Lecture 6: Encryption in Practice

6.5610 - MIT
Spring 2023
Henry Corrigan-Gibbs

Plan

- Recap: AES-GCM
- Three constructions
    - AES
    - DES
    - [Stretch break]
    - Chacha20
- MITM attack on 2DES

Logistics

* Pset 1 due Friday 5pm
* Friday recitation is all about project. IMPORTANT!
* Next week: meet TA/instructor re: project (ideally OH)
* Think on groups for project.

Encryption used everywhere?

- Phone
- Computer
- Satellite
- ⋮

} Essentially any net com today

- NIST publishes standards for encryption
  ↳ Widely used, reqd often for selling to govts
  → Not only standards org (IETF also, ISO, ...)

- NIST Ciphers

		Block size	
- DES (1975)	$\lvert\mathcal{K}\rvert = 2^{56}$	$n = 64$	✗
- 3DES	$\lvert\mathcal{K}\rvert = 2^{168}$	$n = 64$	Still used!
- AES (1998)	$\lvert\mathcal{K}\rvert \in \{2^{128}, 2^{192}, 2^{256}\}$	$n = 128$	Still used!

N.B. DES key size is far too small.

in U.S. SECRET: AES-128/192/256 } Algs are public!
TOP SECRET: AES-192/256

Are PRPs, but some common primitives (eg. ChaCha20) are not

- While you will never need to implement these primitives yourself, worthwhile to understand design.

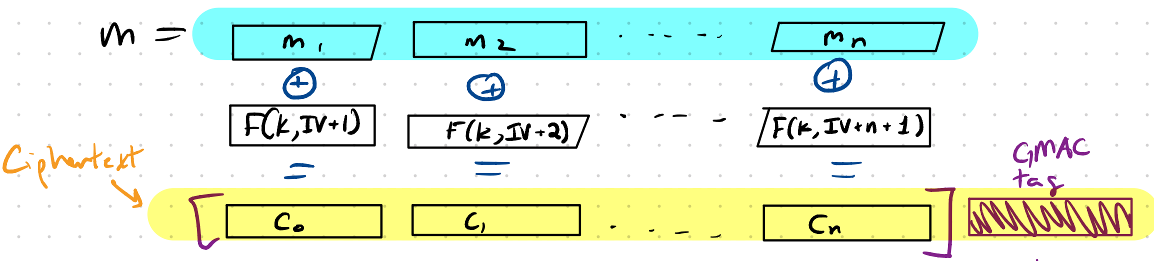
- Hash functions are coming up, but not today...

Recap: AES-GCM (Authenticated encryption)

CPA-secure encryption
Secure MAC
} = Authenticated encryption
⇒ CCA security

Encrypt then MAC.

Uses AES as PRF: $F: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ (n = 128)



$r \leftarrow F(k, \text{"}0000\ldots0\text{"})$

$tag \leftarrow F(k, IV) \oplus \sum_{i=1}^{n} c_i r^{n-i}$

(Also need to include msg length in hash but I'm omitting it for simplicity.)

→ Single pass over the message

→ Careful use of PRF lets use same key for enc & MAC

↑ NOT safe in general

Other notes on AES-GCM

- CPUs have HW support for AES (GBs per second)
  (AES-NI) 0.5 cycles/byte
  ↳ Essentially "for free" today.

- As we discussed, AES is PRP but used here as a PRF.
  ↳ why is that safe?! PRP ≠ PRF

"PRF Switching Lemma" (See Boneh-Shoup)

Let $P: \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ be a PRF

Then for any PRP adv $\mathcal{A}_{PRP}$, $\exists$ PRF adv $\mathcal{A}_{PRF}$ s.t.

$$|\mathcal{A}_{PRP} - \mathcal{A}_{PRF}| \leq \frac{q^2}{2^{n+1}}.$$

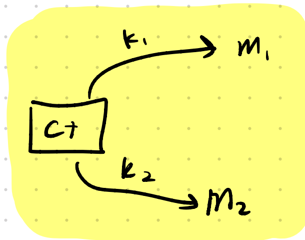
Intuition:

