Introduction to Cryptography
Lecture 3

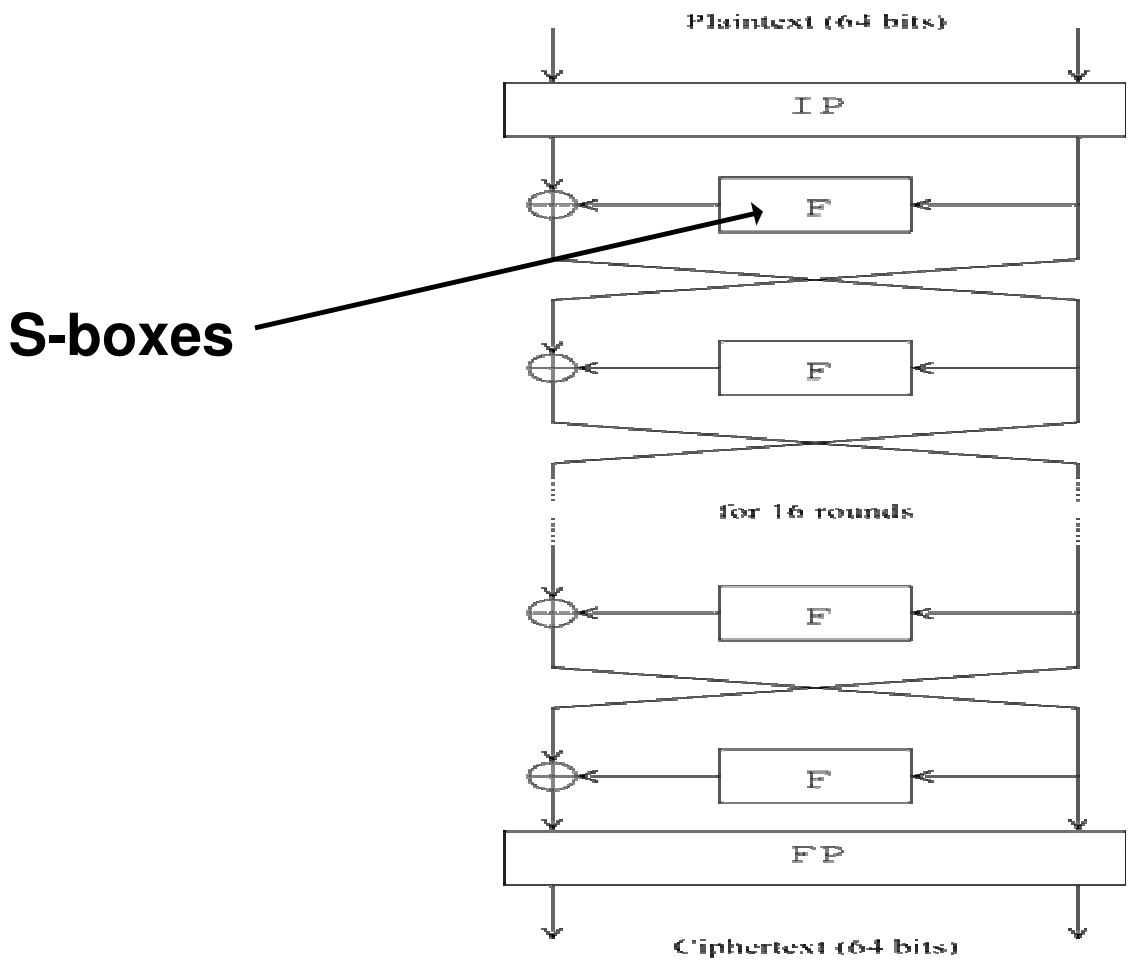
Differential cryptanalysis
Message authentication

Benny Pinkas

Plan

- Today
  - Differential cryptanalysis
  - Message authentication
  - Hash function
- Next week
  - No class
- Following that
  - Public key cryptography
  - Number theory

DES diagram


Plaintext (64 bits)
IP
F
S-boxes
F
for 16 rounds
F
F
FP
Ciphertext (64 bits)

Differential Cryptanalysis [Biham-Shamir]

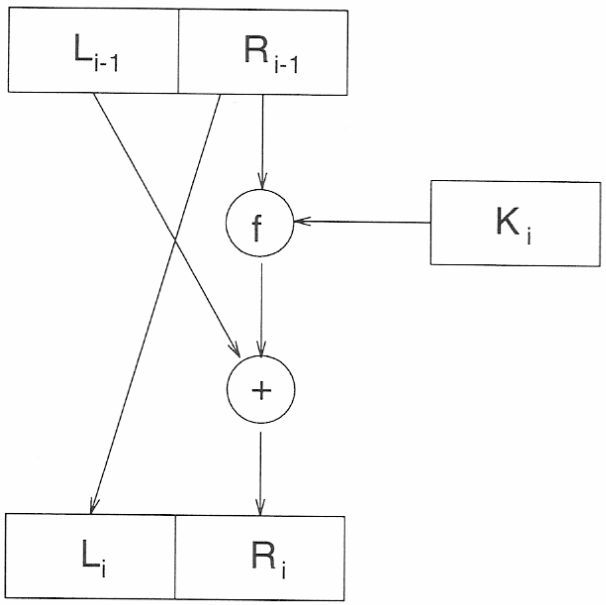
- The first attack to reduce the overhead of breaking DES to below exhaustive search
- Very powerful when applied to other encryption algorithm

