DES

Basic

Block size: 64bit

Divid into two block, left and right, each of them has 32 bit.

Key size is: 64bit, only 56bit in use, other 8bits for examination.

S-Box: input is 6bit and output is 4bit.

Permutation-Box: Duplicate or reorder the bits.

Encryption Algorithm

```
Li+1 = Ri
Ri+1 = f(Ri,Ki)+Li
```

Decryption Algorithm

When decrypt, need change the order of left and right.

```
Ri = Li+1
Li = f(Li+1,Ki)+Ri+1
   = f(Ri,Ki)+f(Ri,Ki)+Li
```

Block Encryption

Electronic Codebook

Con: same plain text got the same cipher text.

CBC - Cipher Block Chain

```
C0 = Ek(M0 + IV);
C1 = Ek(M1 + C0);
Ci = Ek(Mi + Ci−1)
```

CBC as Authenticator

Use the last block as MAC.

CFB - Cipher Feedback[1]

```
Xi = Xi−1[S:]
Ci = Ek(Xi)[:S]+Mi
```

[1] XOR the message with Encryption

```
(SHIFT−REPLACE)−ENC−(SELECT−ADD)−
SELECT (SRESAS)
```

Collision-Resistant Hash Function

Use CFB as hash function then use the last block as hash output.

Diffie-Hellman Key Distribution

```
g^(q\∗k) mod p
= g^(k\∗q) mod p
```

Basic of Communication

Protection Goals

Confidentiality

1. The content of message should be confidential, and cannot been seen by others
2. Sender & recipient information should be secret
3. Current location of sender/recipient should be confidential (for mobile purpose)

Solution: End-to-end encryption. Sender encrypted the data and send, receiver decrypt the message.

Integrity

1. change of the message should be observe
2. sender can prove himself sends the correct message
3. recipient has to prove where the message from

Solution: Authentication

Availability

Allow all participant to send the message

Solution: divisive network

Unobservability*

Ensures that a user can use a resource or service without others being able to observe that the resource or service is being used. Parties not involved in the communication can observe neither the sending nor the receiving of messages.

Anonymity*

Ensures that a user can use a resource or service without disclosing his identify. Not even the communicants can discover the identity of messages.

Observability

Prevent observing Traffic data

Link-to-link encryption: it provides protection of connection between two neighboring station. Attacker cannot get any information of traffic data. He won't know what has been transferred though the networking and who send to whom.

However for powerful attacker, he can attack the exchange stations. Because the data at station are not encrypted, so attacker can get **Content Data** he wants.

Besides, as the networking developing, encrypt all the data thought the network and decrypt them at station is impossible.

Prevent observing Content data

End-to-end encryption: send encrypted the data and send the data to the recipient, recipient will decrypt the data. Attacker cannot get the **content data** but he still can get the traffic data. Even though he doesn't know what is inside the message, but he knows this message sends from whom to who.

Prevent observing Content data and Traffic data

Use end-to-end and link-to-link together.

Protection measures outside the communication network

Public node

The sender and receiver addresses become meaningless if various public nodes are used.

E.g. Public phone station

Time independent

The time when a message is in a communication network becomes almost meaningless if a network node would request information not when the user wants it but **at some randomly chosen point of time** before the request

Local choice (Preference choice)

To protect selection data you can request information in larger chunks and pick the information that you are interested in later

E.g. If a reader orders multiple newspapers of different political directions instead of a specific article, then no one could imply the political interests and opinions of the reader.

Unlinkability, Unobservability, Anonymity

For an event E, if the possibility of occurrence of E is equal before and after every observation O. $P(E) = P(E|O)$ [2] For attacker A: 0< P(E|O) < 1 if P(E|O) = 1, then every event happens, attacher observed the event.

Unobservability of events can be viewed as unlinkability of observations and the events behind them.

Anonymity can be viewed as unlinkability between instances and events.

Broadcasting

Guarantees perfect information-theoretical unobservalility of **receiver** and unlinkability.

Implicit addressing

When Station want to send specified information to **certain** recipients, will use implicit addressing.

