

Υπολογιστική Θεωρία Αριθμών και Κρυπτογραφία

Εισαγωγή - Κλασσικά κρυπτοσυστήματα

Άρης Παγουρτζής – Στάθης Ζάχος

Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Μυστικότητα / Ιδιωτικότητα (Secrecy / Privacy)

- ▶ Κρυπτογράφηση: μετασχηματισμός απλού κειμένου / μηνύματος (*plaintext*) σε κρυπτοκείμενο (*ciphertext*), συνήθως με χρήση κλειδιού.
- ▶ Αποκρυπτογράφηση: μετασχηματισμός κρυπτοκειμένου στο αρχικό κείμενο, συνήθως με χρήση κλειδιού.
- ▶ Παραγωγή / Διανομή κλειδιού
- ▶ Συμμετρικά κρυπτοσυστήματα / Ιδιωτικού κλειδιού (κρυπτογραφία διπλής κατεύθυνσης): μονοαλφαβητικά, πολλαλφαβητικά, τμήματος, ροής, DES, AES
- ▶ Κρυπτοσυστήματα δημοσίου κλειδιού (κρυπτογραφία μονής κατεύθυνσης): Knapsack, RSA, ElGamal, Elliptic Curves

Επισκόπηση Κρυπτογραφίας: authentication, integrity

Έλεγχος γνησιότητας / αυθεντικοποίηση (Authentication)

- ▶ Data / message origin: ψηφιακές υπογραφές, κυρίως βασισμένες σε συστήματα δημοσίου κλειδιού αλλά και Message Authentication Codes (MACs)

Μη αποκήρυξη (Non-Repudiation): κανείς δεν μπορεί να αποποιηθεί την υπογραφή του

- ▶ Entity / User: Identification Schemes, πρωτόκολλα ταυτοποίησης (Interactive Proofs (IP), Zero Knowledge (ZK))

Ακεραιότητα (Integrity)

- ▶ Συνήθως περιλαμβάνεται στην αυθεντικοποίηση
- ▶ Hash Functions (επίσης έχουν μεγάλη χρήση στις ψηφιακές υπογραφές)
- ▶ Συνδυασμός με αυθεντικοποίηση (MACs = keyed hash functions)

Επισκόπηση Κρυπτογραφίας: keys, protocols

Διαχείριση κλειδιών (Key Management)

- ▶ Παραγωγή
- ▶ Διανομή
- ▶ Έμπιστη αρχή

Πρωτόκολλα (πολλών συμμετεχόντων)

- ▶ Broadcast
- ▶ Consensus
- ▶ Mental poker
- ▶ Secure Function Evaluation (SFE), Secure Multiparty Computation (S-MPC)
- ▶ Voting / Elections
- ▶ Interactive Proofs / Zero Knowledge / User Authentication

Τύποι κρυπταναλυτικών επιθέσεων

Θεμελιώδης αρχή (Kerckhoffs): όλοι οι αλγόριθμοι είναι γνωστοί, **μόνο το κλειδί είναι άγνωστο** (μην υποτιμάς τον αντίπαλο!).

1. Κρυπτοκείμενο μόνο (**ciphertext only – CO**). Ο κρυπταναλυτής διαθέτει μόνο το κρυπτοκείμενο.
2. Γνωστό αρχικό κείμενο (**known plaintext attack – KPA**). Ο κρυπταναλυτής διαθέτει κάποια ζεύγη αρχικού κειμένου–κρυπτοκειμένου.
3. Επιλεγμένο αρχικό κείμενο (**chosen plaintext attack – CPA**). Ο κρυπταναλυτής διαθέτει κάποια ζεύγη αρχικού κειμένου–κρυπτοκειμένου, με αρχικά κείμενα της επιλογής του.
4. Επιλεγμένο κρυπτοκείμενο (**chosen ciphertext attack – CCA**). Ο κρυπταναλυτής διαθέτει κάποια ζεύγη αρχικού κειμένου–κρυπτοκειμένου για ορισμένα κρυπτοκείμενα της επιλογής του (ισοδύναμα, έχει προσωρινή δυνατότητα αποκρυπτογράφησης).

The indistinguishability game (Το παίγνιο της μη-διακρισσιμότητας)

1. Μια (σχετικά) νέα θέωρηση: ακόμη και αν κάποιος γνωρίζει ότι ένα κρυπτοκείμενο c αντιστοιχεί σε ένα από δύο μόνο συγκεκριμένα αρχικά κείμενα m_0, m_1 , δεν θα πρέπει να είναι σε θέση να ξεχωρίσει ποιο από τα δύο είναι το σωστό με πιθανότητα σημαντικά μεγαλύτερη από $\frac{1}{2}$.
2. Μπορεί να διατυπωθεί σαν παίγνιο μεταξύ ενός αντιπάλου και ενός κρυπτοσυστήματος.
3. Μπορεί να εφαρμοστεί σε όλες τις επιθέσεις. Περισσότερο γνωστό για IND-CPA και IND-CCA (και IND-CCA2) στα πλαίσια της κρυπτογραφίας δημοσίου κλειδιού (όπου εξ ορισμού ο αντίπαλος έχει δυνατότητα CPA τουλάχιστον).
4. Τυπικοί ορισμοί αργότερα.

Κλασικά κρυπτοσυστήματα

- ▶ **Κρυπτοσυστήματα Αντικατάστασης (substitution ciphers):** κάθε γράμμα (ή ομάδα γραμμάτων) του αρχικού κειμένου αντικαθίσταται με ένα ή περισσότερα γράμματα.
- ▶ **Κρυπτοσυστήματα Μετάθεσης / Αναδιάταξης (transposition ciphers):** τα γράμματα του αρχικού κειμένου αναδιατάσσονται (συνήθως κατά ομάδες).

Συνήθως αφορούν σε κρυπτογράφιση κειμένου φυσικής γλώσσας.

Κρυπτοσυστήματα αντικατάστασης

- ▶ *Μονοαλφαβητικά*: κάθε γράμμα του αρχικού κειμένου κωδικοποιείται πάντοτε με το ίδιο γράμμα (γενικότερα: με τον ίδιο τρόπο).

Κρυπτοσυστήματα: αντικατάστασης (substitution cipher), ολίσθησης (shift cipher: π.χ. Καίσαρα), παραλλαγή Καίσαρα με χρήση λέξης-κλειδί, PLAYFAIR, affine cipher.