- Depends on the structure of the encryption algorithm
- Observation: all operations except for the s-boxes are linear
- Linear operations:
  - $a = b \oplus c$
  - a = the bits of b in (known) permuted order
  - a = the bits of a in (known) permuted order
- Linear relations can be exposed by solving a system of linear equations

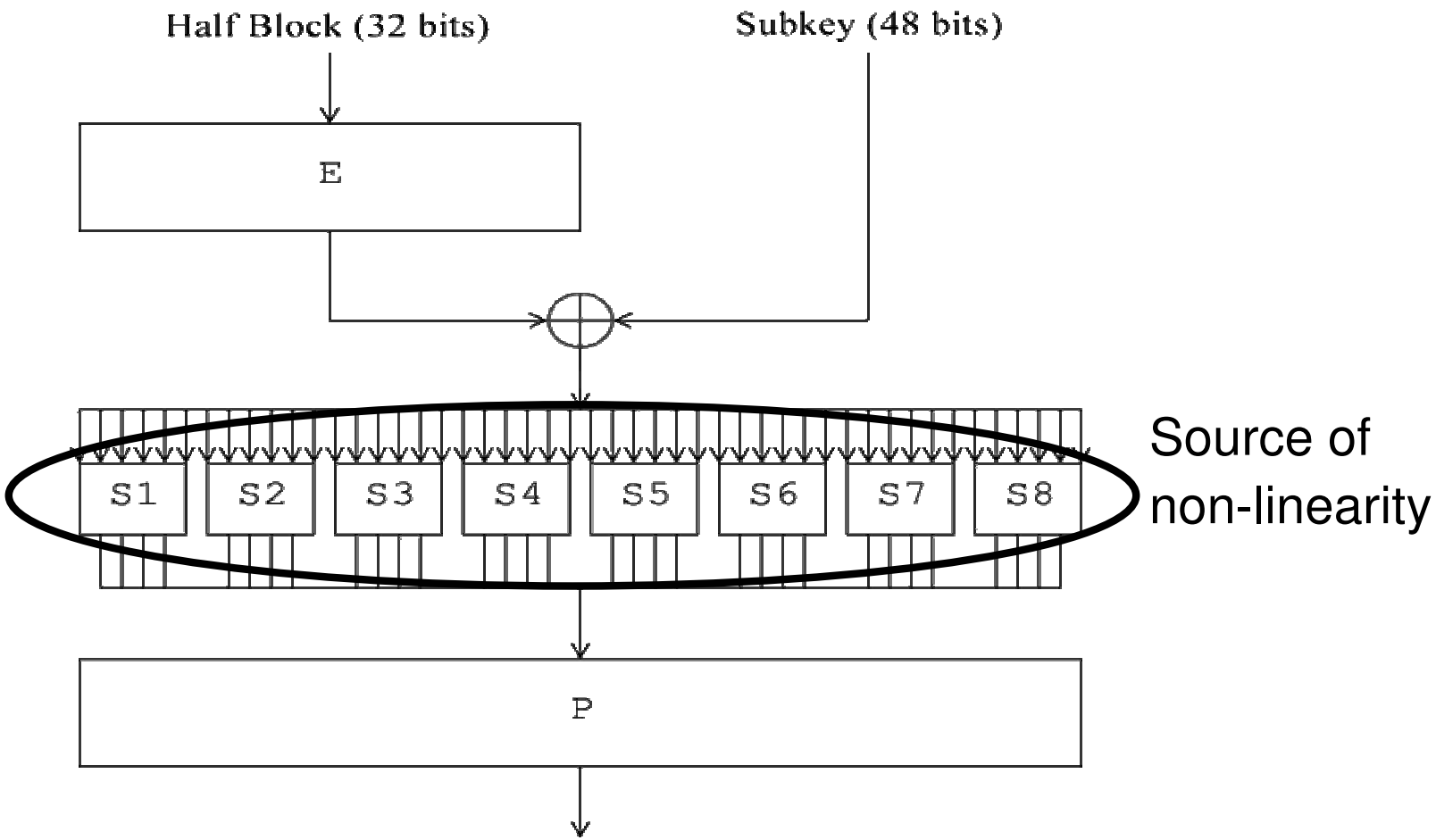
A Linear F in a Feistel Network?

- Suppose F is linear
- E.g., $F(R_{i-1},K_i) = R_{i-1} \oplus K_i$
- Then $R_i = L_{i-1} \oplus R_{i-1} \oplus K_i$

  $L_i = R_{i-1}$
- Write L_{16}, R_{16} as linear functions of L_0, R_0 and K.
  - Given L_0R_0 and $L_{16}R_{16}$ Solve and find K.
- F must therefore be non-linear.
- It is the only source of non-linearity in DES.



