CSC 580
Cryptography and Computer Security

Block Ciphers, DES, and AES

February 6, 2018

Overview

Today:

- HW2 solutions review
- Block ciphers, DES, and AES
  - Textbook sections 4.1, 4.2, 4.4 plus AES handout

To do before Thursday:

- Study for quiz over HW2 material
- Read textbook sections 7.1-7.6

DES and AES for CSC 580

We will focus on *how to use block ciphers securely*.

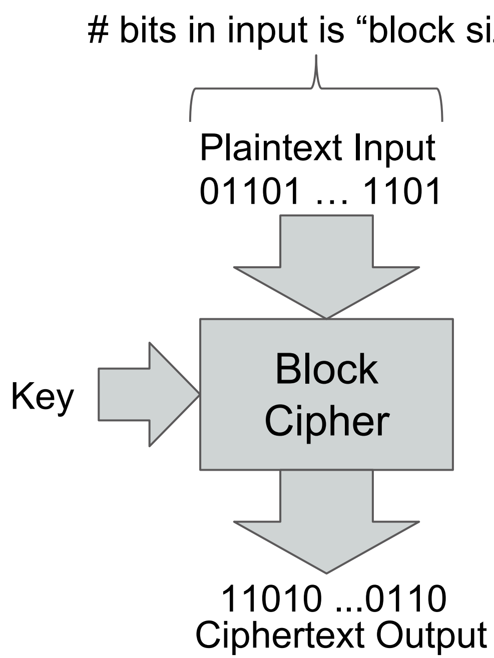
Important to understand big picture issues:

- What parameters describe block ciphers?
- What properties does a good block cipher have?
- How do parameters affect those properties?
- How did parameters change historically as capabilities grew?

How block ciphers work (internals):

- We will view as a “black box” with certain I/O behavior
- Internals are interesting, but avoided here to save time

Block Ciphers - General



Properties of a block cipher

- Must supply a full block of input bits in order to evaluate
- Typical block sizes: 64 or 128 bits
- Every execution of the block cipher is independent of others (stream ciphers typically carry forward state)
  - However - block ciphers ***used*** in ways that carry state forward - more on modes later
- A good block cipher can be modeled as a pseudo-random permutation
  - Appears random to adversary, so no cryptanalysis - stuck doing brute force

This fits nicely with our "view symmetric ciphers as secure black boxes" approach.

Random Block Ciphers

The ideal (and impractical) case

A general encryption function replaces plaintexts with ciphertexts and must be reversible.

Picking a random function is like picking a random permutation of the message space.

- Permutation because 1-to-1
- Number of permutations: $|\mathcal{P}|!$

For a b-bit block cipher, $|\mathcal{P}| = 2^b$

- Number of permutations is $(2^b)!$

For b=3, there are 8! = 40,320 permutations

For b=8, there are $256! \approx 10^{507} \approx 2^{1684}$

To specify one of 256! permutations you a need $\log_2(256!) \approx 1684$ bit long key

3-bit block example:

Input		Output
(0) 000	→	011 (3)
(1) 001	→	101 (5)
(2) 010	→	111 (7)
(3) 011	→	000 (0)
(4) 100	→	110 (6)
(5) 101	→	010 (2)
(6) 110	→	001 (1)
(7) 111	→	100 (4)

