Class CS 6903, Lecture n. 6

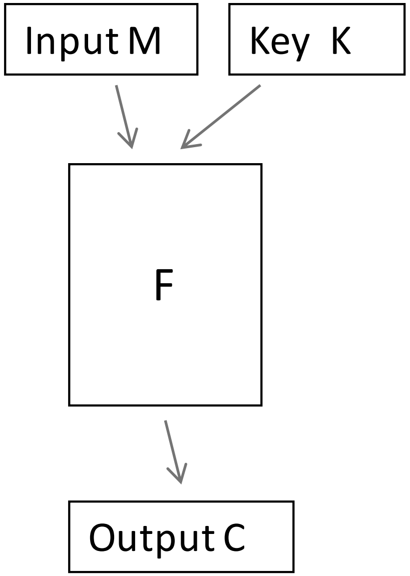
- Welcome to Lecture 6!
- In Lectures 1-5 we studied:
  - Classical cryptography, encryption with perfect secrecy
  - Background on algorithms, complexity theory. Modern cryptography: principles, primitives, and a public-key cryptosystem
  - Algorithmic number theory, number theory and cryptographic assumptions, reductions, proofs by reductions, number theory candidates for cryptographic primitives and schemes
  - Pseudo-random generators, functions, permutations and applications
  - Symmetric Encryption: notions, schemes and proofs

Summary of Lecture 6

- Block ciphers
  - Motivations, notion, examples
  - Attacks and security definitions
- Substitution/Permutation Networks
  - AES
- Feistel Networks
  - DES, Double and Triple DES
- Cryptanalysis
  - Meet-in-the-middle attack
  - Differential and Linear cryptanalysis
- Modes of operations
  - ECB, CBC, OFB, CTR

Block cipher: introduction

- Main motivation: looking for more time-efficient solutions for symmetric encryption



- Block cipher notion:
  - (public and fully-specified) map F associating a k-bit key K and an n-bit plaintext M to an n-bit ciphertext C, and such that, for each key K, map F(K,.) is a permutation
  - k is the “key length” (e.g., 56, 128, 256 bits)
  - n is the “block length” (e.g., 64, 128, 256 bits)
- Intuition: using random K, F scrambles M into C so that no information about M is leaked by C
- Examples: DES, 3DES, AES, IDEA, SIMON, etc.

Block ciphers: attacks and security notion

- Intuition: Block ciphers are supposed to implement (a finite version of) pseudo-random permutations
- Security notion: how close is the block cipher's output to the output of a random permutation, when adversary can run any one of the following:

Attack type	Adversary's resources gained during attack
Ciphertext-only attack	Values F(k,x(i)) for unknown x(i), i=1,2,…
Known-plaintext attack	Pairs (x(i), F(k,x(i))), i=1,2,…
Chosen-plaintext attack	(x(i), F(k,x(i))) for chosen x(i)
Chosen-ciphertext attack	(x(i), F(k,x(i))) and (F^{-1}(k,y(i)), y(i)) for chosen x(i),y(i)

- No asymptotic security here (note: block ciphers are "breakable in O(1)")
- Concrete security: block cipher is (t,ε)-secure if t-time algorithms can only "break" (i.e., distinguish from random permutation using one of the above attacks) with probability < ε
- Goal: design block cipher that cannot be broken in reasonable time
  - E.g.: the time to compute 2^{56} decryptions may be efficient (today) but 2^{100} may not be so

Substitution/Permutation networks: basics

- Intuition: would like to construct a block cipher (a short-representation object) that behaves like a random permutation (a huge-representation object)
- The confusion-diffusion paradigm:
  - Construct a random-looking permutation with a large block length from many other ones operating on shorter blocks
  - Let $F_k(x) = f_1(x_1),\ldots, f_c(x_c)$ for some constant c, where $x= x_1|\ldots|x_c$ and the f_i can be random permutations; this introduces confusion
  - Output bits are permuted or mixed; this introduces diffusion
- Propagating confusion and diffusion
  - Two steps of confusion and diffusion represent a “round”
  - Propagation is achieved by sequentially iterating multiple rounds

