the user choose
IMPORTANT: in the CBC-mac, only the last cypher block is called mac

7.2.2 HMAC

$HMAC_k(x) = SHA\text{-}1((K \oplus opad) \,||\, SHA\text{-}1((K \oplus ipad) \,||\, x))$
where
opad = 3636...36 (outer pad)
ipad = 5c5c...5c (inner pad)
(the above are in hexadecimal)

use a hash function, use opad(outer pad) and ipad(inner pad)
SHA1 can be replaced by other SHA
Because the HMAC is based on SHA, if the collision is found, then the HMAC is found

7.2.3 Why the mac should be based on some existing cryptographic(AES ans hash, if the AES and hash are broken the mac is broken))

A lot of the encryption is implemented in the hardware, if the hardware has already has a AES encryption, applying the existing hardware to implement mac saves cost

7.3 Single Sign-On

Single Sign-On(SSO) is an authentication scheme that allows a user to login with a single ID and password to any serveral related, yet independent, software systems. True single sign-on allows the user to log in once and access services without re-entering authentication factors
Implementation:

1. Simple version: Over IP networks using cookies but only if the sites share a common DNS parent domain
2. SSO workss based upon a trust relationship set up between applications or service providers. This trust relationship is often based upon a certificate that is exchanged between the identity provider and the service provider

7.3.1 The step flow of SSO

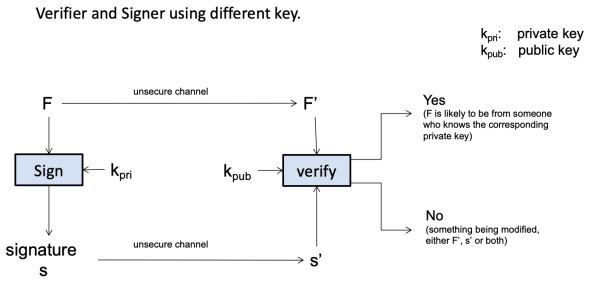
1. A user browser to the application or website they want access to,aka, the service provider
2. Thr service provider sends a token that contains some information about the user, like their email address, to the SSO system ,aka, the identity provider, as part of a request to authenticate the user.
3. The identity provider first checks to see whether the user has already been authenticated, in which case it will grant the user access to the Service provider application and skip to step 5
4. If the user has not logged in, they will be prompted to do so by providing the credentials required by the identity provider. This could simply be a username and password or it might include some other form of authentication like One-time password(OTP)
5. Once the idenitty provider validates the crednetials provided, it will send a token to the service provider confirming a successful authentication
6. Thistoken is passes through the user's browsewr to the service provider
7. the token that is received bny the service provider is validated according to the trust relationship that was set up between theservice provider and identity provider during the initial configuration
8. The user is granted access to the service provider

7.3.2 The SSO token

An SSO token is a collection of data or information that is passed from one system to another during the SSO process. The data can simply be a user ' s email address and information about which system is sending the token. Tokens must be digitally signed for the token receiver to verify that the token is coming from a trusted source. The certificate that is used for this digital signature is exchanged during the initial configuration process. reference:https://www.onelogin.com/learn/how-single-sign-on-works

8 Signature

The mac has a key shared by sender and receivers(symmetric), but there are cases where there is no such shared key, and an asymmetrical authentication(k_{pri} and k_{pub}) can be applied



Security requirement: without knowing the private key, it's hard to form the signature

8.1 Step flow

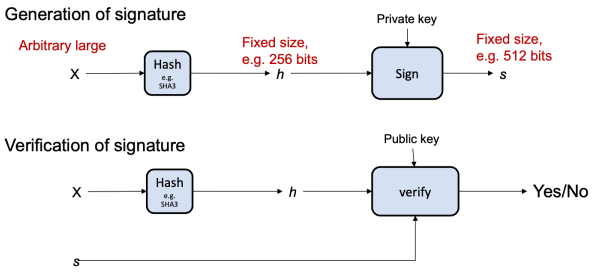
1. There is a broadcast channel for agents to broadcast their public key
2. There is a file, the Alice tell the whole world her public key
3. Alice use the F as the input and sign is with her private key
4. The Bob receive the F' and the s', and run through the verify algorithm using Alice's public key to know whether the file is modified

The computed signature is typically appended to F and store as a single file

8.1.1 Non-reputation

Assurance that someone cannot deny previous commitments or actions
This can be achieved by signature(because only the owner herself knows the private key and can sig)
But mac cannot achieve this, because many people may has the key

8.2 Design of signature scheme



Most signature scheme consist of

1. unkeyed hash
2. sign/verify algorithm

Step flow of signature

1. use hash function to make arbitrary length text into a fixed size text
2. use private key to sign the fixed size text

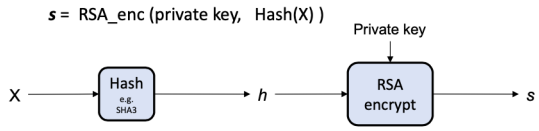
Step flow of verification

1. use hash function to make arbitrary length text into a fixed size text
2. use public key to verify the signature

Almost all signature mechanisms are implemented in the above way
If the collision of the hash function can be found, then can forge the signature
IMPORTANT:Not necessary to have "encryption" in signature scheme, popular schemes such as DSA employ do not "encrypt" (there isn't a way to decrypt). RSA use encryption, but just a special case of **hash-and-sign**.