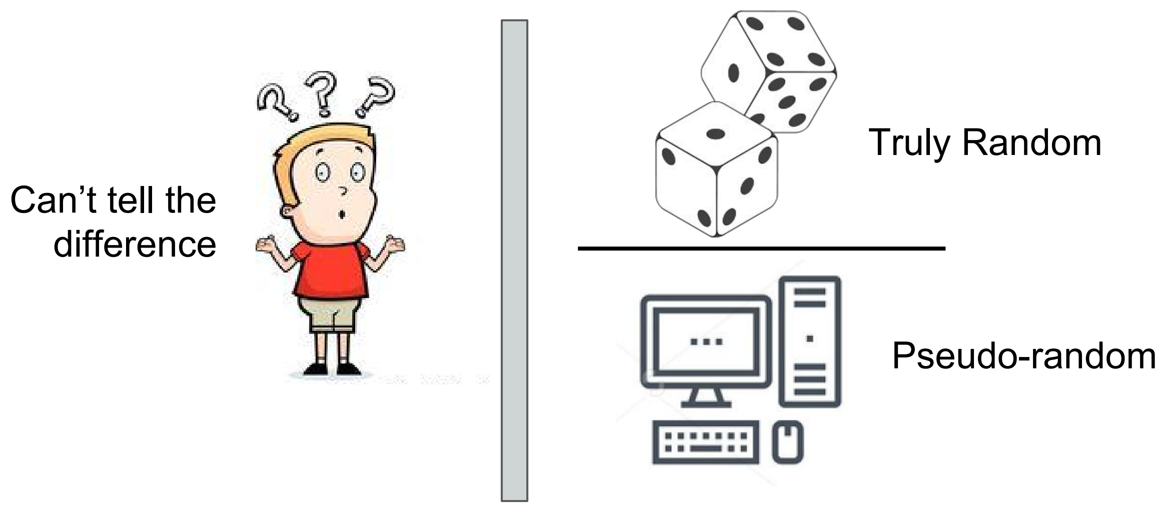
Pseudorandom vs Random

How big a key do you need to specify a permutation of 64-bit values?

Answer: $\log_2(2^{64})! \approx 10^{21}$ bits - the key alone is 1000 million TB

Consequence: Can't pick a random permutation

- Picking from a limited domain of permutations: **pseudorandom permutation**
- Uses a small random seed (key!) to compute random-looking data



We can formalize this into a rigorous definition - and we will later!

Some Pre-DES Historical Notes

Claude Shannon

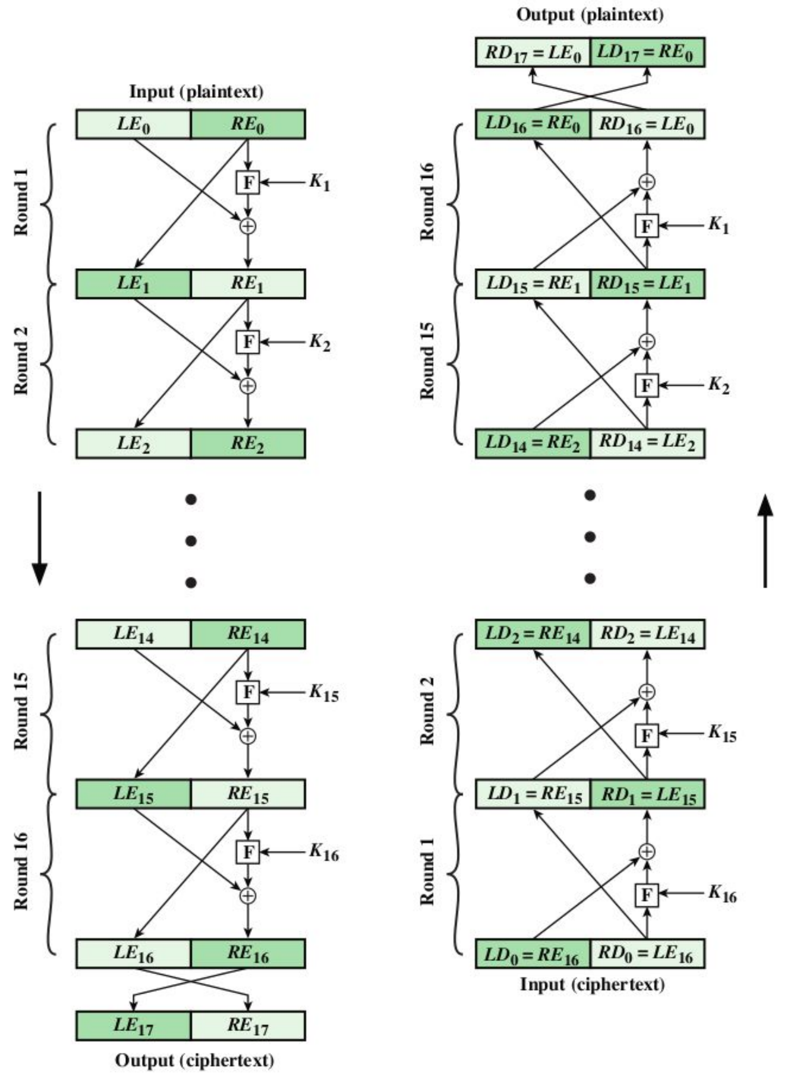
- Worked for the National Defense Research Committee during WWII
- Moved to Bell Labs in 1945
- Wrote classified paper “A Mathematical Theory of Cryptography” in 1945
  - Proved security of one-time pad and the necessity of certain OTP properties for perfect security (*<u>any cipher with perfect security</u>* will be similar to a OTP).
  - Declassified version “Communication Theory of Secrecy Systems” - 1949
  - Defined “unicity distance” - basically how much ciphertext is needed for brute force attacker to recognize plaintext unambiguously
- Very influential paper “A Mathematical Theory of Communication” in 1948
  - Established the field of Information Theory
  - Formalized notions such as “entropy” and measuring information in bits

