Το κρυπτοσύστημα RSA

Παναγιώτης Γροντάς
ΕΜΠ

Κρυπτογραφία

2025 – 2026

Με βάση παλιότερες διαφάνειες Άρη Παγουρτζή

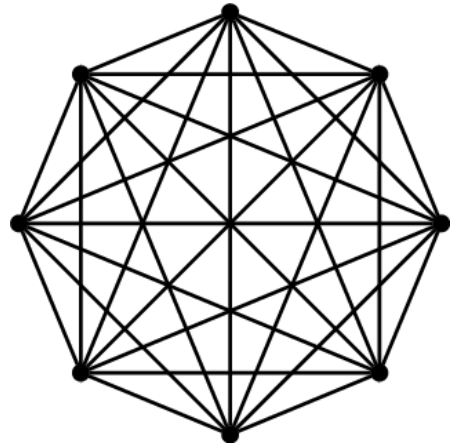
1. Κρυπτογραφία Δημοσίου Κλειδιού
2. Ορισμός RSA
3. Ασφάλεια RSA
4. Μοντελοποίηση- Ιδιότητες Ασφάλειας
5. Παραλλαγές με ισχυρότερη ασφάλεια

Περιεχόμενα

Κρυπτογραφία Δημοσίου Κλειδιού

Διαχείριση και διανομή κλειδιών

- Διαφορετικά κλειδιά ανά ζεύγος χρηστών
  - Για n χρήστες χρειάζονται $\frac{n(n-1)}{2}$ κλειδιά.
- Πρέπει να υπάρξει ‘συνάντηση’ για ανταλλαγή κλειδιών
- Εύκολο σε κλειστά περιβάλλοντα – δύσκολο σε ανοιχτά.
- Κύκλος ζωής κλειδιών
  - π.χ. έκδοση νέων
- Δυσκολίες αποθήκευσης



Μειονεκτήματα Συμμετρικών Κρυπτοσυστημάτων

Η λύση μετά από 2500 χρόνια προσπαθειών:

Whitfield Diffie, Martin Hellman New Directions in Cryptography - (1976) με σημαντική βοήθεια από: *Ralph Merkle*

Ίσως και νωρίτερα (σύμφωνα με αποχαρακτηρισμένα έγγραφα):

- James H. Ellis (1970- GHCQ)- no secret encryption
- Clifford Cocks (1973- GHCQ)- RSA
- Malcolm J. Williamson (1974- GHCQ)- Diffie Hellman Key Exchange

Δεν εφαρμόστηκαν λόγω της **κλειστής** φύσης των στρατιωτικών εφαρμογών

Κρυπτοσυστήματα Δημοσίου Κλειδιού

Κρυπτοσυστήματα Δημοσίου Κλειδιού

Αναλογία: Κλειδωμένο γραμματοκιβώτιο με σχισμή ή κάλπη

- οποιοσδήποτε μπορεί να εισάγει ένα γράμμα (κρυπτογράφηση- δημόσια λειτουργία)
- για άνοιγμα (αποκρυπτογράφηση) χρειάζεται προσπάθεια από οποιονδήποτε τρίτο…
- *εκτός* από τον κάτοχο του κλειδιού (αποκρυπτογράφηση- ιδιωτική λειτουργία)

3 καινοτόμες ιδέες - 1 κατασκευή

1. Ανταλλαγή Κλειδιού Diffie- Hellman
    - Δημιουργία κοινού κλειδιού πάνω από δημόσιο-μη ασφαλές κανάλι (online)
2. Κρυπτογραφία Δημοσίου Κλειδιού
    - Το κλειδί κρυπτογράφησης είναι δημόσιο
    - Το κλειδί αποκρυπτογράφησης είναι ιδιωτικό
    - n χρήστες, $2n$ κλειδιά - Εύκολη διανομή
3. Ψηφιακή Υπογραφή
    - Δημιουργία με ιδιωτικό
    - Επαλήθευση με δημόσιο κλειδί
    - Ακεραιότητα, Αυθεντικότητα, Μη Αποκήρυξη με ασύμμετρο τρόπο

Υλοποίηση μόνο για το 1

New directions in cryptography

Συναρτήσεις μονής κατεύθυνσης

Μία συνάρτηση f λέγεται **μονής κατεύθυνσης** εάν είναι εύκολο να υπολογιστεί το $f(x)$ δεδομένου του x, ενώ ο αντίστροφος υπολογισμός του x δεδομένου του $f(x)$ είναι απρόσιτος.

Trapdoor Functions

Μια συνάρτηση **μονής κατεύθυνσης** f για την οποία ο υπολογισμός της f^{-1} είναι εύκολος … όταν δίνεται μια μυστική πληροφορία (secret trapdoor)

Trapdoor functions

Επίλυση προβλήματος διανομής κλειδιού σε **αυθεντικοποιημένα** (authenticated) κανάλια

- Κρυμμένη υπόθεση: Ανήκει το pk σε αυτόν που νομίζουμε ότι ανήκει;
- Μετατροπή προβλήματος διανομής κλειδιού σε πρόβλημα αυθεντικότητας κλειδιού
- Λύση πάλι με ασύμμετρη κρυπτογραφία

Υπερβολικά αργή- πρόβλημα για μεγάλο όγκο δεδομένων σε μη ισχυρές συσκευές

- Hybrid encryption
- Διανομή συμμετρικού κλειδιού με ασύμμετρη κρυπτογραφία
- Κρυπτογράφηση μηνυμάτων με συμμετρική

Κριτική

RSA

Η πρώτη **κατασκευή** κρυπτοσυστήματος δημοσίου κλειδιού

- Ron **R**ivest,
- Adi **S**hamir,
- Leonard **A**dleman

Πατέντα μέχρι το 2000

RSA (1977)

- **Δημιουργία κλειδιών** $\boldsymbol{KGen}(1^{\lambda}) \rightarrow (\boldsymbol{pk}, \boldsymbol{sk})$
  - Επιλογή πρώτων p, q μηκους $\frac{\lambda}{2}$ bits
  - Υπολογισμός $n \leftarrow p \cdot q$
  - Επιλογή e: $1 < e < \phi(n)$ και $\gcd(e, \phi(n)) = 1$
  - Υπολογισμός d: $d \leftarrow e^{-1} (mod\ \phi(n))$
  - Δημόσιο κλειδί $pk = (e, n)$ ιδιωτικό κλειδί $\boldsymbol{sk} = (d, (p, q))$
- **Κρυπτογράφηση** $\mathbb{Z}_n^* \rightarrow \mathbb{Z}_n^*$
  - $Enc_{e,n}(m) \rightarrow m^e mod\ n$
- **Αποκρυπτογράφηση** $\mathbb{Z}_n^* \rightarrow \mathbb{Z}_n^*$
  - $Dec_d(c) \rightarrow c^d\ mod\ n$

$\mathbb{Z}_n^*: \{x \in \mathbb{Z}_n | \gcd(x, n) = 1\}$

Textbook RSA (1977)

KGen

$$(p, q) = (17,23)$$
$$n = 391$$
$$\phi(n) = 352$$
$$e = 3$$
$$d = 235$$

Enc

$$Enc_3(158) = 158^3 \; mod \; 391 = 295$$

Dec

$$Dec_{235}(295) = 295^{235} \; mod \; 391 = 158$$

Ένα παράδειγμα

Πρέπει να δείξουμε ότι:

$$\forall m \in \mathbb{Z}_n^*: Dec_d\big(Enc_e(m)\big) = m, (e, d) \leftarrow KGen()$$

Πράγματι:

$$Dec_d\big(Enc_e(m)\big) = Dec_d(m^e) = m^{ed} =$$

$$m^{k\phi(n)+1} = m^{k\phi(n)} \cdot m = m \ (mod\ n)$$

λόγω Θ. Euler και αφού $m \in \mathbb{Z}_n^*$.

