CS 1657 Privacy in the Electronic Society

William Garrison
bill@cs.pitt.edu
6311 Sennott Square
https://bill-computer.science/1657

03: Symmetric cryptography

Today's topics: Crypto basics and uses

Security is relative

Basic cryptographic primitives

- Symmetric cryptography
  - Modes of operation
- Hashing
- Public-key cryptography
  - Hybrid crypto

Security modeling: The process of using these primitives

What is cryptography?

Cryptography

kryptos: hidden, secret

grafís: writing

Informally, cryptography is the study of methods for encoding and decoding secret messages

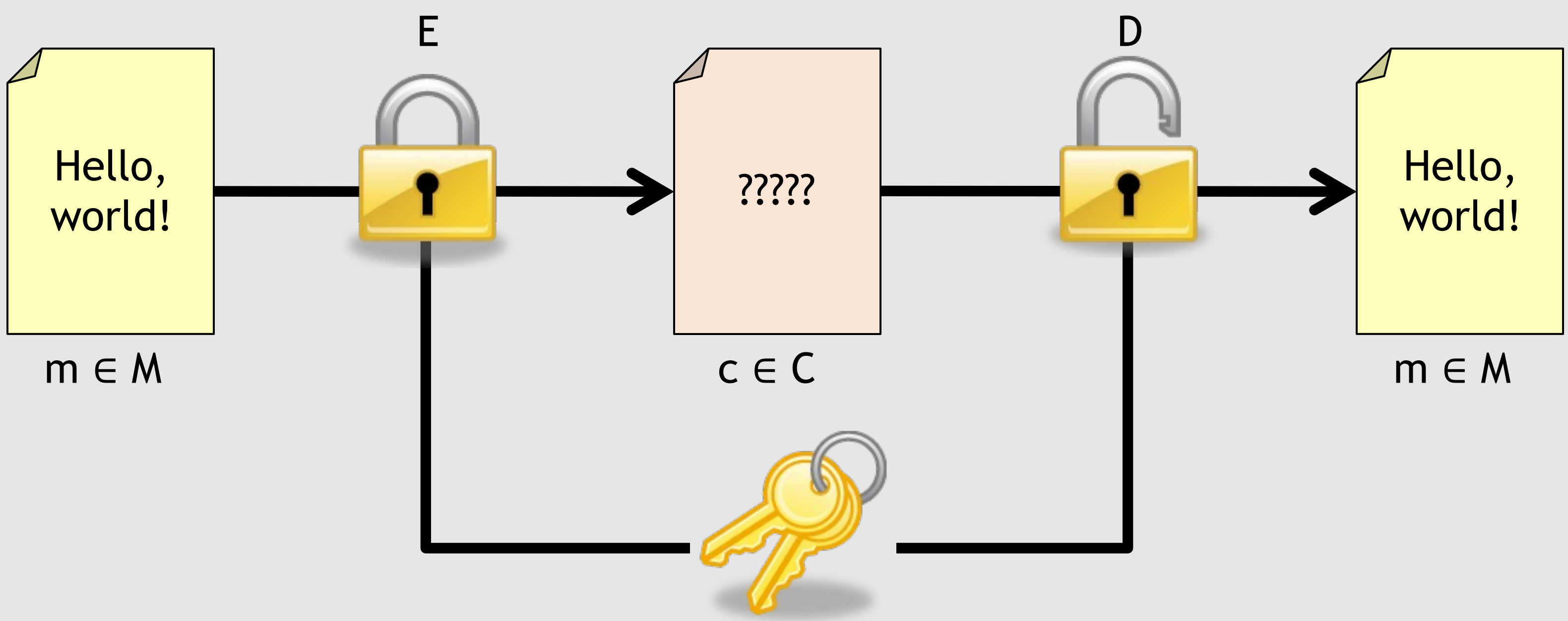
---

Plaintext → Encryption → Ciphertext → Decryption → Plaintext

Cryptography, more formally

A cryptosystem can be represented as the 5-tuple (E, D, M, C, K)

