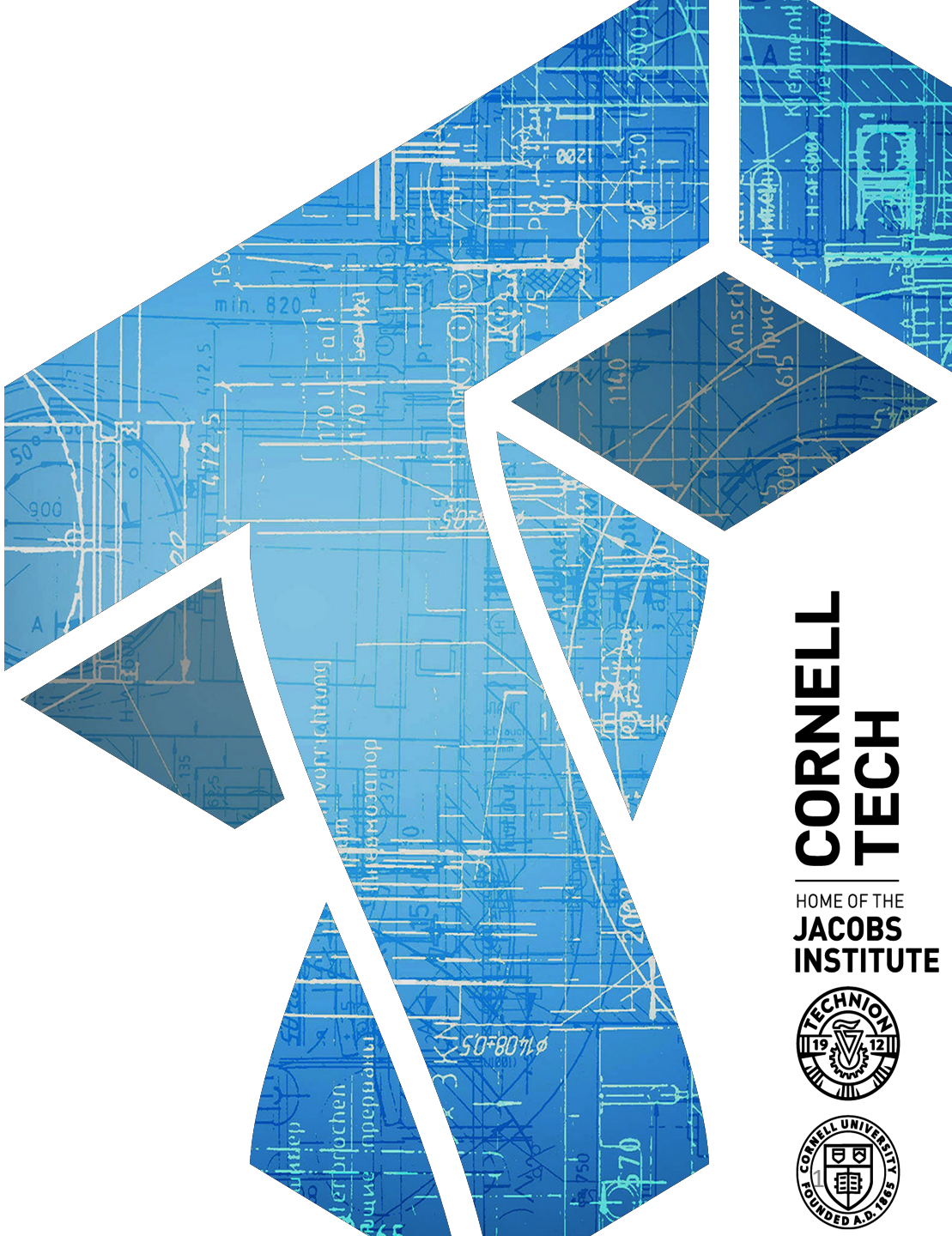
CS 5830
Cryptography



CORNELL TECH

HOME OF THE JACOBS INSTITUTE

Recap and where we're at

- Stream ciphers as computationally secure one-time pads
  - RC4 example of custom-built stream cipher
    - More modern examples: ChaCha, Xsalsa, Trivium, etc.
- Can build stream ciphers from blockciphers
  - CTR mode encryption
- Today: more on blockciphers

Block ciphers

Family of permutations, one permutation for each key

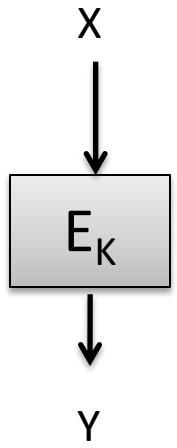
$$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

Use notation $E(K,X) = E_K(X) = Y$

Define inverse $D(K,Y) = D_K(Y) = X$

s.t. $D(K,E(K,X)) = X$

E,D must be efficiently computable



Pick K uniformly at random from $\{0,1\}^k$

K_1

X	Y
00	10
01	11
10	01
11	00

K_2

X	Y
00	11
01	10
10	00
11	01

K_3

X	Y
00	11
01	10
10	00
11	01

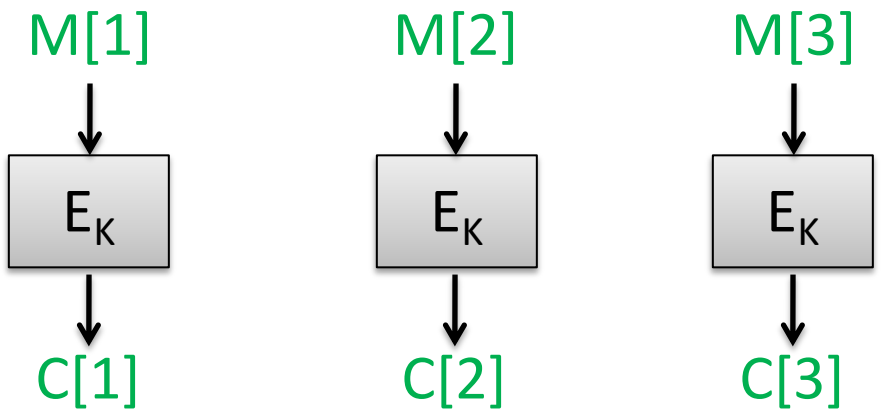
⋮

Block cipher modes of operation

Electronic codebook (ECB) mode

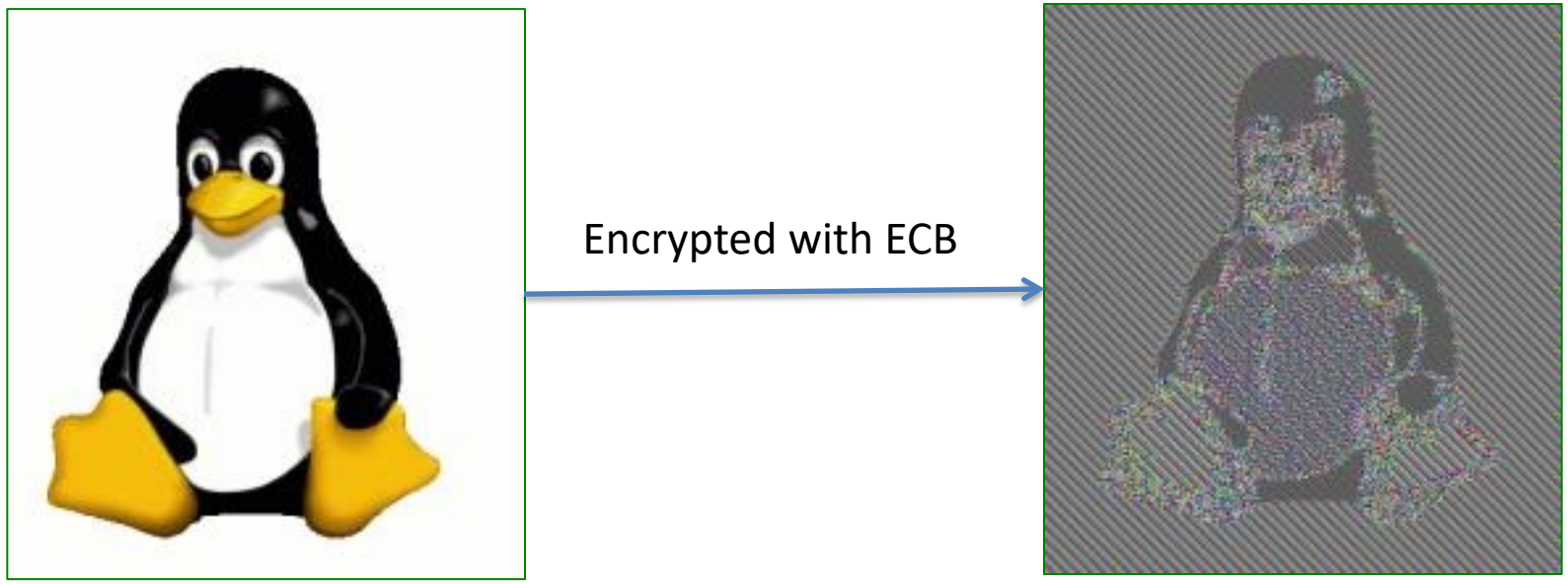
Pad message M to M[1],M[2],M[3],... where each block M[i] is n bits

Then:



How can we decrypt?