Implicit address doesn't have any linkability to the real physical location of recipient. *attacker cannot find the information of recipient based on the implicit addressing*

For example, station (S) want to send message to recipient to (R), and the implicit address (IA) of R is based on a random number (10101001). **S** will broadcast message with **IA**, other recipient will compare the IA with their IA. If it is not for them they will ignore it. *(also a usual way to implement visible implicit address)*

Visible Implicit Address: if an implicit address is visible to anyone, no encryption. E.g. using random numbers.

Invisible Implicit Address: if an implicit invisible is invisible (encrypted, only the right recipient can decrypt it)

---

[2] 事件 E 发生的频率等于 O 和 E 同时发生的频率

Address Distribution

Public Address: Address is publicly known. E.g. yellow book, it is only for the first time contact.

Private Address: assigned to single communication partners

Implementation of invisible implicit address

The usual implementation of invisible implicit addressing employs redundancy within the message content and **an asymmetric encryption system**.

`E.g. Enc(|implicit addr.|message content|)`

Every message is encrypted completely or partially with the encryption key of the addressed participant (key distribution: the first option is an end-to-end encryption).

After the decryption with the corresponding key the user station of the participant addressed can determine (with the redundancy inside the message, think this part is implicit address) whether the message was designated for him.

Symmetric Authentication: messages are authenticated symmetrically. So the MAC is the appended redundancy. The potential addressees check if the message was authenticated correctly from their point of view.[3]

Fault tolerance on broadcast

Recipients that receive error prone information units or do not receive them at all should insist on a repeated but error free transmission (even if they have no need of the information) unless the reaction could reveal the recipient.

Query and Superpose

Basics

For broadcast: when station wants to send certain data to specific recipient, it has to broadcast the certain data to all recipients with specific recipient's address.

But with **query and superpose**, the participants can query the messages superposed from the **servers**. Others are not able to find out which information was queried. That is because the superposed messages are superposed locally what produces the final message.

Goal: protecting the receiver.

Process

Prerequisite: given system has 5 servers and each server which can contain 4 messages (and stores the same messages in same order).

Presume: we decide to query message from 3 (it can be any number between 1-5) different servers.

Steps

Step 1. Generate **2 random request** index messages, and the 3rd one is generate with **Expected Index Message**(e.g. want request message 2) XOR with other **2 random queries**;

```
req.1: 1001 random
req.2: 1011 random
req.0: 0100 want request message 2
XOR
req.3: 0110
```

Step 2. Send request to 3 random server, because the server has the same sequence of messages;

```
// example message on server
msg.1: 1010
msg.2: 1101
msg.3: 1001
msg.4: 0101
```

Step 3. Server XOR requested messages, and send back;

```
res1: 1010 xor 0101 = 1111
res2: 1010 xor 1001 xor 0101 = 0110
res3: 1101 xor 1001 = 0100
```

Step 4.Recipient XOR all received message.

```
1111 xor 0110 xor 0100
= 1101 (msg.2)
```

Optimization

1. Sending a Pseudo-Random Bit Generation (PRBG) seed instead of the random vectors (can generate the random index message more efficient).
2. Using padding keys and a local master to do the superposing (If attacker get all the information from step 3 to step 4, he can also do the sum, and he will get the information).

[3] Invisible Public Address can use asymmetric encryption; Invisible Private Address can use symmetric encryption.

Invisible implicit addresses using Query and Superpose

If a message is intent for only certain participant, so the offset(or index of that message should not be know by other participants). So we can use invisible implicit address for that index.

Fault tolerance and attack mode

In order to handle the intentionally behavior on server side, such as not respond to the request or delivers wrong response, the recipients can do:

1. If the server doesn't reply, the client can send the same request to other servers (selected random)
2. The message sends from server should be authenticated by the server. Cause clients can send the same request to different servers (do this step twice). The client side local sum will be different if some server cheated.

RING-Network

Receiver anonymity: cause every station receive the message at least one time.

Sender anonymity: every station sends at least with the summed up rate of its actual sending rate.

