Κρυπτογραφία

Κρυπτοσυστήματα ροής

Πέτρος Ποτίκας

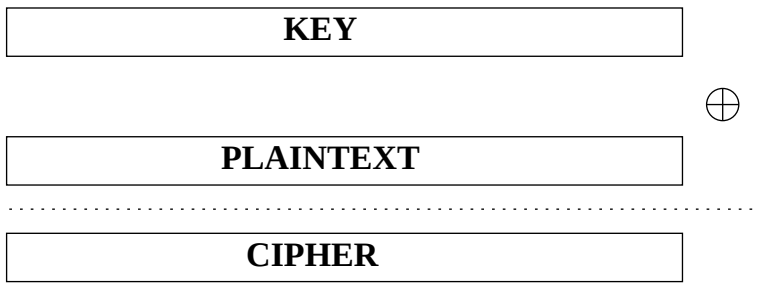
Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Περιεχόμενα

Εισαγωγή

- Ένα καλό σύστημα κρυπτογράφησης: One Time Pad (OTP): $|M| = |K| = |C| = \{0,1\}^n$,

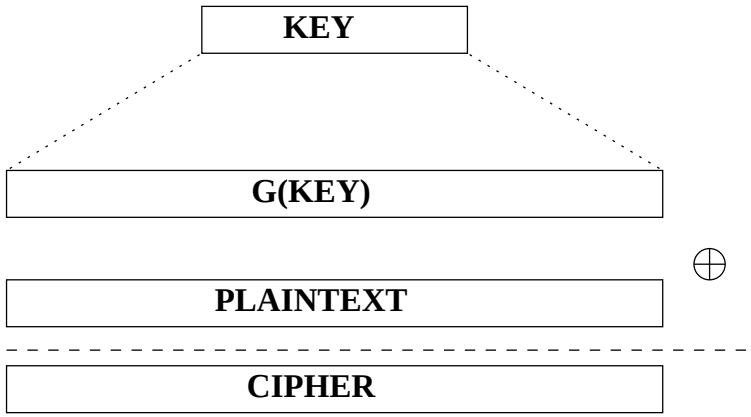
$$Enc_k(m) = k \oplus m, Dec_k(c) = k \oplus c$$



- Το OTP έχει τέλεια μυστικότητα, αλλά πρέπει $|key| \geq |plaintext|$
- Μη ρεαλιστικό!

Κρυπτοσυστήματα ροής

- Ιδέα: από ένα μικρό, πραγματικά "τυχαίο" κλειδί, σπόρο (seed) φτιάχνω ένα μεγάλο, "ψευδοτυχαίο" κλειδί και έτσι κρυπτογραφώ μεγάλου μεγέθους δεδομένα:



$$c = Enc_k(m) = m \oplus G(k)$$

$$m = Dec_k(c) = c \oplus G(k)$$

- Πώς; Με *γεννήτριες ψευδοτυχαιότητας*

Υπολογιστική ασφάλεια

Τι είναι ασφαλές;

Σύγχρονη προσέγγιση: Υπολογιστική μυστικότητα έναντι τέλειας μυστικότητας

Χαλάρωση απαιτήσεων: αντίπαλος πολυωνυμικά περιορισμένος και με μικρή πιθανότητα "σπάει" το σύστημα

1. Η μυστικότητα διατηρείται για *αποδοτικούς* αντιπάλους που διαθέτουν εφικτό χρόνο
2. Οι αντίπαλοι μπορούν να "πετύχουν" μόνο με *πολύ μικρή πιθανότητα* (σχεδόν απίθανο να συμβεί)

Στηρίζεται σε μη αποδεδειγμένες υποθέσεις (προβλήματα για τα οποία δεν έχουμε αποδοτικούς αλγορίθμους ... μέχρι σήμερα π.χ. διακριτός λογάριθμος, παραγοντοποίηση ακεραίων σε πρώτους αριθμούς)

Παίκτες και αντίπαλος χρησιμοποιούν πιθανοτικούς αλγορίθμους πολυωνυμικού χρόνου ως προς την παράμετρο ασφαλείας n (οι αντίπαλοι έχουν πολύ μεγαλύτερη δύναμη)

Αντίπαλος μπορεί να πετύχει με "μικρή πιθανότητα" (αμελητέα συνάρτηση: για κάθε σταθερά c, η πιθανότητα επιτυχίας του αντιπάλου μικρότερη από n^{-c}, για αρκετά μεγάλες τιμές του n)

Κρυπτογραφία ιδιωτικού κλειδιού

Ορισμός

Ένα *σχήμα κρυπτογράφησης ιδιωτικού-κλειδιού (private-key encryprtion scheme)* είναι μια πλειάδα πιθανοτικών αλγορίθμων πολυωνυμικού χρόνου (Gen, Enc, Dec) έτσι ώστε:

1. Gen ο αλγόριθμος παραγωγής κλειδιού: $k \leftarrow Gen(1^n)$, όπου 1^n η παράμετρος ασφάλειας.
2. Enc ο αλγόριθμος κρυπτογράφησης: $c \leftarrow Enc_k(m)$, όπου $m \in \{0,1\}^*$
3. Dec ο αλγόριθμος αποκρυπτογράφησης Dec: $m = Dec_k(c)$

Για κάθε n, κάθε κλειδί $k \leftarrow Gen(1^n)$ και κάθε $m \in \{0,1\}^*$, πρέπει $Dec_k(Enc_k(m)) = m$.