DES F functions



Differential Cryptanalysis

- The S-boxes are non-linear
- We study the differences between two encryptions of two different plaintexts

- Notation:
  - The plaintexts are P and P*
  - Their difference: $P' = P \oplus P^*$
  - Let X and X* be two intermediate values, for P and P*, respectively, in the encryption process.
  - Their difference is $x' = x \oplus x^*$

The advantage of looking at XORs

- It's easy to predict the difference of the results of linear operations
- Unary operations, (e.g. P is a permutation of the bits of X)
  - $(P(X))' = P(x) \oplus P(x^*) = P(x')$
- XOR
  - $(X \oplus Y)' = (x \oplus y) \oplus (x^* \oplus y^*) = x' \oplus y'$
- Mixing the key
  - $(X \oplus K)' = (x \oplus k) \oplus (x^* \oplus k) = x \oplus x^* = x'$
  - The result here is key independent (the key disappears)

Differences and S-boxes

- S-box: a function (table) from 6 bit inputs to 4 bit output
- X and X* are inputs to the same S-box, and we know their difference $X' = X \oplus X^*$.
- $Y = S(X)$
- When $X'=0$, $X=X^*$, and therefore $Y=S(X)=S(X^*)=Y^*$, and $Y'=0$.
- When $X' \neq 0$, $X \neq X^*$ and we don't know Y' for sure, but we can investigate its distribution.
- For example,

Distribution of Y' for S1

- X'=110100
- 2^6=64 input pairs, { (000000,110100), (000001,110101),…}
- For each pair compute xor of outputs of S1
- E.g., S1(000000)=1110, S1(110100)=1001. Y'=0111.

0000	0001	0010	0011	0100	0101	0110	0111
0	8	16	6	2	0	0	12
1000	1001	1010	1011	1100	1101	1110	1111
6	0	0	0	0	8	0	6

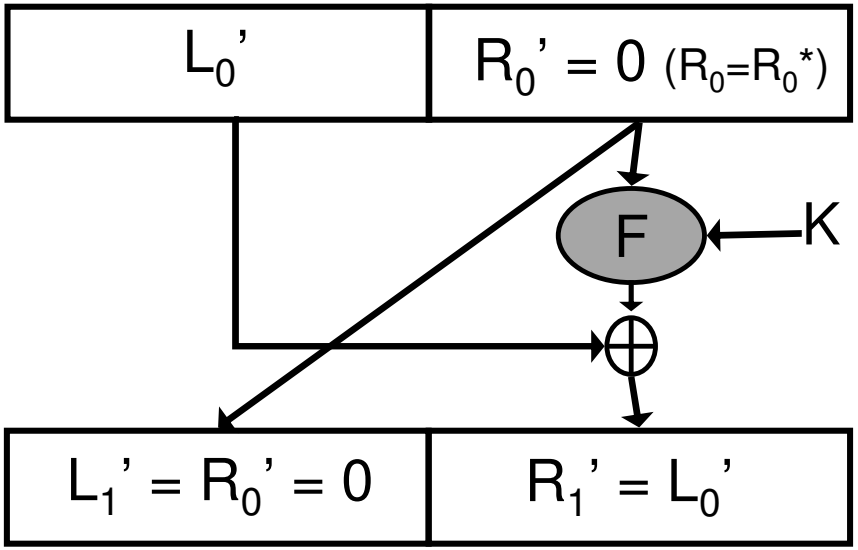
Differential Probabilities

- The probability of X' $\Rightarrow$ Y' is the probability that a pair of difference X' results in a pair of difference Y' (for a given S-box).
- Namely, the entries in the table divided by 64.

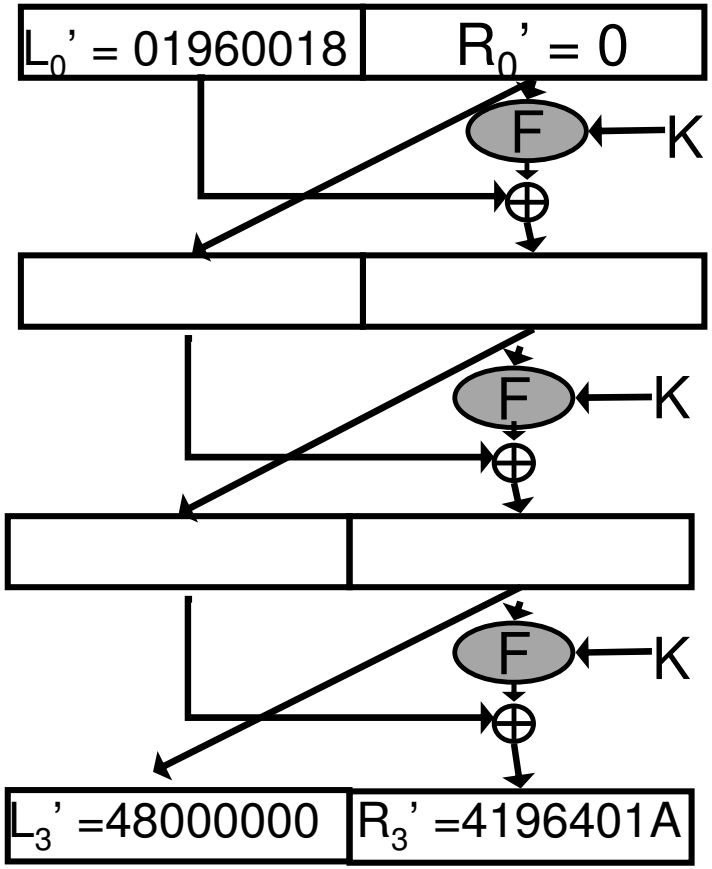
- Differential cryptanalysis uses entries with large values
  - X'=0 $\Rightarrow$ Y'=0
  - Entries with value 16/64.