N-anonymous

There is no situation where an attacker encircling n consecutive stations with any desired amount of attacking stations can identify one station as a sender or a receiver.

如果有N个连续的Station被攻击者环绕，但是无法得知谁是发送送者或者接收者。

Prove 2-anonymous ring-network

If attackers encircled 2 stations, the message passed thought two stations is digital generated, and after digital regeneration messages is not related to original message. So for the attackers they didn't know which one of station 1 or station 2 is sender or receiver. For each information unit there is at least one alternative, on that station 1,2 sends the information unit.

Fault tolerance of the RING-network

Braided Ring

It has two path. Outer ring path: stations connect one by one, inner ring path: stations connect to second next neighbor station.

More details on note.

DC-Network

Basics

Protection of **sender**

Side note: Query & Superpose uses the same idea, but it protects receiver's anonymity.

Anonymity of the sender: If stations are connected by keys the value of which is completely unknown to the attacker, tapping all lines does not give him any information about the sender.

Superposed Sending

1. Exchange the key between the stations in security channel.
2. Message character adds keys create by self, subtract keys create by communication parter, other station could local sum 0 with keys.
3. To get the real message, global sum all messages from different stations.

For station i and j, the key pair follow K(i-j) = - K(i-j)

The idea is, when sum up all messages, all key should be counteract each other

For binary superposed sending, the key between two station is the same. (For addition mod of number of alphabet, e.g. `mod 16: 4 + E (14) = 18 mod 16 = 2`)

Reservation scheme

Station choose **randomly** the time frame he wants, send it to the dc-server, observation the message replied. if the sum of each bit is not greater than 1, means there is not collision, the message can be send in that time frame.

Superposed receiving

Pairwise superposed receiving

Setup: 2 station receiving the message. **Without pairwise:** 2 stations have to wait all the message arrived then

calculate the global sum. **With pairwise:** when 2rd station got the global sum, they can subtract their own message to get another message.

Global superposed receiving

In the global superposed receiving, all member stations **store** the unusable message after a superposition-collision. Only n−1 message need to be re-sent: the last message can be gained by subtraction of the n−1 messages form the unusable message.

Global superposed receiving with average algorithm

More details: A.2

Prove of sender anonymous

More details: A.3

Fault tolerance and attacks

DC+ Net

Basic idea: if broadcast error then uniformly distributed modification of keys. Keys depend on global sums from previous rounds. If only one station receives a corrupted message in one round, its global sum will be corrupted too, meaning that in the next round, its keys will be corrupted, and as a result, it will broadcast a corrupt local output and the global sum will be garbage. Availability is violated, but anonymity is not!

Attack model

Attacker can disturb the DC-Network by sending meanness messages. The global sum will be corrupted, and other stations cannot transmit correct information.

Solution: Reserve Blobs with Trap

Reserve Blobs with Trap

1. Each station has to set one randomly bit (we call it bit index) to 1, the selected position will be the order to send message.
2. Encrypt bit index and random message
3. Based on reserve order, send the encrypted random message

If an attacker damaged the random message (check the global sum) will reveal the encrypted reservation blobs. And participants can know who is the cheater and discard the shared keys with them.

More details: A.4

MIX-Network

Basic Idea

Multiple times encrypted message send to MIX. Each MIX decrypt one of the encryption (like onion layer), then reorder (shuffle) MIX all messages. Send message to next MIX do the same procedure.

Provide unlink-ability: the attacker cannot tell which income message is outcome message. But each MIX knows the incoming message and out message.

Aim: Protection of communication relation: all other senders and receivers of messages that were MIXed together in the batches of the MIX or all MIX that were processed by one message **have to work together** in order to reveal the communication relations against the will of the sender and receiver.

Compare to DC-Network

1. More efficient. In DC-Network, if only one station wants to send message, all other station also has to send message.
2. Less overhead.

Deeper in MIX

Discard Repeats

An attacker could copy a message he has gotten from user before and send copies tot the Mix. These messages would take the same way though the network, cause the return address and description are the same. Attacker can track these information and find the relation between sender and receiver.