Όμως δεν απαιτείται $m \in \mathbb{Z}_n^*$. Το RSA δουλεύει ακόμα και αν $m \in \mathbb{Z}_n$.

Ορθότητα

Το RSA δουλεύει ακόμα και αν $m \in \mathbb{Z}_n$

Αφού $m \in \mathbb{Z}_n$ ισχύει ότι $\gcd(m, n) \in \{p, q\}$.

Αν $\gcd(m, n) = p$ τότε:

$$m = kp \Rightarrow m^{ed} = 0 \ (mod\ p)$$

Επιπλέον:

$$m^{ed} = m \cdot m^{ed-1} = m \cdot m^{k\varphi(n)} = m \cdot m^{k(p-1)(q-1)} = m \cdot 1 \ (mod\ q)$$

Λόγω του Θ. Fermat - στο $\mathbb{Z}_q$.

Ομοίως και αν $\gcd(m, n) = q$

Ορθότητα

Συστάσεις για χρήση modulus:

- 2048 bits: βραχυχρόνια ασφάλεια (≈ 80 bit AES key)
- 3072bits: μακροχρόνια ασφάλεια (≈ 128 bit AES key)

RSA Factoring Challenge (RSA numbers – Wikipedia)

Παραγοντοποίηση Modulus 768bit (2009)

RSA-768 = *1 230 186 684 530 117 755 130 494 958 384 962 720 772 853 569 595 334 792 197 322 452 151 726 400 507 263 657 518 745 202 199 786 469 389 956 474 942 774 063 845 925 192 557 326 303 453 731 548 268 507 917 026 122 142 913 461 670 429 214 311 602 221 240 479 274 737 794 080 665 351 419 597 459 856 902 143 413* =

33 478 071 698 956 898 786 044 169 848 212 690 817 704 794 983 713 768 568 912 431 388 982 883 793 878 002 287 614 711 652 531 743 087 737 814 467 999 489 ×

36 746 043 666 799 590 428 244 633 799 627 952 632 279 158 164 343 087 642 676 032 283 815 739 666 511 279 233 373 417 143 396 810 270 092 798 736 308 917

Παράμετρος ασφάλειας

Παραγοντοποιήθηκε στις 2/12/2009 μετά από 10^{20} υπολογιστικά βήματα

Διάρκεια υπολογισμού:

2+ ημερολογιακά χρόνια χρησιμοποιώντας παράλληλη επεξεργασία **Εκτίμηση:**

2000 χρόνια σε single core system (2.2 GHz AMD Opteron με 2GB RAM) (Factorization of a 768-bit RSA modulus)

Επίσης έχει παραγοντοποιηθεί modulus 795 bits (`RSA240-2019`) και RSA modulus 829 bits (`RSA250-2020`)

Παράμετρος ασφάλειας

Επιλογή περιττού x ($\frac{\lambda}{2}$ bits)

- Εφαρμογή Primality test (Miller Rabin)
- Επανάληψη μέχρι επιτυχία:

$$\Pr[x\ \pi\rho\acute{\omega}\tau o\varsigma] \approx \frac{\frac{x}{lnx}}{\frac{x}{2}} = \frac{2}{lnx}$$

Δηλ. για $|p| = 1024$: $\Pr[x\ \pi\rho\acute{\omega}\tau o\varsigma] \approx \frac{2}{\ln 2^{1024}} = \frac{1}{512\ln 2} = \frac{1}{356}$

Συστάσεις:

- p, q ίδιου μήκους αλλά όχι πολύ κοντά
- p, q safe primes δηλ. $p - 1, q - 1$ έχουν μεγάλους πρώτους παράγοντες
- Αλλά και $p + 1, q + 1$ έχουν μεγάλους πρώτους παράγοντες
- Προσοχή στην τυχαιότητα: επιλογή p, q ομοιόμορφα και ανεξάρτητα

Λειτουργία $KGen$ – Δημιουργία πρώτων

Θέλουμε ταχύτατη κρυπτογράφηση

Εύκολος Υπολογισμός Δύναμης Με **Repeated Squaring (Square και Multiply)**

- Αναπαράσταση e στο δυαδικό
- Για κάθε 0 ύψωση στο τετράγωνο
- Για κάθε 1 ύψωση στο τετράγωνο και πολλαπλασιασμός

Τεράστια διαφορά πχ. υπολογισμός x^e για |e| = 1024 bits

Χωρίς **repeated squaring**: $2^{1024} \approx 10^{300}$ πολλαπλασιασμοί

Με **repeated squaring** περίπου: $1.5 \cdot 1024 = 1536$ πολλαπλασιασμοί

Ελαχιστοποίηση Πολλαπλασιασμών: Low Hamming Weight

- Μπορεί e να είναι πρώτος
- Ανεξάρτητη επιλογή από p, q - πάντα ίδιος
- Παράδειγμα: $e \in \{3, 17, 65537 = 2^{16} + 1\}$

Απόδοση

Πρόβλημα: Το κλειδί αποκρυπτογράφησης δεν μπορεί να είναι μικρό

- Επιθέσεις brute force
- Εξειδικευμένες επιθέσεις
- $|d| > \frac{\lambda}{3}$

Επιτάχυνση αποκρυπτογράφησης με ‘συνιστώσες’ CRT

- Υπολογισμός $c_p = c \ mod \ p, c_q = c \ mod \ q$
- Υπολογισμός $d_p = d \ mod \ (p - 1), d_q = d \ mod \ (q - 1),$
- Υπολογισμός $m_p = c_p^{d_p} \ mod \ p, m_q = c_q^{d_q} mod \ q$
- Συνδυασμός με CRT για m

Βελτίωση ταχύτητας:4 φορές

Απόδοση

Ασφάλεια RSA

Το πρόβλημα RSA (e-οστή ρίζα)

- Δίνεται $n = pq$, e με $(\gcd(e, \varphi(n)) = 1$ και $c \in \mathbb{Z}_n^*$
- Ζητείται $c^{e^{-1}=d} \; mod \; n$