Warmup

Inputs: L_0R_0, $L_0{*}R_0{*}$, s.t. $R_0=R_0{*}$.
Their xor is $L_0'R_0'$



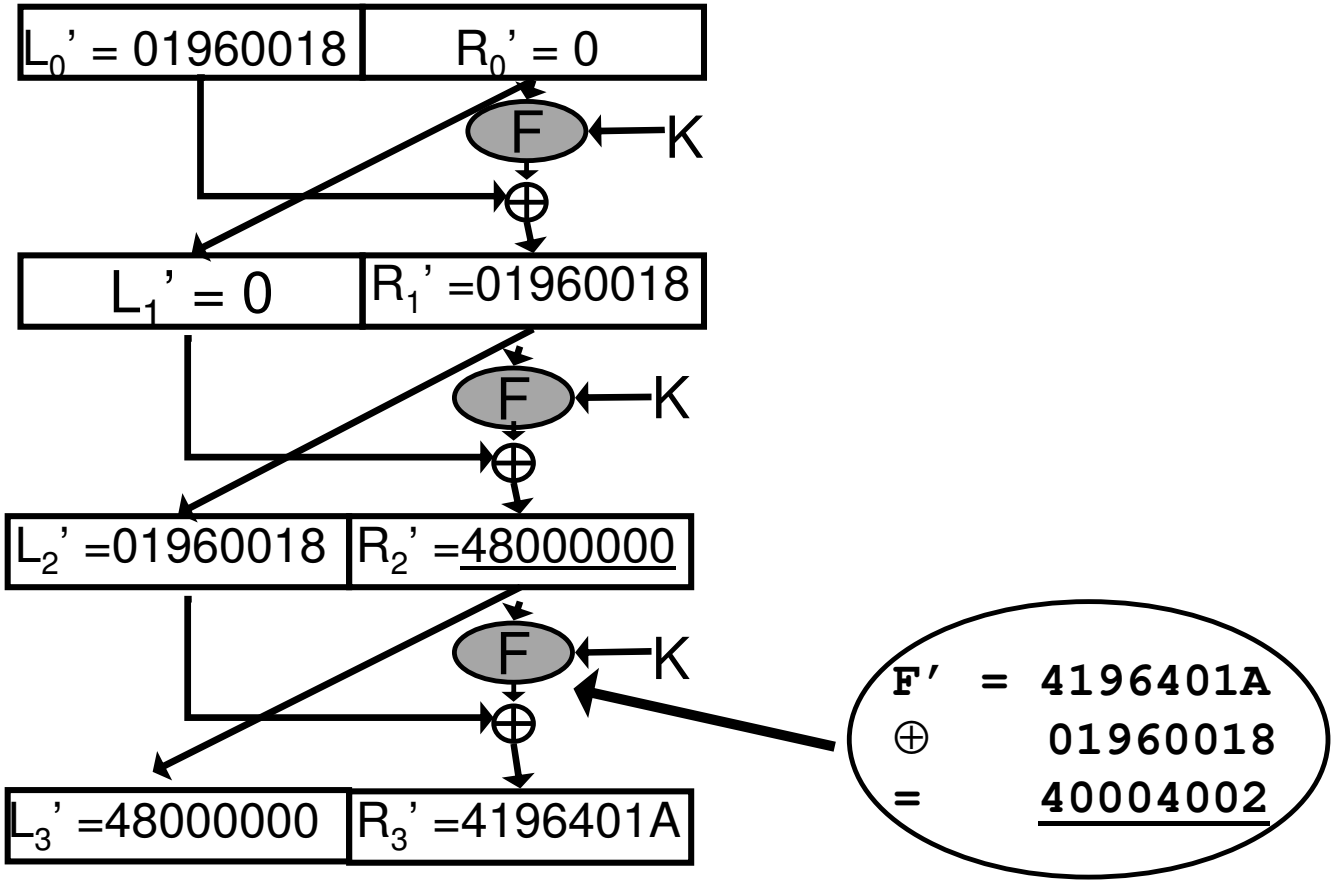
3 Round DES



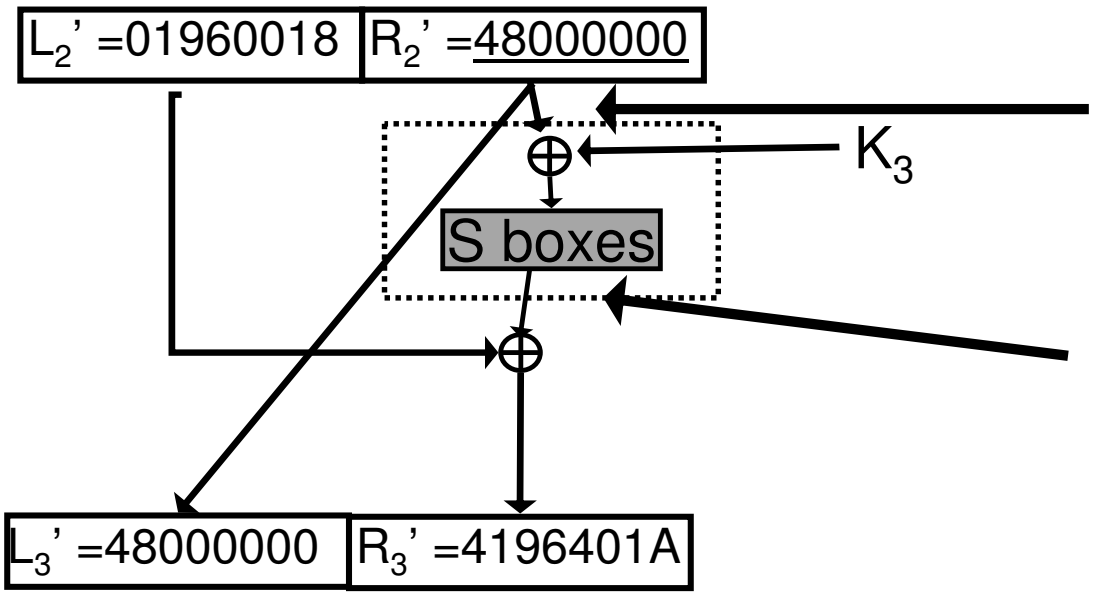
Known plaintext/ciphertext differences

Also, save actual ciphertext values

Intermediate differences equal to plaintext/ciphertext differences



Finding K



The actual two inputs to F are known

Output xor of F (i.e., S boxes) is 40004002

⇒Table enumerates options for the pairs of inputs to S box

Find which K_3 maps the inputs to an s-box input pair that results in the output pair!

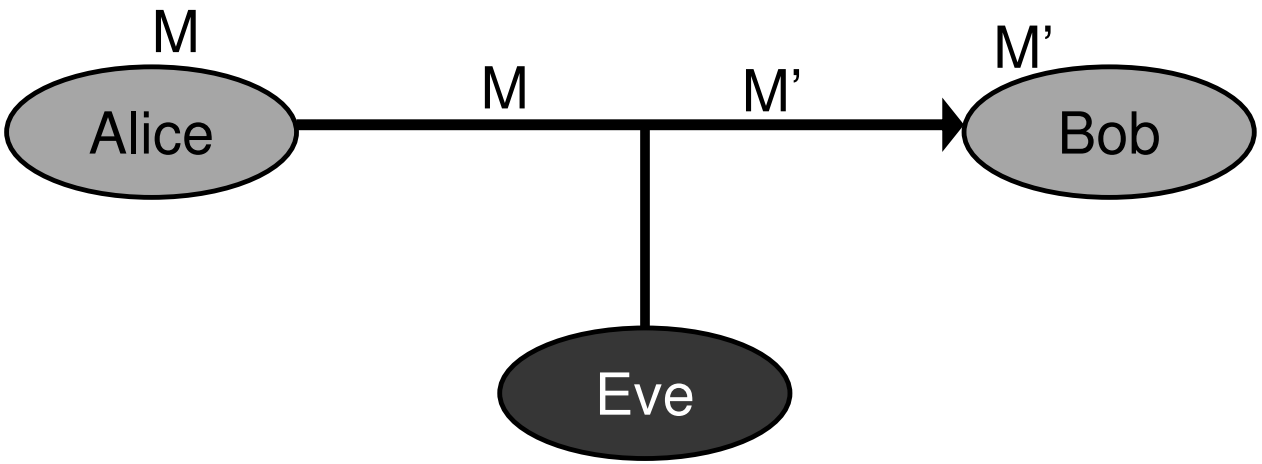
DES with more than 3 rounds

- Carefully choose pairs of plaintexts with specific xor, and determine xor of pairs of intermediate values at various rounds.
- E.g., if $L_0'=40080000_x$, $R_0'=04000000_x$
  Then, with probability ¼, $L_3'=04000000_x$, $R_3'=4008000_x$

- 8 round DES is broken given 2^{14} chosen plaintexts.
- 16 round DES is broken given 2^{47} chosen plaintexts...

Data Integrity, Message Authentication

- Challenge: an *active* adversary might change messages exchanged between Alice and Bob



- Authentication is orthogonal to secrecy. A relevant challenge regardless of whether encryption is applied.

