Nasa hack: AnonSec attempts to crash $222m drone, releases secret flight videos and employee data

By Mary-Ann Russon
February 1, 2016 13:05 GMT

Logistics

- HW0 up on github
  - "Due" next Tuesday
  - Not graded
  - First real homework HW1 will be assigned Tuesday. Homework teams should be pairs or if needed singletons
- Thursday Feb 11: in-class studio session to get over hurdles on homework. Bring your questions, Rahul will be there to help groups.
- Piazza "homeworks" are going to count towards participation credit
  - For this Thursday: Finish descriptions from last semester
  - Come to class prepared to discuss distillation of critical crypto problems seen across all the posts

Today in Cryptography (5830)

One time pad review
Block ciphers
Ideal ciphers / functions
Computational OTP-like security
Pseudorandom functions & permutations
DES, AES

One-time pads

Fix some message length n bits

Key generation: output random n-bit string K

$$E(K,M) = M \oplus K \qquad\qquad D(K,C) = C \oplus K$$

Shannon's security notion (1949)

Def. A symmetric encryption scheme is perfectly secure if for all messages M,M' and ciphertexts C

$$\Pr[\ E(K,M) = C\ ] \quad = \quad \Pr[\ E(K,M') = C\ ]$$

where probabilities are over choice of K

Thm. OTP is perfectly secure

For any C and M of length L bits

$$\Pr[\ K \oplus M = C\ ] \quad = \quad 1 / 2^n$$

$$\Pr[\ K \oplus M = C\ ] \quad = \quad \Pr[\ K \oplus M' = C\ ]$$

Shannon's security notion (1949)

Def. A symmetric encryption scheme is perfectly secure if for all messages M,M' and ciphertexts C

$$\Pr[\,E(K,M) = C\,] \quad = \quad \Pr[\,E(K,M') = C\,]$$

where probabilities are over choice of K

Thm. OTP is perfectly secure

Thm. Perfectly secure encryption requires $|K| \geq |M|$

Block ciphers

Family of ~~permutations~~ functions, one ~~permutation~~ function for each key

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

$E(K,X) = Y$ $D(K,Y) = X$

Ignore this for now. In fact, assume we can't invert.

Ideal block cipher

- Imagine Alice and Bob share an idealized version of block cipher: random look-up table function that is (magically) efficient to evaluate (unit cost)
- We can encrypt using this idealized cipher. How?

Adversary knows the cipher

- Adversary can compute by making unit cost evaluations of cipher as well.

- Does scheme achieve perfect security? No. Why?
- What is best attack you can think of?

Brute-force attacks

- Get one or more input-output examples of encryption
- Try decrypting with each possible key
  - Each decryption = 1 unit cost
- Expected run time: 2^{k-1} (for key size k)
- Worst-case run time: 2^k

Computational security

We aim for encryption schemes secure only against computationally limited adversaries.

In fact, our little scheme provably leaks nothing about plaintexts for computationally bound adversaries.

Referred to as *semantic security*

Random IV instead of counter

- The counter reveals how many messages encrypted. Let's get rid of it
- Pick a random value each time we encrypt

- Security before: counters always unique
- Now no guarantee of uniqueness. How many messages can we encrypt before security fails?

The birthday bound

Throw q balls in 2^n bins uniformly. What is probability that no bin has two balls?

$$\Pr[\mathrm{Coll}] \leq q^2 / 2^n$$

Implication:

Security holds up to q a bit less than $2^{n/2}$

Variable length messages

We have been assuming each message is n bits

How can we do variable length messages?

Still relying on idealization

Idea: instantiate keyed random function with computationally efficient block cipher

Design block cipher to “behave like” random function

- **Random oracle**: function is random for every key
- **Pseudorandom function (PRF)**: function indistinguishable from a random function for a uniform, secret key

Block ciphers

- Family of permutations, one permutation for each key

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

$$E(K,X) = Y \qquad\qquad D(K,Y) = X$$

Random permutations

- Like random functions but respect permutivity
- Keyed random permutation (ideal cipher):
  - Each key selects random permutation
- Pseudorandom permutation
  - Indistinguishable from random permutation under uniform, secret key

How do we build efficient block ciphers that are close to ideal?