Το πρόβλημα RSA-KINV

- Δίνεται $n = pq$, e με $(\gcd(e, \varphi(n)) = 1$ και $c \in \mathbb{Z}_n^*$
- Ζητείται $d = e^{-1} mod \; \varphi(n)$

Το πρόβλημα FACTORING

- Δίνεται $n = pq$
- Να βρεθούν τα p, q

Το πρόβλημα COMPUTE ϕ(n)

- Δίνεται $n = pq$
- Να βρεθεί το $\phi(n)$

Σχετιζόμενα δύσκολα προβλήματα

RSAP ≤ RSA-KINV

Αν μπορώ να βρω d μπορώ να κάνω κανονικά την αποκρυπτογράφηση $c^d \ mod \ n$

RSA-KINV ≤ FACTORING

Αν μπορώ να βρω p, q από n μπορώ να βρω με EGCD να βρω $\varphi(n)$ και κατά συνέπεια $d = e^{-1} \ (mod \ \varphi(n))$

FACTORING = COMPUTE-$\varphi(n)$

Αν μπορώ να λύσω **COMPUTE-$\varphi(n)$** τότε γνωρίζοντας $n, \phi(n)$ μπορώ να βρω p, q

Πώς:

$n = pq$ και

$\varphi(n) = (p-1)(q-1) = n - (p+q) + 1$

Από την εξίσωση

$$x^2 + \left(n + 1 - \phi(n)\right) + n = 0$$

μπορώ να βρω p, q.

Αντίστροφο προφανές.

RSAP ≤ RSA-KINV≤FACTORING = COMPUTE-$\varphi(n)$

Σχέσεις προβλημάτων

Αν γνωρίζουμε το $d = e^{-1}$ μπορούμε να κατασκευάσουμε αλγόριθμο παραγοντοποίησης του n με βάση τον Miller Rabin.

1. Υπολογίζουμε $s = ed - 1 = k\phi(n) = 2^t l$ με l μονό
2. Επιλέγουμε $\alpha \in \{2, \cdots, n-1\}$ με $\gcd(a, n) = 1$
3. Από Θ. Euler $a^s = 1 \ (mod\ n)$
4. Υπολογίζουμε $r_1 = a^{\frac{s}{2}}$.
5. Τότε $r_1^2 - 1 = (r_1 - 1)(r_1 + 1) = 0 \ (mod\ n)$
   - Αν $r_1 \neq \pm 1 \ (mod\ n)$ τότε $(p, q) = (\gcd(r_1 - 1, n), \gcd(r_1 + 1, n))$
   - Αλλιώς επαναλαμβάνουμε $r_2 = a^{\frac{s}{4}}, r_3 = a^{\frac{s}{8}}$, ... μέχρι να βρούμε $r_i \neq \pm 1 \ (mod\ n)$ (παραγοντοποιήσαμε το n) ή
   - το πολύ t φορές ($\frac{s}{2^t} \neq 0 \ (mod\ 2)$). Τότε προσπαθούμε με άλλο a.
   - Από θεωρία αριθμών: $\Pr_{\alpha}[success] \geq \frac{1}{2}$ και για m επαναλήψεις: $\Pr_{\alpha}[success] \geq 1 - \frac{1}{2^m}$

FACTORING$\leq^R$ RSA-KINV

Παράδειγμα:

$$n = 1441499, e = 17, d = 507905$$
$$s = e \cdot d - 1 = 8634385, a = 2$$

$$r_1 = a^{s/2} = 2^{4317192} = 1\ (mod\ n)$$
$$r_2 = a^{s/4} = 2^{2158596} = 1\ (mod\ n)$$
$$r_3 = a^{s/8} = 2^{1079298} = 119533\ (mod\ n)$$

$$p = \gcd(r_3 - 1, n) = \gcd(119532, 1441499) = 1423$$
$$q = \gcd(r_3 + 1, n) = n/p = \gcd(119534, 1441499) = 1013$$

FACTORING$\leq^R$ RSA-KINV

Αργότερα (May, 2004) $\boldsymbol{FACTORING \leq RSA - KINV}$

Τελικά:

$$\boldsymbol{RSAP \leq RSA - KINV = FACTORING = COMPUTE - \varphi(n)}$$

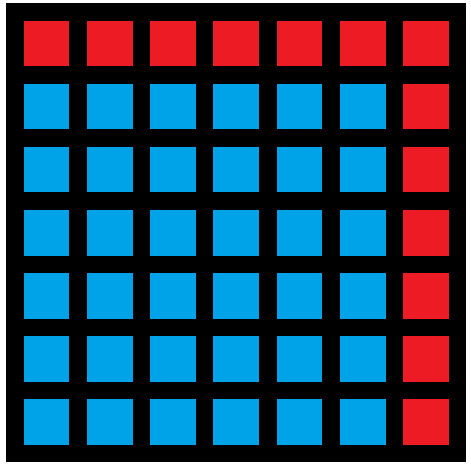
Το RSAP λοιπόν δεν είναι δυσκολότερο από το FACTORING

Υπόθεση RSA: Το RSAP είναι υπολογιστικά απρόσιτο.

Συνολική εικόνα προβλημάτων

Επιλογή κοντινών $p, q \approx \sqrt{n}$ επιτρέπει την παραγοντοποίηση του n:

- $\exists a, b : n = a^2 - b^2 = (a + b)(a - b) = p \cdot q$
- Άρα $b^2 = a^2 - n$ (ξέρουμε n)
- Αρχίζουμε να μαντεύουμε a από $a = \sqrt{n}$
- Υπολογίζουμε b^2 και ελέγχουμε αν είναι τέλειο τετράγωνο
- Επαναλαμβάνουμε με $a = a + 1$



Δυστυχώς πολλές υλοποιήσεις ακόμα και σήμερα είναι ευάλωτες - Fermat Attack on RSA

Παραγοντοποίηση Fermat για κοντινά p, q

Ιδέα: Χρήση κοινού n για να μειωθεί το κόστος πράξεων modulo

Σενάριο: Key Distribution Center διαθέτει $n = pq$ και μοιράζει στους χρήστες $\boldsymbol{\mathcal{A}}, B$ τα κλειδιά (e_A, d_A) και (e_B, d_B).