- ▶ *Πολυαλφαβητικά*: κάθε γράμμα του αρχικού κειμένου μπορεί να κωδικοποιείται με διαφορετικό τρόπο σε διαφορετικά σημεία του κειμένου.

Κρυπτοσυστήματα: Vigenère, AUTOCLAVE, Hill, rotor, Enigma, Vernam (one-time pad), κρυπτοσυστήματα πακέτου (block ciphers: DES, AES), κρυπτοσυστήματα ροής (stream ciphers),.

Κρυπτοσύστημα Καίσαρα

Caesar cipher: ολίσθηση κατά 3 (γενικότερα κατά k)

Αρχικό: A B C D E F G H I J K L M N O P Q R S T U

Κρυπτ/νο: D E F G H I J K L M N O P Q R S T U V W X

Τα κείμενα και το κλειδί αποτελούνται από κεφαλαία γράμματα της Αγγλικής γλώσσας (χωρίς κενά), τα οποία αντιστοιχίζουμε στους αριθμούς από 0 έως 25.

Παράδειγμα

CRYPTOGRAPHY \rightarrow FUBSWRJUSKV

Κρυπτανάλυση

Εύκολη αν το αρχικό κείμενο ανήκει σε φυσική γλώσσα: δοκιμές, συχνότητες εμφάνισης. Αδύνατη για τελείως τυχαίο αρχικό κείμενο. Ισχύει για όλα τα μονοαλφαβητικά συστήματα.

Κρυπτοσύστημα Καίσαρα με κλειδί

Keyword-CAESAR cipher

Κλειδί: ακέραιος $k \in [0, 25]$ (π.χ. $k = 7$) και κωδική λέξη (π.χ.

TENFOUR)

Αρχικό:	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
Κρυπτ/vo:	P	S	V	W	X	Y	Z	T	E	N	F	O	U	R	A	B	C	D	G	H	I

Κρυπτανάλυση

Το πλήθος των δοκιμών αυξάνεται πάρα πολύ. Αλλά με μέτρηση συχνοτήτων είναι εφικτή, για αρχικό κείμενο σε φυσική γλώσσα.

Άμυνα: με χρήση ομοφώνων (homophones).

Affine Cipher

- ▶ Key: (a, k) τ.ω. $\gcd(a, 26) = 1$
- ▶ $Enc(x) = a \cdot x + k \pmod{26}$
- ▶ $Dec(y) = a^{-1}(y - k) \pmod{26}$.

Ορθότητα αποκρυπτογράφησης:

$$y \equiv ax + k \Rightarrow y - k \equiv ax \Rightarrow a^{-1}(y - k) \equiv x \pmod{26}.$$

$a^{-1} \in \mathbb{Z}_{26}$ ($= \{0, \dots, 25\}$): πολλαπλ/κός αντίστροφος του a modulo 26, δηλ. $a \cdot a^{-1} \pmod{26} = 1$

Υπάρχει (και είναι μοναδικός) ανν $\gcd(a, 26) = 1$.

‘1-1’ κρυπτογράφηση: $ax_1 + k \equiv ax_2 + k \pmod{26}$

$$\Rightarrow a(x_1 - x_2) \equiv 0 \pmod{26} \Rightarrow 26 \mid a(x_1 - x_2)$$

αλλά επειδή $\gcd(26, a) = 1$, προκύπτει $26 \mid x_1 - x_2 \Rightarrow x_1 = x_2$.

Μονοαλφαβητικό σύστημα, κρυπτανάλυση με μέτρηση συχνοτήτων.

Κρυπτόςστημα Vigenère

Ορισμός

- ▶ $K = (k_0, k_1, \dots, k_{r-1})$: κλειδί, r χαρακτήρων
- ▶ $X = (x_0, x_1, \dots, x_{n-1})$: αρχικό κείμενο (plaintext), n χαρακτήρων
- ▶ $C = (c_0, c_1, \dots, c_{n-1})$: κρυπτοκείμενο (ciphertext), n χαρακτήρων
- ▶ $c_i = E_K(x_i) = (x_i + k_{i \bmod r}) \bmod 26, 0 \leq i \leq n - 1$: κρυπτογράφηση
- ▶ $x_i = D_K(c_i) = (c_i - k_{i \bmod r}) \bmod 26, 0 \leq i \leq n - 1$: αποκρυπτογράφηση

Κρυπτανάλυση

Η κρυπτανάλυση συνίσταται στην εύρεση του μήκους του κλειδιού πρώτα και κατόπιν στην εύρεση του ίδιου του κλειδιού.

Κρυπτανάλυση Vigenère

Εύρεση μήκους κλειδιού: 2 τρόποι

- ▶ *Kasiski test*: εύρεση patterns που επαναλαμβάνονται. Πιθανή περίοδος: ΜΚΔ των αποστάσεων μεταξύ επαναλαμβανόμενων patterns. Βασική ιδέα: ίδιες λέξεις του αρχικού κειμένου σε απόσταση πολλαπλάσια του r (μήκος κλειδιού), κωδικοποιούνται με ίδιο τρόπο.
- ▶ *Index of Coincidence (Δείκτης Σύμπτωσης)*: εκφράζει την πιθανότητα δύο τυχαίοι χαρακτήρες ενός κειμένου να ταυτίζονται.
Η τιμή του σε κείμενο φυσικής γλώσσας διαφέρει σημαντικά από την τιμή του σε τυχαίο κείμενο.

Δείκτης Σύμπτωσης

Σε κείμενο X , όπου f_i το πλήθος εμφανίσεων του γράμματος i :

$$IC(X) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \sum_{i=0}^{25} \frac{f_i(f_i - 1)}{n(n - 1)}$$

Σημαντική ιδιότητα: **αναλλοίωτος** σε ολίσθηση του κειμένου κατά k .

Σε άγνωστο κείμενο αγγλικής X : $E[IC(X)] \cong \sum_{i=0}^{25} p_i^2 \cong 0.065$
(p_i : η στατιστική συχνότητα του γράμματος i)

Σε εντελώς τυχαίο κείμενο με αγγλικούς χαρακτήρες:

$$E[IC(X)] \cong \sum_{i=0}^{25} \left(\frac{1}{26}\right)^2 = \frac{1}{26} \cong 0.038$$

Μπορούμε με μεγάλη πιθανότητα να ξεχωρίσουμε ένα τυχαίο κείμενο με αγγλικούς χαρακτήρες από ένα κανονικό αγγλικό κείμενο.