ECB mode is a more complicated looking substitution cipher

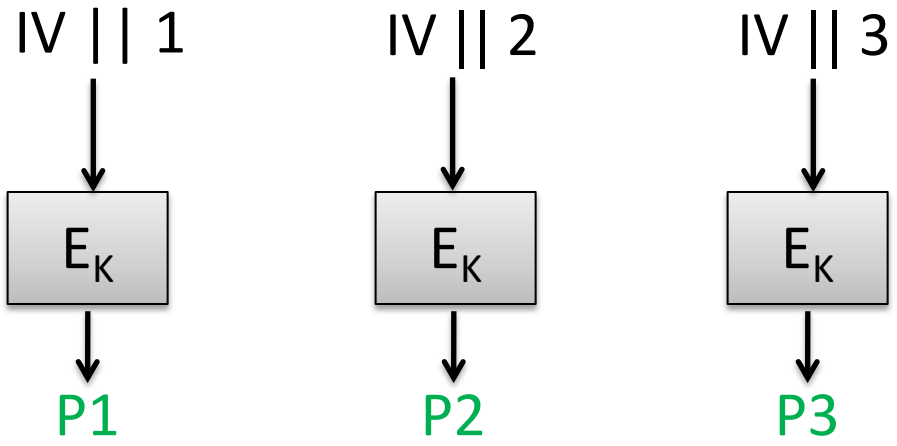


Images courtesy of
http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

CTR mode stream cipher

Counter mode stream cipher:

- Kg outputs random k-bit key for block cipher
- $G(K,IV,L) = E_K(IV \,||\, 1) \,\|\, E_K(IV \,||\, 2) \,\|\, \ldots \,\|\, \text{trunc}(E_K(IV \,||\, m))$
  where $m = \text{ceil}(L / n)$
- Here |IV| smaller than n bits, say n/2 bits



Truncate P3
to get L total bits

CTR-mode SE scheme

Kg():
$K \xleftarrow{\$} \{0,1\}^k$

Enc(K,M):
$L \leftarrow |M|$; $m \leftarrow \lceil L/n \rceil$
$IV \xleftarrow{\$} \{0,1\}^n$
$P \leftarrow E_K(IV \,||\, 1) \,\|\, \cdots \,\|\, \mathrm{trunc}(E_K(IV \,||\, m))$
Return $(IV, P \oplus M)$

Dec(K,(IV,C)):
$L \leftarrow |C|$; $m \leftarrow \lceil L/n \rceil$
$P \leftarrow E_K(IV \,||\, 1) \,\|\, \cdots \,\|\, \mathrm{trunc}(E_K(IV \,||\, m))$
Return $(IV, P \oplus C)$

Should be able to call Enc with same K for many messages. What could go wrong?

Pick a random key

ceil() is standard ceiling function
trunc() truncates last block

What security properties do we need from the block cipher?

Assume ciphertext can be parsed into IV and remaining ciphertext bits

One-time pad as a blockcipher

Family of permutations, one permutation for each key

$$E : \{0,1\}^k \times \{0,1\}^n \rightarrow \{0,1\}^n$$

Let $E_K(X) = X \oplus K = Y$ Then $D_K(Y) = Y \oplus K$

This defines a family of permutations, one for each key.

Efficient to compute

So this can be considered a blockcipher, functionally speaking

But is this secure?

Blockcipher security: key recovery

We would want a blockcipher to *at least* hide the key

Key recovery under chosen-message attack (KR-CMA):

- Adversary can see plaintext/ciphertext pair
- Goal is to recover secret key K

KR-CMA(SE, $\mathcal{A}$):
X <-$ $\mathcal{A}$
K <-$ $\{0,1\}^k$
K' <-$ $\mathcal{A}(E_K(X))$
Return (K = K')

Can you give adversary that breaks OTP as a blockcipher?

$E_K(X) = X$

Can you give blockcipher for which key can't be recovered?
(Hint: just focus on correctness)

KR security is necessary, but is it sufficient? No

Pseudorandom function (PRF) security

F is a random function:

X	Y
00	10
01	11
10	10
11	00

Choose each Y value at random, with replacement

No efficient adversary can distinguish between E_K and random function

- Even given chosen-messages attack: can query X of choosing and get Y, many times

Pseudorandom function (PRF) security

Func(n) is set of all functions
$\{0,1\}^n \rightarrow \{0,1\}^n$

O is called an ***oracle***.
A subroutine that adversary can make calls to

PRF(E, $\mathcal{C}$):
K <-$ $\{0,1\}^k$
F <-$ Func(n)
b <-$ {0,1}
b' <-$ $\mathcal{C}^O()$
Return (b = b')

O(X):
If b = 1 then
  Return $E_K(X)$
Return F(X)

(t,q,ϵ)-pseudorandom function:
no attacker $\mathcal{C}$ limited to time t and q queries to O can distinguish between E_K and random function with advantage greater than ϵ

Measuring advantage of adversary:

$$\Pr[\text{PRF}(G,L,\mathcal{C}) = 1] \leq 1/2 + \epsilon$$

Example adversary

Give an adversary $\mathcal{C}$ that achieves high advantage against E = OTP

$E_K(X) = X \oplus K$

$\mathcal{C}^{\mathcal{O}}$

$X \leftarrow \{0,1\}^n$

$Y \leftarrow \mathcal{O}(X)$ // $1^n \oplus K$

$K \leftarrow Y \oplus X$ // K

$Y' \leftarrow \mathcal{O}(K)$ // $K \oplus K$

If $Y' = 0^n$ then Return 1

Return 0

Func(n) is set of all functions $\{0,1\}^n \rightarrow \{0,1\}^n$

PRF(E, $\mathcal{C}$):
K <-$ $\{0,1\}^k$
F <-$ Func(n)
b <-$ {0,1}
b' <-$ $\mathcal{C}^{O}$()
Return (b = b')

O(X):
If b = 1 then
  Return $E_K(X)$
Return F(X)

Example adversary

Give an adversary $\mathcal{C}_B^{\mathcal{O}}$ that achieves high advantage against any E in time 2^k

Func(n) is set of all functions $\{0,1\}^n \rightarrow \{0,1\}^n$

$\mathcal{C}_B^{\mathcal{O}}$

$X \leftarrow 0^n$

$Y \leftarrow \mathcal{O}(X)$

For $i = 0$ to $2^k - 1$:

$K \leftarrow \langle i \rangle$

$y' \leftarrow E_K(X)$

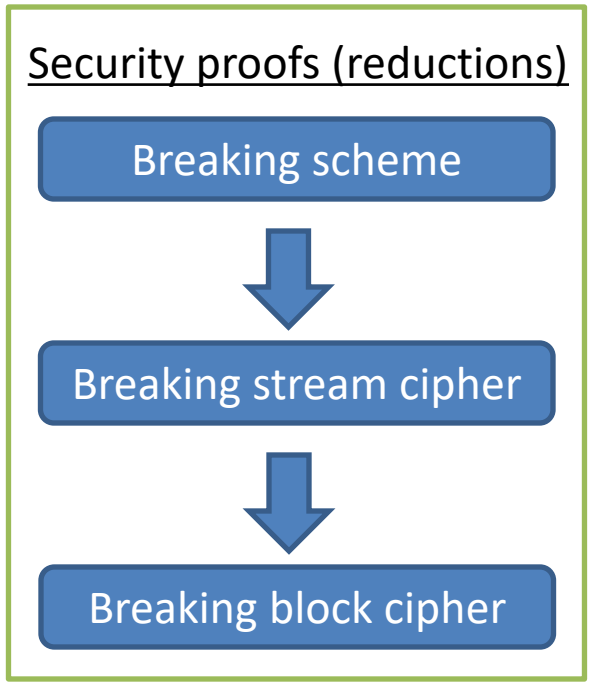
If $y = y'$ then Found $K = \tilde{K}$

PRF(E, $\mathcal{C}$):
K <-$ {0,1}^{k}
F <-$ Func(n)
b <-$ {0,1}
b' <-$ $\mathcal{C}^{\mathcal{O}}$()
Return (b = b')

O(X):
If b = 1 then
  Return $E_K(X)$
Return F(X)

Reduction-based security analysis

Goal: show that if blockcipher is secure, then CTR encryption is secure



Attacker $\mathcal{A}$ violates
(t, ϵ)-indistinguishability security

For any $\mathcal{A}$ need to give $\mathcal{B}$ for which $t' \approx t$ and $\epsilon' \approx \epsilon$

Attacker $\mathcal{B}$ that violates
(t', ϵ')-pseudorandom generator security of G

For any $\mathcal{B}$ need to give $\mathcal{C}$ with $t'' \approx t'$, $\epsilon'' \approx \epsilon'$, q smallish

Attacker $\mathcal{C}$ that violates
(t'', q, ϵ'')-pseudorandom function security of E

Reduces security analysis task to analyzing block cipher

Confidence in block cipher security gives confidence in scheme's security

This is a bit simplistic: in fact want multi-message security

CTR-mode SE scheme

Kg():
K <-$ $\{0,1\}^k$

Pick a random key

Enc(K,M):
L <- |M| ; m <- ceil(L/n)
IV <-$ $\{0,1\}^n$
P <- $E_K(IV \oplus 1) \| \cdots \| \text{trunc}(E_K(IV \oplus m))$
Return (IV, P $\oplus$ M)

Should be able to call Enc with same K for many messages. What could go wrong?

Dec(K,(IV,C)):
L <- |C| ; m <- ceil(L/n)
P <- $E_K(IV \oplus 1) \| \cdots \| \text{trunc}(E_K(IV \oplus m))$
Return (IV, P $\oplus$ C)

Assume ciphertext can be parsed into IV and remaining ciphertext bits

Many-message attack against CTR

Adversary obtains q CTR encryptions (IV_1, C_1) , (IV_2, C_2) , … , (IV_q, C_q) under same key K

Look for $i \neq j$ such that $IV_i = IV_j$

Output $C_i \oplus C_j = M_i \oplus M_j$

$\Pr[IV_1 = IV_2] = \frac{1}{2^n}$

This reveals a lot of partial information about plaintext

How big does q need to get before attack succeeds with good probability?