Εσωτερική Επίθεση (από γνώστη του $\boldsymbol{d_A}$)

- Ο $\boldsymbol{\mathcal{A}}$ αφού γνωρίζει το d_A μπορεί να παραγοντοποιήσει το n (αναγωγή FACTORING ≤r RSA-KINV)
- Υπολογισμός $\phi(n)$
- Ευρεση $d_B = e_B^{-1} \ (mod\ \phi(n))$ με EGCD
- Διάβασμα όλων των μηνυμάτων του B

Επίθεση κοινού γινομένου

Εξωτερική Επίθεση

Ο $\boldsymbol{\mathcal{A}}$ γνωρίζει $(n, e1), (n, e2)$

Μπορεί να ανακτήσει οποιοδήποτε m κρυπτογραφηθεί και με τα δύο δημόσια κλειδιά

Δηλ. ο $\boldsymbol{\mathcal{A}}$ διαθέτει c_1, c_2 με

$$c_1 = m^{e_1} \, mod \, n$$
$$c_2 = m^{e_2} \, mod \, n$$

Αν $gcd(e1, e2) = 1$ $(πολύ \, πιθανό)$ τότε με τον EGCD μπορούν να βρεθούν αποδοτικά t_1, t_2:

$$e_1 t_1 + e_2 t_2 = 1$$
$$c_1^{t_1} \, c_2^{t_2} = m^{e_1 t_1 + e_2 t_2} = m$$

Επίθεση κοινού γινομένου

Συλλογή δημοσίων κλειδιών (e_i, n_i)

- Υπολογισμός $\gcd(n_i, n_j) \forall (i, j)$
- Αν $\gcd(n_i, n_j) \neq 1$τότε (n_i, n_j) μπορούν να παραγοντοποιηθούν
- 2 στα 1000 πραγματικά δημόσια κλειδιά έχουν κοινό πρώτο
- Μάλλον οφείλεται σε πρόβλημα γεννήτριας τυχαίων πρώτων
- Ή λάθος παραμετροποίηση της συσκευής

Ron was wrong, Whit is right(2012)

Επίθεση μικρού δημόσιου εκθέτη

Κακή ιδέα: Χρήση $e = 3$ για να μειωθεί το κόστος κρυπτογράφησης
Τρία δημόσια κλειδιά $k_1 = (3, n_1), k_2 = (3, n_2), k_3 = (3, n_3)$

Ο $\mathcal{A}$ γνωρίζει 3 κρυπτογραφήσεις του μηνύματος-στόχου m

- $c_1 = Enc_{k_1}(m) = m^3 mod\ n_1$
- $c_2 = Enc_{k_2}(m) = m^3 mod\ n_2$
- $c_3 = Enc_{k_3}(m) = m^3 mod\ n_3$

Χρήση CRT για υπολογισμό του $m^3\ mod\ n_1 n_2 n_3$
Αλλά $m^3 < n_1 n_2 n_3$ αφού $m < n_1$ και $m < n_2$ και $m < n_3$
Εύρεση μηνύματος ως $m = \sqrt[3]{m^3}$

Παρατήρηση: Η επίθεση γενικεύεται και για $e > 3$ (Coppersmith)

ΘΕΩΡΙΑ – Αναπαράσταση με συνεχή κλάσματα

Έστω $x \in \mathbb{R}$. Υπάρχουν $\alpha_0, \alpha_1, \ldots \in \mathbb{Z}$ ώστε $x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cdots}}}$.

Συμβολίζουμε την αναπαράσταση με $[\alpha_0, \alpha_1, \ldots]$

Μια προσέγγιση του x μπορεί να δοθεί με ένα υποσύνολο των όρων.
Αν $x \in \mathbb{Q}$ τότε η αναπαράσταση είναι πεπερασμένη.

ΘΕΩΡΗΜΑ

Αν $\gcd(a, b) = 1$ και $\left|x - \frac{a}{b}\right| < \frac{1}{2b^2}$ τότε το κλάσμα $\frac{a}{b}$ εμφανίζεται στην προσέγγιση με συνεχή κλάσματα του x.

Αν το d ($3d < n^{\frac{1}{4}}$) είναι μικρό και $q < p < 2q$ τότε μπορούμε να το βρούμε μέσω της αναπαράστασής του με συνεχή κλάσματα.

Επίθεση μικρού ιδιωτικού εκθέτη

Αφού $n = pq > q^2$ έχουμε $q < \sqrt{n}$ και $(q < p < 2q)$

$$n - \phi(n) = pq - (p - q)(q - 1) = p + q - 1 < 3q < 3\sqrt{n} \ (1)$$

Ο $\mathcal{A}$ γνωρίζει το e και ότι $\exists k: ed = 1 + k\phi(n)$. Επίσης ισχύει ότι:

$$e < \phi(n) \Rightarrow ke < k\phi(n) < 1 + k\phi(n) = ed \Rightarrow k < d \ (2)$$

Επίσης έχουμε:

$$\left|\frac{e}{n} - \frac{k}{d}\right| = \left|\frac{ed - kn}{dn}\right| = \left|\frac{1 + k\phi(n) - kn}{dn}\right| = \left|\frac{1 - k(n - \phi(n))}{dn}\right| \leq$$

$$\frac{1 + k(n - \phi(n))}{dn} <^{(1)} \frac{3k\sqrt{n}}{dn} = \frac{3k}{d\sqrt{n}} <^{(2)} \frac{3}{\sqrt{n}}$$

Επίθεση μικρού ιδιωτικού εκθέτη

Από την υπόθεση για το μέγεθος του d έχουμε:

$$3d < n^{\frac{1}{4}} \Rightarrow d^2 < \frac{\sqrt{n}}{9} \Rightarrow 2d^2 < \frac{2\sqrt{n}}{9} < \frac{\sqrt{n}}{3} \Rightarrow \frac{3}{\sqrt{n}} < \frac{1}{2d^2}$$

Δηλαδή:

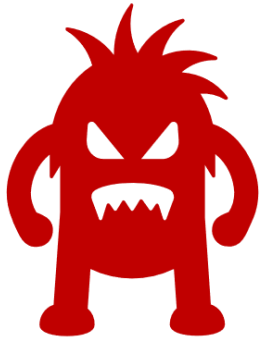
$$\left|\frac{e}{n} - \frac{k}{d}\right| < \frac{1}{2d^2}$$

Επειδή $\gcd(k, d) = 1$ το κλάσμα $\frac{k}{d}$ εμφανίζεται στην προσέγγιση του $\frac{e}{n}$ με συνεχή κλάσματα.

