

ΔΕΥΤΕΡΗ ΑΣΚΗΣΗ

ΣΤΗΝ

ΚΡΥΠΤΟΓΡΑΦΙΑ

ΛΙΒΑΘΙΝΟΣ ΝΙΚΟΛΑΟΣ 2291

8/7/2003

Άσκηση 1^η

Σε ένα σύστημα RSA υποκλέπουμε το κρυπτόγραμμα $C=5347$ που στέλνεται σε παραλήπτη με δημόσιο κλειδί $(e,n)=(5,16781)$. Ποιος είναι ο ακέραιος M που μεταδίδεται; (Δίνεται ότι $16781=97 \cdot 173$).

Προκειμένου να βρούμε το αρχικό μήνυμα πρέπει να υπολογίσουμε το μυστικό κλειδί (d,n) . Δηλαδή πρέπει να βρούμε το d .

Όπως γνωρίζουμε ισχύει ότι $ed=1 \bmod \varphi(n)$, όπου $\varphi(n)=(p-1)(q-1)$ και $n=pq$. Άρα σύμφωνα με τα δεδομένα της άσκησης έχουμε ότι $p=97$, $q=173$. Επομένως $\varphi(n)=16512$ και η εξίσωση που πρέπει να λύσουμε είναι η

$$5d=1 \bmod 16512$$

Ακολουθώντας τη συνάρτηση Modular-Linear-Equation-Solver βρίσκουμε ότι $d=6605$. Έτσι το αρχικό μήνυμα βρίσκεται ως: $M=C^d \bmod n \Rightarrow$

$$M=5347^{6605} \bmod 16781$$

Για να υπολογίσουμε την παράσταση χρησιμοποιούμε την συνάρτηση Modular-Exponentiation. Έτσι τελικά βρίσκουμε $M=16657$.

Άσκηση 2^η

Θεωρήστε τη μέθοδο $\text{RSA mod } N$. Ένας ακέραιος αριθμός M , $1 \leq M \leq N-1$, είναι σταθερό σημείο, αν η κρυπτογράφησή του δίνει τον εαυτό του. Αποδείξτε ότι αν το M είναι σταθερό σημείο, τότε και το $N-M$ είναι σταθερό σημείο

Για να δείξουμε ότι και το $N-M$ είναι σταθερό σημείο πρέπει να δείξουμε ότι

$$(N-M)^e \bmod N = (N-M) \bmod N.$$

Μπορούμε να αναπτύξουμε το $(N-M)^e$ σύμφωνα με το διωνυμικό ανάπτυγμα, οπότε και έχουμε:

$$(N-M)^e \bmod N = (a_0 N^e + a_1 N^{e-1} M + \dots + a_{e-1} N^e M + a_e M) \bmod N \quad (1)$$

Από την (1) παρατηρούμε ότι από όλους τους όρους του αθροίσματος όσοι έχουν συντελεστή το N , θα μηδενίζονται εφόσον $Nx \bmod N = 0$. Επομένως τελικά έχουμε:

$$(N-M)^e \bmod N = a_e M \bmod N \quad (2)$$

Για τους συντελεστές a_i γνωρίζουμε ότι είναι της μορφής $(-1)^k \binom{e}{e-k}$, όπου $0 \leq k \leq e$.

Εφόσον γνωρίζουμε πως το e είναι πάντα μονός (έτσι ορίζει το RSA), τότε $a_e = -1 \binom{e}{e} = -1$.

Άρα:

$$(N-M)^e \bmod N = -M \bmod N \quad (3)$$

Επίσης έχουμε ότι $-M \bmod N = N-M \bmod N$, εφόσον $N \bmod N=0$. Άρα μπορούμε να γράψουμε ότι:

$$(3) \Rightarrow (N-M)^e \bmod N = N-M \bmod N \quad (4)$$

Οπότε και αποδείξαμε το ζητούμενο.

Άσκηση 3^η

Αν το 2^n+1 είναι περιττός πρώτος για κάποιον ακέραιο, δείξτε ότι το n είναι δύναμη του 2

Προκειμένου να αποδείξουμε το ζητούμενο θα προχωρήσουμε σταδιακά. Αρχικά θεωρούμε ότι το n είναι περιττός και θα καταλήξουμε σε άτοπο.

Αν το n είναι περιττός τότε ισχύει η ταυτότητα

$$x^k+1=(x+1)(x^{k-1}-x^{k-2}+\dots+x^0) \quad (1)$$

Εφαρμόζοντας θα έχουμε ότι $2^n+1=(2+1)(2^{n-1}-2^{n-2}+\dots+2^0)$, αυτό όμως σημαίνει ότι το 2^n+1 θα έχει για παράγοντα το 3, άρα αποκλείεται να είναι πρώτος και καταλήξαμε σε άτοπο.