- M is a message space
- K is a key space
- E : M × K → C is an encryption function (we'll sometimes write $E_k(m)$ to mean E(m, k))
- C is a ciphertext space
- D : C × K → M is a decryption function



Brief history of cryptography

Steganography (hiding messages) predates crypto (to antiquity) and is still used in some contexts

- 440 BCE Herodotus describes “scalp messages”
- Secret inks, knots in yarn, writing under postage stamp…
- Conceal data in lowest bits, reorder elements of a set, mimic functions…

Ancient ciphers are closer to the ideas of modern cryptography

- Scytale, rail-fence (7th century BCE transposition)
- Atbash (6th century BCE substitution)
- Caesar cipher (1st century BCE substitution, probably much earlier)

Unfortunately, classical cryptography is insecure today

These (and other) algorithms were designed in a time when people were break ciphers by hand, and few people used cryptography

What does this mean?

- Relatively small keyspaces were OK
- Relying on an algorithm remaining secret was a less dubious assumption

Today: Widespread mechanical and digital computing device and pervasive communication invalidate the above assumptions

- Machines can systematically try all possible keys
- Security by obscurity breaks down
- Security is relative!

How can we define and measure the security of a cryptosystem?

We evaluate a cryptosystem's security relative to the capabilities of an attacker

In a ciphertext-only attack, the adversary is assumed to have stored some amount of ciphertext that can be analyzed offline to attempt to break the cipher.

In a known-plaintext attack, the adversary is assumed to have collected some number of (plaintext, ciphertext) pairs that can be used to guide their attempt at breaking the cipher.

In a chosen-plaintext attack, the adversary has access to the cryptographic algorithm and may encrypt anything that they choose. The resulting (plaintext, ciphertext) pairs are then used to guide attempts at breaking the cipher.

So what cryptosystems are secure?

One-time pad: Invented in the early 20th century as a telegraph cipher at AT&T

How does it work?

- Choose a key that is as long as the plaintext that you wish to encrypt
- $E(p) = p \oplus k$, $D(c) = c \oplus k$

Example (using modular addition/subtraction instead of XOR):

- Plaintext: `BUYTENSHARES`
- Key: `HIENVWKNUQCF`
- Ciphertext: `ICCGZJCUUHGX`

Note: A single ciphertext can recover any plaintext string!

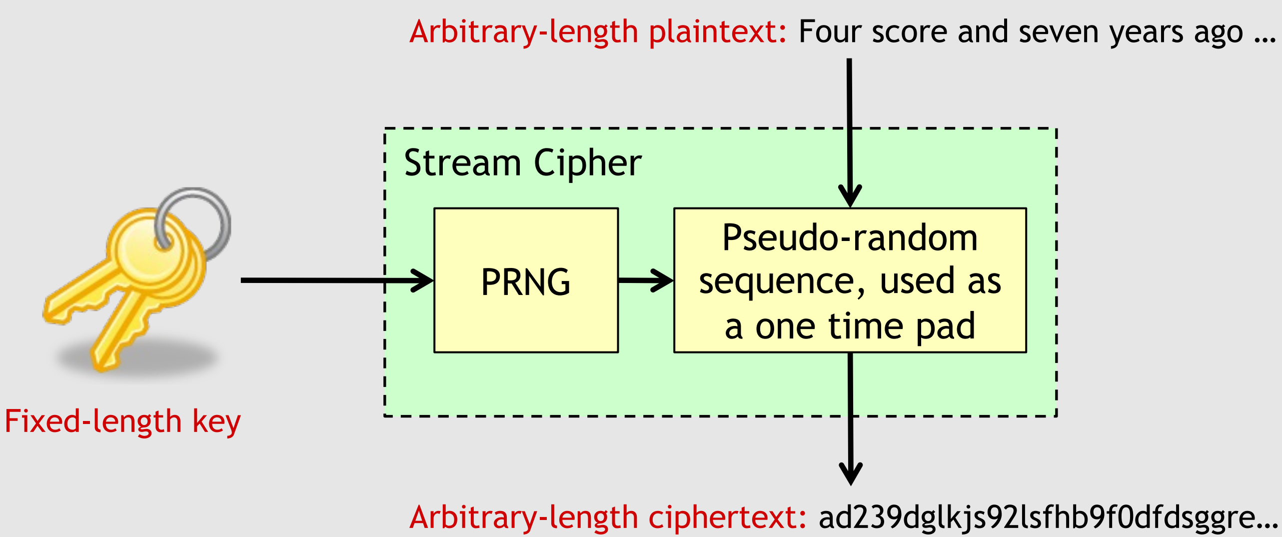
- With key `LGAEMVEKRKAH`, the text `TICKLEGEORGE` can be recovered from the above ciphertext, and `GCJOZSYODDGE` yields `CATSAREGREAT`!