Data encryption standard (DES)

Originally called Lucifer

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

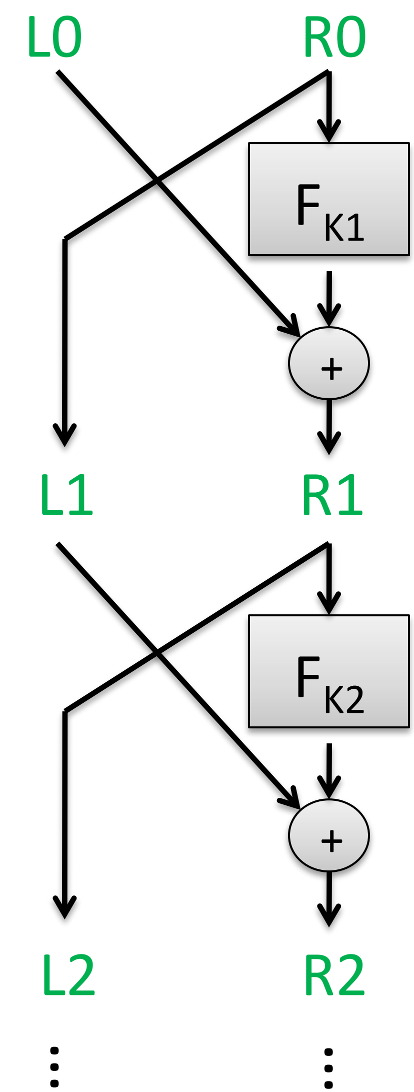
$n = 64$
$k = 56$

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

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

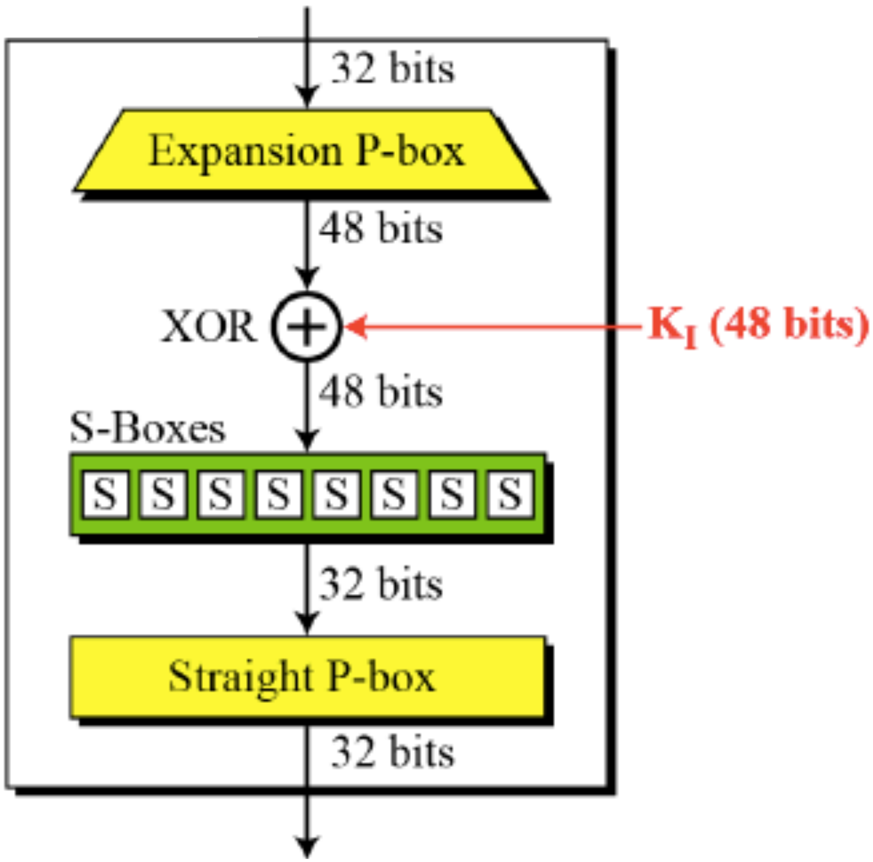
Repeat Feistel round 16 times

Each round applies function F using separate round key



Round functions

X



F(Ki,X)