Επομένως το n θα είναι άρτιος. Ωστόσο πρέπει να δείξουμε ότι είναι και δύναμη του 2. Αν υποθέσουμε ότι το n είναι άρτιος αλλά όχι δύναμη του 2 τότε θα μπορούμε να το γράψουμε στη μορφή $n=2^k \cdot \text{odd}$, όπου odd είναι ένας περιττός.

Αντικαθιστώντας το n έχουμε $2^n + 1 = 2^{2^k \cdot \text{odd}} + 1 = \left(2^{2^k}\right)^{\text{odd}} + 1$, έτσι όμως μπορούμε πάλι να εφαρμόσουμε την ταυτότητα (1), οπότε βρίσκουμε πάλι ότι το 2^n+1 έχει για παράγοντα το $2^{2^k} + 1$, άρα το 2^n+1 δεν είναι πρώτος, που και πάλι είναι άτοπο.

Επομένως τελικά καταλήγουμε ότι το n είναι οπωσδήποτε δύναμη του 2.

Άσκηση 4^η

Δείξτε ότι ο αριθμός των θετικών ακεραίων μικρότερων του n , των οποίων οι πρώτοι παράγοντες ανήκουν σε ένα δοσμένο σύνολο πρώτων $\{p_1, p_2, \dots, p_m\}$ είναι τουλάχιστον

$$n^{r/r!} \text{ με } r = \left\lfloor \frac{\log n}{\log p_m} \right\rfloor \text{ και } p_1 < p_2 < \dots < p_m.$$

Έστω a ένας θετικός ακέραιος που διασπάται σε γινόμενο πρώτων παραγόντων ως:

$$a = p_1^{q_1} p_2^{q_2} \dots p_m^{q_m} \quad (1)$$

Αρχικά θα προσπαθήσουμε να βρούμε τις διαφορετικές τιμές που μπορεί να πάρει ο a . Για αυτό το σκοπό ας υποθέσουμε ότι το άθροισμα των εκθετών στην παράσταση (1), φράσσεται από το k . Δηλαδή

$$\sum_{i=1}^m q_i = k \quad (2)$$

Μπορούμε να δούμε το πρόβλημα της εύρεσης των διαφορετικών τιμών του a , ως ένα ισοδύναμο πρόβλημα κατανομής σφαιριδίων σε κελιά. Το γεγονός ότι το άθροισμα των εκθετών κάνει k μοιάζει σαν να έχουμε k κενές θέσεις, τις οποίες θέλουμε να τις γεμίσουμε με στοιχεία m διαφορετικών κατηγοριών. Μπορούμε λοιπόν να θεωρήσουμε ότι έχουμε m σάκους με σφαιρίδια που ο κάθε σάκος έχει άπειρα σφαιρίδια μέσα. Θέλουμε να βρούμε με πόσους διαφορετικούς τρόπους μπορούμε να καταναείμουμε τις σφαίρες στα κελιά έτσι ώστε κανένα κελί να μην μείνει άδειο και δίεχως να μας ενδιαφέρει η διάταξη των σφαιριδίων στα κελιά.

Εάν λαβαίναμε υπόψη μας τη διάταξη θα είχαμε για κάθε κελί m διαφορετικές επιλογές να το γεμίσουμε, άρα θα είχαμε συνολικά m^k τρόπους. Εφόσον δεν μας ενδιαφέρει η διάταξη, οι τρόποι περιορίζονται σε $m^k/k!$.

Δηλαδή έχουμε $m^k/k!$ διαφορετικές τιμές για το a . Ωστόσο θέλουμε να βρούμε και μια συνθήκη που θα εξασφαλίζει ότι το a θα είναι μικρότερο από το n . Για να το επιτύχουμε αυτό αρκεί να βρούμε τη συνθήκη που εξασφαλίζει ότι η μέγιστη τιμή που μπορεί να λάβει το a είναι μικρότερη από το n .

Προφανώς η μεγαλύτερη τιμή του a είναι το p_m^k , εφόσον το p_m είναι ο μεγαλύτερος πρώτος του συνόλου μας. Άρα λοιπόν έχουμε:

$$p_m^k < n \Rightarrow \log(p_m^k) < \log(n) \Rightarrow k < \left\lfloor \frac{\log(n)}{\log(p_m)} \right\rfloor$$

Άρα η μέγιστη επιτρεπτή τιμή του k είναι $\left\lfloor \frac{\log(n)}{\log(p_m)} \right\rfloor$, έστω r .

