
COMPUTER SECURITY

Cryptography: more advanced topics ([2](#))

Attack models ([2](#))

Cryptographic Models ([5](#))

Randomness ([7](#))

General enciphering schemes ([11](#))

Cipher modes of operation ([14](#))

Base method ([14](#))

Some operation modes ([17](#))

Padding ([24](#))

(to be continued...) ([32](#))

Cryptography: more advanced topics

Attack models¹

Definition

- specification of the kind of access a cryptanalyst has when attempting to break a targeted cryptographic system²

Attack types

- normal
 - cryptanalyst can only obtain ciphertexts that are available (and that he does not control)
- known plaintext (“passively” obtained)
 - cryptanalyst has obtained the enciphered counterparts of some plaintexts

1 or: classification of attacks

2 Cryptanalyst limitations due to time and computational power are not considered (or are assumed to be the maximum reasonably possible).

...Attack models...

- chosen plaintext (“actively” obtained)
 - cryptanalyst can obtain the ciphertexts of any plaintext she chooses¹
 - trivial with public key cryptography! (Why?)
 - Exercise: show how to use this attack to obtain the key used by a (not so truly) *one-time pad*.
- adaptive chosen plaintext (“interactively” obtained)
 - cryptanalyst can obtain the ciphertexts of additional plaintexts after seeing previous pairs (plaintext, ciphertext); usually this implies a (real time?) interaction with encryption oracle
- chosen ciphertext
 - cryptanalyst can gather information by decipherment of chosen ciphertexts²
- adaptive chosen ciphertext
 - cryptanalyst may also be able to choose ciphertexts to be deciphered based on previous decipherments; usually this implies a (real time?) interaction with decryption oracle

1 i.e. from an encryption oracle

2 i.e. from an decryption oracle

...Attack models...

- open key
 - cryptanalyst has obtained sufficient knowledge about the key to allow the decipherment of ciphertexts¹
- side-channel
 - cryptanalyst can gather information not obviously related to the encipherment operations (electronic noise, sound, elapsed time) that allow the grabbing of the key²
- social engineering
 - cryptanalyst tricks some human to decipher ciphertexts or yield the secret key

Study of defense

- the defense from an attack depends on the characteristics of the cryptographic system; the formal³ approach uses *cryptographic models*

1 of course, made with that key

2 or, at least, the decipherment of some ciphertexts

3 kind of theoretical

Cryptographic Models

Definition

- formal descriptions of the security properties and assumptions of cryptographic systems. Should define: adversarial capabilities; security goals¹; security assumptions (environmental and operation details²)...

Models

- standard
 - the cryptanalyst is only limited by the amount of time and computational power she has available (so, access to all operation details and capabilities is granted)

1 e.g. confidentiality

2 such as computing resources

...Cryptographic Models...

- random oracle¹
 - the cryptanalyst may use an (ideal) function (or black box) that
 - for each input, outputs a unique and (truly) random value, uniformly distributed in the (infinite) co-domain
 - is deterministic: always outputs the same value every time the same input is submitted
- generic group
 - the cryptanalyst is given access to a randomly chosen "encoding" of a group, instead of specific "encodings" (eventually used in practice)
- common reference string
 - the cryptanalyst is given access to a special string, taken from some distribution, that is shared by all involved parties
 - common random string
 - when the string is taken from a uniform random distribution

1 Oracle is a "black box" that is able to produce a (true) solution for any instance of a given computational problem (i.e. a decision problem).

Randomness

- essential in Cryptography!
 - one time pad, IV (initialization values), stream cipher seeds
 - hashes
 - *nonces*, key generation...
- generation
 - excellent: physical source
 - inherent: radioactive decay, brownian movement, ...
 - depending on initial conditions: (non-biased) roulette or dice, ...
 - reasonable: algorithmic-based with physical seed
 - cryptographically secure pseudorandom number generators
 - use physical (hopefully random) sources (e.g. mouse movements)
 - Linux's `getrandom()` (`/dev/random`, `/dev/urandom`)
 - bad: algorithmic-based
 - pseudorandom number generators
 - POSIX's `random()`

Evaluation

- frequency analysis
 - determine the frequency distribution of digits or bit patterns of a sequence of values:
 - if (truly) random, each digit or bit occurs with approximately equal frequency
- entropy measurement¹
 - measure of the unpredictability of the values in sequence:
 - if values are (truly) random, unpredictability is maximum, and so is entropy

1 Calculation of entropy varies: in computing, if values occur with equal probability, $E = \log_2$ (no. of possible values) ; if value is a bit, it can be 0 and 1; then $E = 1$ (bit).