Κρυπτανάλυση Vigenère

Μέθοδος για εύρεση r

Δοκιμή για $r = 1, 2, \dots$. Χωρίζουμε το κρυπτοκείμενο σε r στήλες:
στήλη $C_i = \{c_{i+jr} \mid 0 \leq j \leq \lceil \frac{n}{r} \rceil - 1\}$

Υπολογισμός $IC(C_i)$. Αν έχουμε βρει σωστό μήκος, τιμές κοντά στο **0.065**, αλλιώς συμπεριφορά τυχαίου κειμένου (συνήθως < 0.050 ακόμη και σε σχετικά μικρά κείμενα).

Κρυπτανάλυση Vigenère: εύρεση κλειδιού

- ▶ 1ος τρόπος: στατιστική κρυπτανάλυση στις στήλες με βάση τη συχνότητα εμφάνισης των γραμμάτων, διγραμμάτων, κ.λπ. της αγγλικής (ή γενικότερα της γλώσσας του αρχικού κειμένου).
- ▶ 2ος τρόπος: βρίσκουμε το σχετικό shift μεταξύ της πρώτης στήλης και της m -οστής στήλης (για $2 \leq m \leq r$). Έχοντας τα σχετικά shift της πρώτης στήλης με τις υπόλοιπες είμαστε ουσιαστικά αντιμέτωποι με μονοαλφαβητικό σύστημα.
 - ▶ Δοκιμάζουμε ολισθήσεις της πρώτης στήλης κατά $j = 1, 2, \dots, 25$.
 - ▶ Χρήση **δείκτη αμοιβαίας σύμπτωσης** μεταξύ της ολισθημένης πρώτης στήλης και της m -οστής στήλης.

Δείκτης Αμοιβαίας Σύμπτωσης (Index of Mutual Coincidence – IMC)

$$IMC(C_{1 \gg j}, C_m) = \sum_{i=0}^{25} \frac{f_{(1 \gg j)}(i) f_{(m)}(i)}{|C_1| |C_m|}$$

$f_{(1)}(i)$: # εμφανίσεων χαρακτήρα i στην στήλη 1.

$f_{(1 \gg j)}(i) = f_{(1)}((i - j) \bmod 26)$: # εμφανίσεων χαρακτήρα i στην στήλη 1, μετά από ολίσθηση της στήλης κατά j .

- ▶ Αντιστοιχεί στην πιθανότητα δύο τυχαίοι χαρακτήρες από δύο κείμενα να ταυτίζονται.
- ▶ Παρόμοιες ιδιότητες με Δείκτη Σύμπτωσης: η τιμή του **διαφέρει σημαντικά** μεταξύ αγγλικών κειμένων (ή προερχόμενων από αγγλικά κείμενα, με την ίδια ολίσθηση) και τυχαίων κειμένων (ή προερχόμενων από αγγλικό κείμενο, με διαφορετική ολίσθηση).

Μπορούμε να βελτιώσουμε το Vigenère;

- ▶ Αυξάνοντας το μήκος του κλειδιού;
- ▶ Ιδανικά: κλειδί ίσου μήκους με αρχικό κείμενο.
- ▶ Αυτή είναι ουσιαστικά μια μορφή του περίφημου **One Time Pad** (Vernam, 1917).

Τέλεια μυστικότητα (Shannon, 1949)

Ας θεωρήσουμε το αρχικό κείμενο M , το κλειδί K και το κρυπτοκείμενο C σαν τυχαίες μεταβλητές που παίρνουν τιμές αντίστοιχα από τα σύνολα $\mathcal{M}, \mathcal{K}, \mathcal{C}$. Οι M και K είναι ανεξάρτητες, ενώ η C εξαρτάται από τις άλλες δύο.

Ο ορισμός του Shannon

$$\forall x \in \mathcal{M}, y \in \mathcal{C} : \Pr_{M \in \mathcal{M}, K \in \mathcal{K}} [M = x \mid C = y] = \Pr_{M \in \mathcal{M}} [M = x]$$

Το κρυπτοκείμενο δεν παρέχει **καμμία πληροφορία** για το αρχικό κείμενο (*a posteriori* πληροφορία ίδια με την *a priori*).

Παράδειγμα

Έστω το παρακάτω κρυπτοσύστημα με
 $\mathcal{M} = \{0, 1\}$, $\mathcal{C} = \{A, B\}$, $\mathcal{K} = \{K_1, K_2\}$:

	K_1	K_2
0	A	B
1	B	A

$$\text{με } Pr[K_1] = \frac{1}{3}, \quad Pr[K_2] = \frac{2}{3}$$

Παράδειγμα

Έστω το παρακάτω κρυπτοσύστημα με
 $\mathcal{M} = \{0, 1\}$, $\mathcal{C} = \{A, B\}$, $\mathcal{K} = \{K_1, K_2\}$:

	K_1	K_2
0	A	B
1	B	A

με $Pr[K_1] = \frac{1}{3}$, $Pr[K_2] = \frac{2}{3}$

Έχει την ιδιότητα της τέλειας μυστικότητας;

Random SHIFT Cipher

Ορισμός

- ▶ $\mathcal{M} = \mathcal{K} = \mathcal{C} = \{0, \dots, 25\}$
- ▶ Κρυπτογράφηση: $C = enc(M, K) = M + K \bmod 26$
- ▶ Κατανομή $K \in \mathcal{K}$: $\Pr[K = i] = \frac{1}{26}$, $0 \leq i \leq 25$.

$$1. \forall y \in \mathcal{C} : \Pr[C = y] = \sum_{x \in \mathcal{M}} \Pr[M = x] \cdot \Pr[K = y - x \bmod 26] = \frac{1}{26} \sum_{x \in \mathcal{M}} \Pr[M = x] = \frac{1}{26}$$

$$2. \Pr[M = x \mid C = y] = \frac{\Pr[C=y|M=x] \Pr[M=x]}{\Pr[C=y]}$$

3. Από (1) και (2):

$$\forall x \in \mathcal{M}, y \in \mathcal{C} : \Pr[M = x \mid C = y] = \frac{\frac{1}{26} \Pr[M=x]}{\frac{1}{26}} = \Pr[M = x]$$

Τέλεια μυστικότητα! (η απόδειξη επεκτείνεται για οποιοδήποτε μέγεθος κειμένου).