Solution: add timestamp to each message comes into Mix. Within certain interval, the same message will be discard.

Buffer Messages

- Batch buffer: wait until certain mount of messages and then flush the message. **Con:** if there are not enough message, the wait time will be very long; **solution**: add dummy messages or set a unbound.
- Pad Buffer: if new message comes, randomly select certain message. **Pro:** faster; **Con:** you wont know how long the message store in the pool, can't use in low latency system.

Cascade or Mix Network

- Mix Network: user freely choose the Mixes

§ Cascade: user choose a specific chain of Mixes.

In a Mix network, the user decide which Mixes he wants to use. This approach provides good scalability and flexibility. Also, the chosen Mixes are totally random, so attackers cannot observer more efficiently. On the other hand, when user choose the Mixes freely, it increase the possibility of choosing unsafe mixes.

For Cascade, it is vulnerable to denial of service. If one of the Mix failed, the service won't work any more. Compare with the possibility of choosing unsure Mixes, cascade maybe a better choice.

Key distribute mechanism

Never decryption directly after encryption, e.g. MIX 1 encrypt the message and send to MIX 2, MIX 2 decrypt the message and encrypt by himself. Both MIXes can reveal the communication relations.

First MIX may know the sender, last MIX knows the receiver, so it can use symmetric encryption. Asymmetric encryption systems must be used for "middle" MIX's.

Why asymmetric encryption for middle MIX?

The messages ware encrypted by the sender before it send to the MIX, so the keys for further MIX and determined by the sender at the beginning, if use the symmetric encryption system. Further MIXes know sender information by the symmetric keys.

Change order

An appropriate order would be the alphabetical order of the encoded messages.

Why don't use random?

1. faster, more efficient
2. Give no chance to MIX to produce a Trojan Horse, cause the order time is very short.

Maximal protection

All messages of same length in the considered time interval have to pass the MIX's at the same time.

If one message of batch is left over by others, attacker can differ this message from others.

Mix Channel

Anonymity scheme

Sender Anonymity

n: number of MIX; M: Message; C: Asymmetric Encryption; Z: Random number [4]; A: Address of MIX; K: Symmetric key; n+1 is receiver; 0 is sender; e: return address of receiver

Direct

```
Mn+1 = Cn+1(M0)
Mi = Ci(Zi, Ai+1, Mi+1)
```

In-direct

```
Mn+1 = Cn+1(M0)
Mi = Ci(Ki,Ai+1); Ki(Mi+1)
```

Receiver Anonymity

```
// For header is decryption in each
step
Hn+1 = e
Hj = Cj(Kj, Aj+1, Hj+1)
// For conten is encryption in each
step
Ij = Kj−1(Ij−1)
// Recipent will decrypt the message
locally with corrosponding key
```

Mutual Anonymity

Sender chooses K1 - Ks; Receiver choose Ks - Kn; The real return address: e, set n = 5, H = header, B = body, b real body

```
//for header
H5 = C5(K5, e)
H4 = C4(K4, A5, H5)
Hs = Cs(Ks, A4, H4)
// Sender get Hs, decrypt with Ds
// get H4 encrypt with C3 C2 C1
H3 = C3(K3, A4, H4)
H2 = C2(K2, A3, H3) subsituide
```

[4] Why we need Z? Due to the public key is known by everyone. We need the random number to prevent attacker guess the plain text by compare the encrypted message.

```
H1 = C1(k1, A1, H2)
```

MIX decrypts one by one find the final address

```
//for body, message on link
B1 = K1(K2(K3(Ks(b)))
B2 = K2(K1(b))
B3 = K3(Ks(b))
B4 = Ks(b)
B5 = K4(B4)
B6 = K5(B5)
```

Return Address

It's the address of recipient.

Maintaining message length

- If the input message's length is different from the output message, it will decrease the anonymous of the system. Because attacker can compare to message's length and distinguish different messages.
- In order to keep the output message has the same length, we need to add some random data in the message.
  - One can choose add the data between Header and Message, but the MIX has to know where is the HEAD section and Message section. Cause HEAD is always decrypt.
  - or at the end of the message. If the encryption algorithm satisfied K-1(K(M)) = K(K-1(M)) = M. Then the Mix doesn't need to differ which part should be use encrypt or decrypt.
- Implementation of MIXes using RSA without redundancy predicate and with contiguous bit strings is insecure.