One Tine Pad

- OTP is a perfect cipher, yet provides no authentication
  - Plaintext $x_1x_2\ldots x_n$
  - Key $k_{1k2}\ldots k_n$
  - Ciphertext $c_1=x_1\oplus k_{1,}$ $c_2=x_2\oplus k_2,\ldots,c_n=x_n\oplus k_n$

- Adversary changes, e.g., c_2 to $1\oplus c_2$
- User decrypts $1\oplus x_2$

- Error-detection codes are insufficient

Definitions

- Scenario: Alice and Bob share a secret key K.
- Authentication algorithm (Message Authentication Code): $MAC_K(m)$.
- Verification algorithm: $V_K(m, a)$.
  - $V_K(m, MAC_K(m)) = accept.$
  - For $a \neq MAC_K(m)$, $V_K(m,a) = reject.$

- Common usage:
  - Every message m is sent as $(m, MAC_K(m)).$
  - Recipient receives (m,a), computes $MAC_K(m).$
  - $V_K(m,a) = 1$ iff $MAC_K(m)=a.$

Requirements

- Security: The adversary,
  - Knows the MAC algorithm (but not K).
  - Is given many pairs $(m_i, MAC_K(m_i))$, where the m_i values might also be chosen by the adversary (chosen plaintext).
  - Cannot compute $(m, MAC_K(m))$ for any new m $(\forall i\ m \neq m_i)$.
  - The adversary must not even compute $MAC_K(m)$ for a message m which is "meaningless" (since we don't know the context of the attack).