- Collisions in outputs is only diff b/w PRF & PRP.
- Until $q \approx 2^{n/2}$ will not expect to see collisions by Birthday paradox.
- After that, can distinguish!
  ↳ See "Sweet32 attack" 768 GB traffic on 3DES (64-bit block)

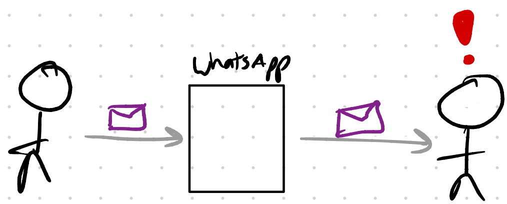
Properties that AES-GCM doesn't provide

- Nonce-reuse protection
  - ↳ Some modes of operation do (at some cost)
  - Reusing nonce reveals equality & nothing more

- Commitment
  - ↳ Can find (k_1, k_2, c)
  - s.t.



$$\left.\begin{array}{l} \mathrm{Dec}(k_1, c) = msg_1 \\ \mathrm{Dec}(k_2, c) = msg_2 \end{array}\right\} \text{Control} \sim 1/2 \text{ bits of each msg}$$

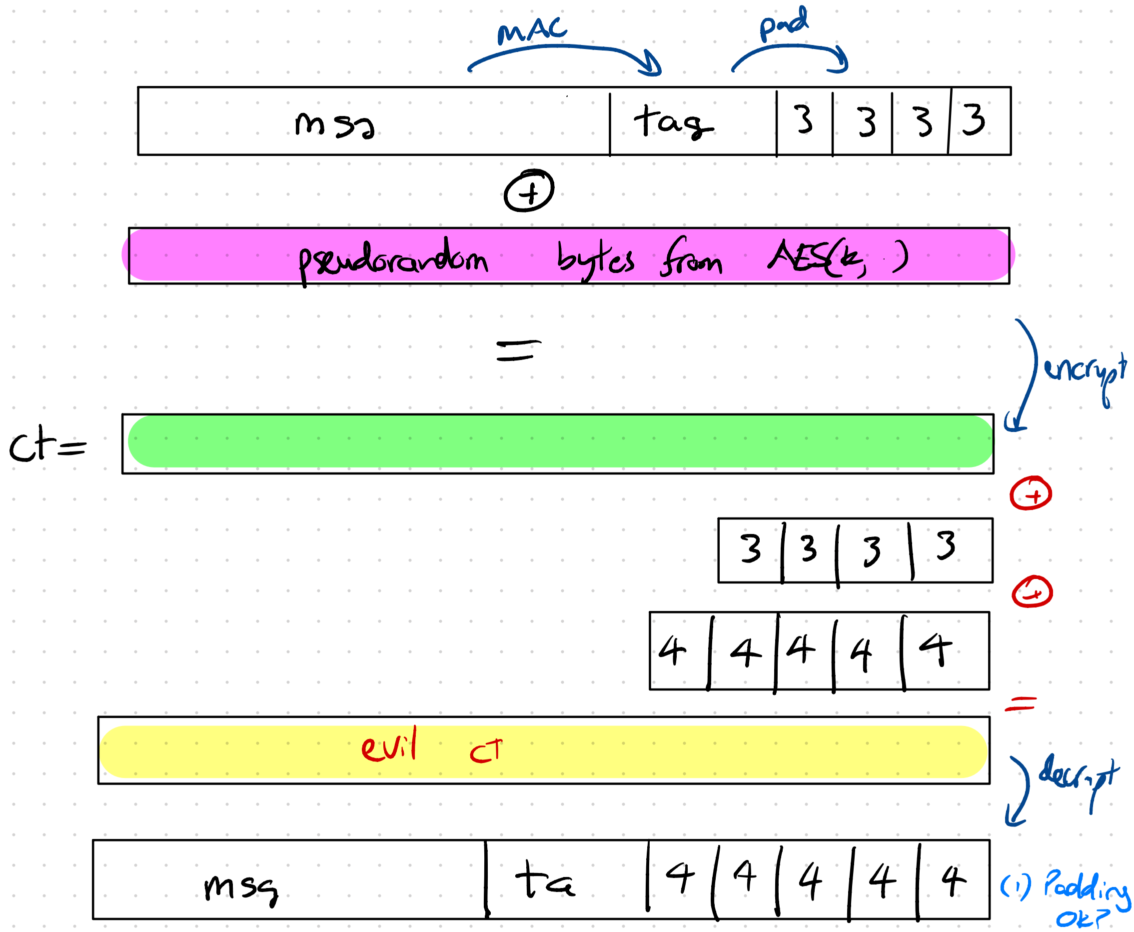


[Dodis, Grubbs, Ristenpart, Woodage '18]

Why MAC-then-encrypt is bad:

* Some enc schemes (CBC mode) require plaintext be multiple of block size, e.g. 16 bytes
  ↳ Convenient & sometimes necessary

* Pad msg with n indicating "truncate n+1 bytes"



If adv can learn whether padding is valid, learns one byte of msg! → Timing, error msg, etc.