Πώς μπορεί να το εκμεταλλευτεί ο αντίπαλος;

Επίθεση μικρού ιδιωτικού εκθέτη

Η επίθεση



- Επιλογή μηνύματος m από τον $\mathcal{A}$ και κρυπτογράφηση σε c
  - Γνωστό δημόσιο κλειδί
- Κατασκευή αναπαράστασης του $\frac{e}{n}$ με συνεχή κλάσματα
- Ύψωση c σε κάθε έναν από τους παρονομαστές της
- Ο παρονομαστής που επιτυγχάνει σωστή αποκρυπτογράφηση είναι το d

Επίθεση μικρού ιδιωτικού εκθέτη

Παράδειγμα

Έστω $(e, n) = (207031, 242537)$

Η αναπαράσταση με συνεχή κλάσματα είναι $[0,1,5,1,4,1,10,2,5,1,2,1,4,2]$

Έστω $c = 46578 = 8^{207031} \; mod \; 242537.$

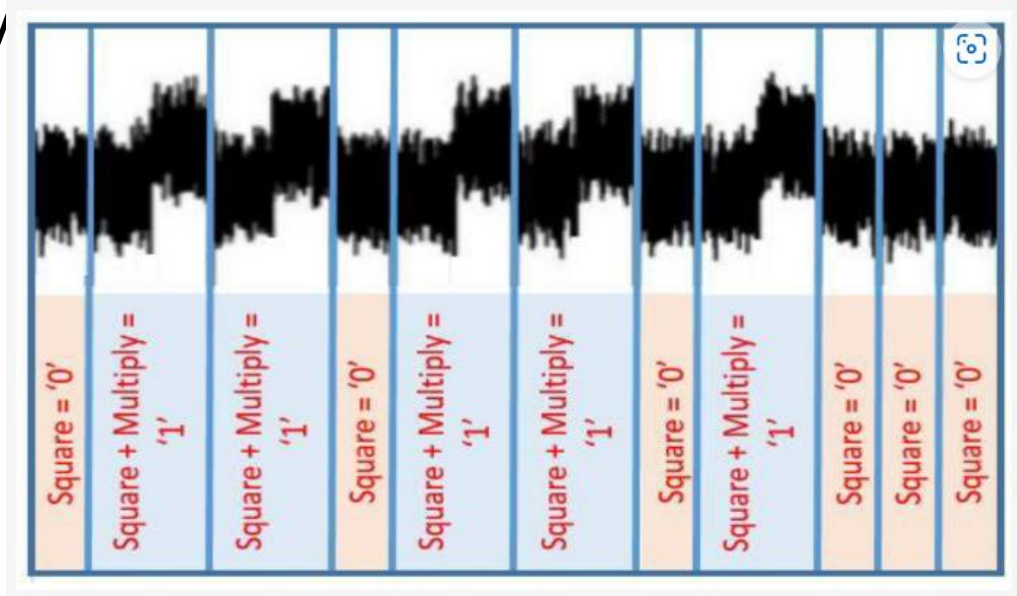
- $[0,1] = 0 + \frac{1}{1} = \frac{1}{1} \rightarrow 46578^{1} = 46578 \neq 8 \; (mod \; 242537)$
- $[0,1,5] = 0 + \frac{1}{1+\frac{1}{5}} = \frac{5}{6} \rightarrow 46578^{6} = 175938 \neq 8 \; (mod \; 242537)$
- $[0,1,5,1] = 0 + \frac{1}{1+\frac{1}{5+\frac{1}{1}}} = \frac{6}{7} \rightarrow 46578^{7} = 8 \; (mod \; 242537)$

Άρα $d = 7$

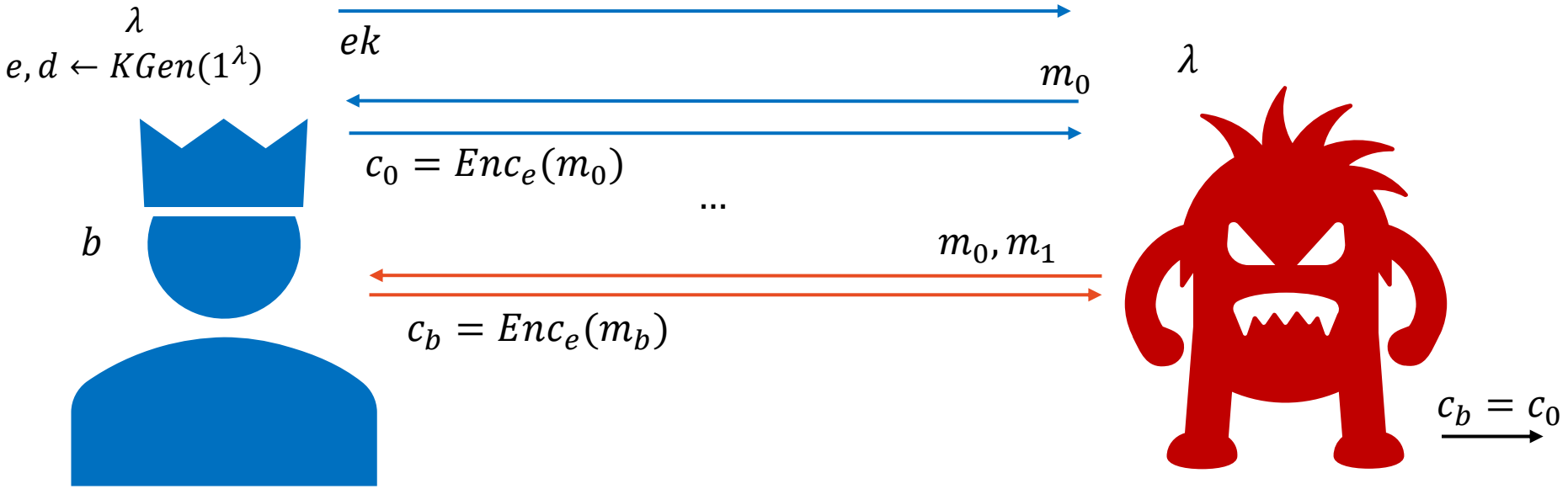
Επίθεση μικρού ιδιωτικού εκθέτη

Side-Channel Attacks

- Φυσική πληροφορία που διαρρέει από την επεξεργασία κρυπτοκειμένου - κλειδιού
- Κατανάλωση ρεύματος
- Αυξομειώσεις συχνότητας επεξεργασίας
- Χρόνος απάντησης Απαιτείται πρόσβαση στη συσκευή (όχι πάντα)
- Μπορεί να αποκαλυφθεί ολόκληρο το κλειδί
- Λύση: Dummy πράξεις



Μοντελοποίηση Ασφάλειας



Γιατί είναι ντετερμινιστικό

Ο $\mathcal{A}$ μπορεί να ξεχωρίσει κρυπτογραφήσεις μηνυμάτων τις οποίες μπορεί να παράγει ο ίδιος (δημόσιο κλειδί)

Δεν διαθέτει ούτε IND-CCA

Δουλεύει η ίδια επίθεση χωρίς να χρησιμοποιηθεί το decryption oracle

Δεν διαθέτει IND-CPA, IND-CCA

Πολλαπλασιαστικός ομομορφισμός

$$Enc_{e,n}(m_1) \cdot Enc_{e,n}(m_2) = m_1^e \, mod \, n \cdot m_2^e \, mod \, n =$$
$$(m_1 \cdot m_2)^e mod \, n = Enc_{e,n}(m_1 \cdot m_2)$$

Στο παίγνιο CCA

- **Στόχος:** Αποκρυπτογράφηση του $c_b = m_b^e \, mod \, n$
- Μπορεί να αποκρυπτογραφήσει το $c' = c_b \cdot x^e \, mod \, n$ όπου
- το x είναι δικής του επιλογής
- Ανακτά το $m_b = m' \cdot x^{-1}$
- Αν $m_b = m_0$ επιστρέφει $b^* = 0$ αλλιώς επιστρέφει $b^* = 1$

Malleability

Διαρροή Jacobi symbol

$$\left(\frac{c}{n}\right) = \left(\frac{m^e}{n}\right) = \left(\frac{m^e}{p}\right)\left(\frac{m^e}{q}\right) = \left(\frac{m}{p}\right)\left(\frac{m}{q}\right) = \left(\frac{m}{n}\right)$$

- Το textbook RSA δεν έχει σημασιολογική ασφάλεια
  - θα μπορούσαμε να διακρίνουμε δύο μηνύματα με βάση το Jacobi symbol τους.
- Μικρή σημασία
  - Ετσι κι αλλιώς μπορούμε να τα διακρίνουμε μέσω της απουσίας ιδιότητας IND-CPA.