In information theory (Shannon!) E (in bits) = $-\sum_i [(probability\ of\ occurrence\ of\ value\ i) * \log_2 (probability\ of\ occurrence\ of\ value\ i)]$, where i is a value from a possible set. Again, if i is a bit, and its 0 or 1 value occurs with equal probability, $E = 1$ (bit).

...Randomness: evaluation...

- statistical tests
 - examination of properties such as uniformity, independence and distribution of sequence values. Examples: Chi-square¹, Kolmogorov-Smirnov², RUNS³.
 - if sequence is (truly) random, results depend on specific test performed
- serial correlation measurement
 - check for correlations between successive values:
 - if (truly) random sequence of values, correlation should be zero
- *randomness tests*
 - run specialized tests. Examples of test suites: NIST Statistical⁴, Dieharder⁵, ENT⁶.
 - if sequence is (truly) random, results depend on specific test performed

1 en.wikipedia.org/wiki/Chi-squared_test

2 en.wikipedia.org/wiki/Kolmogorov%E2%80%93Smirnov_test

3 en.wikipedia.org/wiki/Wald%E2%80%93Wolfowitz_runs_test

4 nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-22r1a.pdf

5 webhome.phy.duke.edu/~rgb/General/dieharder.php

6 www.fourmilab.ch/random/

...Randomness: evaluation...

ENT, A Pseudorandom Number Sequence Test Program

- battery of tests:
 - frequency (ideal: all values with same number of occurrences)
 - entropy (ideal: 8 bits per byte)
 - compression (ideal: 0 % compression)
 - Chi-square (ideal:] ~10%, ~90% [)
 - arithmetic mean (ideal: 50% of possible values)
 - Monte Carlo value for Pi (ideal: Pi with very "low" error)
 - Serial correlation coefficient (ideal: 0)
- used in a SEED lab!

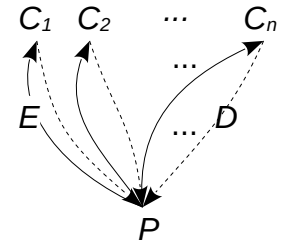
General enciphering schemes

Definition

- sets of algorithms and protocols used to transform plaintext (clear data) into ciphertext (concealed data) in such a way that unauthorized users cannot reverse the transformation.

Types

- deterministic encipherment
 - the same ciphertext is always produced for a given plaintext and key
- probabilistic encipherment
 - different ciphertexts are, in general, produced for a given plaintext and key¹
- format-preserving encipherment
 - ciphertext is produced in the same format² as the plaintext



¹ An example is ElGamal's encryption system.

² The meaning of "format" varies: only letters from English alphabet are used; n -bit block cipher (only n -bit numbers are accepted and produced), etc.

...General enciphering schemes...

- perfect secrecy encipherment
 - the ciphertext reveals no information at all about the plaintext
- semantic security encipherment
 - the ciphertext could reveal some information about the plaintext, but it cannot be feasibly extracted
- indistinguishable encipherment
 - a ciphertext does not reveal information to allow distinguishing which plaintext produced it from a group of chosen plaintexts; or the distinction is no better than that of random guessing
- malleable encipherment
 - the ciphertext produced for a given (possibly unknown) plaintext can be transformed into another ciphertext which deciphers to a plaintext related to the first

...General enciphering schemes...

- homomorphic encipherment

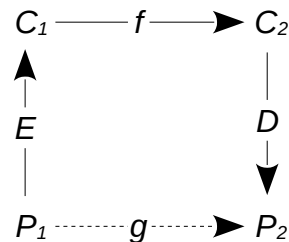
- the ciphertexts are able to suffer computations that, when deciphered, are identical to related computations on the corresponding plaintexts
- Ex.: RSA public key with modulus n , encryption exponent e and plaintext P :

$E(P) = P^e \bmod n$. The homomorphic property is then:

$$E(P_1) * E(P_2) = P_1^e * P_2^e \bmod n = (P_1 * P_2)^e \bmod n = E(P_1 * P_2)$$

- (perfect) forward secrecy encipherment¹

- being able to grab a session key (and so being able to decipher the session) does not allow the decipherment of previous sessions. (Also, knowledge of a long-term key does not allow the decipherment of past sessions.)²



1 this has to do more with key exchange schemes than with the encipherment operations by themselves

2 However, the breaking of the encipherment *algorithm*, in the sense of being able to operate it without a cryptographic key, might allow the decipherment of past sessions.