The Birthday Bound

Throw q balls randomly into N bins. Let Coll(N,q) be event that some bin has (at least) two balls. Then:

$$\frac{0.3q(q-1)}{N} \leq \Pr[\text{Coll}(N,q)] \leq \frac{q(q-1)}{2N}$$

See Appendix A of: https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf

The Birthday Bound

Throw q balls randomly into N bins. Let Coll(N,q) be event that some bin has (at least) two balls. Then:

$$\frac{0.3q(q-1)}{N} \leq \Pr[\text{Coll}(N,q)] \leq \frac{q(q-1)}{2N}$$

Proof of upper bound: Let Coll_i be event that collision occurs when throwing ball i

logical OR — union bound

$$\Pr[\text{Coll}(N,q)] = \Pr[\bigvee_{i=1}^{q} \text{Coll}_i] \leq \Pr[\text{Coll}_1] + \Pr[\text{Coll}_2] + \ldots + \Pr[\text{Coll}_q]$$

$$\leq 0/N + 1/N + \ldots + (q-1)/N$$

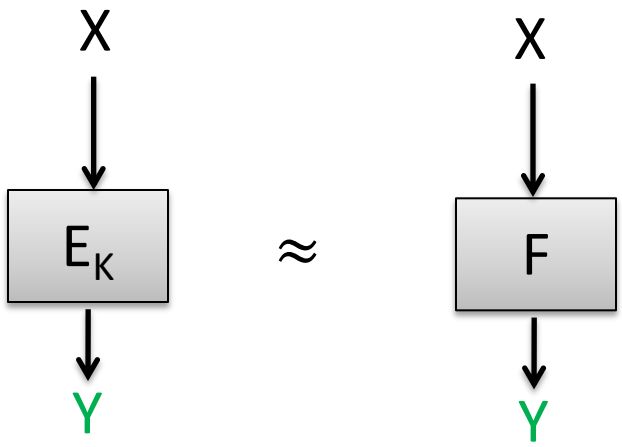
$$= q(q-1)/2N$$

See Appendix A of: https://web.cs.ucdavis.edu/~rogaway/classes/227/spring05/book/main.pdf

Birthday PRF attack against *any* block cipher

E_K is a ***permutation***

X	Y
00	10
01	11
10	01
11	00



F is a random ***function***:

X	Y
00	10
01	11
10	10
11	00

Choose each Y value at random, with replacement

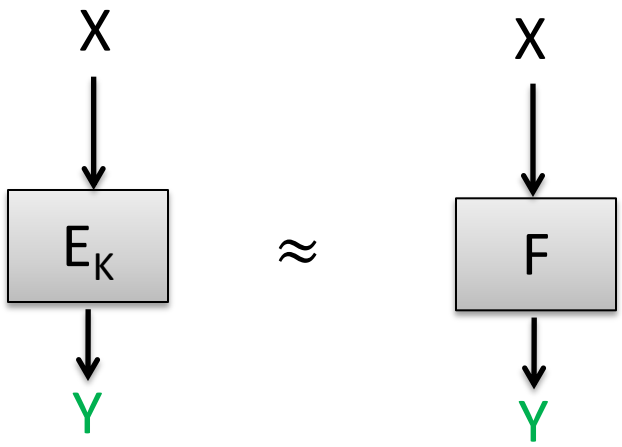
For any sequence of queries by adversary what is guaranteed to ***never*** happen when interacting with E_K that could happen when interacting with F?

If n is bit length of Y values, how many outputs of F do we have to see before we are likely to see repeat?

Birthday PRF attack against *any* block cipher

E_K is a ***permutation***

X	Y
00	10
01	11
10	01
11	00



F is a random ***function***:

X	Y
00	10
01	11
10	10
11	00

Choose each Y value at random, with replacement

Generic PRF adversary: query oracle O on q distinct inputs $X_1, \ldots, X_q$ to get outputs $Y_1, \ldots, Y_q$. If exists $Y_i = Y_j$ guess that you are interacting with F.

Succeeds if $q \approx 2^{n/2}$

This means we need n to be large enough to make $2^{n/2}$ intractably large

More on block cipher security

- Need key length k to make 2^k computationally intractable
- Need block length n to make $2^{n/2}$ computationally intractable

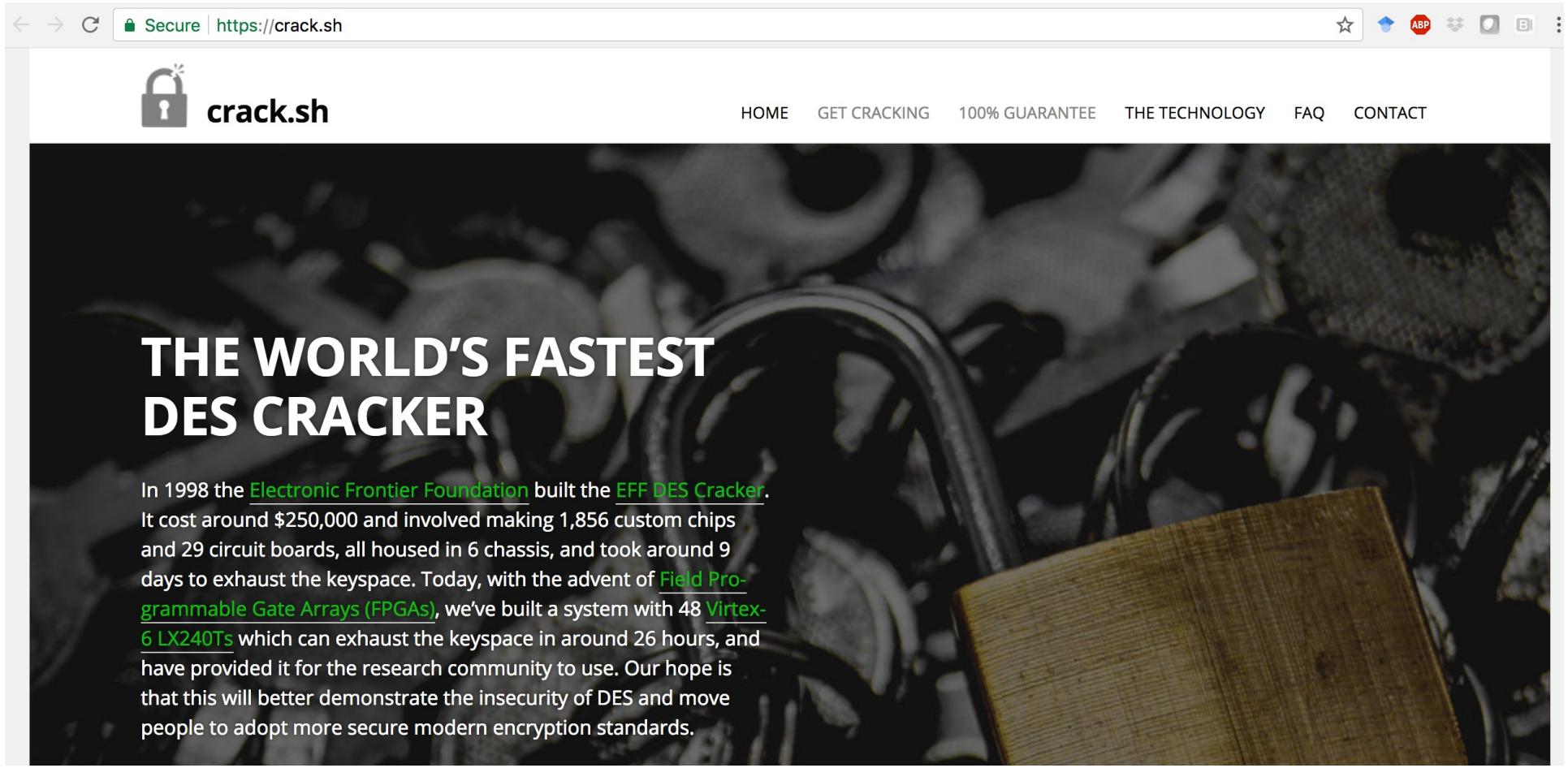
- Pseudorandom permutation (PRP) security similar to PRF, but indistinguishability from random permutation
  - We want PRF security in CTR application, so focused on that
  - Birthday bound proof equivalence of PRP/PRF notions for $q \ll 2^{n/2}$
- Chosen-ciphertext attack variants
  - Strong PRF and PRP security gives access to inverse oracle to which Y values of adversary's choosing can be queried

Block cipher design

- A big topic with ~50 year history
- Data Encryption Standard (DES) designed in 1970s
  - Uses Feistel network structure
- Advanced Encryption Standard (AES) designed in 1990s
  - Uses substitution-permutation network structure

The History

- DES (under name Lucifer) designed by IBM in 1970s
- NIST standardized it
  - NSA evaluated it and made suggested changes to shorten key length to 56 bits and changes to S-boxes
  - Many public criticisms of these changes, though S-boxes change actually strengthened DES
- AES competition run by NIST (1997-2000)
  - Many good submissions (15 total submissions)
  - AES chosen as winner


Secure https://crack.sh
crack.sh
HOME
GET CRACKING
100% GUARANTEE
THE TECHNOLOGY
FAQ
CONTACT
THE WORLD'S FASTEST DES CRACKER
In 1998 the Electronic Frontier Foundation built the EFF DES Cracker. It cost around $250,000 and involved making 1,856 custom chips and 29 circuit boards, all housed in 6 chassis, and took around 9 days to exhaust the keyspace. Today, with the advent of Field Programmable Gate Arrays (FPGAs), we've built a system with 48 Virtex-6 LX240Ts which can exhaust the keyspace in around 26 hours, and have provided it for the research community to use. Our hope is that this will better demonstrate the insecurity of DES and move people to adopt more secure modern encryption standards.