Διαρροή πληροφοριών

RSA Hardcore Bits

Έστω $c = m^e \ mod \ n$. **Ορίζουμε τα παρακάτω predicates:**

- $parity((e,n),c) = m \ mod \ 2$
  - τελευταίο bit (LSB) του plaintext
- $loc\big((e,n),c\big) = m > \frac{n}{2}$
  - plaintext κάτω μισό / πάνω μισό του $\mathbb{Z}_n$

Η διαρροή τους οδηγεί σε 'σπάσιμο' του RSA.

Διαρροή πληροφοριών

Θεώρημα (Goldwasser, Micali, Tong)

Για κάθε στιγμιότυπο του RSA (e, n), και $c = m^e \ mod \ n$ τα παρακάτω είναι ισοδύναμα:

1. Υπάρχει ένας αποδοτικός αλγόριθμος $\mathcal{A}$ τέτοιος ώστε
$$\mathcal{A}(c) = m, \forall m \in \mathbb{Z}_n$$
2. Υπάρχει ένας αποδοτικός αλγόριθμος που υπολογίζει την συνάρτηση *parity*.
3. Υπάρχει ένας αποδοτικός αλγόριθμος που υπολογίζει την συνάρτηση *loc*.

Διαρροή πληροφοριών

Θεώρημα (Goldwasser, Micali, Tong)

Θα δείξουμε ότι: $\boldsymbol{loc(c) = parity(c \cdot Enc(2))}$

Έχουμε:

$$parity\big(c \cdot Enc(2)\big) = parity\big(Enc(2 \cdot m)\big) = (2m \bmod n) \bmod 2$$

$loc(c) = 1 \Rightarrow m > \frac{n}{2} \Rightarrow 2m > n$

Αφού $n < 2m < 2n$: $2m \bmod n = 2m - n$

Αφού n μονός θα είναι και $2m - n$ μονός δηλ. $parity(c \cdot Enc(2)) = 1$

$loc(c) = 0 \Rightarrow m \leq \frac{n}{2} \Rightarrow 2m \leq n$

Άρα $(2m \bmod n) \bmod 2 = 0$ δηλ. $parity(c \cdot Enc(2)) = 0$

Συνέπεια: $parity(c) = loc(c \cdot Enc(2^{-1}))$

Διαρροή πληροφοριών

Θεώρημα (Goldwasser, Micali, Tong) (συνέχεια)

Προφανώς (1) ⇒ (3) (αν μπορώ να αποκρυπτογραφήσω ξέρω loc)

Για το (3) ⇒ (1) Δυαδική αναζήτηση, για το m, σε $\log_2 n$ βήματα χρησιμοποιώντας την loc και διαδοχικές 'ολισθήσεις' προς τα αριστερά. Δηλαδή

- $loc(Enc(m)) = 0 \Leftrightarrow m \in [0, n/2)$
- $loc(Enc(2m)) = 0 \Leftrightarrow m \in [0, n/4) \cup (n/2, 3n/4)$
- $loc(Enc(4m)) = 0 \Leftrightarrow m \in [0, n/8) \cup (n/2, 5n/8) \ldots$

Άρα αν $loc(Enc(m)) = 0$ και $loc(Enc(2m)) = 0$ και $loc(Enc(4m)) = 0$ τότε $m \in [0, \frac{n}{8})$

Διαρροή πληροφοριών

Ασφαλείς παραλλαγές RSA (με IND-CPA)

Απαιτείται τυχαιότητα στην κρυπτογράφηση

Δηλαδή ένα μήνυμα να αντιστοιχεί σε πολλά πιθανά ciphertexts.

- Προσθήκη ψηφίων τυχαιοποίησης *r* στο μήνυμα.
- Κρυπτογράφηση $f(m, r)$
- Αποκρυπτογράφηση
- Αντιστροφή f (πρέπει να γίνεται εύκολα)

Οι λεπτομέρειες καθορίζουν την ασφάλεια…

padded RSA

- Πριν την κρυπτογράφηση μηνύματος m μήκους l δημιουργείται το μήνυμα:

$m' = r||m$, όπου r είναι μια τυχαία συμβολοσειρά από $\lambda - l$ bits.

- Μετατροπή του m' σε ακέραιο
- Η κρυπτογράφηση γίνεται (κανονικά) ως: $c' \leftarrow m'^{e} \ mod \ n$
- Η αποκρυπτογράφηση γίνεται (κανονικά) ως $c'^{d} mod \ n \rightarrow m'$
- Από το m' κράταμε μόνο τα l bits χαμηλότερης τάξης.

Διαθέτει ασφάλεια IND-CPA, όχι όμως IND-CCA.

- μπορούμε να εκμεταλλευτούμε την δομή του padded plaintext

PKCS1 v1.5 (SSL 3.0)

Padding oracle

Ένα σύστημα το οποίο μπορεί να αποφανθεί, αν το κρυπτοκείμενο έχει δημιουργηθεί με το σωστό padding.

Επίθεση με padding oracle

Κατά την αποκρυπτογράφηση, ο παραλήπτης ελέγχει αν το μήνυμα έχει το σωστό padding

- Αν **ναι**, το επεξεργάζεται - αν **όχι**, το απορρίπτει.

