

Admin:

Pset #2 due. Pset #3 to be posted tonight.

Project ideas posted on Piazza (team & propos) by 3/18)

Thanks to Nicolai & Tim!

Talk today: 2PM E14-LH-633 (Media Lab)
Robert Hannigan (GCHQ Dir.)
"Encryption, Privacy, & Partnerships"

Today:

Block ciphers:

AES

Modes of operation:

common: ECB, CTR, CBC, CFB, ...

ideal (IND-CCA)

Desai's "UFE" mode

Readings:

Wikipedia "Block cipher modes of operation"

Katz Chapter 3 (esp. 3.6.2 & 3.7)

Paar Chapter 5 (esp. 5.1)

Ferguson Chapter 4

Wikipedia "Ciphertext stealing"

AES

"Advanced Encryption Standard" (U.S. govt)

Replaces DES

AES "contest" 1997-1999:

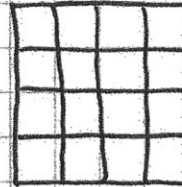
15 algorithms submitted: RC6, Mars, Twofish, Rijndael, ...
 Winner = Rijndael (by Joan Daemen & Vincent Rijmen, (Belgians))

Specs: 128-bit plaintext/ciphertext blocks
 128, 192, or 256-bit key
 10, 12, or 14 rounds (dep. on key length)

Byte-oriented design (some math done in Galois field $GF(2^8)$)

View input as 4x4 byte array:

$4 \times 4 \times 8 = 128$



For version with 128-bit keys, 10 rounds:

- Derive 11 "round keys", each 128 bits (4x4x byte)

- In each round:
 - ① XOR round key
 - ② Substitute bytes (lookup table)
 - ③ Rotate rows (by different amts)
 - ④ Mix each column (by linear opn)

- Output final state

(last round has another round key XORed in instead of mix-column)

See readings for details.

There are very fast implementations. Also Intel has put supporting hardware into its CPU's.

Security: Good; perhaps # rounds should be a bit larger...

For practical purposes, can treat AES as ideal block cipher:

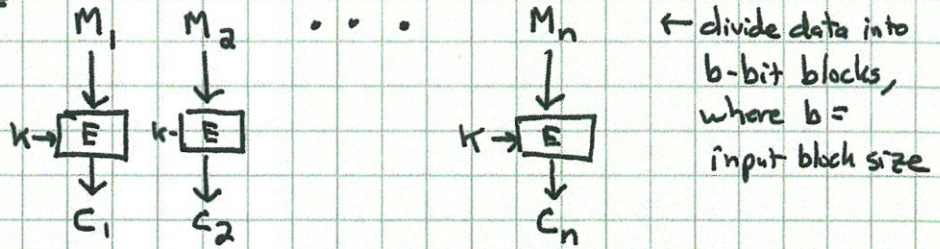
[For each key, mapping $Enc(K, \cdot)$ is a random independent permutation of $\{0,1\}^{128}$ to itself.

Modes of Operation:

How to encrypt variable-length messages? (using AES)

- "ECB" = "Electronic code book"
- "CTR" = "Counter mode"
- "CBC" = "Cipher-block chaining" (& CBC-MAC)
- "CFB" = "Cipher feedback"
- ...
- (others...)

ECB:



To handle data that is not a multiple of b bits in length:

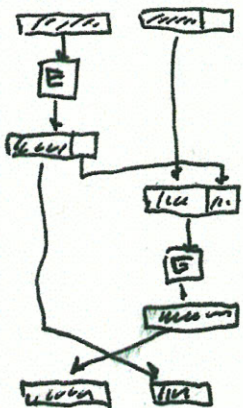
- Append a "1" bit (always)
- Append enough "0" bits to make length a multiple of b bits.

This gives invertible (1-1) "padding" operation. Pad before encryption; unpad after decryption.

ECB preserves many patterns: repeated message blocks ⇒ repeated ciphertext blocks