Three constructions

AES - Substitution permutation

DES - Feistel network

Chacha20 - "Even-Mansour" (?) PRF

Some of crypto is based on "nice" assumptions

↳ "win win". E.g., factoring

↳ Nice things cost too much

PRF/PRP design is messier in some ways:

* Design to resist best known attacks
* Try to get others to break (NIST competitions)
* Patch when broken

→ Surprise: No serious break of 3DES (beyond obvious ones)

Difficult part (in some sense) isn't security, it's getting security with good performance on all h/w.

(example of 3rd grader)

Warning!

Do NOT attempt to build or implement a block cipher (or mode of operation yourself!)

- ↳ timing & cache attacks, ... cryptanalysis, etc.
- ↳ Takes many years of effort to gain confidence in design.

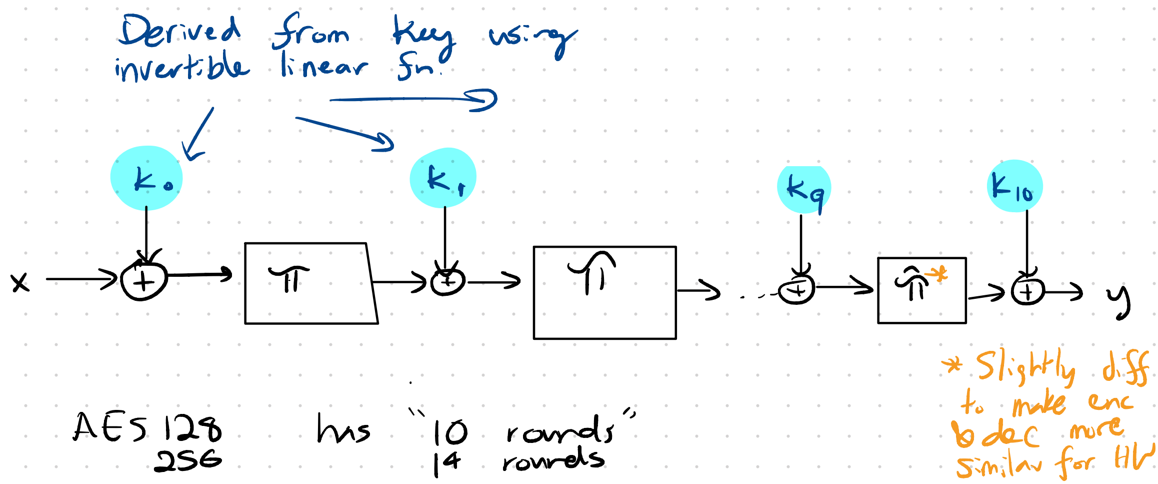
Design of AES (PRP/ block cipher)

AES is an "Iterated Even-Mansour cipher"

Uses invertible $\pi : \{0,1\}^{128} \to \{0,1\}^{128}$

* Very simple – substitution, linear ops, etc.