Data encryption standard (DES)

Originally called Lucifer
- team at IBM
- input from NSA
- standardized by NIST in 1976

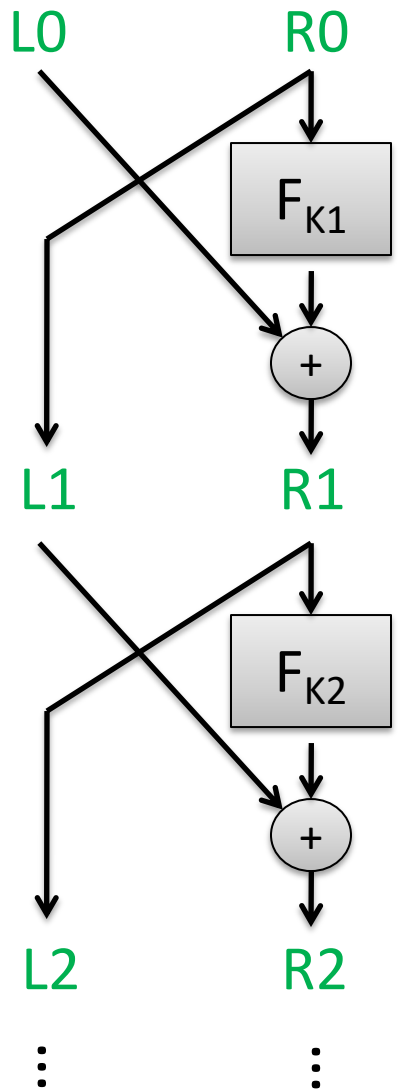
n = 64
k = 56

Number of keys:
72,057,594,037,927,936

Split 64-bit input into L0,R0 of 32 bits each

Repeat Feistel round 16 times

Each round applies function F using separate round key



(Omitting initial and final permutations)

Key schedule expands 56-bit key into 16 48-bit round keys

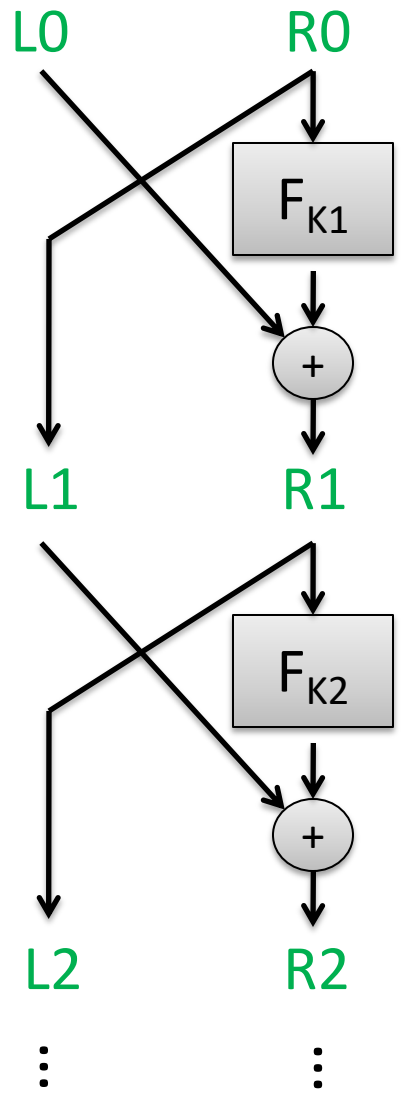
Feistel functions: intuition

Converts any function into a permutation

If we repeat enough Feistel rounds and F is a secure PRF, then block cipher is secure PRF

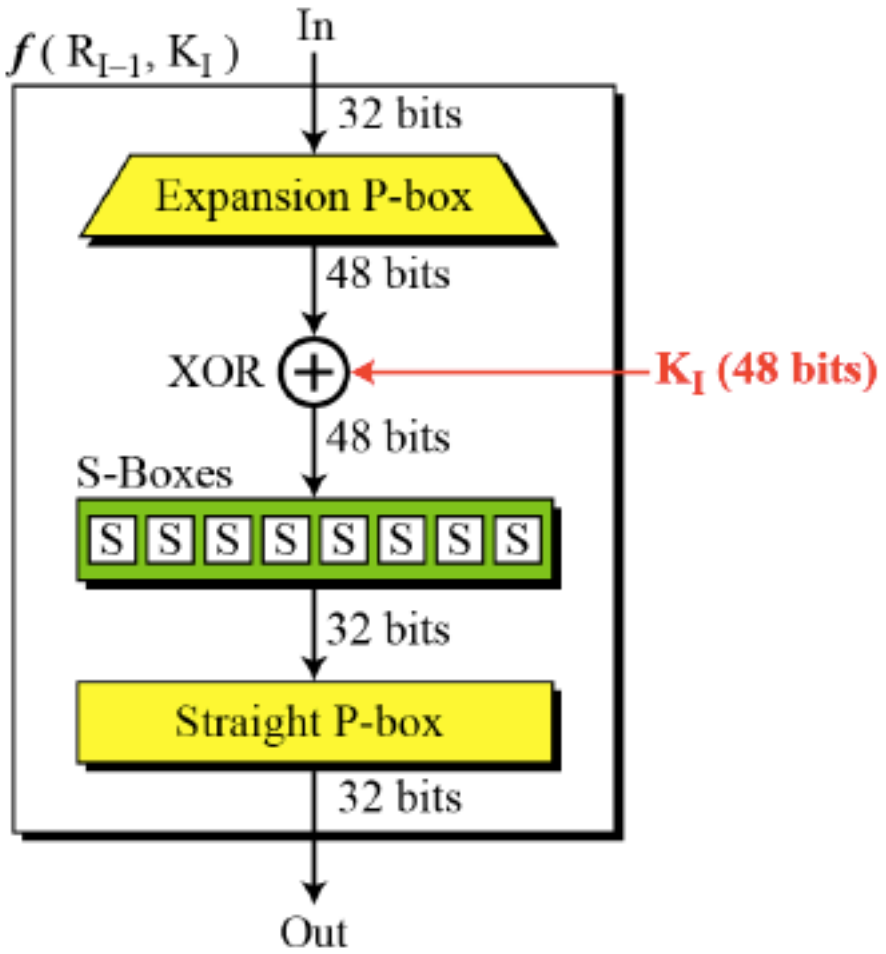
How many rounds do we need?

Luby-Rackoff proved that 3 rounds suffice for chosen-plaintext attacks, assuming at most $2^{n/4}$ uses under same secret key



DES round functions

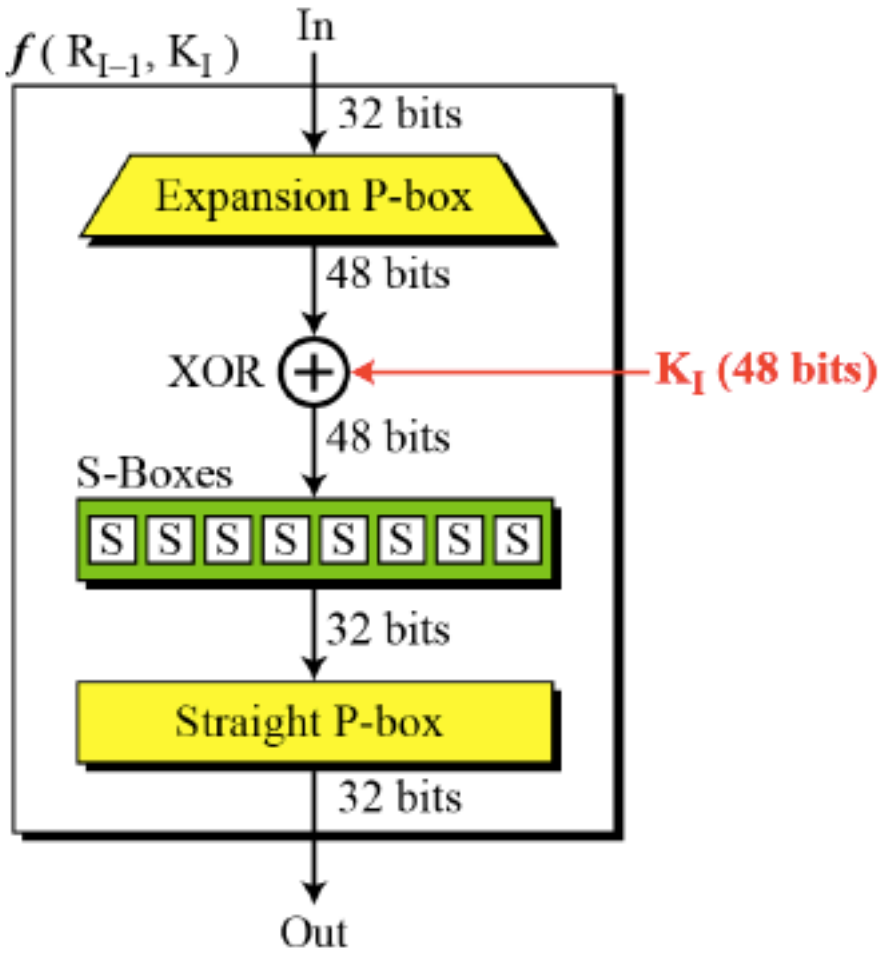
- P-box expands 32 bits to 48 bits and permutes
  - Copies over most bits, duplicates some
- 8 S-Boxes:
  - Each a 6-bit to 4-bit lookup table
- XOR in round key
  - 16 48-bit round keys derived via key schedule from 56 bit key deterministically
- How S-boxes chosen? Why particular permutations?
  - ***Confusion***: each output bit should depend on every bit of the key
  - ***Diffusion***: each output bit should depend on every input bit
  - Related term: ***avalanche effect***