Important civilian post-WWII, pre-1970 cryptography work done at IBM

- Key players: Horst Feistel, Don Coppersmith, Alan Hoffman, Alan Konheim

Feistel Network

Based on Figure 4.3 from the textbook (corrected!)



If “F” is a pseudorandom function indexed by key K_1, transforms right-side data into a pseudo-one-time-pad for left-side.

In one round, left side is modified (substitution) then sides swapped (permutation).

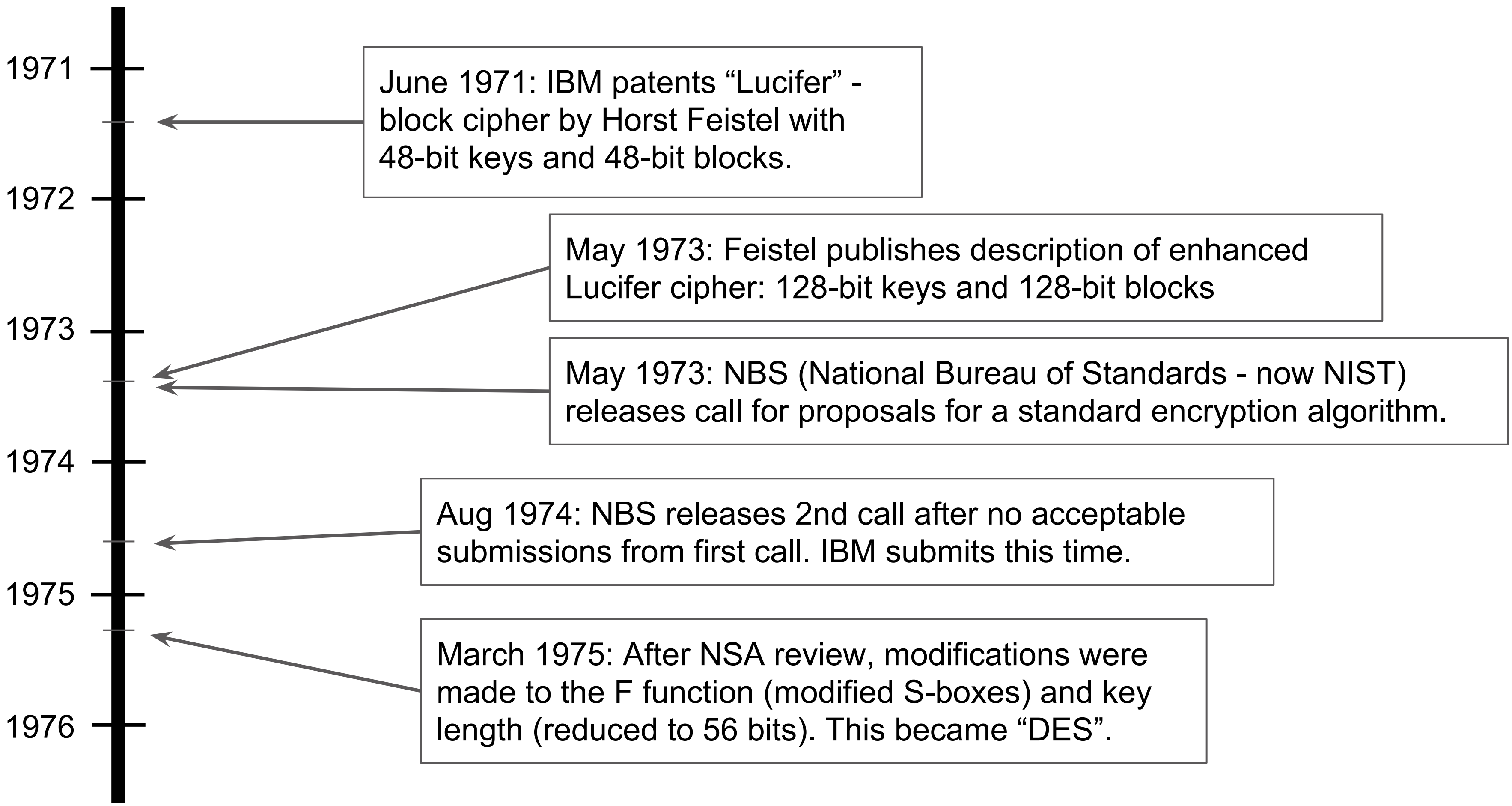
- One round clearly not secure since half just carried forward
- Since one side affects the other, transformation “spreads out” (diffusion) over multiple rounds