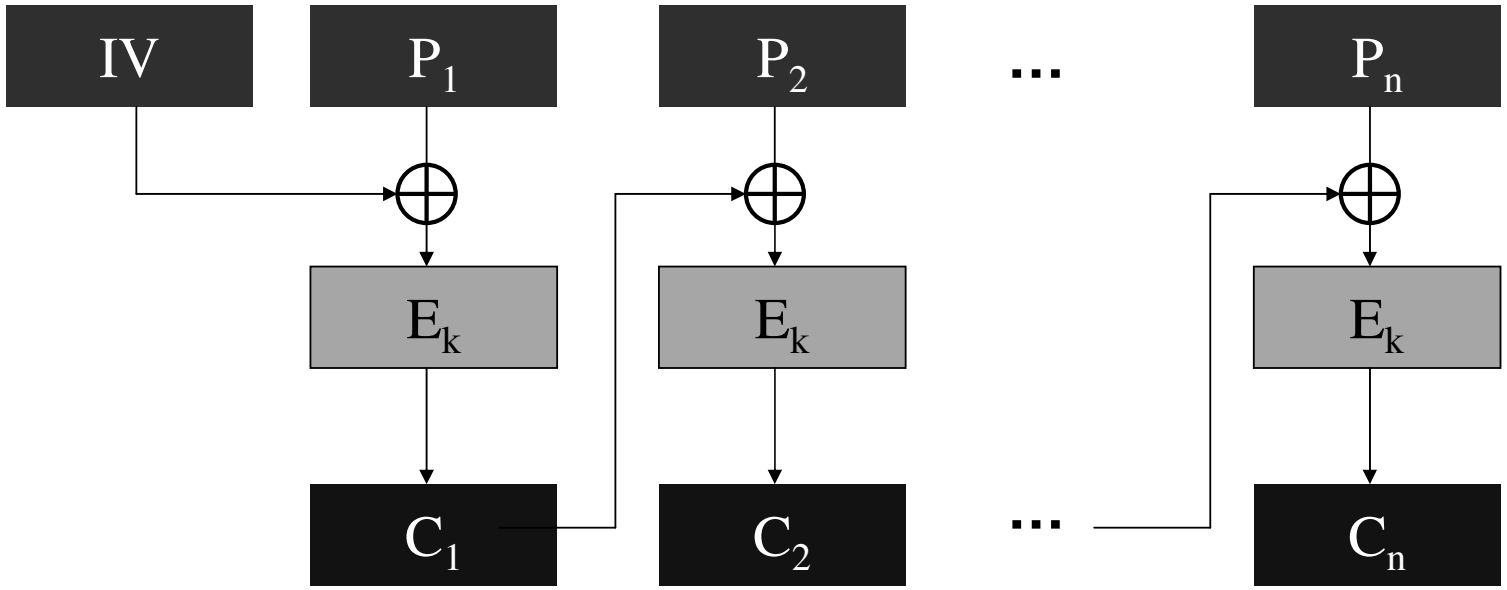
- Efficiency: output must be of fixed length, and as short as possible.
  - $\Rightarrow$ The MAC function is not 1-to-1.

Constructing MACs

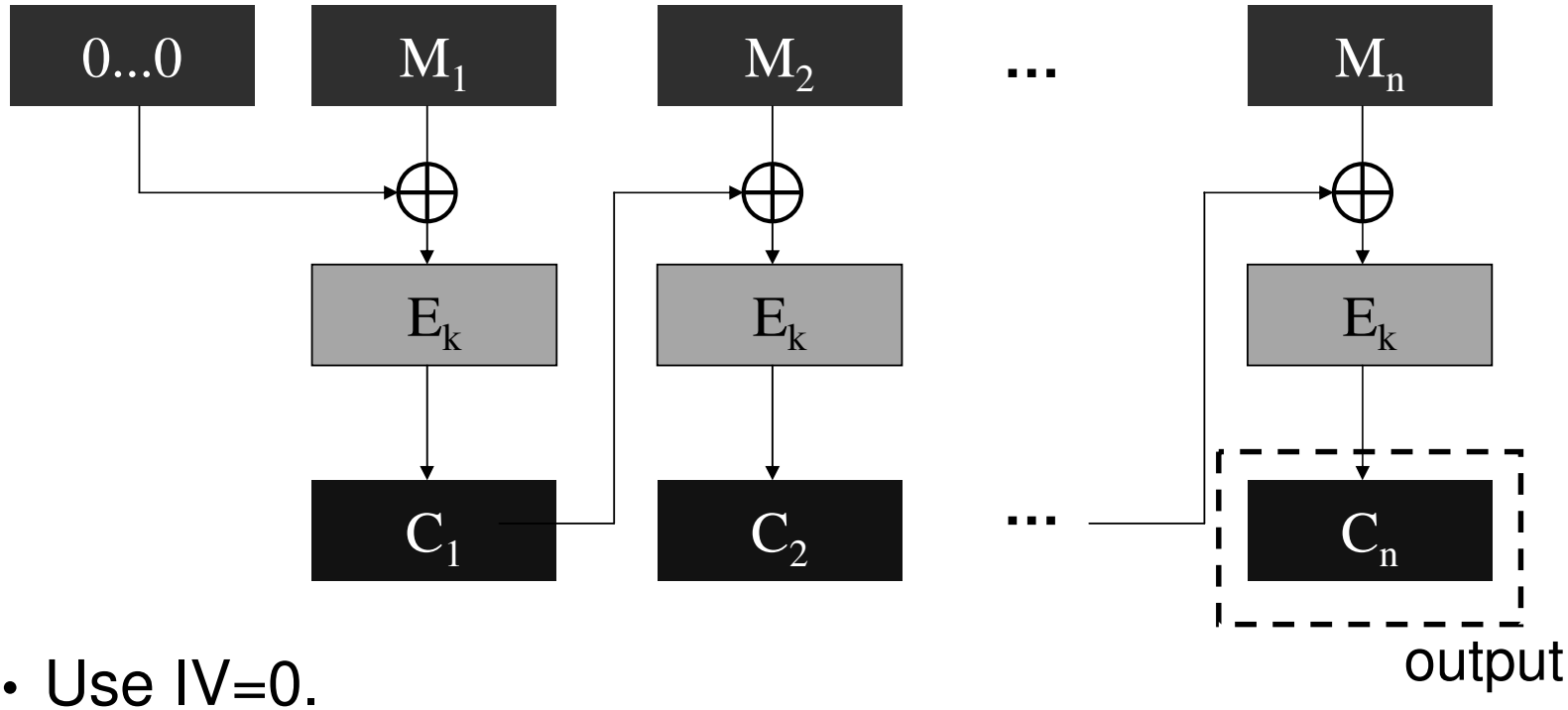
- Based on block ciphers (CBC-MAC)
- Based on hash functions
  - More efficient
  - At the time, encryption technology was controlled (export restricted) and it was preferable to use other means when possible