Cipher modes of operation¹

Base method

- $P = P_1 P_2 \dots$ parts (blocks) of equal size
 - block size: 1 b, 1 B, 8 B (typical), 16 B (typical)...
- enciphering methods:
 - stream
 - $K = K_1 K_2 \dots : C = E_{K_1}(P_1) E_{K_2}(P_2) \dots =^2 K_1(P_1) K_2(P_2) \dots$
 - block
 - $K : C = K(P_1) K(P_2) \dots$
 - “mix” of previous
 - $K, v_1, v_2 \dots^3 : C = E_K(P_1, v_1) E_K(P_2, v_2) \dots = K_{v_1}(P_1) K_{v_2}(P_2) \dots$

1 Necessary for the symmetric encipherment of “long” texts. But, in practice, almost any text is “long”!...

2 for simplicity

3 real single key with additional (and different) information per block: overall, like a different “virtual” key per block

...Cipher modes of operation...

Rationale for "operation modes"¹

- stream
 - Pro: most secure²
 - Con: long, one-time usable, (random) key
- block
 - Pro: simplicity and single (random) key
 - Con: same plaintext, same ciphertext
 - if $P_1 = P_2$, then $C_1 = C_2$ [FIG]
- mixed
 - Pro: single (random) key
 - Con: added complexity
 - several possibilities

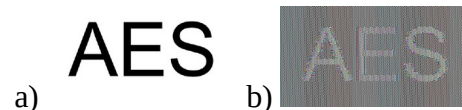


Fig. a) original picture;
b) enciphered with AES 256b, ECB mode

- 1 Goal is *confidentiality* protection; *integrity* protection is not guaranteed: with some modes, even the "mixed", modifications of ciphertext might go undetected; for confidentiality and integrity protection, authenticated encipherment is used.
- 2 even *provable* secure with *One-time pad*

...Cipher modes of operation...

Pictures' notation

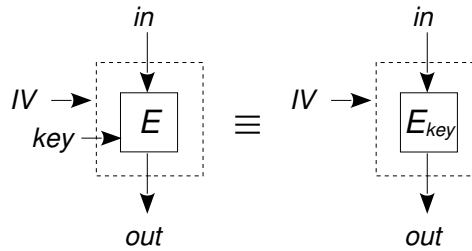


Fig. IV is Initialization Value (or Vector), public value that, as a rule, should be random.

Some operation modes

Stream method

- Some properties:
 - usually, $E = D = \text{XOR}^1 (\oplus)$
 - no padding of last block
 - parallelizable en/deciphering
 - ultimate security: K_i random, one-time value
- Formulas:
 - $C_i = E_{k_i}(P_i)$, $i > 0$
 - usually, $P_i = E_{k_i}(C_i)$
- Error propagation:²
 - exercise!

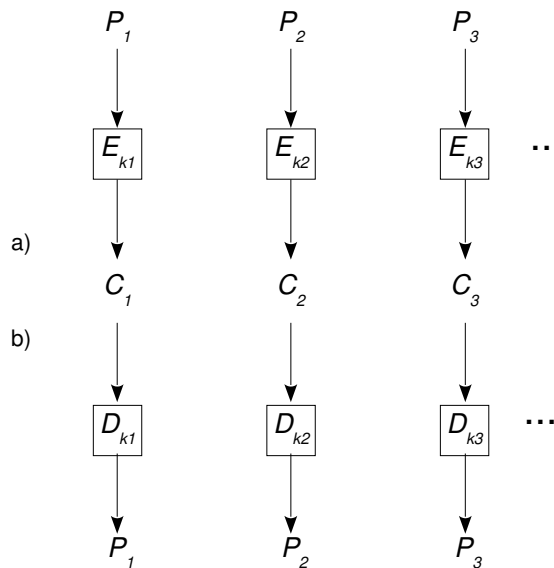


Fig. Use of plain stream method: a) enciphering; b) deciphering

1 bitwise

2 When at least one bit/byte of C_i is garbled, how that is reflected in following blocks.

...Cipher modes of operation ...