Ισοδύναμες Συνθήκες Τέλειας Μυστικότητας

1. $\forall x \in \mathcal{M}, y \in \mathcal{C} : \Pr[C = y] = \Pr[C = y \mid M = x]$

δηλαδή, η πιθανότητα εμφάνισης ενός κρυπτοκειμένου είναι ανεξάρτητη από το αρχικό κείμενο.

2. $\forall x_1, x_2 \in \mathcal{M}, y \in \mathcal{C} : \Pr[C = y \mid M = x_1] = \Pr[C = y \mid M = x_2]$

(συνθήκη χρήσιμη για ανταπόδειξη)

Τέλεια μυστικότητα: μήκος κλειδιού \geq μήκος κειμένου

Αναγκαία συνθήκη για τέλεια μυστικότητα:

$$|\mathcal{M}| \leq |\mathcal{C}| \leq |\mathcal{K}|$$

- ▶ $|\mathcal{M}| \leq |\mathcal{C}|$: Από απαίτηση για κρυπτογράφηση ‘1-1’.
- ▶ $|\mathcal{C}| \leq |\mathcal{K}|$: Αν $|\mathcal{C}| > |\mathcal{K}|$,
 $\forall x \in \mathcal{M}, \exists y \in \mathcal{C}, \Pr[C = y \mid M = x] = 0 \neq \Pr[C = y]$.

Τέλεια μυστικότητα όταν $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}|$

Θεώρημα

Έστω κρυπτοσύστημα με $|\mathcal{M}| = |\mathcal{C}| = |\mathcal{K}|$. Το σύστημα έχει τέλεια μυστικότητα ανν ισχύουν τα εξής:

- (1) για κάθε $x \in \mathcal{M}, y \in \mathcal{C}$, υπάρχει μοναδικό $k \in \mathcal{K}$, ώστε $enc_k(x) = y$
- (2) κάθε κλειδί επιλέγεται με την ίδια πιθανότητα, συγκεκριμένα $1/|\mathcal{K}|$

Απόδειξη (συνοπτικά):

‘ \Rightarrow ’: Παραβίαση της (1) οδηγεί σε μηδενική δεσμευμένη πιθανότητα κάποιου y με δοσμένο x .

Από την (1) και αρχή Περιστερώνα και ιδιότητα ‘1-1’ της enc_{K_i} :

$$\forall y \in \mathcal{C}, k_1, k_2 \in \mathcal{K}, \exists x_1, x_2 \in \mathcal{M} : enc_{k_1}(x_1) = y, enc_{k_2}(x_2) = y$$

Με χρήση της δεύτερης Ισοδύναμης Συνθήκης προκύπτει ότι τα k_1, k_2 είναι ισοπίθανα.

‘ \Leftarrow ’: άμεση, με χρήση δεύτερης Ισοδύναμης Συνθήκης.

One Time Pad (Vernam, 1917)

Ορισμός

- ▶ Plaintext: $x = (x_0, x_1, \dots, x_{n-1})$, $x_i \in \{0, 1\}$
- ▶ Key: $k = (k_0, k_1, \dots, k_{n-1})$, $k_i \in \{0, 1\}$
- ▶ Ciphertext: $y = (y_0, y_1, \dots, y_{n-1})$, $y_i \in \{0, 1\}$
- ▶ Κρυπτογράφηση: $y_i = x_i \oplus k_i = x_i + k_i \bmod 2$
- ▶ Αποκρυπτογράφηση: $x_i = y_i \oplus k_i$

Ασφάλεια: αν για κάθε bit k_i του κλειδιού ισχύει

$\Pr[k_i = 0] = \Pr[k_i = 1] = 1/2$, τότε το κρυπτοσύστημα έχει τέλεια μυστικότητα (γιατί;).

Άσκηση: Ποιό πρόβλημα ασφάλειας εμφανίζεται αν χρησιμοποιήσουμε το κλειδί και δεύτερη φορά;

Πρώτα Συμπεράσματα

- ▶ Η τέλεια μυστικότητα είναι εφικτή.
- ▶ Η παραγωγή και η ανταλλαγή του κλειδιού όμως είναι πρακτικά ασύμφορες (τεράστιο μήκος, μία χρήση μόνο).
- ▶ Ενδιαφερόμαστε για *πρακτικά εφικτές* λύσεις.

Unicity Distance (Shannon, 1949)

- ▶ Είναι εφικτό να έχουμε ένα επίπεδο πληροφοριοθεωρητικής ασφάλειας, ακόμη και με μικρότερο κλειδί, αν το κλειδί είναι “αρκετά μεγάλο” σε σχέση με το κρυπτοκείμενο. Συγκεκριμένα:
- ▶ Σε ένα κρυπτοκείμενο c μπορεί να αντιστοιχούν τουλάχιστον δύο αρχικά κείμενα (άρα και αντίστοιχα κλειδιά). Ο κρυπταναλυτής χρειάζεται επιπλέον υποθέσεις.
- ▶ Τα μη γνήσια κλειδιά λέγονται *κίβδηλα* (spurious).
- ▶ **Unicity Distance**: το μήκος κειμένου πέρα από το οποίο “εξαφανίζονται” τα κίβδηλα κλειδιά.
- ▶ Παράδειγμα: στο (απλό) Shift Cipher, το ίδιο κρυπτοκείμενο **CTGPC** αντιστοιχεί στα αρχικά κείμενα **ARENA** και **RIVER** με διαφορετικό κλειδί (shift number). Αν αυξήσουμε το κρυπτοκείμενο, πιθανότατα μόνο ένα από τα δύο κλειδιά θα “επιβιώσει”.

Unicity Distance (Shannon, 1949)

- ▶ Η αναμενόμενη τιμή της μπορεί να υπολογιστεί με βάση την **εντροπία** του κλειδιού και τον **πλεονασμό (redundancy)** της φυσικής γλώσσας:

$$U = \frac{H(\mathcal{K})}{D} = \frac{\log(|\mathcal{K}|)}{D}$$

(για ισοπίθανα κλειδιά)

D : ο πλεονασμός της φυσικής γλώσσας, π.χ. για Αγγλικά
 $D \approx 3.2$ bits/character.

- ▶ Έτσι, για Αγγλικά και Shift Cipher, έχουμε $U \approx 2$ χαρακτήρες, για Vigenere $U \approx 1.47 \cdot m$ χαρακτήρες, με m το μήκος του κλειδιού. Για Substitution Cipher (κλειδιά είναι οι $26!$ μεταθέσεις του αλφαβήτου), έχουμε $U \approx 28$: αντιστοιχεί στην εμπειρική παρατήρηση ότι ένας έμπειρος κρυπτογράφος μπορεί να σπάσει το Substitution Cipher αν έχει περίπου 25 χαρακτήρες κρυπτοκειμένου.