Secure against all previously mentioned attacks! … Any issues?

Modern cryptosystems use a fixed-length key to encrypt variable-length data

Ideally, we want something as secure as 1TP, but this is hard

Stream ciphers use PRNG to “stretch” a key

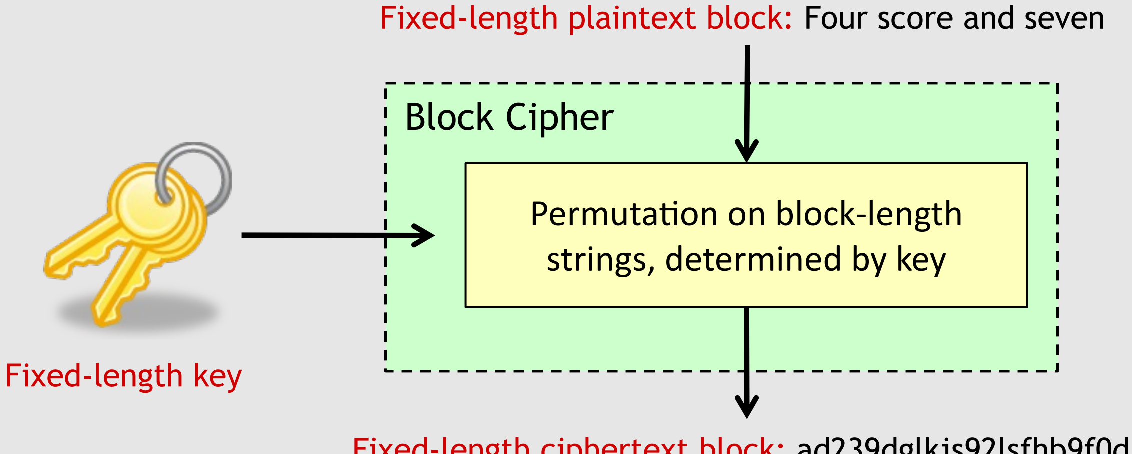


Secrecy depends on strong PRNG

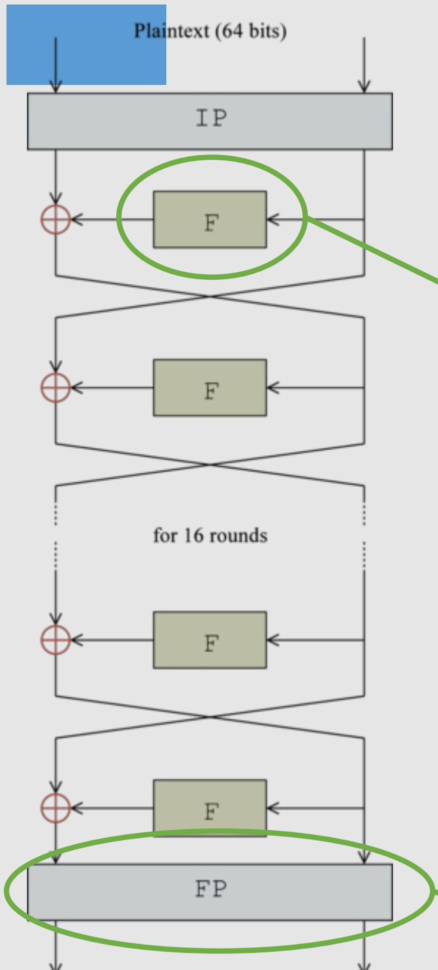
Block ciphers are much more common

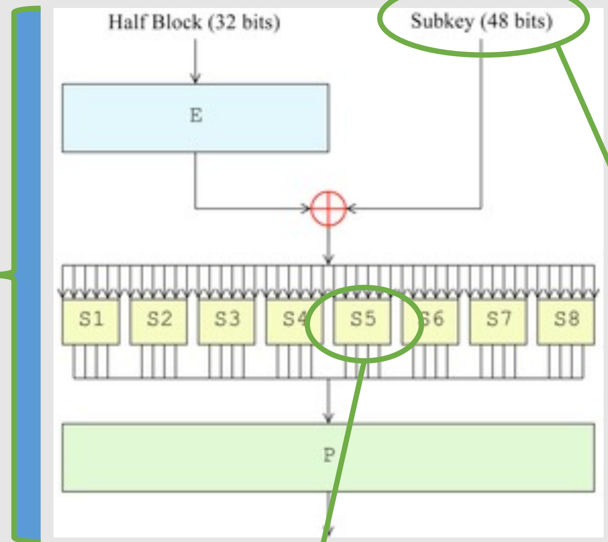
Block ciphers operate on fixed-length blocks of plaintext