Substitution/Permutation networks: principles

- A substitution/permutation network can be seen as a direct implementation of the confusion/diffusion paradigm
  - confusion → substitution via S-boxes (previous f_i)
  - diffusion → permutation via mixing functions
  - A function of the (master) key (i.e., a sub-key) is xor-ed with intermediate results fed as input to each round of the network
  - Key schedule defines how sub-keys are obtained from the (master) key
- Actual choice of S-boxes, mixing permutation and key schedule may imply that a given block cipher is easy to break, or seemingly hard to break, or anything in between
- Design principle 1: invertibility of S-boxes
- Design principle 2: the avalanche effect

Substitution/Permutation networks: security

- Are substitution/permutation networks secure?
- Arguments in favor:
  - Researcher experience
  - Years of cryptanalysis efforts
- However:
  - Number of rounds is very important to assess security
  - Small-round versions are easily breakable because the avalanche effect only happens on a number of bits that, although growing as exponential in the number of rounds, starts as small as 2, 4, 8, 16, etc.
    - Proof idea: To distinguish a substitution/permutation cipher from a random permutation, query oracle on similar inputs; if change is only detected on a few output bits, then this is a cipher, otherwise this is a random permutation

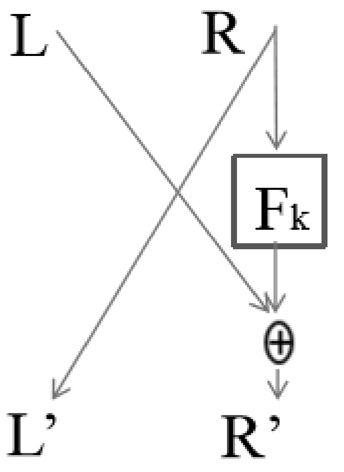
Feistel networks

- Feistel network: same low-level building blocks (S-boxes, mixing permutations, key schedule) but different high-level structure
- Basic idea: constructing an invertible function

  from non-invertible components
- Feistel round: given a non-invertible function f,

  on input (L,R) the transform returns (L',R'), where
  - L'=R and R'=L xor f(k, R)
- Inverse Feistel round: on input (L',R'),

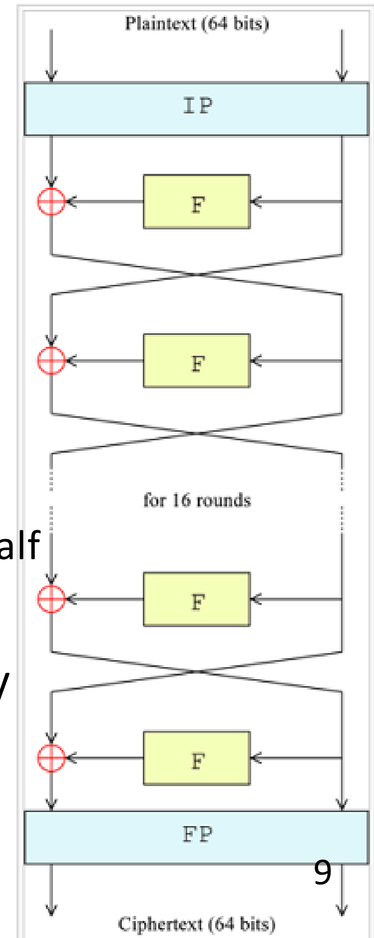
  the inverse transform returns (L,R), where
  - R=L' and L=R' xor f(k, L')



- Results:
  - 3-round Feistel network is pseudo-random if so is f
  - 1-round and 2-round versions are not pseudo-random even if so is f
  - 4-round Feistel network is (super-)pseudo-random if f is pseudo-random (i.e., the 4-round transform is pseudo-random even if adversary A is allowed to query both O and O's inverse)

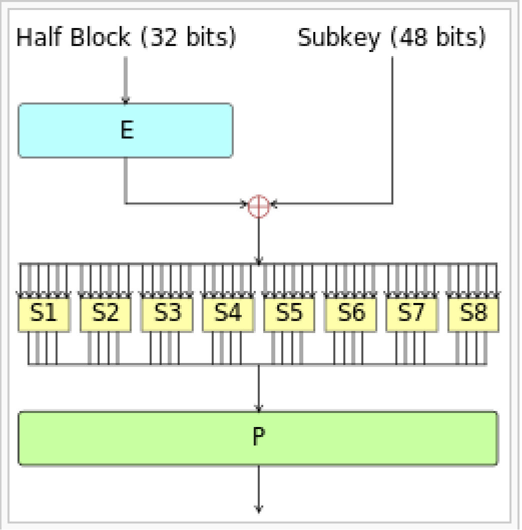
Data Encryption Standard (DES)