CBC

- Reminder: CBC encryption
- Plaintext block is xored with previous ciphertext block



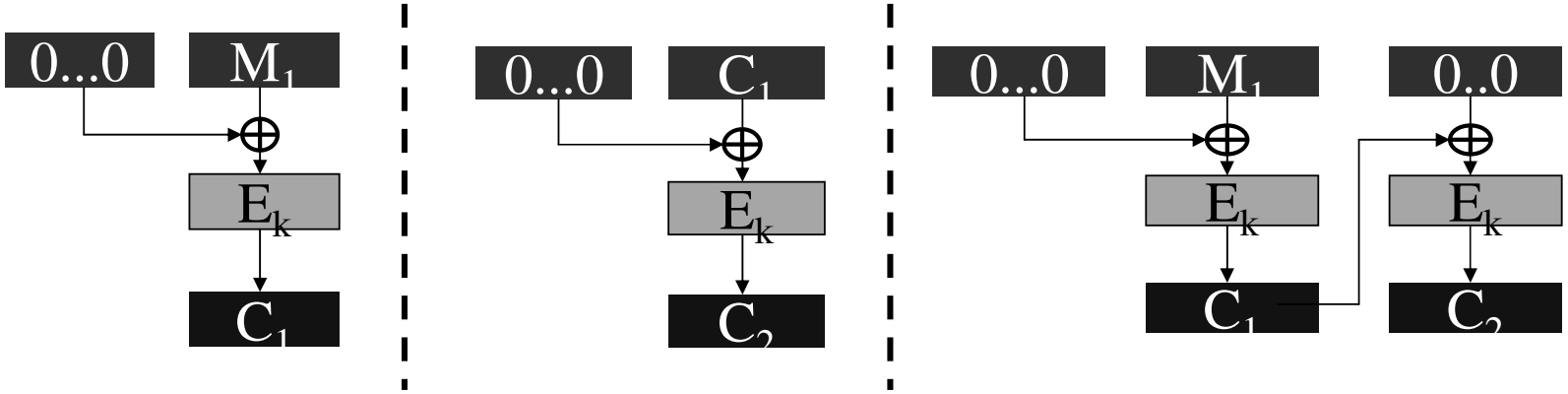
CBC MAC



- Use IV=0.
- Encrypt M in CBC mode, using the MAC key. Discard $C_1,\ldots,C_{n-1}$ and define $MAC_K(M_1,\ldots,M_n)=C_n$.

Security of CBC-MAC

- Claim: if E_K is pseudo-random then CBC-MAC, applied to *fixed length messages*, is resilient to forgery.
- But, insecure if messages of different lengths are allowed
  - Get $C_1 = \text{CBC-MAC}_K(M_1) = E_K(0 \oplus M_1)$
  - Ask for MAC of C_1, i.e., $C_2 = \text{CBC-MAC}_K(C_1) = E_K(0 \oplus C_1)$
    $= E_K(C_1 \oplus 0) = E_K(\, E_K(0 \oplus M_1) \oplus 0) = \text{CBC-MAC}_K(M_1 \mid 0)$



CBC-MAC for variable length messages

- Solution 1: The first block of the message is set to be its length. I.e., to authenticate $M_1,\ldots,M_n$, apply CBC-MAC to $(n,M_1,\ldots,M_n)$.
  - Drawback: The message length (n) has to be known in advance.
- "Solution 2": apply CBC-MAC to $(M_1,\ldots,M_n,n)$
  - Message length does not have to be known is advance
  - But, this scheme is broken
- Solution 3: (preferable)
  - Use a second key K'.
  - Compute $MAC_{K,K'}(M_1,\ldots,M_n) = E_{K'}(MAC_K(M_1,\ldots,M_n))$
  - Essentially same overhead as CBC-MAC

Hash functions

- A hash function h:X->Z maps long inputs to fixed size outputs. (|X|>|Z|)
- No secret key. The hash function algorithm is public.
- There are collisions *(x≠y* for which *h(x)=h(y))*.