Linear cryptanalysis

[Matsui 1993]

- Approximate S-box behavior by linear functions, e.g.,:
  - X1 + X2 + X6 = Y1 + Y2 + Y4
- S-boxes exhibit some biases, meaning some linear functions are satisfied with higher probability (over uniform inputs)
- Can carefully "pile up" linear approximations across multiple DES rounds
- Use linear approximations to efficiently narrow down search for key, given lots of encryptions of uniform messages



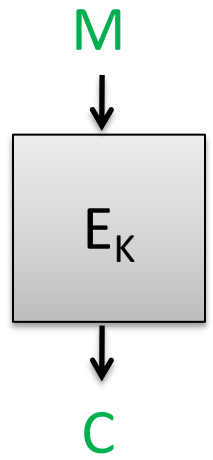
http://www.cs.bc.edu/~straubin/crypto2017/heys.pdf

Linear cryptanalysis

[Matsui 1993]

Approximate E_K by linear boolean function satisfied w/ probability different from ½ (over choice of random message)

For string X and set S ⊆ {0,…,n-1}, let X[S] = $\bigoplus_{i \in S} X_i$



Suppose we know S_m , S_c , S_k such that

$$\Pr\left[\, M[S_m] \oplus C[S_c] = K[S_k] \,\right] = \frac{1}{2} + \epsilon$$

where $C = E_K(M)$ and M chosen uniformly

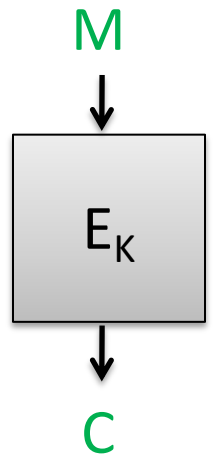
Then this allows recovering 1 bit of K using ***known plaintext attack***

Linear cryptanalysis: recovering one bit

$$\Pr\left[\, M[S_m] \oplus C[S_c] = K[S_k] \,\right] = \frac{1}{2} + \epsilon \qquad (1)$$

Let M_1 , … , M_q be uniform messages, $C_i = E_K(M_i)$

Let $K[S_k] = \text{Maj}(\{ M_i[S_m] \oplus C_i[S_c] \}_i)$



Theorem 1 *Let $\mathcal{E}$ be a cipher such that (1) holds with $\epsilon > 0$, and let $K \in \mathcal{K}$. Let $M_1, \ldots, M_q$ be sampled uniformly from $\{0,1\}^n$ and let $C_i = E_K(M_i)$ for $i \in \{1, \ldots, q\}$. Then*

$$\Pr\left[\, K[S_k] = \text{Maj}\left(\{M_i[S_m] \oplus C_i[S_c]\}_{i=1}^{q}\right) \,\right] \geq 1 - e^{-q\epsilon^2/2} \,.$$

Chernoff bound

Finding linear approximations

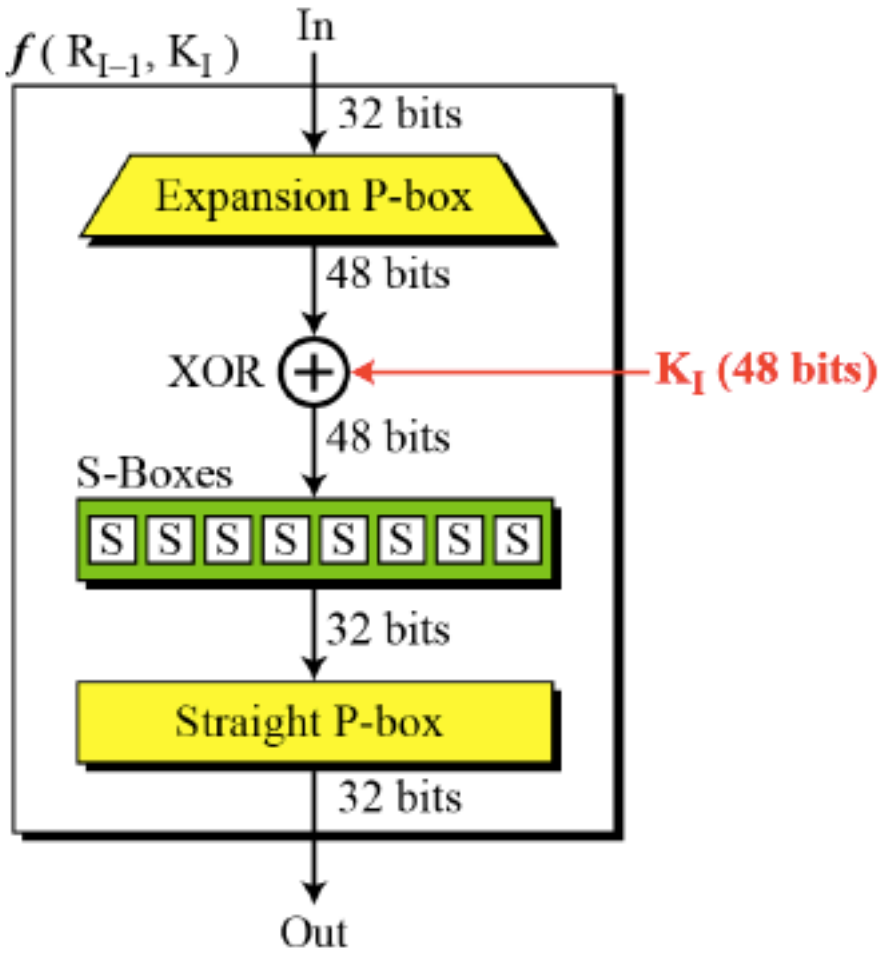
$$\Pr\left[\, M[S_m] \oplus C[S_c] = K[S_k] \,\right] = \frac{1}{2} + \epsilon$$

Need to determine a good linear approximation of cipher

For DES, only non-linearity is due to S-boxes

	S[5]															
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13

S[5] : $(x_0, x_1, x_2, x_3, x_4, x_5) \rightarrow (y_0, y_1, y_2, y_3)$
$(1, 0, 1, 0, 0, 0)$: row 2, column 4, $S[5](1, 0, 1, 0, 0, 0) = 2 = (0, 0, 1, 0)$



Can we find linear approximation of S-Box?

Finding linear approximations

$$\Pr\left[\, X[S_x] \oplus \mathrm{Sbox}(X)[S_y] = 0 \,\right] = \frac{1}{2} + \epsilon$$

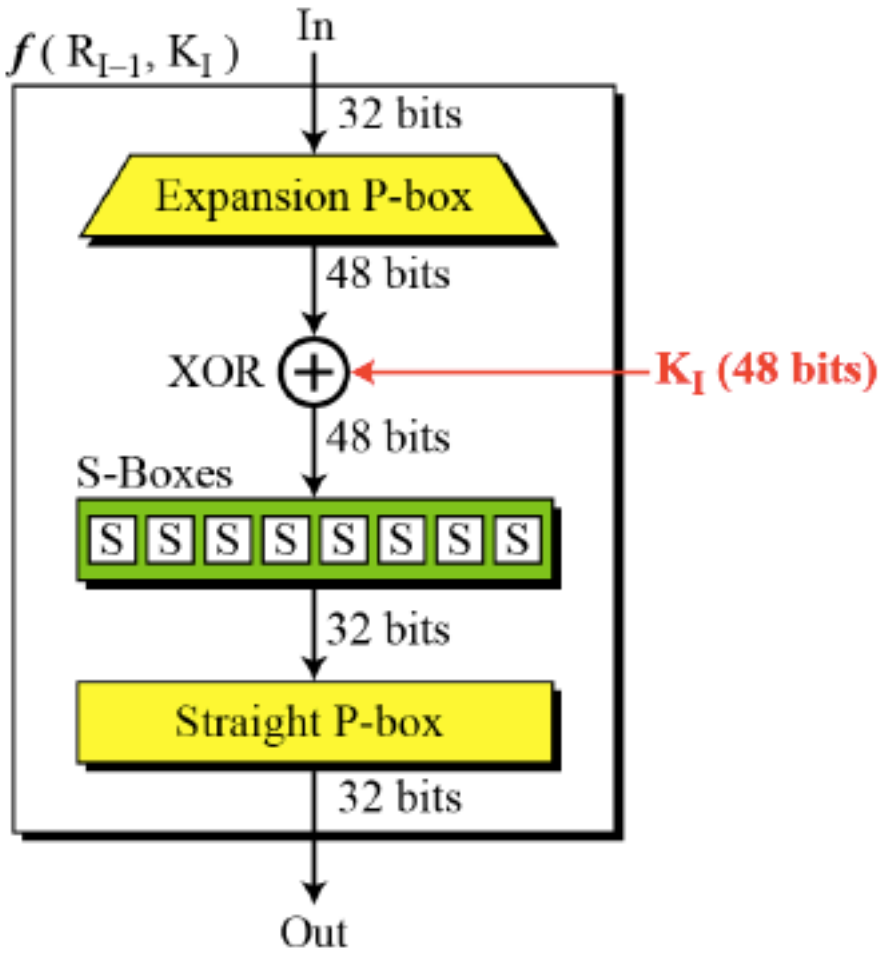
Can we find linear approximation of an SBox?