- History:
  - Developed in 1970s at IBM (with help from National Security Agency)
  - Adopted in 1977 as a Federal Information Processing Standard (FIPS) for US
  - Great historical significance, one of most studied cryptographic algorithms
- High-level construction (See figure 5.3 in [KL]):
  - 16-round Feistel network with block length = 64 and key length =56
  - Each round uses a (mangler) function f, using a sub-key k(i)
  - Key schedule (entirely public, except for master key):
    - each sub-key k(i) (input to one application of function f in a round) is a permuted subset of 48 bits from master key
    - 56-bit master key is divided into 28-bit left half and 28-bit right half
    - Left-most 24 bits of sub-key k(i) are some subset of above 28-bit left half
    - Right-most 24 bits of sub-key k(i) are some subset of above 28-bit right half
  - Round functions f are non-invertible
  - f is essentially a 1-round substitution/permutation network, with very carefully designed S-boxes (with help from National Security Agency); actually, slightly modified or random S-boxes make DES breakable)



DES: design details

- The DES mangler function f:
  - f computes f(k(i),R), where |k(i)|=48 and |R|=32
  - First of all it goes through an expansion step that expands R from 32 bits to a 48 bits R’
  - R’ is xor-ed with the key k(i) and the result goes through 8 S-boxes, each mapping 6 bits to 4 bits (like a table with 4 rows, 16 columns, and 4 bits in each cell)
  - Finally a mixing permutation is applied



- The S-boxes are public non-invertible functions with following properties:
  - Each S-box is a 4-to-1 function
  - Each row in the table contains each of 16 possible 4-bit strings exactly once
  - Changing one input bit results in >=2 output bits changed
- DES avalanche property is obtained by combining this latter property, the Feistel transform and the permutation step

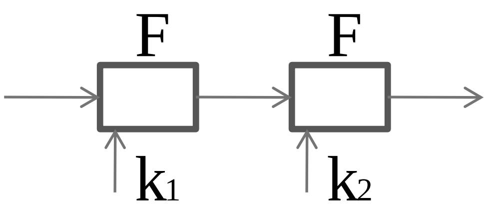
DES Cryptanalysis

- Any small-round variant of DES can be broken (similarly as for substitution/permutation networks) because of small avalanche factor
  - Here, “broken” means that DES is distinguished in time t from a random permutation
- If we consider 1 or 2 or 3-round versions, DES can be broken in that the actual key can be found by the adversary
  - Attack basic idea: use chosen message attack to derive S-boxes outputs, note that from S-boxes inputs one can immediately compute key
  - 1-round: 2 known plaintexts/ciphertexts and time 2^{16}
  - 2-round: 4 known plaintexts/ciphertexts and time 2^{17}
  - 3-round: 2^{13} known plaintexts/ciphertexts and time 2^{30}
- After about 30 years, best known attack is exhaustive search of the key space, which was estimated to need a 20M$ computer in 1977 but only needs a relatively inexpensive computer today (!)
  - There is a history of challenges broken by distributed and special-purpose computing
- Short block length n is also an issue as certain schemes based on DES can be broken using $2^{n/2}$ work
- The Advanced Encryption Standard (AES) was designed to avoid these issues

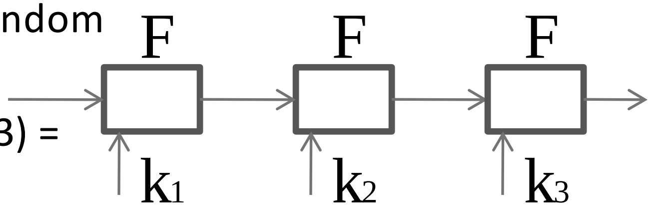
Increasing key length of block ciphers

- Consideration: If only known weakness of DES is relatively short key, we may try to use it as a building block into a cipher with longer key
- In fact, this composition approach makes sense for any block cipher

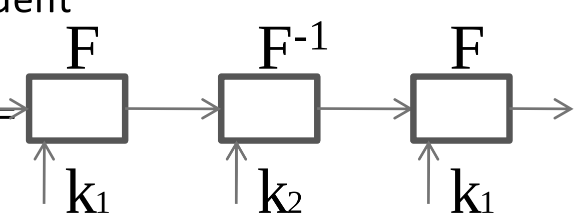
- Let F be a block cipher and let G be Double-F, where G(k,m) is defined as
  - Write k as k(1) | k(2), where k(1), k(2) are random & independent keys
  - Set c(1) = F(k(1),m) and c(2) = F(k(2),c(1)), and return c(2)