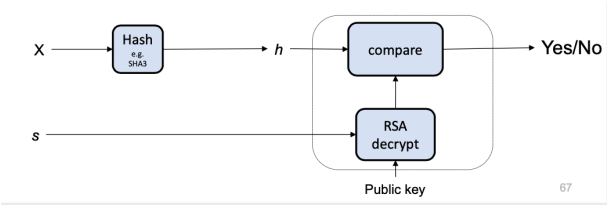
8.2.1 Two vision

1. RSA: The size of key has to be longer; when use RSA, it's actually using the RSA encryption algorithm



   (a) hash the message into a fixed size string

(b) use the private key to encrypt/sign the RSA



If Hash(x)=RSA_dec(public key,s) then accept, else reject

IMPORTANT: the RSA-based signature used private key to encrypt and public key to decrypt(different to normal encryption) IMPORTANT, for RSA, hash is necessary

2. DSA: based on **discrete log**, which means it can use any arithmetic groups, if the Elliptic Curve Cryptography (ECC) is used, the size of the key can be shorter; For DSA is not using the ElGamal encryption(base on Diffie–Hellman key exchange),The Digital Signature Algorithm (DSA) is a variant of the ElGamal signature scheme, which should not be confused with ElGamal encryption.

   IMPORTANT, for DSA, hash is **NOT** necessary if the text is short already

9 Birthday attack

In NIST, AES is safe with 128 bits, the hash with the same security should be 256 bits, the reason is birthday attack

Hashes are designed to make collision difficult to find. (two different x_1 and x_2 such that h(x_1)=h(x_2) and $x_1 \neq x_2$)

The birthday attack is like (exhaustive search)

For a 128 bits hash, there are two ways

1. ```
   for i=1 to Infinity:
       random pick two x1 and x2
       check whether h(x1)=h(x2)
   ```

   $\frac{1}{2^{128}} \times \frac{1}{2^{128}}$

   Very low chance

2. ```
   for i=1 to Infinity:
       random pick 2^{64}(This number can be changed) message
       check whether there is collision among them(the probability is very high, with
           0.5)
   ```

There contains only 2^{64} computations each time

Hence the hash with 256 bits has the same security as AES with 128 bits

9.1 Birthday problem

Suppose we have M messages, and each message is tagged with a value randomly chosen from 1,2,3,...,T. If M>$1.17T^{0.5}$, then with probability more than 0.5, there is a pair of messages tagged with the same value; the probability that two events locate at the same seat is:$1 - e^{-M^2/2T}$

Interesting fact: the mac is also recommended to 256 bits

Because the signature uses hash-and-sign mechanism, the length should also be double

9.2 Vulnerability: Using encryption for the purpose of authentication

IMPORTANT: encryption only achieves confidentiality but not guarantee integrity, availability and authentication

Encryption schemes may provide false sense of security. Consider this design of a mobile apps from a company XYZ.

The mobile phone and a server share a secret 256-bit key *k*. The server can send instructions to the mobile phone via sms. (Note that sms only consist of readable ascii characters. We assume that there is a way to encode binary string using the readable characters). The format of the instruction is:

X P

where X is an 8-bit string specifying the operation, and P is a 120-bit string specifying the parameter. So, an instruction is of size 128 bits. If an operation doesn't need a parameter, P will be ignored. There is a total of 15 valid instructions.

E.g.

00000000 P : send the GPS location to phone number P via sms. If P is not a valid phone number, ignore.

11110000 P : rings for P seconds. If P>10, ignore.

10101010 : self-destruct now!

- An instruction is to be encrypted using AES CBC-mode with 256-bit key, encoded to readable characters and sent as sms. (recap: block size of AES is 128 bits).
- After a mobile phone received a sms, it decrypts it. If the instruction is invalid, it ignores the instruction. Otherwise, it executes the instruction.
- The company XYZ claims that "*256-bit AES provides high level of security, and in fact is classified as Type 1 by NSA. Hence the communication is secure. Even if the attackers have compromised the base station, they are still unable to break the security*".