Number of possible S_x , S_y ? 64 x 16

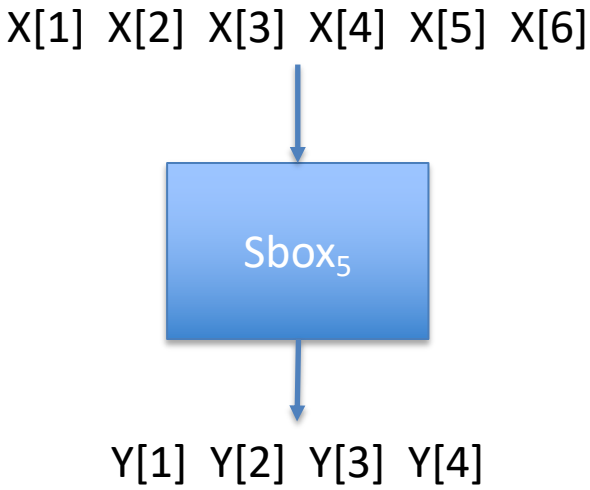
Directly compute bias of each possible S_x , S_y

# of X for which $X[S_x] \oplus \mathrm{Sbox}(X)[S_y] = 0$

Subtract 32 to get bias



S_x \ S_y	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	4	-2	2	-2	2	-4	0	4	0	2	-2	2	-2	0	-4
3	0	-2	6	-2	-2	4	-4	0	0	-2	6	-2	-2	4	-4
4	2	-2	0	0	2	-2	0	0	2	2	4	-4	-2	-2	0
5	2	2	-4	0	10	-6	-4	0	2	-10	0	4	-2	2	4
6	-2	-4	-6	-2	-4	2	0	0	-2	0	-2	-6	-8	2	0
7	2	0	2	-2	8	6	0	-4	6	0	-6	-2	0	-6	-4
8	0	2	6	0	0	-2	-6	-2	2	4	-12	2	6	-4	4
9	-4	6	-2	0	-4	-6	-6	6	-2	0	-4	2	-6	-8	-4
10	4	0	0	-2	-6	2	2	2	2	-2	2	4	-4	-4	0
11	4	4	4	6	2	-2	-2	-2	-2	-2	2	0	-8	-4	0
12	2	0	-2	0	2	4	10	-2	4	-2	-8	-2	4	-6	-4
13	6	0	2	0	-2	4	-10	-2	0	-2	4	-2	8	-6	0
14	-2	-2	0	-2	4	0	2	-2	0	4	2	-4	6	-2	-4
15	-2	-2	8	6	4	0	2	2	4	8	-2	8	-6	2	0
16	2	-2	0	0	-2	-6	-8	0	-2	-2	-4	0	2	10	-20
17	2	-2	0	4	2	-2	-4	4	2	2	0	-8	-6	2	4
18	-2	0	-2	2	-4	-2	-8	4	6	4	6	-2	4	-6	0
19	-6	0	2	-2	4	2	0	4	-6	4	2	-6	4	-2	0
20	4	-4	0	0	0	0	0	-4	-4	4	4	0	4	-4	0
21	4	0	-4	-4	4	-8	-8	0	0	-4	4	8	4	0	4
22	0	6	6	2	-2	4	0	4	0	6	2	2	2	0	0
23	4	-6	-2	6	-2	-4	4	4	-4	-6	2	-2	2	0	4
24	6	0	2	4	-10	-4	2	2	0	-2	0	2	4	-2	-4
25	2	4	-6	0	-2	4	-2	6	8	6	4	10	0	2	-4
26	2	2	-8	-2	4	0	2	-2	0	4	2	0	-2	-2	0
27	2	6	-4	-6	0	0	2	6	8	0	-2	-4	-6	-2	0
28	0	-2	2	4	0	-6	2	-2	6	-4	0	2	-2	0	0
29	4	-2	6	-8	0	-2	2	10	-2	-8	-8	2	2	0	4
30	-4	-8	0	-2	-2	-2	2	-2	2	-2	6	4	4	4	0
31	-4	8	-8	2	-6	-6	-2	-2	2	-2	-2	-8	0	0	-4
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



$$X[5] \oplus \mathrm{Sbox}(X)[1,2,3,4] = 0$$

Holds for for 12 / 64 of inputs X

Bias of 12-32 = -20

Finding linear approximations

$$\Pr\left[\, X[S_x] \oplus \mathrm{Sbox}(X)[S_y] = 0 \,\right] = \frac{1}{2} + \epsilon$$

Can we find linear approximation of S-Box?

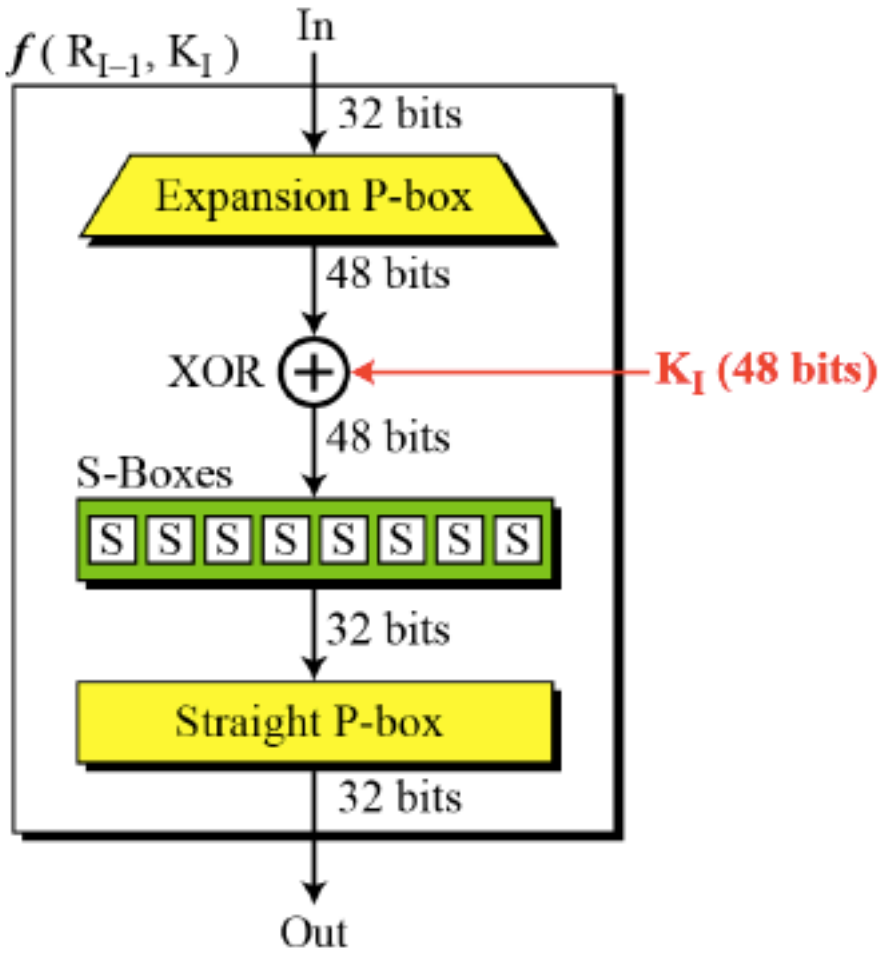
Number of possible S_x , S_y ? 64 x 16

Directly compute bias of each possible S_0 , S_1

# of X for which $X[S_x] \oplus \mathrm{Sbox}(X)[S_y] = 0$ minus 32

Use Sbox approximation within full round function to get ***one-round linear approx***:

$$\Pr\left[\, X[15] \oplus F_K(X)[7, 18, 24, 29] = K[22] \,\right] = \frac{1}{2} - \frac{20}{64}$$

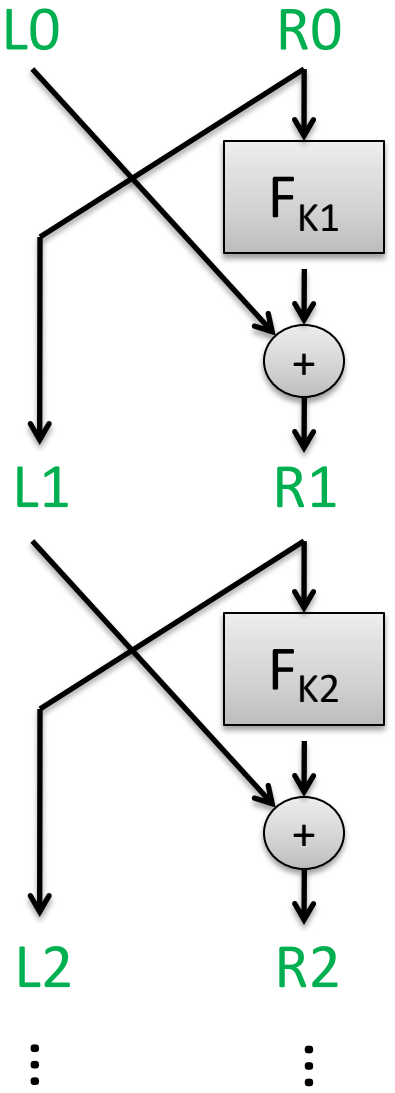


Finding linear approximations

Combine linear approximations for different rounds together, to get function of known values and key bits