- Let F be a block cipher and let G be Triple-F, where G(k,m) is defined as
  - Write k as k(1) | k(2) | k(3), where k(1), k(2), k(3) are random and independent keys
  - Set c(1) = F(k(1),m), c(2) = F(k(2),c(1)) and c(3) = F(k(3),c(2)), and return c(3)

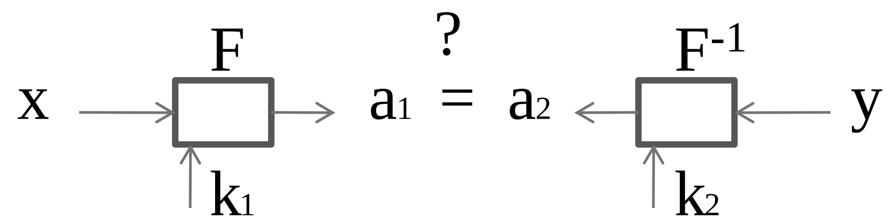


- Let F be a block cipher, let G be 2Key-Triple-F, where G(k,m) is defined as
  - Write k as k(1) | k(2), where k(1), k(2) are random & independent keys
  - Set c(1) = F(k(1),m), c(2) = F^{-1}(k(2),c(1)) and c(3) = F(k(1),c(2)), and return c(3)



The Meet in the Middle Attack

- Perhaps surprisingly, Double-F is only marginally stronger than F
- Consider the so-called "Meet in the middle attack" on oracle O where an adversary is trying to decide whether O is Double-F or a random permutation:
  - Choose input x(0),x(1),x(2) in $\{0,1\}^n$ and compute y(i)=O(x(i)), i=0,1,2
  - For each k(1) in $\{0,1\}^s$, compute a(1) =F(k(1),x(0)), store (a(1),k(1)) in list L
  - For each k(2) in $\{0,1\}^s$, compute a(2) =F^{-1}(k(2),y(0)), store (a(2),k(2)) in list L'
  - If there exists a,k(1),k(2) such that (a,k(1)) is in L and (a,k(2)) is in L' then
    - If F(k(1),x(i))=F^{-1}(k(2),y(i)) for i=0,1,2 then return 1 (for F) else return 0 (for random permutation)



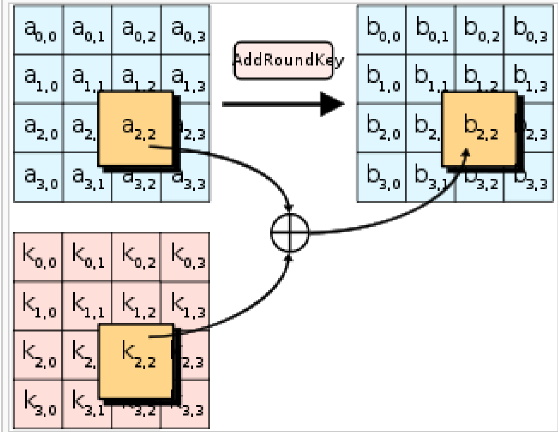
- Analysis (sketch):
- If O is Double-F, then the equation F(k(1),x(i))=F^{-1}(k(2),y(i)) holds for all i=0,1,2
- If O is a random permutation, then the equation holds with probability roughly 2^{-n} for each i and each pair (k(1),k(2)), and the probability that there exists one pair (k(1),k(2)) for which the equation hold for i=0,1,2, is at most $2^{2s} / 2^{3n}$, which is very small (as we can assume the key length s is <= the block length n).
- Thus the adversary can distinguish Double-F from a random permutation with probability $1-2^{2s} / 2^{3n}$ (i.e., almost 1) and in time 2^s

Advanced Encryption Standard (AES)