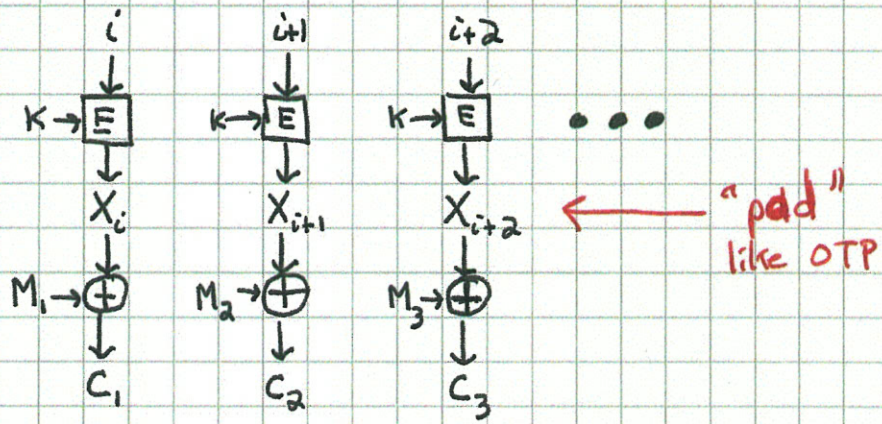
ECB really only good for encrypting random data (e.g. keys)

Ciphertext stealing



CTR (Counter mode):

Generate a PR (pseudorandom) sequence by encrypting $i, i+1, \dots$
XOR with message to obtain ciphertext.



Initial counter value can be transmitted first:

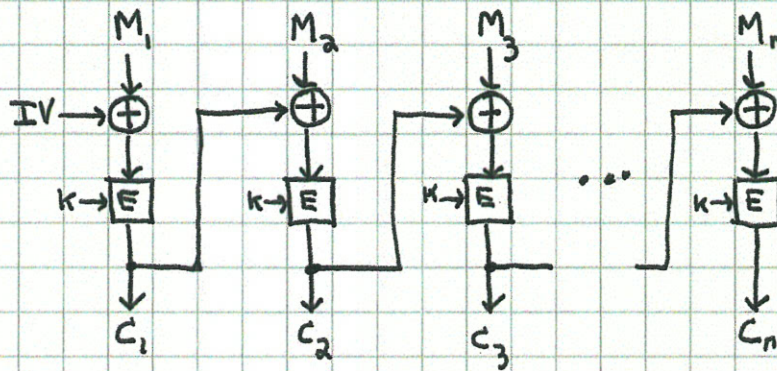
i, C_1, C_2, \dots

Of course, no counter value should be re-used!

CBC (Cipher-block chaining):

Choose IV ("initialization value") randomly, then use each C_i as "IV" for M_{i+1} . Transmit IV with ciphertext:

$$IV, C_1, C_2, \dots, C_n$$



Decryption easy, and parallelizable (∴ little error propagation)

Lookup "ciphertext stealing" for cute way of handling messages that are not a multiple of b bits in length. This method give ciphertext length = message length.

Last block C_n is the "CBC-MAC" (CBC Message Authentication code) for message M . [A fixed IV is used here.] The MAC is a "cryptographic checksum" (more later...) (If messages have variable length then key for last block should be different.)

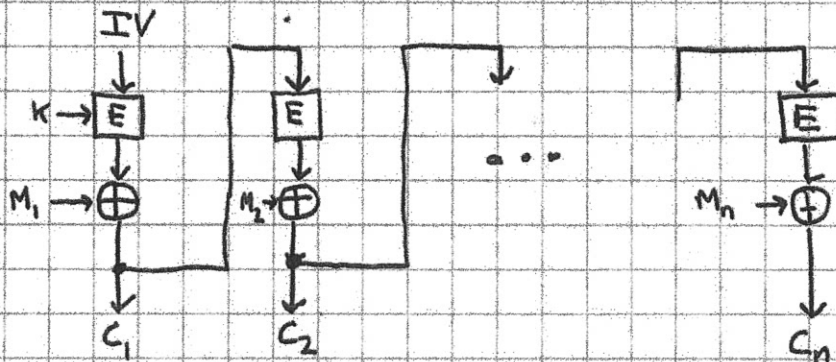
IV might be $Enc(k, msg \#)$ or $Enc(k, nonce)$
 saves space if $msg \#$ does not need to be transmitted, or is short.

MAC should use a different key than that used for encryption: requires 2 passes to do CBC-Enc, then CBC-MAC over ciphertext.



CFB (Cipher feedback mode)

Similar to CBC mode. Uses random IV transmitted with ciphertext.



If M is not a multiple of b bits in length, can just transmit shortened ciphertext. (No need for ciphertext stealing.)

Goal →

Are these modes good ones? What do we want?

If block cipher is indistinguishable from ideal block cipher then mode provides indistinguishability based on chosen ciphertext attack (IND-CCA):

- Define as game with adversary.
- Mode is IND-CCA secure if adversary can win with probability at most $\frac{1}{2} + \epsilon$ for "negligible" ϵ .

Let K be randomly chosen key.

Let E_K denote encryption (using mode) with key K .