Block method

- *ECB, Electronic Code Book*
- Some properties:
 - padding of last block
 - parallelizable en/deciphering
- Formulas:
 - $C_i = E_k(P_i)$, $i > 0$
 - Write the decipherment formula. :-)
- Error propagation:
 - exercise!

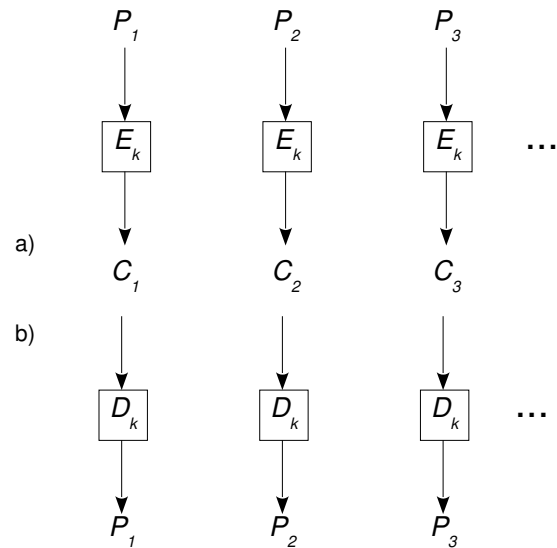


Fig. Use of (plain) block method: a) enciphering;
b) deciphering.

...Cipher modes of operation ...

“Mix” method: CTR

- CTR, Counter Mode
- Some properties:
 - IV^1 (random + counter)
 - no padding
 - parallelizable en/deciphering
- Formulas:
 - Write the en/decipherment formulas.
- Error propagation:
 - exercise!

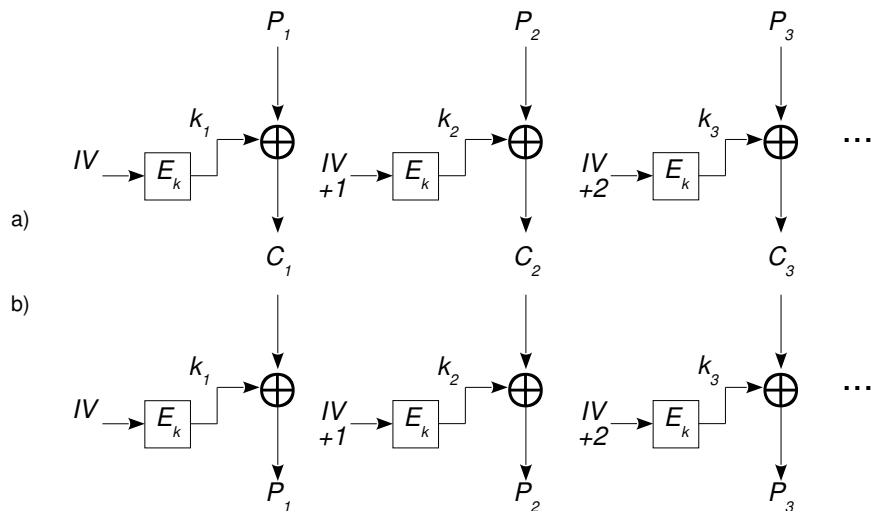


Fig. Use of “mixed” method CTR: a) enciphering; b) deciphering.
(Notice the virtual keys k_i .)

1 public value that, as a rule, should be random

...Cipher modes of operation ...

“Mix” method: CFB

- *CFB, Cipher FeedBack*
- Some properties:
 - *IV* (random)
 - no padding
 - not parallelizable enciphering; parallelizable deciphering
- Formulas:
 - $C_0 = IV$;
 - $C_i = P_i \oplus E_k(C_{i-1})$, $i > 0$
 - Write the decipherment formula.
- Error propagation:
 - exercise!