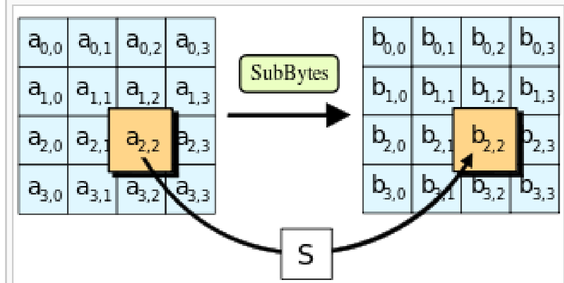
- Selected by a public competition announced by NIST in January 1997, which encouraged quick scrutinizing of candidates and increased confidence in winner
- AES has a 128-bit block and can use 128-, 192-, or 256-bit keys and has, respectively, 10, 12 or 14 rounds; most known is 128-bit variant
- While DES was a Feistel network, AES is a substitution/permutation network
- During AES computation, a 4-by-4 array of bytes, called state, is modified
- State is initially set to 128-bit input
- Each round comes in 4 stages and has the following structure:
  - 1 (AddRoundKey): The 128-bit round sub-key is derived from master key, interpreted as a 4-by-4 array of bytes and xor-ed with the state array
  - 2 (SubBytes): Each byte of state array is replaced by another byte according to a substitution lookup table S (a bijection over $\{0,1\}^8$) that is fixed for all rounds
  - 3 (ShiftRows): Bytes in each row of state array are cyclically shifted to the left as follows: for i=1,…,4, the i-th row is shifted i-1 places to the left
  - 4 (MixColumns): an invertible linear transformation (i.e., a matrix multiplication over a field) is applied to each column
- Viewing stages 3 and 4 are a "mixing" step, each round is a subs/perm network
- Attacks are known for reduced-round AES versions or under not realistic assumptions on the attack scenario, but best way to attack AES remains exhaustive search

AES design details

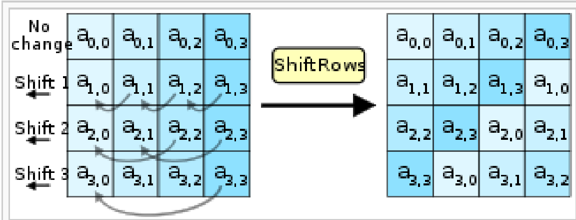
- Each round comes in 4 stages and has the following structure:
  - 1 (AddRoundKey): The 128-bit round sub-key is derived from master key, interpreted as a 4-by-4 array of bytes and xor-ed with the state array
  - 2 (SubBytes): Each byte of state array is replaced by another byte according to a substitution lookup table S (a bijection over $\{0,1\}^8$) that is fixed for all rounds
  - 3 (ShiftRows): Bytes in each row of state array are cyclically shifted to the left as follows: for i=1,…,4, the i-th row is shifted i-1 places to the left
  - 4 (MixColumns): an invertible linear transformation (i.e., a matrix multiplication over a field) is applied to each column



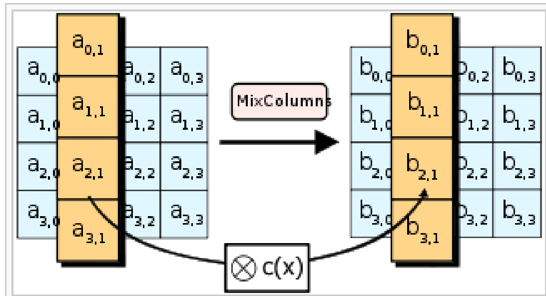
In the AddRoundKey step, each byte of the state is combined with a byte of the round subkey using the XOR operation (⊕).



In the SubBytes step, each byte in the state is replaced with its entry in a fixed 8-bit lookup table, S; $b_{ij} = S(a_{ij})$.



In the ShiftRows step, bytes in each row of the state are shifted cyclically to the left. The number of places each byte is shifted differs for each row.



In the MixColumns step, each column of the state is multiplied with a fixed polynomial $c(x)$.

Question set 14

- Compare the following two values:
  - The number of possible permutations with a domain/range of all 128-bit strings,
  - the number of possible permutations with a domain/range of all 128-bit strings defined by a fixed block ciphers and any possible 128-bit key
- Give 5 principles to design a block cipher based on
  - substitution/permutation networks
  - Feistel networks
- Which block cipher would you use as part of the implementation of a secure system (to maximize security and efficiency) among the following: DES, Double DES, Triple DES, AES? Which key length would you use for your block cipher (to maximize security and efficiency) among the following: 128, 256, 512, 1024?
- Define "generalized meet-in-the-middle attacks" for
  - Triple DES
  - 2-key Triple DES
  - m-tuple DES (not easy)
- Analyze the success probability of all these attacks, by assuming the atomic ciphers are random permutations (not easy)