Ο αποστολέας μπορεί να καταλάβει τον **λόγο** απόρριψης μέσω:

- ειδικού μηνύματος λάθους
- side channel: χρόνος απάντησης (απόρριψη χρειάζεται λιγότερο χρόνο)

Η επίθεση:

- Τροποποιούμε ένα ciphertext (CCA) επαναληπτικά και **adaptively** (δηλ. λαμβάνοντας υπ' όψιν και απαντήσεις σε προηγούμενα ciphertexts)
- Ώστε να πάρουμε πληροφορίες για τη δομή του μηνύματος μέσω του padding και τελικά να μάθουμε κάτι και για το ίδιο το μήνυμα

Padding oracle attacks

Ακριβής Μορφή padded μηνύματος στο pkcs1:

`PKCS(r, m) = 0x 00||02||r||00||m`

Επεξεργασία μετά την αποκρυπτογράφηση:

- Έλεγχος πρώτου byte για την τιμή **0**
- Έλεγχος δεύτερου byte για την τιμή **2 (mode encryption)**
- Παράκαμψη τυχαιότητας **r (τουλάχιστον 8 bytes)** με αναζήτηση του byte **0**
- Ανάκτηση του **m** (οτιδήποτε ακολουθεί)

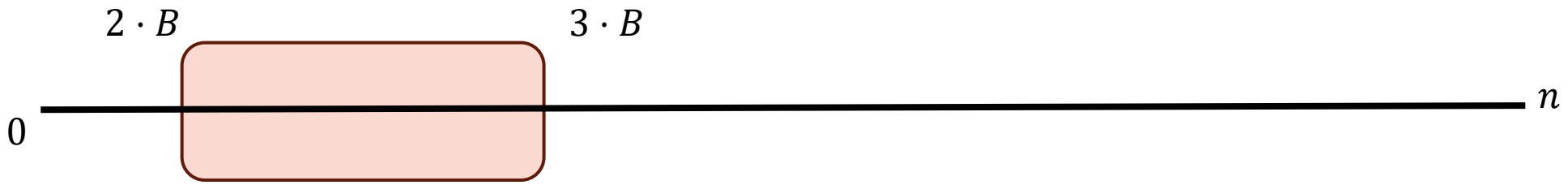
Αποδεκτό Plaintext ⟺ έχει ακριβώς αυτή τη μορφή.

Bleichenbacher Million Message Attack (1998)

Η επίθεση

Έστω κρυπτοκείμενο $c = PKCS(m, r)^e (mod\ n) = c_0$ το οποίο με κάποιον τρόπο έχει πέσει στα χέρια του A, ο οποίος **επιθυμεί να το αποκρυπτογραφήσει. Από τις προδιαγραφές του PKCS γνωρίζει τα εξής:**

- $|n| = \lambda$ (π.χ. 2048 bits)
- Τα πρώτα δύο bytes είναι 0x 0002
- Θέτουμε $\mathrm{B} = 2^{\lambda - 16}$ και $m_0 = PKCS(m, r)$
- $2 \cdot B \leq m_0 < 3 \cdot B$ αν θεωρήσουμε το $PKCS(m, r)$ ως αριθμό



Bleichenbacher Million Message Attack (1998)

Ο αντίπαλος:

- Διαλέγει τυχαία $s_1 \leftarrow_{\$} \mathbb{Z}_n$ **γνωστό** στον $\boldsymbol{\mathcal{A}}$.
- **Blinding:** Σχηματίζει $c_1 \leftarrow c_0 s_1^e \ mod \ n$ (το (e, n) είναι δημόσιο). Από malleability RSA:
  - $c_1 = \ c_0 s_1^e = (m_0 \cdot s_1)^e (mod n)$
- Το c_1 στέλνεται στον SSL server για επεξεργασία
- $m_1 = m_0 \cdot s_1 \ (mod \ n) \Rightarrow m_1 = m_0 \cdot s_1 - kn$ για κάποιο $k \in \mathbb{Z}$
- Padding oracle: Διαρρέει η ορθότητα του padding.
- Τα περισσότερα θα έχουν λάθος padding.
- Όμως αν κάποια έχουν σωστό padding τότε ανακτάται χρήσιμη πληροφορία.

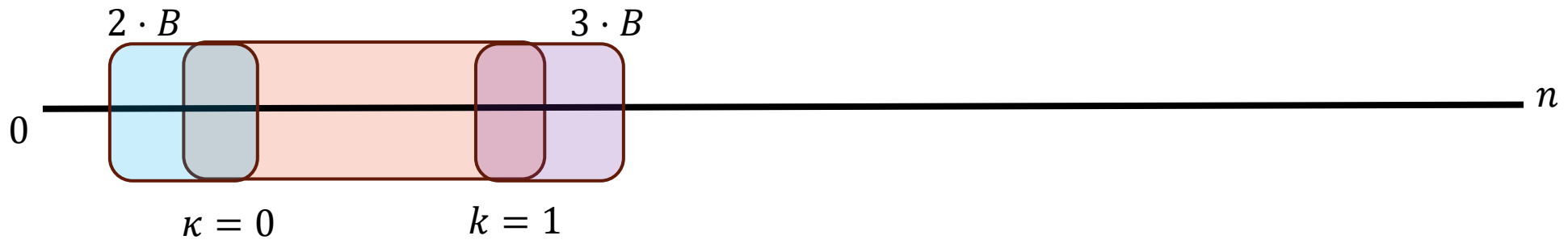
Bleichenbacher Million Message Attack (1998)

- Χρήσιμη πληροφορία από σωστό padding: $2 \cdot B \leq m_1 < 3 \cdot B$

Δηλαδή $\frac{2 \cdot B + kn}{s_1} \leq m_0 < \frac{3 \cdot B + kn}{s_1}$ (1)

Αφού $k = \frac{s_1 m_0 - m_1}{n}$ προκύπτει ότι: $\frac{2Bs_1 - 3B + 1}{n} \leq k \leq \frac{(3B-1)s_1 - 2B}{n}$

- Δοκιμάζουμε όλες τις δυνατές τιμές του k (πρακτικά είναι λίγες - λόγω του $2 \cdot B \leq m_0 < 3 \cdot B$)
- Βρίσκουμε τα σημεία τομής του $[2B, 3B)$ με τα $[\frac{2 \cdot B + kn}{s_1}, \frac{3 \cdot B - 1 + kn}{s_1}]$



Bleichenbacher Million Message Attack (1998)

Γενικεύοντας: Μετά από i επαναλήψεις θα έχουμε

$m_0 \in M_i = \bigcup_{(a,b,r)} \left\{ \left[max\left(a, \left\lceil \frac{2B+kn}{s_i} \right\rceil\right), min\left(b, \left\lceil \frac{3B-1+kn}{s_i} \right\rceil\right)\right]\right\}$ όπου

$[a, b] \in M_{i-1}$ και $\frac{as_i - 3B + 1}{n} \leq k \leq \frac{bs_i - 2B}{n}$

- Αν M_i περιέχει τουλάχιστον δύο υποδιαστήματα τότε θέτουμε $s_i \leftarrow s_i + 1$ μέχρι το $c_{i+1} \leftarrow c_0 s_{i+1}^e$ να μην δώσει padding error.
- Αν M_i περιέχει ακριβώς ένα διάστημα:

$a \leq m_0 < b$ και $\frac{2 \cdot B+kn}{m_0} \leq s_i < \frac{3 \cdot B+kn}{m_0} \Rightarrow \frac{2 \cdot B+kn}{b} \leq s_i < \frac{3 \cdot B+kn}{a} \Rightarrow$

- Διαλέγουμε $k \geq \frac{2(bs_i-2B)}{n}$ και προκύπτει $s_{i+1} \geq \frac{2 \cdot B+2(bs_i-2B)}{b} = 2s_i$
- Η σύγκλιση θα είναι πιο γρήγορη για το συγκεκριμένο διάστημα.