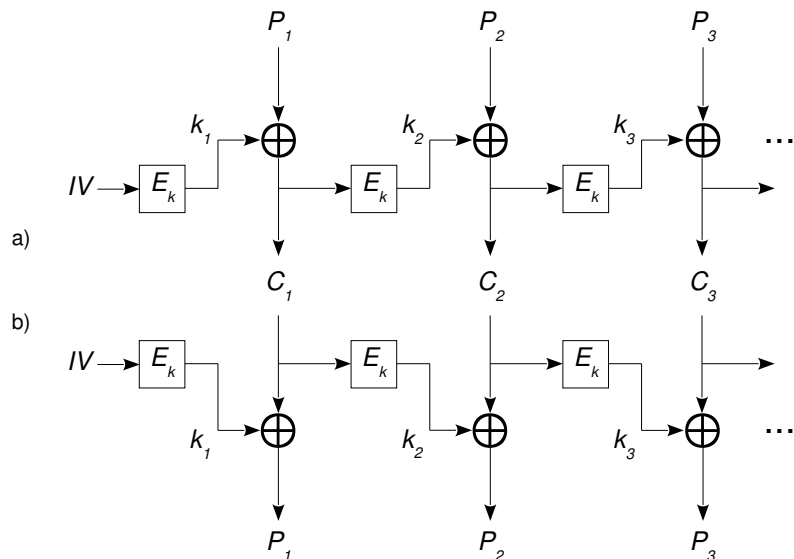


Fig. Use of “mixed” method CFB: a) enciphering; b) deciphering.
(Notice the virtual keys k_i .)

...Cipher modes of operation ...

“Mix” method: OFB

- OFB, Output FeedBack
- Some properties:
 - IV (random)
 - no padding
 - not parallelizable en/deciphering, but successive $E_k^i(IV)$ can be done in advance
- Formulas:
 - $C_i = P_i \oplus E_k^i(IV)$, $i \geq 0$
 - Write the decipherment formula.
- Error propagation:
 - exercise!

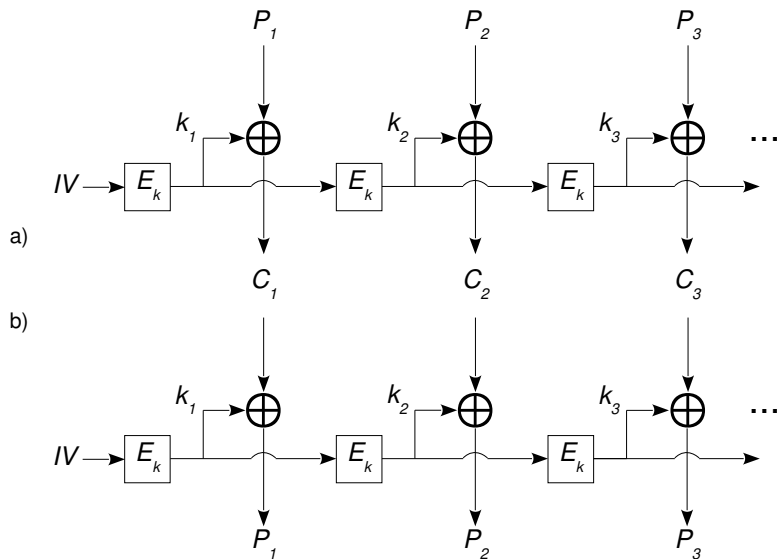


Fig. Use of “mixed” method OFB: a) enciphering; b) deciphering.
(Notice the virtual keys k_i .)

...Cipher modes of operation ...

“Mix” method: CBC

- *CBC, Cipher Block Chaining*
- Some properties:
 - *IV* (random) or explicit initialization by (phony) 1st block!
 - padding
 - not parallelizable enciphering;
parallelizable deciphering
- Formulas:
 - $C_0 = IV$; $C_i = E_k(P_i \oplus C_{i-1}) \quad i > 0$
 - Write the decipherment formula.
- Error propagation:
 - exercise!

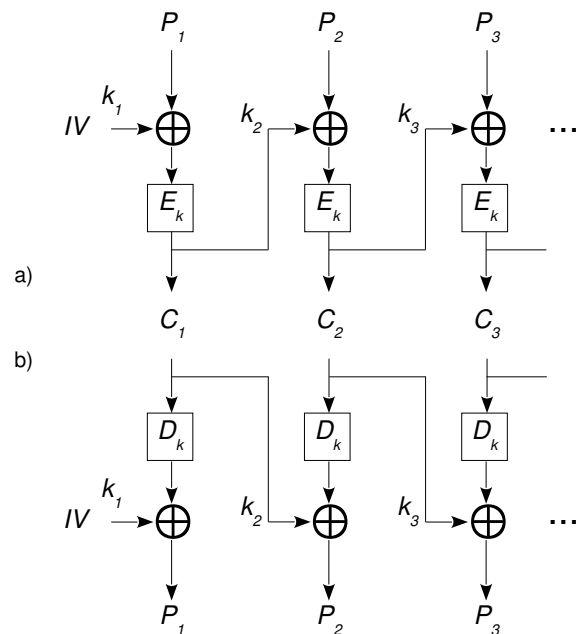


Fig. Use of “mixed” method CBC: a) enciphering;
b) deciphering
(Notice the virtual keys k_i .)