Let D_K denote decryption

Phase I ("Find"):

- Adversary given black-box access to E_K, D_K (can encrypt/decrypt whatever it likes)
- Adversary outputs two messages m_0, m_1 , of same length, plus state information s .

Phase II ("Guess"):

- Examiner secretly picks $d \leftarrow_R \{0, 1\}$
Examiner computes $y = E_K(m_d)$
- Adversary given y, s , access to E_K , and access to D_K (except on y)
- Adversary computes for a while, then must produce bit \hat{d} as its guess for d .
- Adversary's advantage is $|P(\hat{d} = d) - \frac{1}{2}|$.

Encryption secure against CCA attack if advantage is negligible.

Fact: To be IND-CCA secure, method must be randomized!

(else Adv can encrypt m_0, m_1 , & compare to y)

& randomization should not be evident to Adv (i.e. not usable for decryption)

Previous modes are not IND-CCA secure!

ECB: not randomized

CTR: starting counter value might be random, but it is transmitted in clear.

In any case, it is legal for Adv to ask for decryption of prefix of y (giving prefix of m_d)

CBC: similar to CTR: IV might be random, but it is transmitted in clear. Decryption of prefix of y also works.

CFB: Same. IV in clear; prefix argument works.

Can one design a IND-CCA scheme?

Theorem: Modes ECB, CTR, CBC, CFB are
not IND-CCA secure.

Proof: Adversary picks $m_0 = 0^x$, $m_1 = 1^x$ for large x .

Then $y = E_k(m_d)$.

Let $z = 1^x$ half of y .

Since $z \neq y$, Adversary allowed in
phase II to ask for $D_k(z)$.

This gives first half of m_d , revealing d .

Adversary always wins. \square

Can one design a IND-CCA scheme?

Given a ciphertext y for a message m ,

Adversary should not be able to construct a

ciphertext z for a related (e.g. truncated) message.

(nonmalleability)

CRYPTO
2006

Here is a sketch of one IND-CCA secure method,
(due to Desai. UFE = "Unbalanced Feistel encryption")

M = long message, sequence M_1, M_2, \dots, M_n of b -bit blocks.

$K = (K_1, K_2, K_3)$ Three indep. keys for block ciphers.

$r \xleftarrow{R} \{0, 1\}^b$ starting counter value

pad $P = P_1, P_2, \dots, P_n$ where $P_i = E_{K_1}(r+i)$ ← (CTR mode)

ciphertext $C = C_1, C_2, \dots, C_n$ where $C_i = M_i \oplus P_i$

CBC-MAC: $X_0 = 0^b$

$$X_i = E_{K_2}(X_{i-1} \oplus C_i) \quad 1 \leq i < n$$

$$X_n = E_{K_3}(X_{n-1} \oplus C_n) \quad (\text{MAC})$$

← last block uses K_3

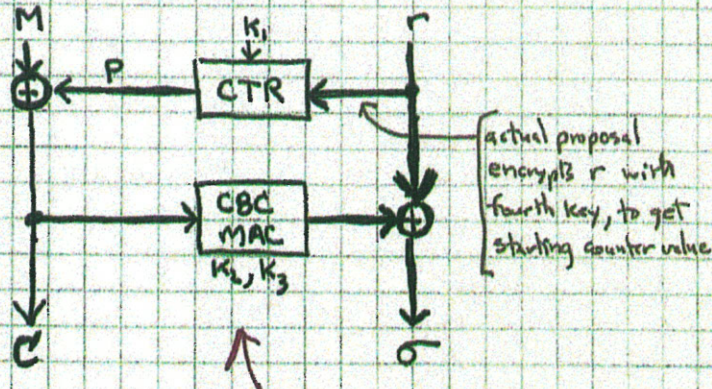
$$\sigma = r \oplus X_n$$

use MAC to mask r
(no message authentication)

Output: $C_1, C_2, \dots, C_n, \sigma$

VO-PRF

VI-PRF



CBC MAC uses K_2 mostly, but K_3 on last block

Pad C with $100\dots 0$ before computing MAC so MAC deals with integral # of blocks.

- Encryption with UFE can be done in single pass ^("online" property) over data, but decryption requires two passes:
 - first to compute mask X_n , then to get r
 - second to decrypt C to get M
- Only designed for confidentiality (there is no way provided for receiver to tell if ciphertext has been tampered with.) (Need to use MAC on top of all of this, or some "combined mode" providing both confidentiality & integrity.)
- Note "unbalanced Feistel structure".
- Length of ciphertext $(C, r) = |M| + |r|$;
expansion only as needed for randomization.
No need for "ciphertext stealing" since we use CTR mode.