- ▶ Τέλεια (πληροφοριοθεωρητική, information theoretic): ανεξάρτητη της ισχύος του αντιπάλου, καμμία νέα πληροφορία δεν μπορεί να προκύψει από την κρυπτανάλυση.
- ▶ Στατιστική: ανεξαρτήτως της ισχύος του αντιπάλου, η πιθανότητα αποκρυπτογράφησης είναι πολύ μικρή (αμελητέα).
- ▶ Υπολογιστική: οποιοσδήποτε αντίπαλος με “λογική” υπολογιστική ισχύ (συνήθως πολυωνυμικού χρόνου) έχει αμελητέα πιθανότητα να σπάσει το κρυπτοσύστημα.

Semantic Security

- ▶ Είναι το αντίστοιχο της κατά Shannon τέλειας μυστικότητας, όταν ο αντίπαλος είναι πολυωνυμικά φραγμένος.
- ▶ Ο αντίπαλος δεν μπορεί *αποδοτικά* να μάθει τίποτε χρήσιμο από το κρυπτοκείμενο παρά μόνο με αμελητέα πιθανότητα.
- ▶ Εάν διαθέτει δύο αρχικά κείμενα, και του δώσουν το κρυπτοκείμενο ενός από αυτά, δεν μπορεί αποδοτικά να βρει ποιο είναι με πιθανότητα σημαντικά μεγαλύτερη του $1/2$.
- ▶ Για κρυπτογραφία δημοσίου κλειδιού αυτό ισοδυναμεί με **IND-CPA** ασφάλεια – προϋποθέτει χρήση τυχαιότητας.

Pseudorandom Generators

- ▶ Επιτρέπουν ένα μικρό τυχαίο κλειδί (seed) να δώσει ένα μεγάλο “ψευδοτυχαίο”, αρκετά τυχαίο για έναν πολυωνυμικά φραγμένο αντίπαλο.
- ▶ Το ψευδοτυχαίο κλειδί μπορεί να χρησιμοποιηθεί σαν κλειδί για το one-time pad (πράξη XOR).
- ▶ Παρεμφερής χρήση: σε κρυπτοσυστήματα ροής.
- ▶ Η ύπαρξη ψευδοτυχαίων γεννητριών σχετίζεται με την ύπαρξη *μονόδρομων συναρτήσεων (one-way functions)*.
- ▶ **RC4** (Rivest '87): μια σημαντική γεννήτρια / κρυπτοσύστημα ροής.

Η γεννήτρια ψευδοτυχαίων RC4

- ▶ Συστατικά: 2 arrays of bytes:
 - ▶ Μετάθεση $P[0..255]$. Αρχικοποίηση:
for all $i \in \mathbb{Z}_{256}$ **do** : $P[i] \leftarrow i$
 - ▶ Κλειδί $K[0..keylen - 1]$, $keylen \leq 256$ – συνήθως $keylen \in [5..8]$.
Επιλέγεται από χρήστη.
- ▶ Δημιουργία σειράς κλειδιών (key-scheduling algorithm – KSA)
Η αρχική (ταυτοτική) μετάθεση P μετατρέπεται μέσω μιας σειράς ανταλλαγών (swar) σε μια (φαινομενικά τυχαία) μετάθεση.
Το “ανακάτεμα” επηρεάζεται από το αρχικό κλειδί K .
- ▶ Παραγωγή ψευδοτυχαίων bytes (pseudorandom generation algorithm – PRGA)
Επαναληπτικός βρόχος. Σε κάθε επανάληψη επιλέγεται κάποιο byte της P ως κλειδί εξόδου με τρόπο που καθορίζεται από τα τρέχοντα περιεχόμενα της P .
Οι επαναλήψεις συνεχίζονται για όσο χρειάζεται (δηλ. μέχρι να τελειώσει το stream). Σε κάθε επανάληψη γίνεται και ένα νέο swar.

Η γεννήτρια ψευδοτυχαίων RC4

Περιγραφή KSA, PRGA

- ▶ Δημιουργία σειράς κλειδιών (KSA)

$j \leftarrow 0$

for all $i \in \mathbb{Z}_{256}$ **do** :

$j \leftarrow (j + P[i] + K[i \bmod \text{keylen}]) \bmod 256$

swap($P[i], P[j]$)

Η γεννήτρια ψευδοτυχαίων RC4

Περιγραφή KSA, PRGA

- ▶ Δημιουργία σειράς κλειδιών (KSA)

$j \leftarrow 0$

for all $i \in \mathbb{Z}_{256}$ **do** :

$j \leftarrow (j + P[i] + K[i \bmod \text{keylen}]) \bmod 256$

swap($P[i], P[j]$)

- ▶ Παραγωγή ψευδοτυχαίων bytes (PRGA)

$i \leftarrow 0; j \leftarrow 0$

while next key needed :

$i \leftarrow (i + 1) \bmod 256 ; j \leftarrow (j + P[i]) \bmod 256$

swap($P[i], P[j]$)

$K_o \leftarrow P[(P[i] + P[j]) \bmod 256]$

output K_o

Η γεννήτρια ψευδοτυχαίων RC4

Περιγραφή KSA, PRGA

- ▶ Δημιουργία σειράς κλειδιών (KSA)

$j \leftarrow 0$

for all $i \in \mathbb{Z}_{256}$ **do** :

$j \leftarrow (j + P[i] + K[i \bmod \text{keylen}]) \bmod 256$

swap($P[i], P[j]$)

- ▶ Παραγωγή ψευδοτυχαίων bytes (PRGA)

$i \leftarrow 0; j \leftarrow 0$

while next key needed :

$i \leftarrow (i + 1) \bmod 256 ; j \leftarrow (j + P[i]) \bmod 256$

swap($P[i], P[j]$)

$K_o \leftarrow P[(P[i] + P[j]) \bmod 256]$

output K_o

Κάθε κλειδί εξόδου K_o χρησιμοποιείται για την κρυπτογράφηση ενός byte αρχικού κειμένου.