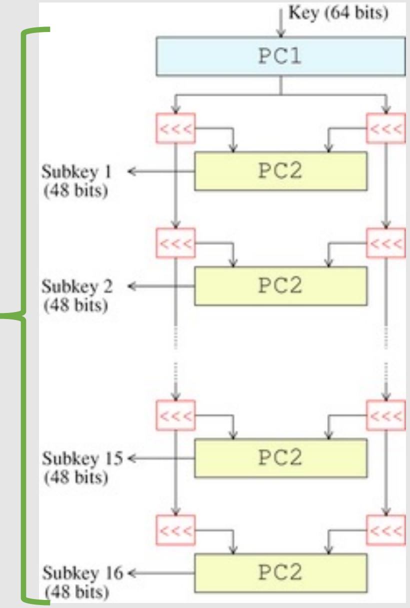
- Typically, 40, 56, 64, 80, 128, 192, or 256 bits
- Usually multiple rounds of simpler functions
  - Bit shuffling, non-linear substitution, linear mixing
  - Confusion and diffusion



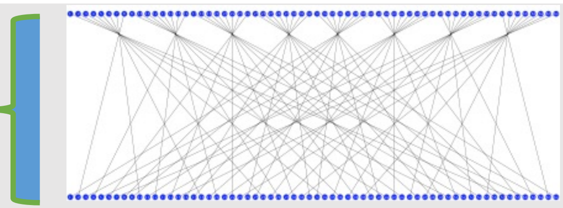
Example: DES







S_5		Middle 4 bits of input															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Outer bits	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