- Weak collision resistance: for any x∈X, given x, it is hard to find x'≠x such that h(x)=h(x'). (Also known as "universal one-way hash", or "target collision resistance").
- Strong collision resistance: it is hard to find any x,x' for which h(x)=h(x').

- Real world hash functions: MD5, SHA-1.

The Birthday Paradox

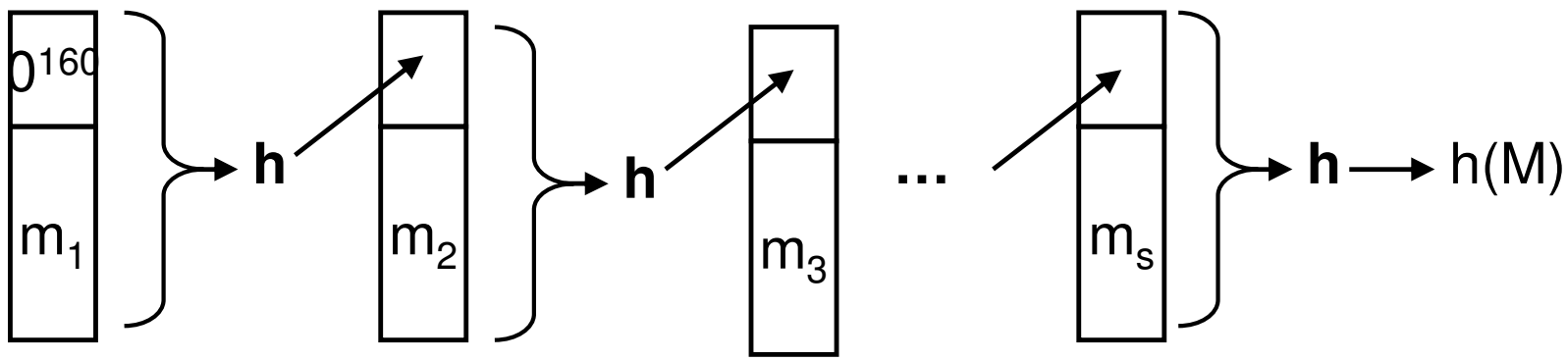
- For 23 people chosen at random, the probability that two of them have the same birthday is ½.
- (compare to: the probability that one or more of them have the same birthday as G.W. Bush is about 23/365 (more accurately, $1-(1-1/365)^{23}$.)

- More generally, for a random h:X->Z, if we choose $s=1.17|Z|^{½}$ elements of Z at random, the probability that two of them are mapped to the same image is > ½.

- Implication: it's harder to achieve strong collision resistance
  - Hash function with a 40 bit output
    - Find x,x' with h(x)=h(x') after about 1,117,000 tries.
    - Find x≠0 s.t. h(x)=h(0) after about 2^{39} attempts.

Hash functions

- Input block length is usually 512 bits ($|X|=512$)
- Output length is at least 160 bits (birthday attacks)
- Extending the domain to arbitrary inputs
  - Suppose $h:\{0,1\}^{512} \rightarrow \{0,1\}^{160}$
  - Input: $M=m_1 \ldots m_s$, $|m_i|=512-160=352$.
  - $y_0=0^{160}$. $y_i=h(y_{i-1},m_i)$. $h(M)=y_s$.
  - Why is it secure? What about different length inputs?



Basing MACs on Hash Functions

- Hash functions are not keyed. MAC_K uses a key.
- Idea: MAC by combining message and a secret key, and hashing with a collision resistant hash function.
  - E.g. $MAC_K(m) = h(k,m)$. (insecure.., given $MAC_K(m)$ can compute $MAC_K(m|m')$.)
  - $MAC_K(m) = h(m,k)$. (insecure.., use a birthday attack to find m,m' such that h(m)=h(m').)

- Difficulty:
  - Want to prove that if MAC is insecure than so is hash function h.
  - Insecurity of MAC: adversary can generate $MAC_K(m)$ without knowing k.
  - Insecurity of h: adversary finds collisions ($x \neq x'$, $h(x)=h(x')$.)

HMAC

- Input: message *m,* a key *K*, and a hash function *h*.
- $HMAC_K(m) = h(K \oplus opad, h(K \oplus ipad, m))$
  - where ipad, opad are 64 byte long fixed strings
  - K is 64 byte long (if shorter, append 0s to get 64 bytes).
- Overhead: as applying h to m, plus an additional invocation to a short string.

- It was proven [BCK] that if HMAC is broken then either
  - h is not collision resistant (even when the initial block is random and secret)
  - The output of h is not “unpredcitable” (when the initial block is random and secret)
- HMAC is used everywhere (SSL, IPSec).

What we learned today

- Differential cryptanalysis of DES
  - Depends on the structure of the cipher

- Message authentication
  - CBC MAC
  - Hash functions
  - The birthday paradox
  - HMAC