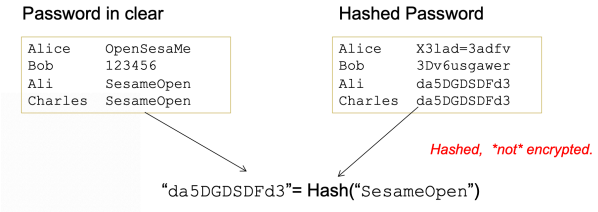
In the example above, though the attacker cannot see the real contends of the message, it can randomly send a message to the receiver, the receiver decrypt it, the change of the first 8 bits is 10101010 would be $\frac{1}{256}$ The proper way is use mac(mac itself is not sufficient)

10 Password hash and password file

The password file stores uid+password

There are many well-known incidents where unprotected or weakly protected password files are leaked, hence, it is desired to add an additional layer of protection to the password file

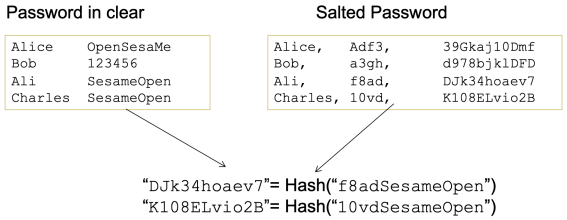
Passwords should be **"hashed"** and stored in the password files.



IMPORTANT: here it is hash rather than encryption.

The encryption is vulnerable to insider attack but no one can reverse hash

In reality the salted hash is applied



The salted IV is randomly chosen

1. to make it longer
2. to make it probabilistic
3. to void rainbow table: weak password is vulnerable

11 Time-memory tradeoff for dictionary attack

Even if H() is one-way, given the digest y, it is still feasible to find a x in D s.t. H(x) =y. This can be done by exhaustively searching the 250 messages in D. Although feasible, this would take days of computing time.

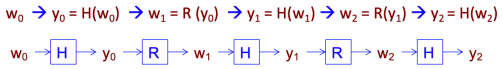
We are allowed to perform pre-processing. One straightforward method is to build a dataset with

250 elements (H(x), x) for x = 0, 1, 2, ..., 2^{50}-1
However, such table is too large. (2^{50} entries = 2^{10} "Tera" entries)

11.1 reduce function R()

Define a reduce function R() that maps a digest y to a word w in the dictionary D. For illustration, if D consists of all 50-bit messages, and each digest is 320 bits, then a possible reduce function simply keeps the first 50 bits of input.
R $(b_0 b_1 ... b_{320}) = b_0 b_1 ... b_{49}$

11.2 Password chain



Call w_0 the starting-point, and y_2 the ending-point. Store the pair (w_0, y_2) in the data-structure ***T***.
The process is repeated with other randomly chosen starting points.

During the query, given a digest y. First search for y in the data-structure ***T***.

- Suppose y is in ***T***. That is, y is one of the ending-point. Let's assume that w_0 is the corresponding starting point (hence $y = y_2$ in the above chain). Note that a pre-image of y is w_2, but at this point, we don't know the value of w_2. Nevertheless, the fact that y is the ending-point implies that w_2 is within the chain starting from w_0. So, we can start to construct the chain w_0, y_0, w_1, y_1, w_2, y_2. When the process hits y_2, we have found the w_2.
- Suppose y is not in ***T***. Compute $y' = H(R(y))$. Search the data-structure for y'. Suppose y' is in ***T***. Let's assume that the starting-point be w_0 (hence $y' = y_2$). With high chances, $y = y_1$. So, a pre-image of y is w_1 (i.e. $H(w_1)=y$). At this point, we don't known w_1. But by constructing the chain from w_0, the pre-image w_1 can be found.

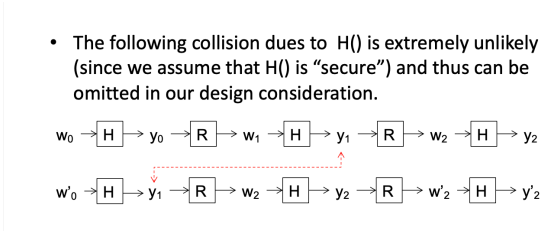
  If y' is not in T, compute y'' = H(R(y')) and repeat the process.

Only the start and the end of each chain is stored, which saves space
If each chain contains k pairs of w and y
Space: reduction of space by a factor of k
Time per query: number of hash operations increases by a factor of k and (k-1) reduce operations
Accuracy: the chains contain repetitions

11.2.1 Potential problem

1. Because the start of each w is randomly chosen, hence hard to make sure every password is covered
2. There is possible of collision, this leads to two issues:
   (a) Even the password is found which meets the requirement, the password may not be the correct password
   (b) Part of the chain may be duplicate

IMPORTANT: the collision is not resulted from the hash function, because SHA3 is applied



11.3 Rainbow table

A optimization of the hash chain

There are multiple reduce function, each reduce function has a color, that's why it's called rainbow table