Concepts to work through from diagram

- Requirements on F (injective? no!)
- Decryption relation to encryption

DES - The Data Encryption Standard

History



DES - The Data Encryption Standard
Basic Parameters, Controversy, and Context

DES parameters:
- Block size: 64 bits
- Key size: 56 bits (8 7-bit characters, with parity bits)
- Feistel network with 16 rounds
- Feistel “F function” based on 4-bit substitutions (S-boxes)

Controversy - why were changes made?
- Warning sign: DES never cleared for secret data - only “confidential”
- Changed S-boxes - do they contain a backdoor for NSA?
  - 1994: Revealed that changes protected against differential cryptanalysis - discovered in “open literature” in 1990
  - To this day: Only really practical attacks on DES are brute force
- Reduced key length - why?
  - 56-bits is “secure enough” against non-nation-state adversaries
  - But the NSA had (and still has!) a big budget for big machines

DES - The Data Encryption Standard

A peek inside

DES F function:

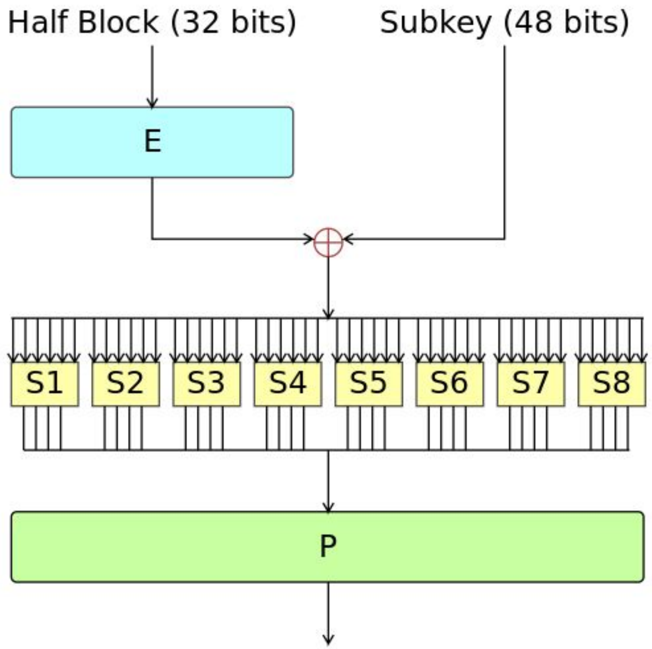


Diagram from Wikipedia

"E" is an expansion function - one input bit can affect two S-box inputs

S-boxes are pseudo-random substitutions (with certain properties)

P is a bit-by-bit permutation

DES - The Data Encryption Standard
A peek inside

What does P look like?

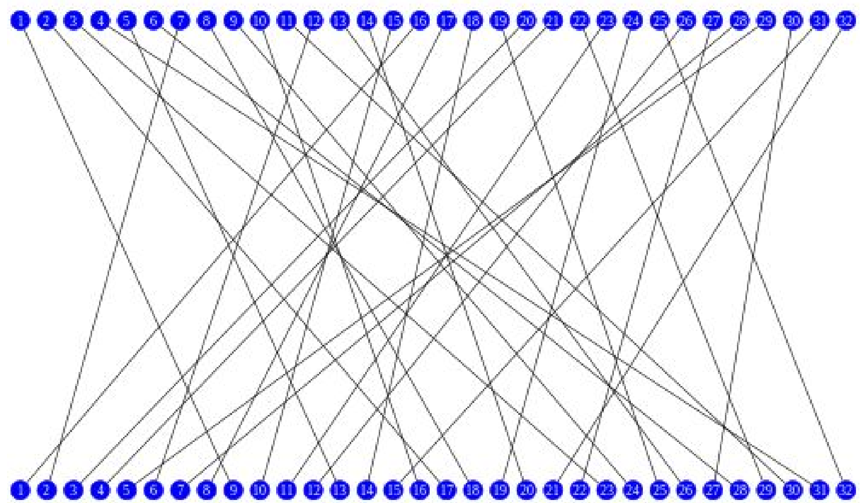


Diagram from Wikipedia

Moves individual bits around.