Summary of Lecture 6

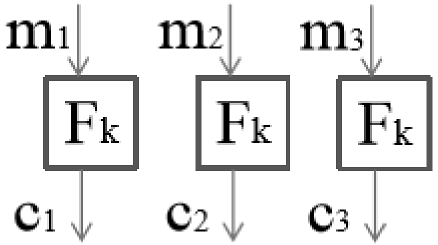
- Block ciphers
  - Motivations, notion, examples
  - Attacks and security definitions
- Substitution/Permutation Networks
  - AES
- Feistel Networks
  - DES, Double and Triple DES
- Cryptanalysis
  - Meet-in-the-middle attack
  - Differential and Linear cryptanalysis
- Modes of operations
  - ECB, CBC, OFB, CTR

Cryptanalysis

- Cryptanalysis is a large set of techniques that attempts to break cryptographic objects like block ciphers
- We have studied the meet-in-the-middle attack, an example of a type of cryptanalysis that focuses on higher-level cipher structures only
- In another large body of research, the math used by the entire cryptographic object is carefully analyzed until a weakness is found
- Two important examples are worth mentioning
- Differential cryptanalysis:
  - Basic idea: specific differences between two inputs propagate in specific differences in the output, with probability greater than for a random cipher
  - For random x(1), x(2), if x(1) xor x(2) = d(x) and, for randomly chosen k, F(k,x(1)) xor F(k,x(2)) =d(y) holds with probability p, we say that differential (d(x),d(y)) appears with probability p
  - Random cipher: $p=2^{-n}$; weak cipher: p is significantly higher
  - Was used to break FEAL-8 cipher, but no practical attack on AES, or DES
- Linear cryptanalysis:
  - Basic idea: consider linear relationship (based on xor) between input bits and output bits, with bias greater than for a random cipher
  - Does not require chosen plaintext attacks, but known plaintext attacks are sufficient

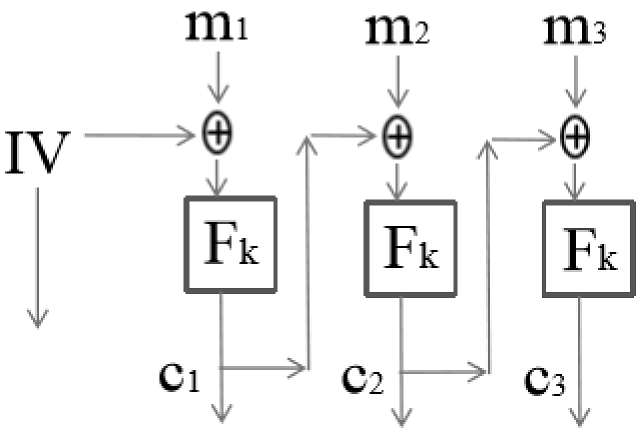
Modes of operation

- Block ciphers can be used to encrypt 1 block (e.g., 64 bits or 128 bits)
- To encrypt longer messages, we combine them using (block cipher) modes of operation
  - Note: the constructions based on pseudo-random generators and pseudo-random functions (resp.) could be used, but they would not be as efficient
- We consider encryption of an (arbitrarily large) integer number of blocks (note: an arbitrary-length message can always be padded with 10…0 so that its length is a multiple of the block length)
- Simplest known mode is the Electronic Code Book (ECB) mode:
  - See also figure 3.5 in [KL]
  - Let m=m(1)..m(n) be an n-block message and let F be a block cipher
  - On input k and m, algorithm E returns F(k,m(1)),…,F(k,m(n))
  - On input k and c=c(1),…,c(n), algorithm D returns F^{-1}(k,c(1)),…,F^{-1}(k,c(n))
- The resulting F-ECB encryption scheme is deterministic, so cannot satisfy IND-CMA-security
- The F-ECB scheme does not even satisfy IND-security