Η γεννήτρια ψευδοτυχαίων RC4

Παρατηρήσεις

- ▶ Με ίδιο αρχικό κλειδί K προκύπτει η ίδια σειρά κλειδιών εξόδου.
- ▶ Απλή και γρήγορη στην υλοποίηση με software (σε αντίθεση με άλλα stream cipher, π.χ. αυτά που βασίζονται σε LFSRs).
- ▶ Χρήση σε πολύ διαδεδομένα πρωτόκολλα: TLS, WEP, WPA.
- ▶ Η ασφάλεια της γεννήτριας RC4 έχει αμφισβητηθεί έντονα. Κάποιοι τρόποι χρήσης ιδιαίτερα ανασφαλείς (π.χ. WEP) – επίθεση Fluhrer, Mantin, Shamir (2001).
- ▶ Άμυνα: απόρριψη αρχικού τμήματος κλειδοροής (RCA4-drop[n]), ενδεικτικά: $n = 768$ bytes, συστήνεται ακόμη και $n = 3072$.

‘Αποδεδειγμένα’ ασφαλείς γεννήτριες ψευδοτυχαίων

- ▶ RSA-based, BBS.
- ▶ Βασίζονται σε (γενικά παραδεκτές) αριθμοθεωρητικές μονόδρομες συναρτήσεις: ύψωση σε δύναμη modulo n , τετραγωνισμός modulo n .
- ▶ Λειτουργία: διαδοχικές εφαρμογές της συνάρτησης, έξοδος κάθε φορά το λιγότερο σημαντικό bit του αριθμού (ή κάποια από τα λιγότερο σημαντικά bit).
- ▶ Είναι ασφαλείς κάτω από την υπόθεση δυσκολίας αντιστροφής της αντίστοιχης συνάρτησης.
- ▶ Απαιτούν μεγαλύτερη υπολογιστική προσπάθεια.

Κρυπτοσυστήματα ροής / ρεύματος (stream ciphers)

Παραγωγή ακολουθίας κλειδιών με βάση κάποιο αρχικό κλειδί, και (πιθανά) το plaintext.

Ορισμός

- ▶ Plaintext: x_0, x_1, \dots, x_{n-1}
- ▶ Ciphertext: y_0, y_1, \dots, y_{n-1}
- ▶ Αρχικό κλειδί: k
- ▶ Βοηθητικές συναρτήσεις: $f_i, 0 \leq i < m$
- ▶ Key stream: $z_i = f_{i \bmod m}(k, x_0, \dots, x_{i-1}, z_0, \dots, z_{i-1})$
- ▶ Κρυπτογράφηση: $y_i = enc_{z_i}(x_i)$
- ▶ Αποκρυπτογράφηση: $x_i = dec_{z_i}(y_i)$

Π.χ. για δυαδικές ακολουθίες:

$$enc_z(x) = x \oplus z = x + z \bmod 2$$

$$dec_z(y) = y \oplus z = y + z \bmod 2$$

Κρυπτοσυστήματα ροής / ρεύματος (stream ciphers)

Διακρίνονται σε **synchronous** (το κλειδί δεν εξαρτάται από το plaintext), και **asynchronous** (λέγονται και **self-synchronizing**).

Επίσης σε **periodic** ($\forall i : z_{i+d} = z_i$, όπου d η περίοδος) και **aperiodic**.

Παράδειγμα: το Vigenère είναι synchronous και periodic.

Κρυπτοσυστήματα ροής: Linear Recurrence Keystream

Αρχικό διάνυσμα κλειδιών: $(z_0, z_1, \dots, z_{m-1})$.

Τα υπόλοιπα κλειδιά υπολογίζονται ως εξής:

$$z_{i+m} = \sum_{j=0}^{m-1} c_j \cdot z_{i+j} \pmod{2}, \quad \forall j, c_j \in \{0, 1\}$$

Εάν το πολυώνυμο $c_0 + c_1x + c_2x^2 + \dots + c_{m-1}x^{m-1} + x^m$ είναι **primitive**, τότε το κρυπτοσύστημα έχει περίοδο $d = 2^m - 1$.

Π.χ. $c_0 = c_1 = 1, c_2 = c_3 = 0$ ορίζουν το πολυώνυμο $x^4 + x + 1$, και με δεδομένο αρχικό κλειδί z_0, \dots, z_3 έχουμε $z_{4+i} = z_i + z_{i+1} \pmod{2}$.

Το κρυπτοσύστημα αυτό έχει περίοδο 15.

Υλοποίηση με **Linear Feedback Shift Register (LFSR)**.

Καταχωρητές Ολίσθησης Γραμμικής Ανάδρασης - LFSRs

- ▶ Δημιουργούν περιοδικές ακολουθίες, με περίοδο το πολύ $2^L - 1$, L το πλήθος των ψηφίων.
- ▶ Αν το αντίστοιχο πολυώνυμο είναι primitive έχουμε **maximum-length LFSR**. Πολλά γνωστά primitive πολυώνυμα.

Καταχωρητές Ολίσθησης Γραμμικής Ανάδρασης - LFSRs

- ▶ Δημιουργούν περιοδικές ακολουθίες, με περίοδο το πολύ $2^L - 1$, L το πλήθος των ψηφίων.
- ▶ Αν το αντίστοιχο πολυώνυμο είναι primitive έχουμε **maximum-length LFSR**. Πολλά γνωστά primitive πολυώνυμα.
- ▶ Σημαντικό μέγεθος για ακολουθίες: **γραμμική πολυπλοκότητα (linear complexity)**. Είναι το ελάχιστο μέγεθος LFSR που παράγει την ίδια ακολουθία.
- ▶ Αλγόριθμος Berlekamp-Massey: υπολογίζει τη γραμμική πολυπλοκότητα και τον αντίστοιχο LFSR.