Best attacks against DES

Attack	Attack type	Complexity	Year
Biham, Shamir	Chosen plaintexts, recovers key	2^{47} plaintext, ciphertext pairs	1992
DESCHALL	Unknown plaintext, recovers key	$2^{56/4}$ DES computations 41 days	1997
EFF Deepcrack	Unknown plaintext, recovers key	~4.5 days	1998
Deepcrack + DESCHALL	Unknown plaintext, recovers key	22 hours	1999

- DES is still used in some places
- 3DES (use DES 3 times in a row with more keys) expands keyspace and still used widely in practice

Advanced Encryption Standard (AES)

Rijndael (Rijmen and Daemen)

n = 128
k = 128, 192, 256

Number of keys for k=128:
340,282,366,920,938,463,463,374,607,431,768,211,456

Substitution-permutation design.
For k=128 uses 10 rounds of:

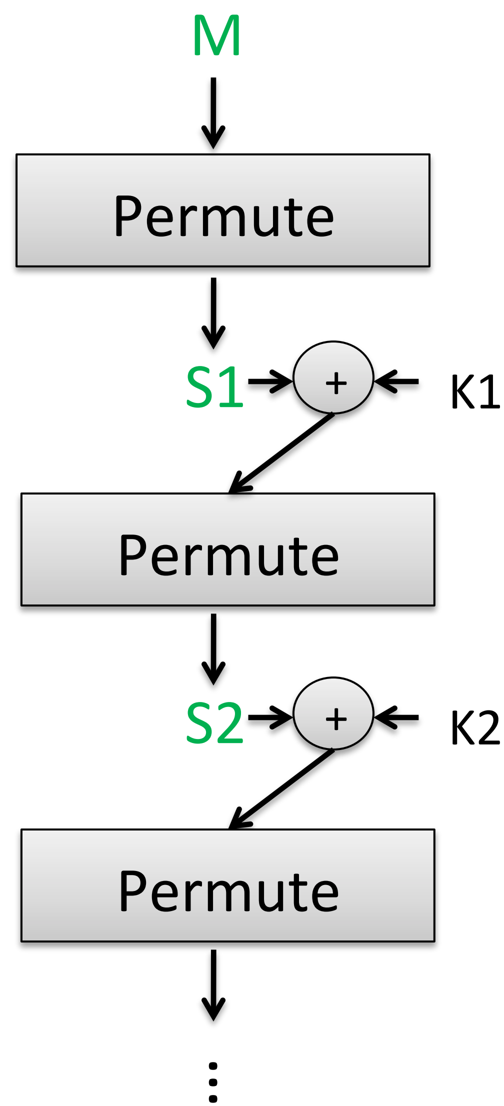
1) Permute:

SubBytes (non-linear S-boxes)

ShiftRows + MixCols (invertible linear transform)

2) XOR in a round key derived from K

(Actually last round skips MixCols)



Best attacks against AES

Attack	Attack type	Complexity	Year
Bogdanov, Khovratovich, Rechberger	chosen ciphertext, recovers key	$2^{126.1}$ time + some data overheads	2011

- Brute force requires time 2^{128}
- Approximately factor 4 speedup

Summary

- Use block ciphers to build stream cipher
  - Secure block cipher means we get OTP-like security against computionally bounded adversaries
  - Brute-force attacks
- A good blockcipher is family of permutations, one permutation per key
  - Ideally behaves like random permutation for all keys
  - PRF and PRP security hold for uniform, secret keys
- We have good blockciphers: 3DES, AES
  - Design of good blockciphers is topic of a whole other class