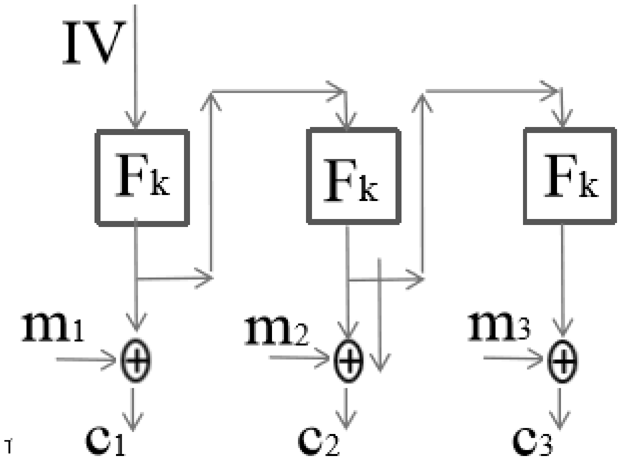
Cipher Block Chaining (CBC)

- Intuition: next ciphertext blocks are dependent on previous ones via chaining
- Construction (when applied to a block cipher F):
  - A random n-bit initial vector (IV) is chosen
  - Let m=m(1)..m(L) be an L-block message and let F be a block cipher
  - On input k and m, algorithm E computes c(0)=IV and c(i)=F(k, c(i-1) xor m(i)), i=1,..,L, and returns: c=(c(0),c(1),…,c(L))
  - On input k and c=(c(0),c(1),…,c(L)), algorithm D computes and returns IV, m(1)=F^{-1}(k,c(1)) xor c(0),…m(n)=F^{-1}(k,c(L)) xor c(L-1)
- Theorem: if F is a pseudo-random permutation then the above F-CBC encryption scheme is IND-CMA-secure
- Drawback: inherently sequential



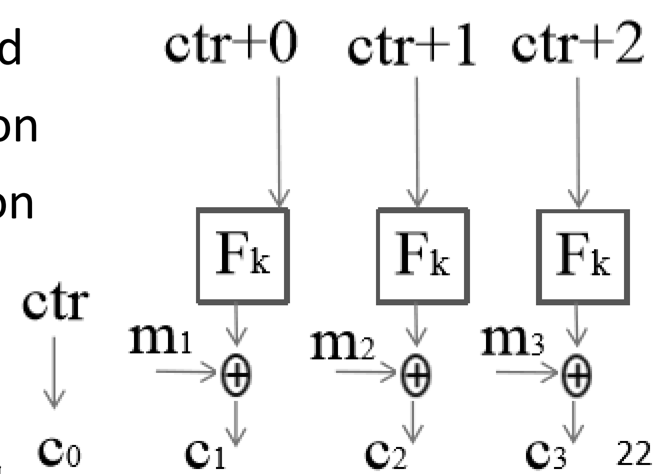
Output Feedback Mode (OFB)

- Intuition: use block cipher to generate a pseudo-random string to be XORed with message
- Construction (when applied to a block cipher F):
  - A random n-bit initial vector (IV) is chosen
  - Let m=m(1)..m(L) be an L-blocks message and let F be a block cipher
  - On input k and m, algorithm E computes r(0)=IV, and r(i)= F(k,r(i-1)), and c(i)=r(i) xor m(i), i=1,…,L, then returns c(0)=IV and c(1),…,c(L)
  - On input k and c=c(0),c(1),…,c(L), algorithm D computes r(0)=IV=c(0), r(i)= F(k,r(i-1)), and m(i)=r(i) xor c(i), i=1,…,L, and returns m(1),…,m(L)
- Theorem: if F is a pseudo-random permutation then the above F-OFB encryption scheme is IND-CMA-secure
- Note: the pseudo-random stream can be computed in a preprocessing phase



Counter Mode (CTR)

- Theorem: F As for OFB, with parallelization and more
- Construction (when applied to a block cipher F):
  - A random n-bit initial vector (IV), also denoted as ctr, is chosen
  - Let m=m(1)..m(L) be an L-block message and let F be a block cipher
  - On input k and m, algorithm E computes ctr = IV, and r(i)= F(k,ctr + i), c(i)=r(i) xor m(i), i=1,…,L, then returns c(0)=IV and c(1),…,c(L)
  - On input k and c=c(0),c(1),…,c(L), algorithm D computes ctr=IV=c(0), r(i)= F(k,ctr+i), m(i)=r(i) xor c(i), i=1,…,L, and returns m(1),…,m(L)
- Theorem: if F is a pseudo-random permutation then the above F-CTR encryption scheme is IND-CPA-secure
- Note: the pseudo-random stream can be computed in a preprocessing phase; encryption and decryption can be fully parallelized, and has random decryption access (i.e., can decrypt i-th block alone)