...Cipher modes of operation ...

Another view of some operation modes

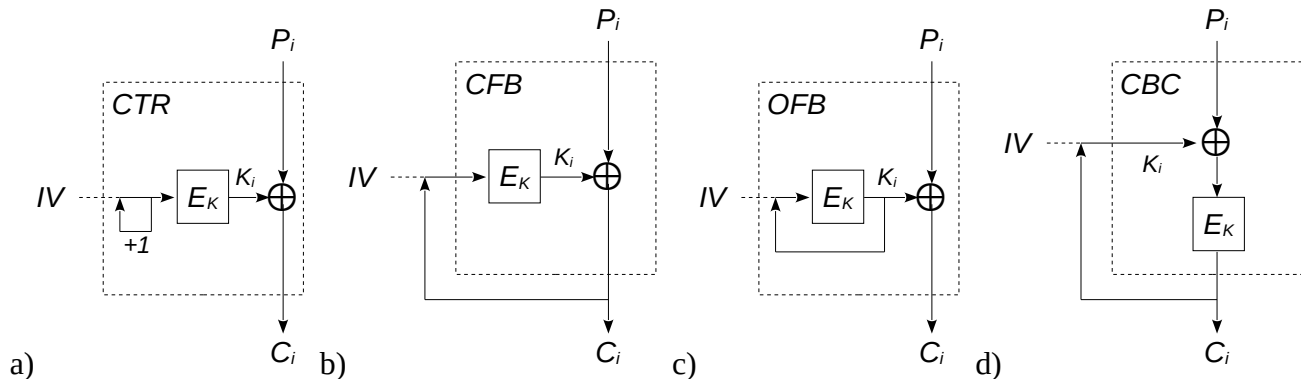


Fig. The software-view of some of the operation modes ($i > 0$). In b) and c) the reason for the modes' names is apparent...

Padding

Need

- size of plaintext varies (just hardly ever is multiple of block size)
 - so, final block might need¹ padding!
 - but, "casual" padding might open an attack path (*see ahead*)!
- harden message deciphering and traffic analysis²
 - by obscuring the size (and content) of ciphertext
 - e.g. avoiding short messages' attack on RSA³
 - e.g. avoiding deterministic ciphering's attack⁴

1 Why?... Also, some "modes of operation" do not need padding... why?

2 interception and examination of communications (ciphered or not) to deduce information (e.g. from patterns)

3 asecuritysite.com/encryption/crackrsa2

4 As same plaintext always produces same ciphertext, a cryptanalyst may build a collection of plaintext/ciphertext pairs and look for cipher matches in communication media; it is specially feasible with "public-key cryptography" (why?)!

...Cipher modes of operation: Padding...

Padding schemes

- several schemes (bit padding or, more usually, byte padding)
 - shared-key cryptography
 - e.g. PKCS¹ #5², #7³ (enciphering) [FIG]
 - one-way cryptography
 - e.g. RFC 6234 (SHA-1, SHA-256) [FIG]
 - e.g. SHA3 (sponge) [FIG]
 - public-key cryptography
 - e.g. PKCS #1 v2 (RFC 8017)
 - RSA's PKCS1-v1_5 [FIG]
 - RSA's OAEP, Optimal Asymmetric Encryption Padding [FIG]
 - Exercise (after analyzing picture): what about deciphering?... does receiver need *seed* and *L*?...

1 Public Key Cryptography Standards, devised and published by RSA Security LLC since the 1990s

2 PKCS #5: Password-Based Cryptography - from a password, generate a (symmetric) key for a following symmetric encipherment.

3 #7 padding just extends 8B block #5 padding to 16B (128b) blocks

...Cipher modes of operation: Padding examples (figs)...