Additive combination of different linear approximations

“Piling up” heuristic: calculate bias across multiple rounds by heuristically treating round estimators as independent



Recovering many key bits

Use a linear approximation of 15 rounds. Let Y_{15} = L15 || R15

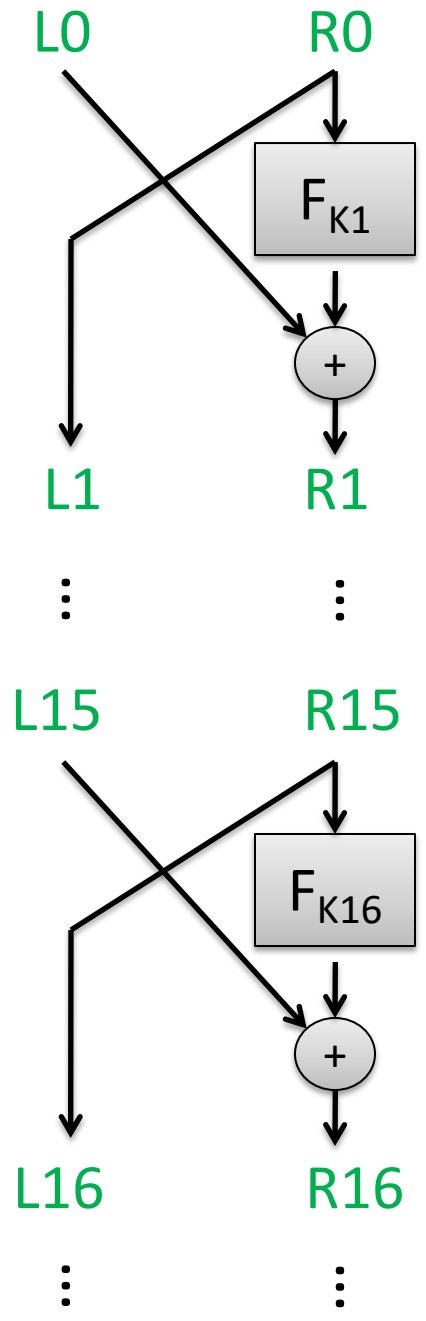
$$\Pr\left[\, M[S_m] \oplus Y_{15}[S_c] = K[S_k] \,\right] = \frac{1}{2} + \epsilon$$

Partially decrypt last round – subset of K16 key bits used

For correct K16 bits, $M[S_m] \oplus Y_{15}[S_c]$ will equal zero for either ½+ε or ½-ε fraction of M, Y_{15} pairs

For incorrect K16 bits, $M[S_m] \oplus Y_{15}[S_c]$ will equal zero for closer to ½ of M, Y_{15} pairs

Gives maximum-likelihood estimator of subset of K16 bits
Similar approach for 1st round, w/ estimator of last r-1 rounds
Combine this approach with brute-force of remaining key bits



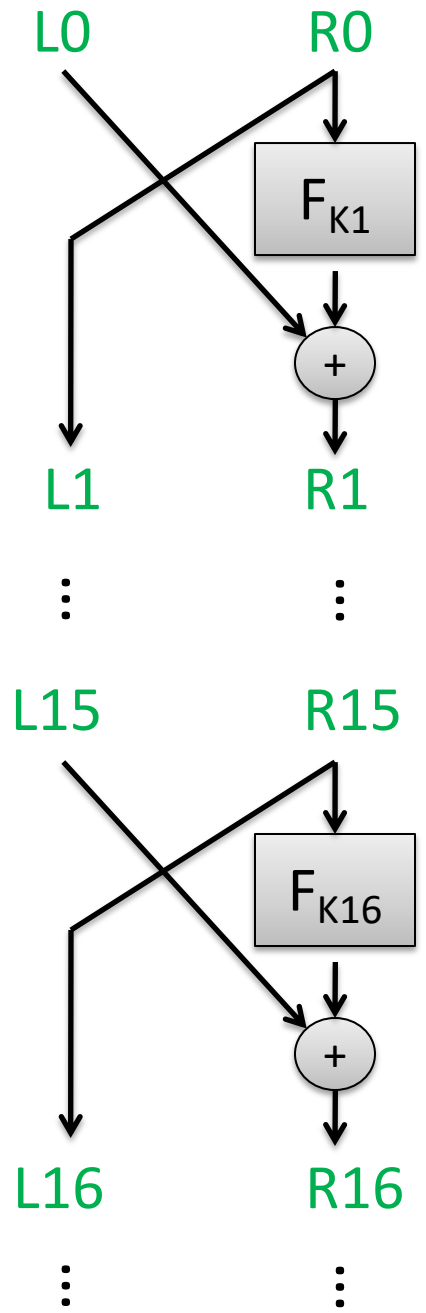
Recovering many key bits

Breaks 8-round DES with:

- 2^{21} known plaintext/ciphertext pairs
- 40 seconds of 1992 computation time (66 Mhz!)

Breaks 16-round DES with:

- 2^{43} known plaintext/ciphertext pairs
- ~2^{41} DES computations in expectation



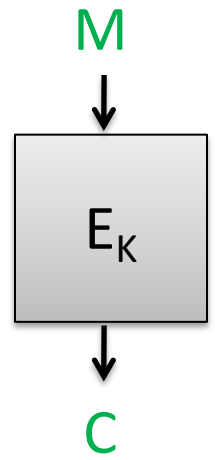
Differential cryptanalysis

[Biham, Shamir 1990]

Look for weaknesses based on pairs of inputs

$$\Delta_C = E_K(M + \Delta_m) + E_K(M)$$

Find Δ_m s.t. Δ_C holds with high probability (over choice of M)



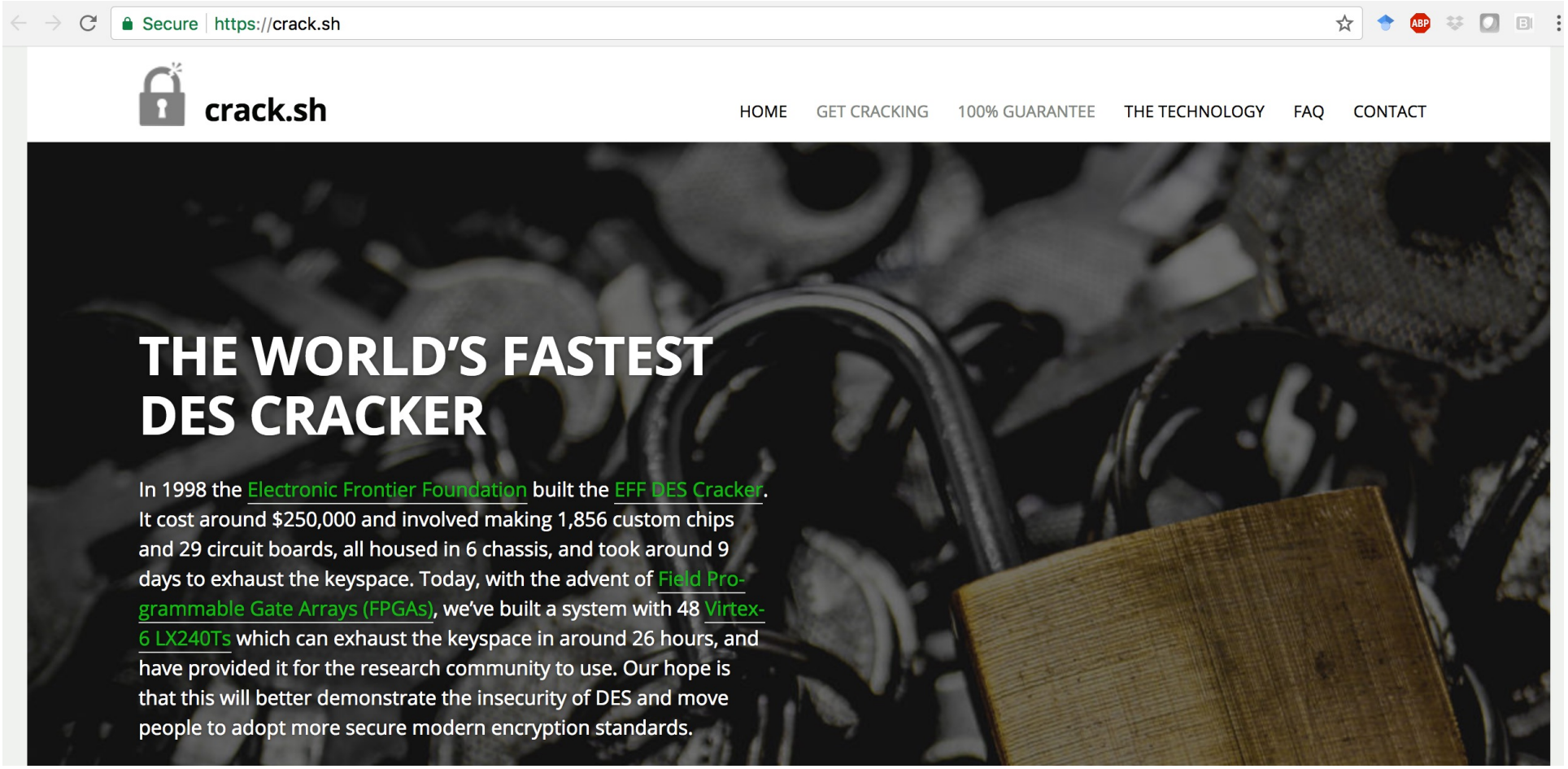
- Analyze S-boxes to find differentials Δ_x, Δ_y such that
  $\Delta_y = Sbox(X + \Delta_x) + Sbox(X)$
  holds with high probability over random choice of X
- Piece together compatible differentials across rounds, providing *differential characteristics*