Think about doing this in software vs hardware - how efficient?

DES also includes a similar bit-by-bit “initial permutation” (and final)

Bottom line: DES is **not** easy/efficient to implement in software.

DES - The Data Encryption Standard

Efficiency and Security

From papers published 1984-1986:

- Proposed (paper) hardware estimated about 1 million encr/sec
- Actual (built) hardware ran around 300,000 enc/sec
- Best software implementation: about 2,500 enc/sec (Vax 11/780)

Question: How long on average for a brute force attack?

Part a: Using one custom HW chip
Part b: Using 1,000,000 custom HW chips
Part c: Using software

Modern technology

- General purpose hardware: approx 10,000,000 enc/sec/core
  - HW: How long to brute force on one core? On 512 cores?
- Special-purpose HW - COPACOBANA ($10,000): 48 billion enc/sec
  - How long now?

DES - The Data Encryption Standard
Bottom Line

Single DES can no longer be considered secure

Triple-DES (3-DES) extends keyspace to 56*3 = 168 bits
- Big enough to be secure against brute force
- Inefficient (times 3!) in software
- Still has a 64-bit block size (bad for certain applications)

Conclusions:
- Good to understand history/evolution of cryptography
- Good introduction to block cipher concepts
- But don't use DES now...

Next: What key parameters need improvement in a replacement?

What Parameters are Important?

Key size: Can brute force a 56-bit key in a matter of days now

Algorithm design: DES is inefficient in software

Block size:

- "Collision attacks" follow "birthday problem" probabilities
  - With just 23 people, 50% chance that two have the same birthday
  - Roughly square-root of "universe size" (sqrt(365) = 19.1…)
- Applies to some applications of block ciphers
  - "universe" is number of possible ciphertext outputs
  - sqrt(2^{64}) = 2^{32} - requirement for both time and space (memory)
  - Trivial by today's standards

What values would be good today?

Key Size
Is 128 bits enough?

2004 Estimate: $100k machine breaks 56-bit DES key in 6 hours

What about a 128-bit key?
$100k machine takes >$10^{18}$ years [the earth is <10^{10} years old]

What if we spent $100,000,000,000?
Would take >10^{12} years

What about Moore's law saying that in 20 years machines will be about 16,000 times faster?
Would take >10^{8} years

OK, what about in 40 years (machines 100 million times faster)?
Would still take >30,000 years

Do you really think Moore's law will last this long?

Block Size

Is 128 bits enough?

Birthday attack:

- Requires sqrt(2^{128}) = 2^{64} time and space
- Space is 2^{64} 128-bit entries, for a total of $16*2^{64} = 2^{68}$ bytes
- One terabyte is 2^{40} bytes → requires 256 million terabytes
- At $25/TB that would cost around $6.4 billion (plus power, …)

Seems pretty safe...

AES Selection Process

1993-1995: Clipper Chip fiasco

1997: Request for proposals for new standard block cipher

- Must use 128-bit block
- Must support 128-bit, 192-bit, and 256-bit keys
- Selection process through open evaluation

1999: 15 good submissions narrowed to 5 finalists

2000: Winner selected

- Winner was an algorithm named Rijndael (limited to 128-bit blocks)
- Invented/submitted by Vincent Rijmen and Joan Daemen (Belgians)

Important points:

- Very open, public process
- No secret modifications
- Not rushed

→ More trust!

AES - Some Final Points

In 20 years, no practical cryptanalytic attacks discovered

Approved for protecting classified information

- 128 bit keys for SECRET
- 192 or 256 bit keys for TOP SECRET
- Note: implementation must be approved

Efficiency

- Works on byte/word units: Efficient in software!
- Widespread standard → special fast CPU instructions now
  - Intel AES-NI instructions: over 10 gigabits/sec on a single core!
  - OpenSSL demo...
- Still simple enough for special-purpose hardware
  - 30+ Gbps possible