(subbytes, shiftrows, mixcolumns)



AES 128 has "10 rounds"
256 14 rounds

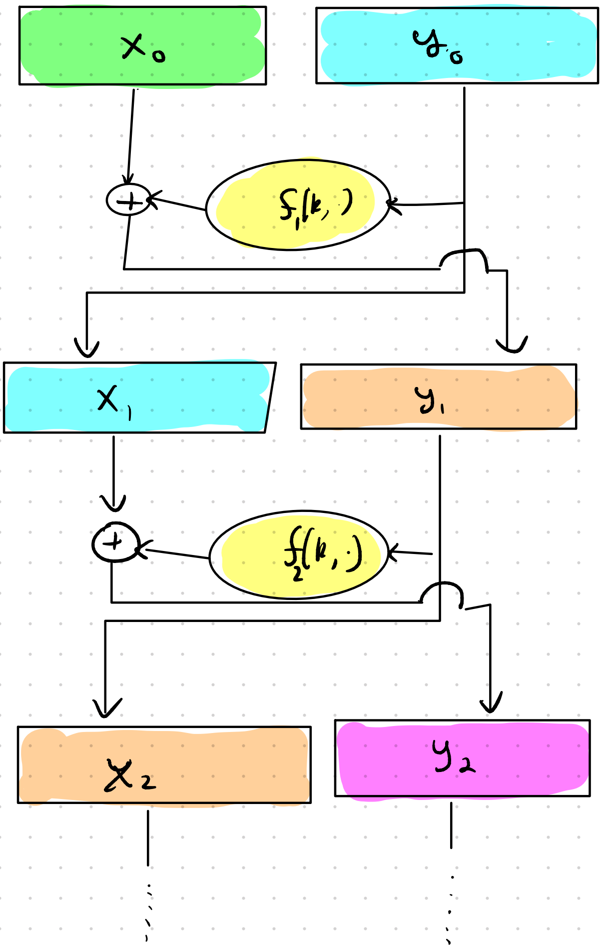
Security justification

* After two decades of cryptanalysis, no great attacks
* If we model π as a random perm ⇒ Prove security.

Stretch Break

DES Cipher (PRP / block cipher)

* Horst Feistel (MIT ugrad) → gov't → IBM
* Lucifer – precursor to DES, was Feistel net
* Also how you get PRF ⟹ PRP



Invertible?

$y_{i-1} = x_i$

$x_{i-1} = y_i \oplus f(x_i)$

Luby & Rackoff showed that if $f(k,\cdot)$ is a secure PRF ⟹ Feistel_f^3 is a secure PRP

[Not obvious]

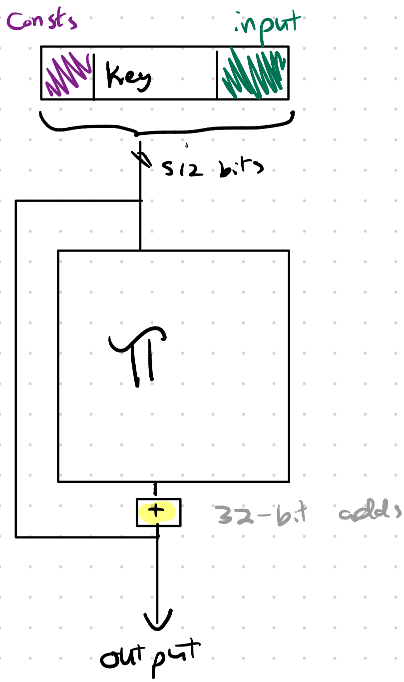
* In practice (e.g. DES) fn f is NOT a PRF
  → But LR analysis gives some justification for design.