Leads to ***chosen plaintext attack***

Best attacks against DES

Attack	Attack type	Complexity	Year
Biham, Shamir	Chosen plaintexts, recovers key (differential cryptanalysis)	2^{47} plaintext, ciphertext pairs	1992
Matsui	Known plaintexts, recovers key (linear cryptanalysis)	2^{43} plaintext, ciphertext pairs	1993
DESCHALL	Brute-force attack	$2^{56/4}$ DES computations 41 days	1997
EFF Deepcrack	Brute-force attack	~4.5 days	1998
Deepcrack + DESCHALL	Brute-force attack	22 hours	1999

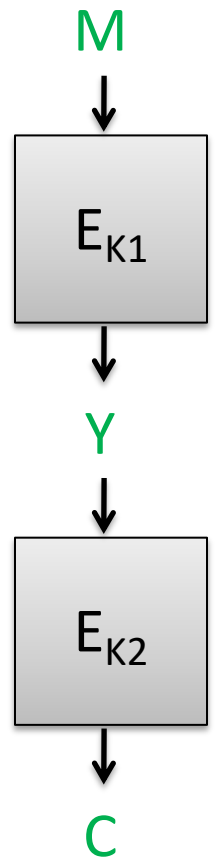
Best practical attack is brute-force key recovery



2DES and 3DES

Increase key length by composing cipher with separate keys

Security not as high as one would expect, due to
meet-in-the-middle attack

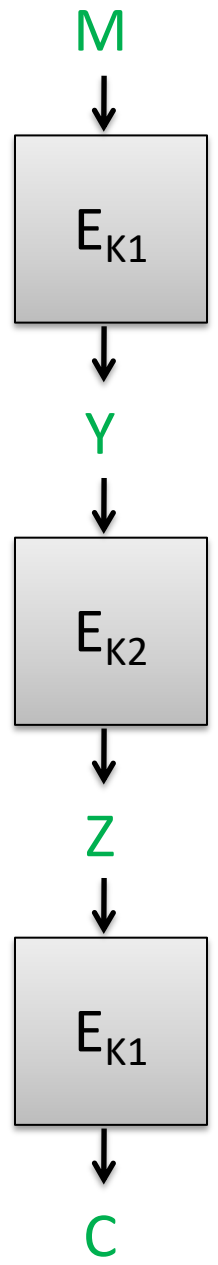


2DES and 3DES

Increase key length by composing cipher with separate keys

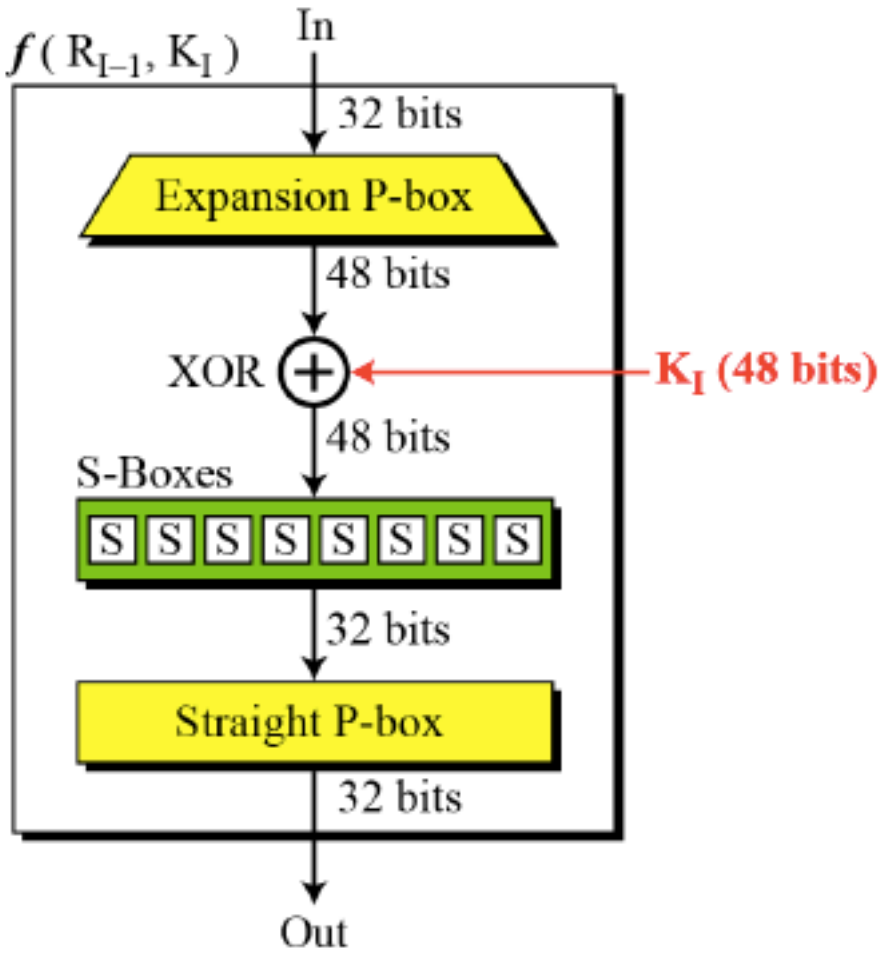
Security not as high as one would expect, due to
meet-in-the-middle attack

Triple-DES provides 112-bits of security



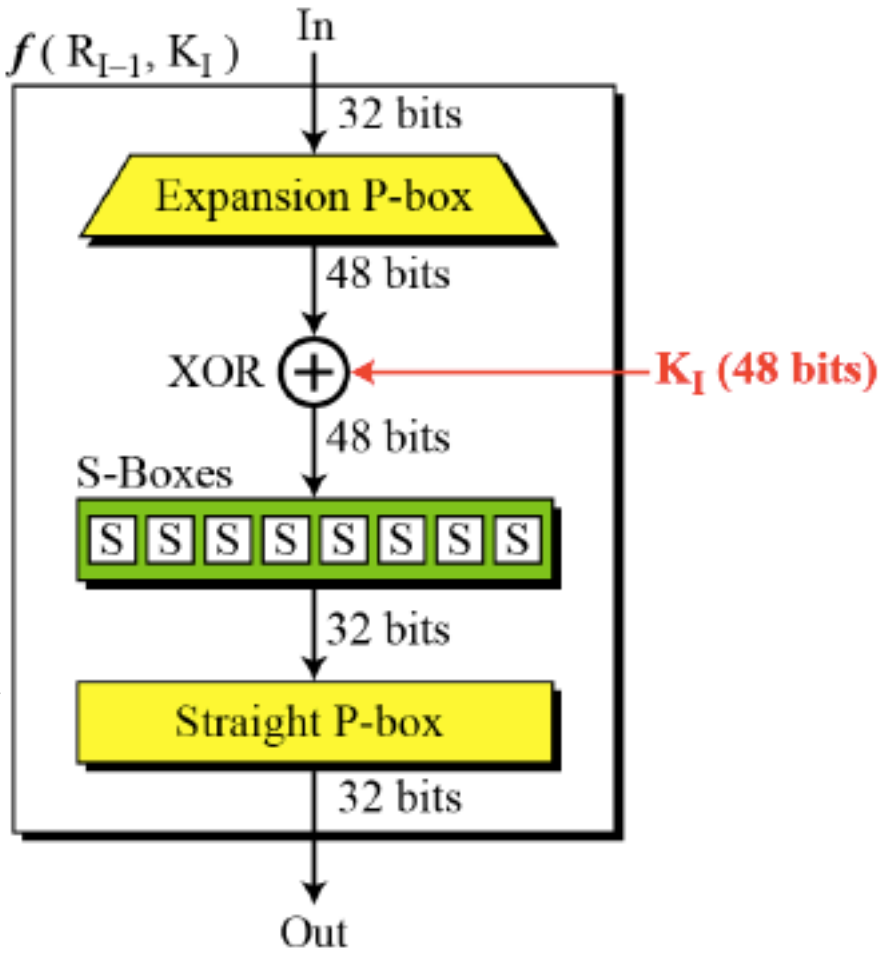
Differential cryptanalysis

- Differential cryptanalysis: [Biham, Shamir 1992]
  - Analyze S-boxes to find differentials Δ_x, Δ_y such that
    - $\Delta_y = Sbox(X + \Delta_x) + Sbox(X)$
  - holds with high probability over random choice of X
  - Piece together such differentials across rounds, providing a differential trail
  - Use to reduce key search space significantly, given lots of encryptions of chosen messages



Linear cryptanalysis

- Linear cryptanalysis: [Matsui 1993]
  - Approximate S-box behavior by linear functions, e.g.,:
    - X1 + X2 + X6 = Y1 + Y2 + Y4
  - S-boxes exhibit some biases, meaning some linear functions are satisfied with higher probability (over uniform inputs)
  - Can carefully "pile up" linear approximations across multiple DES rounds
  - Use linear approximations to efficiently narrow down search for key, given lots of encryptions of uniform messages



http://www.cs.bc.edu/~straubin/crypto2017/heys.pdf

Next up

- Building & analyzing block ciphers
  - Feistel networks and DES
  - AES cipher
  - Cryptanalysis of block ciphers