More details check the note.

Fault Tolerance of MIX

- Simple: sender has alternative disjoint MIX-network
- Better one: use candidate MIX, if one MIX down in the network, can choose another candidate. But it will decrease the anonymity, and sender has to increase the trust number of mix.

Solution: coordination only between neighbor MIXes.

Attack model

1. Power attacker observes the all Mix network;
2. Denial of service
3. N-1 Attack: A batch has N messages, and attacker control N-1 messages. Obviously the attacker can know the information of that message. **Solution:** add interval for sender and receiver. at least some times the attacker cannot send the message, then control the message flow.

Oral Exam Question

Which anonymity technologies are there?

- Broadcast
- Query and Superpose
- RING-Network
- DC-Network
- MIXes

What does broadcasting provide?

Recipient anonymity

What is an implicit address?

The implicit address is unknown to public. It's a attribute that implicate the real recipient.

How can implicit addresses be implemented?

The usual implementation of invisible implicit addressing employs redundancy within the message content and an asymmetric encryption system. Every message is encrypted completely or partially with the encryption key of the addressed participant.

After the decryption with the corresponding key the user station of the participant addressed can determine (with the redundancy inside the message) whether the message was designated for him.

How can a broadcast be attacked?

Attacker can disturb the message. Also he can choose certain recipients, and send the messages waiting for response. Using the method, attacker can decrease the anonymity.

How can one defend against that attack?

Insisting on error free receipt a sending of the user station is necessary.

What are the limitations of broadcasting?

Broadcasting is very inefficient if not supported by transport media; e.g., it's easy to do broadcasts in radio networks, but difficult in the internet.

How does query and superpose work?

On Scripts, P201 5.4.2

What are the optimizations?

- Sending a Pseudo-Random Bit Generation (PRBG) seed instead of the random vectors
- Using padding keys and a local master to do the superposing

If a PRBG seed is sent instead of the random vectors, and only one vector (which is calculated) is sent to a server, does that vector still need to be encrypted?

Yes! If the server that receives the only full vector is the only honest one (as per the attacker model) and all other stations are attackers, anonymity is lost.

How does the DC-net work?

Participants share the key with each other based on topology. Calculating (add the key generated by self and subtract the key generated by other) the local sum on each server and send to other participants publicly.

Calculating the global sum.

What are the optimizations?

- Superposing in local subnets and broadcasting intermediate results
- **Establishing a server**, having that server do the addition (instead of the clients) and broadcast the result to all stations

What is the attacker model?

Attacker can disrupt communication and attack availability, but he won't violate anonymity

How is the DC NET (or DC+ NET) protected against modifying attacks? E.g. the attacker disrupts one line and a station gets a wrong global sum, what happens then? Is anonymity violated? Is availability violated?

In DC+ NET, keys depend on global sums from previous rounds. If only one station receives a corrupted message in one round, its global sum will be corrupted too, meaning that in the next round, its keys will be corrupted, and as a result, it will broadcast a corrupt local output and the global sum will be garbage.

Participant can also use Reservation Blob with trap. If the attacker unluckily damage the trap random message, the global sum will be corrupted. This will cause the reveal the message. The disrupter will be kick out the communication.

Availability is violated, but anonymity is not!

How do MIXes work? In a direct sender anonymity scheme, how does a message look if there is one MIX in between the sender and the recipient?

MIX process: 1. Discard repeat message, encode all the input message; 2. Batch the messages; 3. Recode the message; 4. Shuffle, reorder;

The message could look like Cm(Z2,A,Cr(Z1,M)), where Cm is the public key of the MIX, Cr is the public key of the recipient, Z1 and Z2 are random numbers, A is the address of the recipient and M is the real message.

Exclude Section

- Digital Signature System
- Threshold Scheme
- Electrical Banking