Bleichenbacher Million Message Attack (1998)

- Αποδεικνύεται ότι θα φτάσουμε σε κάποιο διάστημα $[a, a]$. Τότε $m_0 \leftarrow as_1^{-1} \bmod \mathrm{n}$
- Αφαιρούμε την τυχαιότητα ψάχνοντας για μηδενικό byte και προκύπτει το m
- Χρειάζονται περίπου από 300.000 μέχρι $2.000.000$ ciphertexts

Bleichenbacher Million Message Attack (1998)

Αντιμετώπιση:

- Αφαίρεση μηνύματος για λάθος padding
  - Επιστροφή τυχαίου plaintext, αντί για το πραγματικό
  - Δεν χρειάζεται client changes (TLS 1.0)
  - Προσοχή στον χρόνο απάντησης!
- Αλλαγή padding scheme
  - Χρειάζεται client changes (RSA-OAEP)
- Χρησιμοποίηση non-deterministic cryptosystem
  - Στο RSA η τυχαιότητα είναι afterthought
  - Χρειάζεται client changes (RSA-OAEP)
- Πολύ δύσκολες οι αλλαγές σε ένα ελαττωματικό πρωτόκολλο!
  - DROWN Attack
  - The ROBOT Attack - Return of Bleichenbacher's Oracle Threat

Bleichenbacher Million Message Attack (1998)

Βασικές Ιδέες

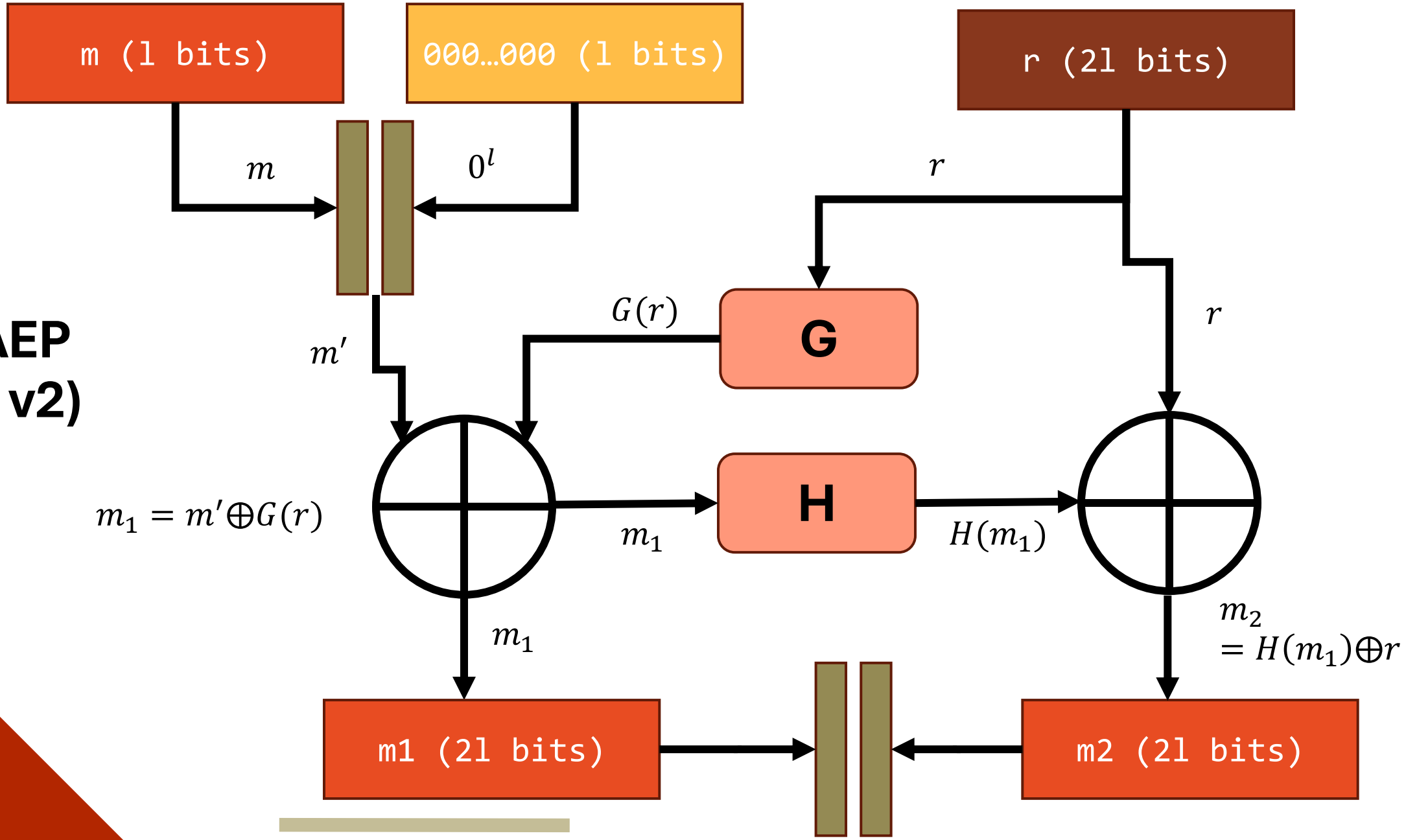
- Τα τυχαία bits πρέπει να *'διαχυθούν'* σε όλο το κρυπτοκείμενο (Δίκτυα Feistel)
- Πρέπει να υπάρχει κάποιου είδους δέσμευση στο αρχικό μήνυμα ενσωματωμένη στο κρυπτοκείμενο (Συνάρτηση Σύνοψης)

<u>Υποθέσεις</u>

- $|m| = l$
- $G, H : \{0,1\}^{2l} \rightarrow \{0,1\}^{2l}$ συναρτήσεις σύνοψης
- $r \in \{0,1\}^{2l}$

RSA-OAEP (PKCS1 v2)

RSA-OAEP (PKCS1 v2)



Κρυπτογράφηση:

1. $m' \leftarrow m||0^l$
2. $r \leftarrow_{\$} \{0,1\}^{2l}$
3. $m_1 \leftarrow m' \oplus G(r)$
4. $m_2 \leftarrow H(m_1) \oplus r$
5. $\bar{m} \leftarrow m_1||m_2$
6. $\bar{c} \leftarrow \bar{m}^e mod\ n$

Αποκρυπτογράφηση:

1. $\bar{m} \leftarrow \bar{c}\ mod\ n$
2. $\bar{m} \leftarrow m_1||m_2$ (χωρίζουμε σε δύο μέρη)
3. $r \leftarrow H(m_1) \oplus m_2$
4. $m' \leftarrow m_1 \oplus G(r)$
5. Έλεγχος l bits χαμηλότερης τάξης
6. Αν είναι 0 ανακτούμε ως μήνυμα τα l bits υψηλότερης τάξης

RSA-OAEP (PKCS1 v2)