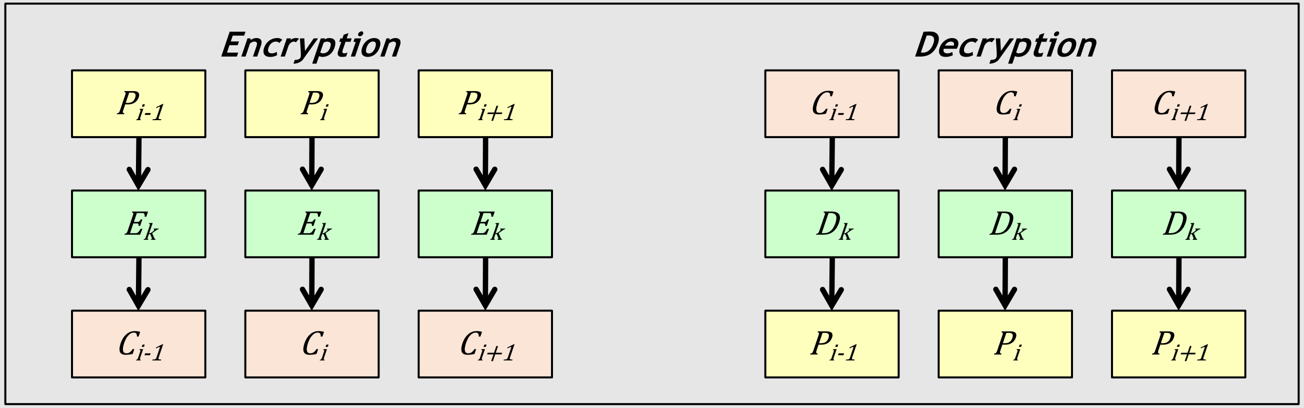


Images from Wikipedia

But, what if we want to encrypt more than one block?

Block ciphers have many modes of operation

Most obvious way: Electronic Codebook mode (ECB)



Benefit: Errors in ciphertext do not propagate past a single block

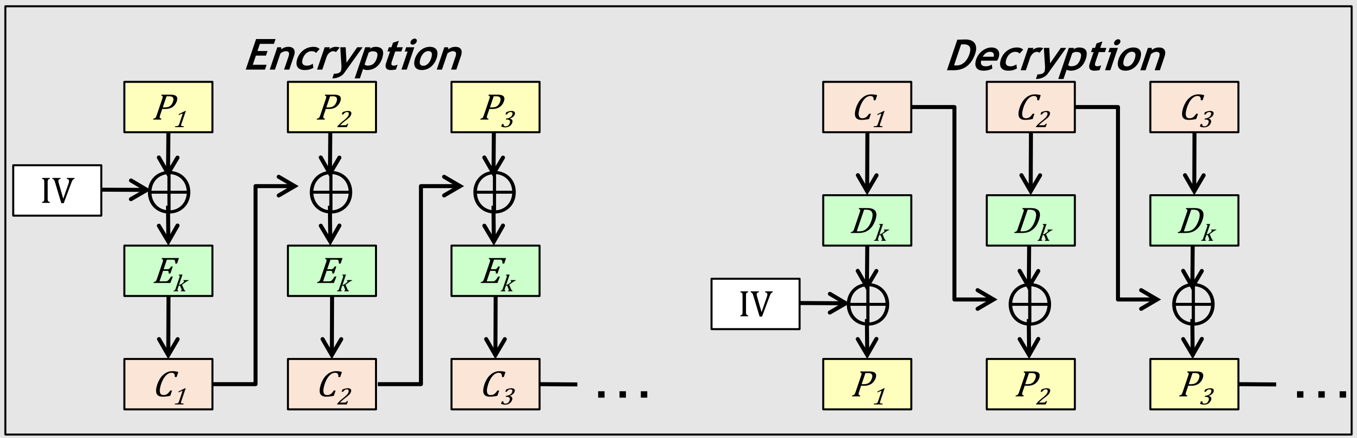
Dangers?

- Known plaintext/ciphertext pairings, block replay/reorder

Cipher Block Chaining (CBC) addresses some of these issues

In CBC mode, each plaintext block is XORed with the previous ciphertext block prior to encryption

- $C_i = \mathrm{E}_k(P_i \oplus C_{i-1}),\ P_i = C_{i-1} \oplus \mathrm{D}_k(C_i)$



Need to encrypt a random block to get things started

- This initialization vector needs to be random, but not secret (Why?)

CBC eliminates block replay attacks

- Each ciphertext block depends on previous block

There are several other useful block modes, as well

Cipher Feedback Mode (CFB)

- Can use shift-register approach to process a smaller block at once
  - Why is this useful?

Output Feedback Mode (OFB)

- Keystream can be generated offline, then XOR with data
  - Why is this useful?

Counter mode (CTR)

- Keystream can be generated offline and in parallel
  - Why is this useful?

So what should I use?

- Consider synchronization, offline generation, cost to decrypt single block, etc.
- Never ECB, use up-to-date best practices

Conclusions

Security is relative to attacker capabilities

Classical ciphers are not secure today, because modern attacker models are (and need to be) more powerful

Modern symmetric cryptosystems are efficient and utilize confusion and diffusion

Block cipher modes of operation allow block ciphers to be more flexible