Οι Gen, Enc, Dec είναι γνωστοί, μυστικό μόνο το κλειδί.

Γεννήτορας ψευδοτυχαίων συμβολοσειρών διαισθητικά

Ιδέα: κάτι που “μοιάζει” με τυχαίο, αλλά δεν είναι πραγματικά

Δε ξεχωρίζει ένα τυχαίο string από ένα που δημιουργείται από τη γεννήτρια ψευδοτυχαιότητας

Εφαρμογή ψευδοτυχαιότητας και αλλού όπως π.χ. παίγνια, δειγματοληψία

Θα την χρησιμοποιήσουμε για να αποδείξουμε την ασφάλεια σχημάτων κρυπτογράφησης ιδιωτικού κλειδιού

Δημιουργία τυχαιότητας

- υλικό, φυσικά φαινόμενα π.χ. θερμικός ή ηλεκτρικός θόρυβος
- λογισμικό π.χ. πάτημα πλήκτρων πληκτρολογίου, κίνηση του ποντικιού

Γενικού σκοπού γεννήτριες τυχαίων αριθμών, μη κατάλληλες για κρυπτογραφία π.χ. random() της C.

Μια κατανομή πιθανοτήτων $\mathcal{D}$ είναι ψευδοτυχαία αν κανένας διαχωριστής πολυωνυμικού χρόνου δεν μπορεί να καταλάβει αν του έχει δοθεί ένα δείγμα της $\mathcal{D}$ ή από ένα string επιλεγμένο ομοιόμορφα τυχαία.

Ορισμός

Έστω $l(\cdot)$ ένα πολυώνυμο και G ένας ντετερμινιστικός αλγόριθμος πολυωνυμικού χρόνου τέτοιος ώστε για κάθε είσοδο $s \in \{0,1\}^n$, ο αλγόριθμος δίνει έξοδο μια συμβολοσειρά μήκους $l(n)$. Λέμε ότι ο G είναι *γεννήτορας ψευδοτυχαιότητας* αν ισχύουν τα ακόλουθα:

1. (Επέκταση) Για κάθε n ισχύει $l(n) > n$
2. (Ψευδοτυχαιότητα) Για όλους τους πιθανοτικούς διαχωριστές πολυωνυμικού χρόνου D, υπάρχει μια αμελητέα συνάρτηση $negl$ έτσι ώστε:

$$|Pr_{r \xleftarrow{R} \{0,1\}^{l(n)}}[D(r) = 1] - Pr_{s \xleftarrow{R} \{0,1\}^n}[D(G(s)) = 1]| \leq negl(n)$$

l: παράγοντας επέκτασης

Παρατηρήσεις: ντετερμινιστικός και αποδοτικός (πολυωνυμικός) αλγόριθμος

Είναι τυχαίο; Καθόλου!

αν $l(n) = 2n$, τότε η ομοιόμορφη κατανομή στο $\{0,1\}^{2n}$, έχει χώρο 2^{2n}, άρα η πιθανότητα να επιλέγει μια συμβολοσειρά είναι 2^{-2n},

$|dom(G)| = 2^n$, $|range(G)| = 2^{2n}$, άρα πιθανότητα μια συμβολοσειρά μήκους $2n$ να επιλέγει είναι $2^n/2^{2n} = 2^{-n}$

Αν ο διαχωριστής είναι εκθετικού χρόνου, τότε με εξαντλητική αναζήτηση μπορεί να ξεχωρίσει ένα ψευδοτυχαίο string από ένα τυχαίο

Ο σπόρος πρέπει να μείνει μυστικός και αρκετά μεγάλος, ώστε να μη μπορεί ο διαχωριστής να δοκιμάσει όλους τους δυνατούς σπόρους (αύξηση του μήκους κλειδιού, αν χρειαστεί)

Υπάρχουν αποδεδειγμένα ασφαλείς γεννήτριες ψευδοτυχαιότητας; Άγνωστο. Υπάρχουν όμως υποψήφιες

Στηρίζεται στην υπόθεση ύπαρξης συναρτήσεων μονής κατεύθυνσης (one-way functions)

Ισχύει: G γεννήτρια ψευδοτυχαιότητας ανν G μη προβλέψιμη

Ορισμός

(Μη προβλέψιμη) Υπάρχει πολυωνυμικός αλγόριθμος Α τέτοιος ώστε:

$$Pr[A(G(K)_{1..i}) = G(K)_{i+1}] > \frac{1}{2} + \epsilon$$

για μη αμελητέο ϵ

Ασφαλές σχήμα κρυπτογράφησης

Ορισμός

Έστω G ένας ψευδοτυχαίος γεννήτορας με παράγοντα επέκτασης l έτσι ώστε:

1. Gen: $k \xleftarrow{R} \{0,1\}^n$
2. Enc: $c = m \oplus Gen(k)$, όπου $m \in \{0,1\}^{l(n)}$
3. Dec: $m = c \oplus Gen(k)$

Ντετερμινιστικοί Enc, Dec

Με βάση τη μη διακρισιμότητα και το IND-EAV της προηγούμενης διάλεξης, έχουμε:

Θεώρημα

Αν G είναι ένας γεννήτορας ψευδοτυχαιότητας, τότε το παραπάνω σχήμα κρυπτογράφησης έχει μη διακρίσιμες κρυπτογραφήσεις στο μοντέλο παθητικού αντιπάλου (IND-EAV).

Απόδειξη.

(Ιδέα) Με αναγωγή: Απόδειξη βασισμένη στην υπόθεση της ασφάλειας του γεννήτορα. Υποθέτουμε ότι έχουμε αντίπαλο $\mathcal{A}$ ο οποίος διακρίνει τις κρυπτογραφήσεις. Χρησιμοποιώντας τον $\mathcal{A}$ μπορούμε να διακρίνουμε την έξοδο του G από ένα πραγματικά τυχαίο. Το συγκρίνουμε με το OTP. Καταλήγουμε σε άτοπο. □

Πιθανοτική κρυπτογράφηση

Μη ασφαλές για πολλαπλά μηνύματα: Two Time Pad

Ανάγκη για πιθανοτική κρυπτογράφηση

Θεώρημα

Έστω $\Pi = (Gen, Enc, Dec)$ *ένα σχήμα κρυπτογράφησης όπου* Enc *είναι ντετερμινιστικό. Τότε το* Π *δεν έχει μη διακρίσιμες πολλαπλές κρυπτογραφήσεις στο μοντέλο παθητικού αντιπάλου.*

Απόδειξη.

$\mathcal{A}$ στέλνει τα $\vec{M}_0 = (0^n, 0^n)$ και $\vec{M}_1 = (0^n, 1^n)$ και παίρνει $\vec{C} = (c^1, c^2)$ □

Λανθασμένη χρήση στα MS Word, Excel (το ίδιο διάνυσμα αρχικοποίησης/κλειδί χρησιμοποιείται όταν τροποποιείται ένα αρχείο)

Blum-Blum-Shub (1986)

Αλγόριθμος

Πάρε δύο μεγάλους πρώτους p, q, με $p \equiv q \equiv 3(\mod 4)$, και θέσε $n = pq$.

Επίλεξε τυχαία ένα s_0 σχετικά πρώτο με το n.

Για $1 \leq i \leq l$ όρισε

$$z_{i+1} = (s_0^{2^i} \mod n) \mod 2$$

Παρατήρηση: σχετικά αργό, αλλά ασφαλές με την υπόθεση ότι η παραγοντοποίηση του n σε πρώτους παράγοντες είναι δύσκολη.

Παράδειγμα BBS

Έστω $n = 192649 = 383 * 503$ και $s_0 = 101355^2 \mod n = 20749$. Τα πρώτα 5 bits που παράγονται από τον BBS είναι

$$11001$$

και προκύπτουν:

i	s_i	z_i
0	20749	
1	143135	1
2	177671	1
3	97048	0
4	89992	0
5	174051	1

Η γεννήτρια ψευδοτυχαίων RC4

- Συστατικά: 2 arrays of bytes:
  - Μετάθεση $P[0..255]$. Αρχικοποίηση:
    for all $i \in \{0..255\}$ **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 (swap) 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 swap.

Η γεννήτρια ψευδοτυχαίων RC4

Περιγραφή KSA, PRGA

- Δημιουργία σειράς κλειδιών (KSA)

  $j \leftarrow 0$
  for $i \leftarrow 0$ **to** 255 **do** :
  $j \leftarrow (j + P[i] + K[i \bmod keylen]) \bmod 256$
  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: TSL, 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 $n = 3072$.

Πραγματικά συστήματα

LFSR (linear feedback shift register): εύκολο στο hardware, κακό γιατί είναι γραμμικό

Χρήση:

1. DVD κρυπτογράφηση (CSS): 2 LFSRs
2. GSM (A5/1,2): 3 LFSRs
3. Bluetooth (E0): 4 LFSRs

Μοντέρνα κρυπτοσυστήματα ροής

Μοντέρνα κρυπτοσυστήματα ροής: eStream (2008)

$$G : \{0,1\}^s \times R \to \{0,1\}^n$$

όπου το R: nonce, δεν επαναλαμβάνεται για το ίδιο κλειδί

$$Enc_k(m, k; r) = m \oplus G(k, r)$$

Ιδέα: επαναχρησιμοποίηση ίδιου κλειδιού k καθώς το (k, r) αλλάζει

- Salsa20, Sosemanuk