Καταχωρητές Ολίσθησης Γραμμικής Ανάδρασης - LFSRs

- ▶ Δημιουργούν περιοδικές ακολουθίες, με περίοδο το πολύ $2^L - 1$, L το πλήθος των ψηφίων.
- ▶ Αν το αντίστοιχο πολυώνυμο είναι primitive έχουμε **maximum-length LFSR**. Πολλά γνωστά primitive πολυώνυμα.
- ▶ Σημαντικό μέγεθος για ακολουθίες: **γραμμική πολυπλοκότητα (linear complexity)**. Είναι το ελάχιστο μέγεθος LFSR που παράγει την ίδια ακολουθία.
- ▶ Αλγόριθμος Berlekamp-Massey: υπολογίζει τη γραμμική πολυπλοκότητα και τον αντίστοιχο LFSR.
- ▶ Αύξηση γραμμικής πολυπλοκότητας: χρήση περισσότερων LFSRs, συνδυασμός εξόδων με μη γραμμικό τρόπο. Π.χ. Geffe generator συνδυάζει 3 maximum-length LFSRs με μήκος L_1, L_2, L_3 και εξόδους x_1, x_2, x_3 :

$$f(x_1, x_2, x_3) = x_1x_2 \oplus (1 \oplus x_2)x_3$$

έχει περίοδο $(2^{L_1} - 1) \cdot (2^{L_2} - 1) \cdot (2^{L_3} - 1)$ και γραμμική πολυπλοκότητα $L = L_1L_2 + L_2L_3 + L_3$

Permutation (Transposition) Cipher

Το κλειδί, μήκους m , είναι μία μετάθεση (*permutation*) του $\{1, \dots, m\}$. Χωρίζουμε το αρχικό κείμενο σε μπλοκ μεγέθους m και σε κάθε μπλοκ εφαρμόζουμε την μετάθεση.

Σημαντικό πρόβλημα: το κρυπτοκείμενο περιέχει τους ίδιους χαρακτήρες με το αρχικό κείμενο. Αντιμετώπιση: *παρεμβολή σκουπιδιών*.

Κάποιες πληροφορίες μπορούν να βοηθήσουν σημαντικά στην κρυπτανάλυση. Παράδειγμα:

ECSEEMDR IAERFRR RITSADEM
ESCOBARA LACAILCD LESHYRCR

Άσκηση: ποιες ιδέες από τα προηγούμενα θα μπορούσαμε να χρησιμοποιήσουμε για κρυπτανάλυση του συστήματος αυτού;

Κρυπτοσυστήματα Γινομένου (Product Cryptosystems)

Προκύπτουν από σύνθεση των συναρτήσεων κρυπτογράφησης δύο ή περισσότερων κρυπτοσυστημάτων:

$$e_k(x) = e_{k_1}(e_{k_2}(x))$$

Συχνά δεν επιτυγχάνεται αύξηση της ασφάλειας.

Idempotent λέγονται τα κρυπτοσυστήματα που το γινόμενο με τον εαυτό τους δίνει το ίδιο κρυπτοσύστημα, π.χ. το Shift Cipher.

Άσκηση: δείξτε ότι το Affine Cipher είναι idempotent.

Ασύμμετρη κρυπτογραφία: κρυπτοσύστημα Σακιδίου Merkle-Hellman

Στηρίζεται σε μια ειδική περίπτωση του προβλήματος του Σακιδίου (Knapsack), συγκεκριμένα στο πρόβλημα Αθροίσματος Υποσυνόλων (Subset Sum).

Πρόβλημα Subset Sum

- ▶ Είσοδος: σύνολο $A = \{a_1, \dots, a_n\} \subseteq \mathbb{N}$, και $k \in \mathbb{N}$.
- ▶ Έξοδος: $A' \subseteq A$ τ.ώ. $\sum_{a_i \in A'} a_i = k$, εάν υπάρχει, αλλιώς 'No'.

Το πρόβλημα είναι **NP**-complete.

Ανήκει όμως στην κλάση **P**, αν το A είναι *υπεραυξητικό* (superincreasing): ταξινομημένο σύνολο όπου κάθε στοιχείο είναι μεγαλύτερο από το άθροισμα όλων των προηγούμενων. Π.χ.,

$$A = \{3, 7, 12, 25, 100, 211, 430\}$$

Περιγραφή του κρυπτοσυστήματος Σακιδίου (i)

Βασική ιδέα: το κρυπτογράφημα μιας δυαδικής ακολουθίας x_1, \dots, x_m μήκους $|A|$, προκύπτει από το άθροισμα $\sum a_i \cdot x_i$.

Π.χ. για το παραπάνω σύνολο,

$$Enc_A(0100110) = 7 + 100 + 211 = 381.$$

Περιγραφή του κρυπτοσυστήματος Σακιδίου (i)

Βασική ιδέα: το κρυπτογράφημα μιας δυαδικής ακολουθίας x_1, \dots, x_m μήκους $|A|$, προκύπτει από το άθροισμα $\sum a_i \cdot x_i$.

Π.χ. για το παραπάνω σύνολο,
 $Enc_A(0100110) = 7 + 100 + 211 = 381$.

Τι πρόβλημα έχει η παραπάνω ιδέα;

Περιγραφή του κρυπτοσυστήματος Σακιδίου (i)

Βασική ιδέα: το κρυπτογράφημα μιας δυαδικής ακολουθίας x_1, \dots, x_m μήκους $|A|$, προκύπτει από το άθροισμα $\sum a_i \cdot x_i$.

Π.χ. για το παραπάνω σύνολο,
 $Enc_A(0100110) = 7 + 100 + 211 = 381$.

Τι πρόβλημα έχει η παραπάνω ιδέα;

Βελτιωμένη ιδέα: Ο παραλήπτης Bob χρησιμοποιεί ως ιδιωτικό κλειδί ένα υπεραυξητικό σύνολο A , το οποίο “καμουφλάρει” σε A' ώστε να φαίνεται στον υπόλοιπο κόσμο σαν τυχαίο, προκειμένου να το χρησιμοποιήσει ως δημόσιο κλειδί. Για το σκοπό αυτό επιλέγει m, t τέτοια ώστε $m > \sum a_i, \gcd(t, m) = 1$:

$$A' = \{a'_i \mid a'_i = t \cdot a_i \bmod m\}$$

Περιγραφή του κρυπτοσυστήματος Σακιδίου (ii)

- ▶ Δημόσιο κλειδί: A'
- ▶ Ιδιωτικό κλειδί: $A, m, t^{-1} \bmod m$
- ▶ $Enc_{A'}(x) = \sum_{i=1}^n a'_i \cdot x_i$
- ▶ $Dec_{A,m,t^{-1}}(y) = Solve_A(t^{-1} \cdot y \bmod m)$
όπου $Solve_A(k)$ ένας αλγόριθμος που λύνει το πρόβλημα Subset Sum για είσοδο (A, k) .