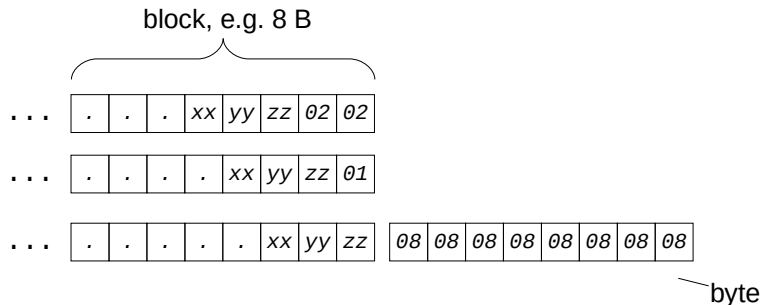


Fig: Shared-key cryptography padding: examples for PKCS #5 (8B blocks); #7 will be similar, but appropriate to 16B blocks.

Algorithm: add $(\text{block_size} - P_length \bmod \text{block_size})$ bytes; all with value equal to number of added bytes: e.g. if 3 bytes are needed to complete last block, each added byte's value is 3.

...Cipher modes of operation: Padding examples (figs)...

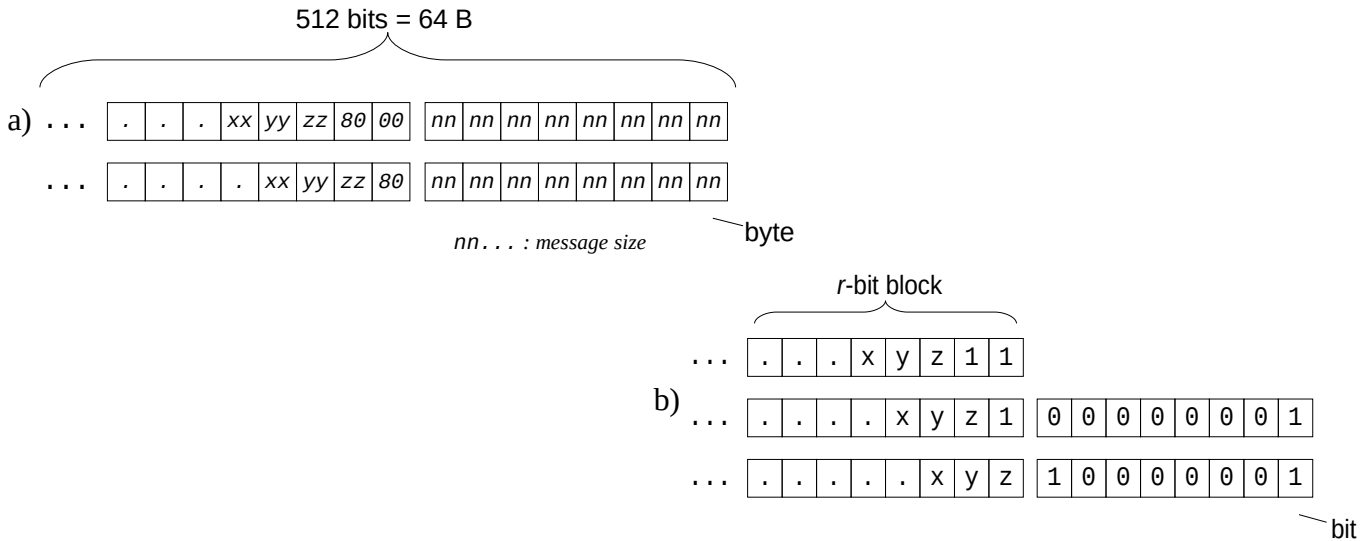


Fig: Instances of one-way cryptography padding:
 a) RFC 6234 padding: (SHA1, SHA256...) - sequence of *nns* is message size;
 b) Sponge *multirate* padding: 10^*1 (*r* is the number of bits of input block).

...Cipher modes of operation: Padding examples (figs)...