Counter Mode - Security Proof

- Recall the F-CTR construction:
  - Let ctr be a random n-bit counter, let m=m(1)..m(L) be an L-block message
  - On input k and m, algorithm E computes r(i)= F(k,ctr + i), c(i)=r(i) xor m(i), i=1,…,L, and returns c(0)=ctr and c(1),…,c(L)
  - On input k and c=c(0),c(1),…,c(n), algorithm D computes ctr=c(0), r(i)= F(k,ctr+i), m(i)=r(i) xor c(i), i=1,…,L, and returns m(1),…,m(n)
- Theorem: if F is a pseudo-random permutation then F-CTR is secure in the sense of indistinguishability with chosen-message attack
- Proof plan: Assume F-CTR is not secure. Then there exists an efficient adversary A for which the experiment Exp(A,ind-cpa) (defined in lecture 6) is successful.
- Let succ(A;F) be event "Exp(A,ind-cpa) is successful when F is used as atomic permutation"
  - define, for a random permutation R, succ(A;R)
- We write prob[succ(A;F)] as $p_1 + p_2$, where
  - p_1 = prob[succ(A;F)] - prob[succ(A;R)] and p_2 = prob[succ(A;R)]
- Claim 1: if F is a pseudo-random permutation, p_1 is negligible in n
  - Proved using proof by reduction and hybrid arguments similarly as in previous proofs
- Claim 2: $p_2 \le 2q^2/2^n$, which is negligible (See next slide)

Counter Mode - Security Proof (2)

- Recall the F-CTR construction:
  - Let ctr be a random n-bit counter, let m=m(1)..m(L) be an L-block message
  - On input k and m, algorithm E computes r(i)= F(k,ctr + i), c(i)=r(i) xor m(i), i=1,…,L, and returns c(0)=ctr and c(1),…,c(L)
  - On input k and c=c(0),c(1),…,c(n), algorithm D computes ctr=c(0), r(i)= F(k,ctr+i), m(i)=r(i) xor c(i), i=1,…,L, and returns m(1),…,m(n)
- Theorem: if F is a pseudo-random permutation then F-CTR is secure in the sense of indistinguishability with chosen-message attack
- Sketch of proof of claim 2: We want to prove that prob[succ(A;R)] is small.
- Note that each ciphertext is distributed like a one-time pad encryption of the plaintext since R is a random function and R's inputs ctr+i are all distinct
  - This suffices to prove indistinguishability (without chosen message attack)
- What can go wrong with a chosen message attack? A additionally obtains encryptions of chosen L-block messages $M_1,\dots,M_q$ and then has to guess which of two chosen L-block messages U_0,U_1 where encrypted under challenge ciphertext C; here, A can obtain $R(ctr_j+i)$ xor $M_j(i)$ in the chosen message attack and R(ctr'+i') xor $U_b(i')$ in the challenge ciphertext such that $ctr_j+i=ctr'+i'$ (same random pad would be used for two messages, one of which is known to A)
- The probability that $ctr_j+i=ctr'+i'$ for some j in {1,…,q} and some random ctr_j can be shown to be smaller than $2q^2/2^n$, which is negligible (see textbook)

Question set 15

- What is leaked by the ECB block cipher mode of operation about the plaintext?
- Which block cipher mode of operation would you choose in the following scenarios?
  - Block cipher mode of operation needs to be secure under some reasonable assumption
  - Output from the mode of operation is further encrypted using another scheme
  - We want to maximize the work done in an off-line phase (before message is available) so to minimize the work done in an on-line phase (after message is available)
  - In addition to the goal in last bullet, we want to take advantage of parallelization
- Find an attack for the following block ciphers modes of operation when the IV is a fixed number (instead of being randomly chosen)
  - CBC
  - OFB
  - Counter

Class CS 6903, End of Lecture n. 6

Reference → / Topic ↓	[KL]	[MOV]	[FSK]
Block ciphers	6.2 (edition 1: 5)	7.1, 7.2, 7.4	3, 4
Modes of operation	3.6 (edition 1: 3.6.3)	7.2.2	
Modes of operation	http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation http://http://csrc.nist.gov/groups/ST/toolkit/BCM/index.html http://csrc.nist.gov/groups/ST/toolkit/BCM/modes_development.html#03		
Cryptanalysis	6.2.6 (edition 1: 5.6)	7.8	