Ορθότητα αποκρυπτογράφησης

Ο πολλαπλασιασμός του $y = \sum_{i=1}^n a'_i \cdot x_i$ με $t^{-1} \bmod m$ “βγάζει τη μάσκα” από τα a'_i :

$$\begin{aligned} Dec_{A,m,t^{-1}}(Enc_{A'}(x)) &= Solve(t^{-1}(\sum_{i=1}^n a'_i x_i) \bmod m) = \\ Solve_A(t^{-1}(\sum_{i=1}^n (t \cdot a_i \bmod m) x_i) \bmod m) &= Solve_A(\sum_{i=1}^n a_i x_i) \end{aligned}$$

Περιγραφή του κρυπτοσυστήματος Σακιδίου (iii)

Παράδειγμα

$$A = \{1, 3, 5, 11\}, m = 23, t = 7.$$

$$\text{Ιδιωτικό κλειδί: } A, m, t^{-1} \bmod m = 10.$$

$$\text{Δημόσιο κλειδί: } A' = 7 \cdot A \bmod 23 = \{7, 21, 12, 8\}.$$

$$\text{Enc}_{A'}(0110) = 33.$$

$$\text{Dec}_{A,23,10}(33) = \text{Solve}_A(10 \cdot 33 \bmod 23) = \text{Solve}_{\{1,3,5,11\}}(8) = 0110$$

Επίθεση Shamir

- ▶ Βασική ιδέα: Αν μπορούμε να βρούμε ti^* , m^* τ.ώ. το $A'' = (t^*)^{-1} \cdot A' \bmod m^*$ να είναι υπεραυξητικό τότε η αποκρυπτογράφηση $Dec_{A'', m^*, (t^*)^{-1}}$ θα δώσει το ίδιο αποτέλεσμα με την $Dec_{A, m, t^{-1}}$!
- ▶ Παράδειγμα: για $t^* = 7$, $m^* = 15$, έχουμε $(t^*)^{-1} \equiv 13 \pmod{15}$, και $A'' = 13 \cdot A' \bmod 15 = \{1, 3, 6, 14\}$: υπεραυξητικό.

Επίθεση Shamir

- ▶ Βασική ιδέα: Αν μπορούμε να βρούμε ti^*, m^* τ.ώ. το $A'' = (t^*)^{-1} \cdot A' \bmod m^*$ να είναι υπεραυξητικό τότε η αποκρυπτογράφηση $Dec_{A'', m^*, (t^*)^{-1}}$ θα δώσει το ίδιο αποτέλεσμα με την $Dec_{A, m, t^{-1}}$!
- ▶ Παράδειγμα: για $t^* = 7, m^* = 15$, έχουμε $(t^*)^{-1} \equiv 13 \pmod{15}$, και $A'' = 13 \cdot A' \bmod 15 = \{1, 3, 6, 14\}$: υπεραυξητικό.

$$Dec_{A'', 15, 13}(33) = Solve_{A''}(13 \cdot 33 \bmod 15) =$$
$$Solve_{\{1, 3, 6, 14\}}(9) = 0110$$

Επίθεση Shamir

- ▶ Βασική ιδέα: Αν μπορούμε να βρούμε t^* , m^* τ.ώ. το $A'' = (t^*)^{-1} \cdot A' \bmod m^*$ να είναι υπεραυξητικό τότε η αποκρυπτογράφηση $Dec_{A'', m^*, (t^*)^{-1}}$ θα δώσει το ίδιο αποτέλεσμα με την $Dec_{A, m, t^{-1}}$!
- ▶ Παράδειγμα: για $t^* = 7$, $m^* = 15$, έχουμε $(t^*)^{-1} \equiv 13 \pmod{15}$, και $A'' = 13 \cdot A' \bmod 15 = \{1, 3, 6, 14\}$: υπεραυξητικό.
 $Dec_{A'', 15, 13}(33) = Solve_{A''}(13 \cdot 33 \bmod 15) =$
 $Solve_{\{1, 3, 6, 14\}}(9) = 0110$
- ▶ Ο Shamir (1984) έδειξε επιπλέον ότι αυτή η επίθεση μπορεί να γίνει γρήγορα.
- ▶ Ένα χρήσιμο συμπέρασμα: η χρήση ενός υπολογιστικά δύσκολου προβλήματος δεν αρκεί από μόνη της.

Επίθεση Shamir

- ▶ Βασική ιδέα: Αν μπορούμε να βρούμε t^* , m^* τ.ώ. το $A'' = (t^*)^{-1} \cdot A' \bmod m^*$ να είναι υπεραυξητικό τότε η αποκρυπτογράφηση $Dec_{A'', m^*, (t^*)^{-1}}$ θα δώσει το ίδιο αποτέλεσμα με την $Dec_{A, m, t^{-1}}$!
- ▶ Παράδειγμα: για $t^* = 7$, $m^* = 15$, έχουμε $(t^*)^{-1} \equiv 13 \pmod{15}$, και $A'' = 13 \cdot A' \bmod 15 = \{1, 3, 6, 14\}$: υπεραυξητικό.
 $Dec_{A'', 15, 13}(33) = Solve_{A''}(13 \cdot 33 \bmod 15) =$
 $Solve_{\{1, 3, 6, 14\}}(9) = 0110$
- ▶ Ο Shamir (1984) έδειξε επιπλέον ότι αυτή η επίθεση μπορεί να γίνει γρήγορα.
- ▶ Ένα χρήσιμο συμπέρασμα: η χρήση ενός υπολογιστικά δύσκολου προβλήματος δεν αρκεί από μόνη της.
- ▶ Άσκηση: δουλεύει η επίθεση του Shamir για οποιαδήποτε t^* , m^* τ.ώ. το $A'' = (t^*)^{-1} \cdot A' \bmod m^*$ να είναι υπεραυξητικό;

- ▶ Η πληροφοριοθεωρητική (τέλεια) μυστικότητα είναι μεν εφικτή αλλά πρακτικά ασύμφορη.
- ▶ Επιπλέον, αφορά μόνο σε επιθέσεις τύπου Ciphertext Only (CO).
- ▶ Σύγχρονη τάση: υπολογιστική ασφάλεια, ισχυρή απέναντι και σε πιο προηγμένες επιθέσεις: KPA, CPA, CCA.
- ▶ Απαραίτητη η μαθηματική τεκμηρίωση. Εργαλεία: γραμμική άλγεβρα, θεωρία πιθανοτήτων, στατιστική, αφηρημένη άλγεβρα (θεωρία ομάδων), θεωρία αριθμών, υπολογιστική πολυπλοκότητα.
- ▶ Κεντρικό ρόλο παίζει η (εκτιμώμενη) **υπολογιστική δυσκολία** αριθμοθεωρητικών και αλγεβρικών προβλημάτων και μάλιστα στην **μέση περίπτωση**.