* Fn f used in DES shares many features w/ AES round fn (substitution, permutation)

ChaCha20 (PRF)

- Essentially a PRF (used as "stream cipher")

  Key = 256 bits $F: \underbrace{\{0,1\}^{256}}_{\text{Key}} \times \{0,1\}^{128} \to \{0,1\}^{512}$



* The permutation Π performs 10 rounds of simple bit operations on 4×4 matrix of 32-bit words (add, rot, xor).
* Design rationale
* Used in CTR mode for CPA-secure encryption.

3DES

- DES 56-bit key too short.
- EFF DES Cracker : 1998 : \$250k of compute
  - ↳ Now \$20, takes a few days

$$3DES((k_1, k_2, k_3), m) := DES(k_3, DES^{-1}(k_2, DES(k_1, m)))$$

[Clever hack: $k_1 = k_2 = k_3 \Rightarrow 3DES = DES$]

* Keylen is 168 bits
* MITM takes $\approx 2^{112}$ time.

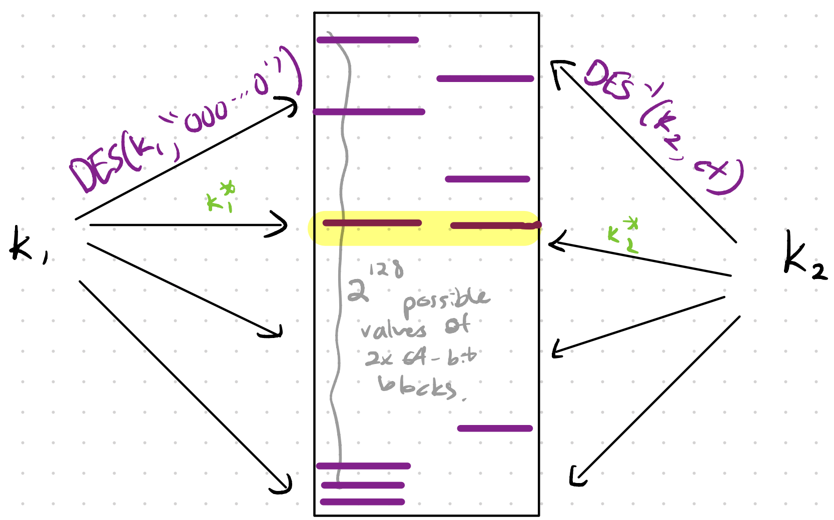
Broken idea: 2DES

- "Meet-in-the-middle" attack.
  - ↳ shows up all over the place.

$$2DES((k_1, k_2), m) := DES(k_2, DES(k_1, m)).$$

- Key is $56 \times 2 = 112$ bits
- Problem: Meet-in-the middle attack

Say attacker gets (m_0, c_0), (m_1, c_1) s.t. $c_i \leftarrow DES(k_2^*, DES(k_1^*, m_i))$



By birthday paradox, expect to find a collision after $\sqrt{2^{128}} = 2^{64}$ time. Space $= 2^{64}$. ← Can reduce?

$\Rightarrow$ Keylen is only effectively 56 bits... no improvement.

How does one break a PRF/PRP?

(See Dan Boneh's CS255 notes.)

Linear cryptanalysis.

$$\Pr_{P,C}\left[P_1 \oplus P_3 \oplus P_5 \oplus C_1 \oplus C_2 \oplus C_7 = k_1 \oplus k_5 \oplus k_{12}\right] \geq \frac{1}{2} + \varepsilon \quad (*)$$

Matsui (1993) found linear relation like this with $\varepsilon = 2^{-21}$

Attack: - Find $1/\varepsilon^2 \approx 2^{42}$ (pt, ct) pairs
- Compute noisy guesses of key bits using (*)
- After $1/\varepsilon^2$ pt/cts, will get correct key bits whp.

↳ Reveals ≈ 13 key bits. Brute force the rest.

$\Longrightarrow$ Small bias causes serious break $2^{56} \to 2^{42}$