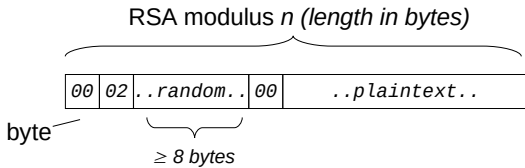
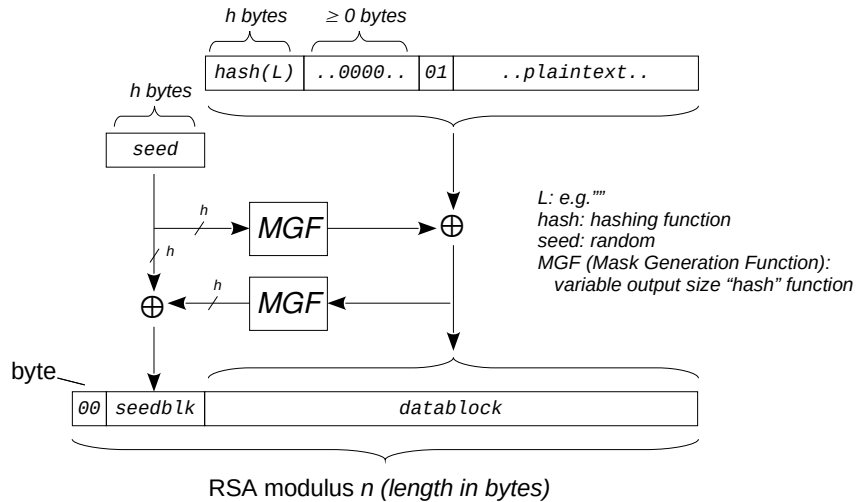


Fig. RSA padding: PKCS1-v1_5.

Fig. RSA padding: OAEP, *Optimal Asymmetric Encryption Padding*. After padding, RSA enciphering proceeds with final data being treated as of n -byte hex number.



...Cipher modes of operation: Padding...

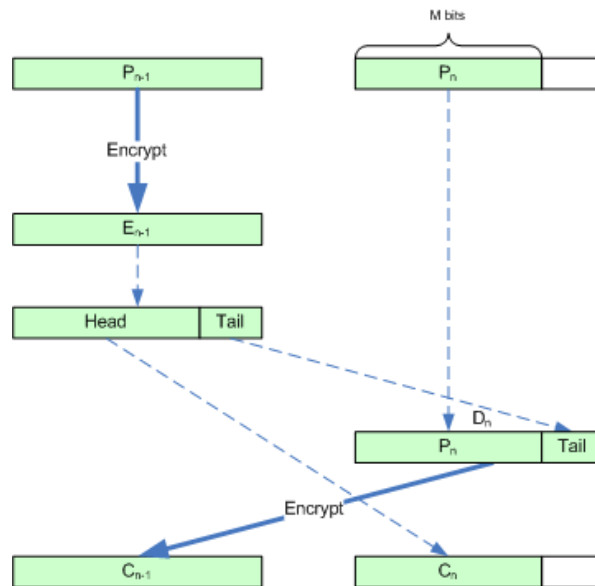
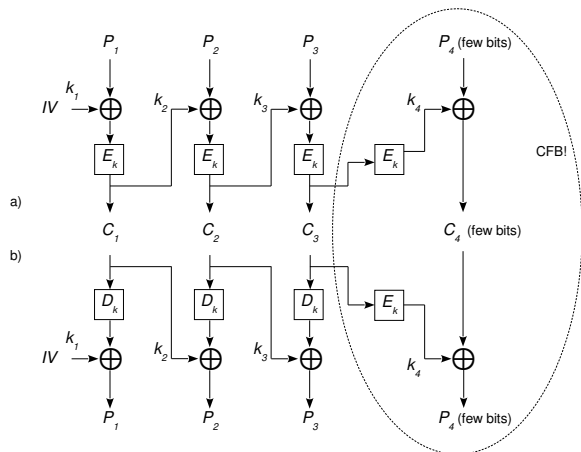
Attack examples

- length extension: one-way cryptography, MAC (if $= h(K||P)$)
 - if $hash(P1) = hash(IV, P1) = hash(hash(IV), P1)$
 $hash(P1||P2) = hash(P1, P2) = hash(hash(P1), P2)$
 - SEED Lab!
- padding oracle: two-way cryptography, CBC mode
 - if attacker can keep testing decipherment with crafted ciphertext
 - if deciphering error code says explicitly "*invalid padding*" instead of a general "*decryption failed*"
 - CBC: $P_i = D_k(C_i) \oplus C_{i-1} \quad i > 0$
 - a byte/bit change in C_{i-1} affects corresponding byte/bit in P_i
 - starting from last C_i block (where padding is), keep changing last byte until padding is valid; then repeat for previous bytes
 - see [FIG] (PKCS #5, #7 padding)

...Cipher modes of operation: Padding...

Real need for padding?

- avoidance:
 - ciphertext stealing [FIG in Wikipedia]
 - residual block termination [FIG]
- will it be worth the trouble?...



(to be continued...)