Κατόπιν παρατηρούμε ότι η συνάρτηση $f(x)=m^x/x!$ είναι φθίνουσα. Αυτό φαίνεται εύκολα εφόσον το παραγοντικό στον παρανομαστή έχει πολύ μεγαλύτερη αύξηση σε σχέση με το εκθετικό στον αριθμητή. Έτσι όταν το k λάβει τη μέγιστη τιμή του r , τότε θα έχουμε και το ελάχιστο πλήθος τιμών του a , το οποίο όμως θα είναι οπωσδήποτε μεγαλύτερο από $m^r/r!$.

Οπότε αποδείξαμε το ζητούμενο.

Άσκηση 5^η

Υποθέστε ότι οι χρήστες Alice, Bob και Carol έχουν τα δημόσια κλειδιά $(n_A, 3)$, $(n_B, 3)$ και $(n_C, 3)$ αντίστοιχα. Υποθέστε πως στέλνουμε χρησιμοποιώντας το RSA το ίδιο μήνυμα M και στους 3. Δείξτε πως μπορεί κάποιος που «ακούει» τα τρία μηνύματα να βρει ποιο ήταν το μήνυμα M . Προτείνετε μια απλή τροποποίηση που εμποδίζει την υποκλοπή

Αρχικά υπολογίζουμε τα κρυπτογραφημένα μηνύματα που θα λάβει ο υποκλοπέας:

$$K1 = M^3 \bmod n1$$

$$K2 = M^3 \bmod n2$$

$$K3 = M^3 \bmod n3$$

Θεωρούμε λοιπόν, ότι ο υποκλοπέας γνωρίζει τα $K_1, K_2, K_3, n_1, n_2, n_3$ και θέλει να βρεί το M .

Όπως γνωρίζουμε μπορούμε να λύσουμε τις παραπάνω εξισώσεις χρησιμοποιώντας το κινέζικο θεώρημα. Φυσικά για να μπορέσουμε να προχωρήσουμε πρέπει τα n_1, n_2, n_3 να είναι πρώτοι μεταξύ τους.

Για να βρούμε το M^3 πρέπει να υπολογίσουμε τα c_i και να χρησιμοποιήσουμε την σχέση (31.28) του βιβλίου.

Υπενθυμίζουμε ότι αυτά ορίζονται ως εξής:

$$\begin{aligned} n &= n_1 n_2 n_3. \\ m_i &= n/n_i \\ c_i &= m_i(m_i^{-1} \bmod n_i) \\ M^3 &= \sum c_i K_i \pmod{n} \end{aligned}$$

Για να συνεχίσουμε την επίλυση των εξισώσεων πρέπει να βρούμε τους πολλαπλασιαστικούς αντιστρόφους των m_i . Για να βρεθεί αυτό πρέπει να λυθούν οι εξισώσεις $m_i m_i^{-1} = 1 \bmod n_i$. Ωστόσο όλα αυτά μπορούν να λυθούν μέσω της Modular-Linear-Equation-Solver.

Έστω λοιπόν ότι έχουμε υπολογίσει μια λύση για το σύστημα λ . Τότε θα ισχύει ότι:

$$\lambda = M^3 \bmod n_1 n_2 n_3$$

Για να προχωρήσουμε πρέπει να έχουμε εξασφαλισμένο ότι $M^3 < n_1 n_2 n_3$. Αν ισχύει αυτό τότε $\lambda = M^3 \bmod n_1 n_2 n_3 = M^3$, δηλαδή δεν επηρεαζόμαστε από το modulo. Θεωρώντας ότι κάτι τέτοιο ισχύει, μπορούμε να υπολογίσουμε το M ως $\sqrt[3]{\lambda}$, κάτι που επίσης μπορεί να γίνει σε πολωνυμικό χρόνο.

Άρα είδαμε ότι είναι εφικτό για κάποιον υποκλοπέα να βρει το μήνυμα. Προκειμένου να αποτρέψουμε κάτι τέτοιο, μπορούμε να εισαγάγουμε στο αρχικό μήνυμα M ένα τυχαίο string. Με αυτή την παραλλαγή τα τρία κωδικοποιημένα μηνύματα θα είναι διαφορετικά μεταξύ τους και επομένως δεν θα είναι εφικτή η δημιουργία συστήματος εξισώσεων.

Φυσικά θα πρέπει οι Alice, Bob, Carol να γνωρίζουν εκ των προτέρων που θα βρίσκεται το ένθετο string και να το αφαιρέσουν μετά την αποκωδικοποίηση.

Άσκηση 6^η

Έστω (n, e) το δημόσιο κλειδί στο RSA και $\text{RSA}_{n,e}(x) = x^e \bmod n$. Υποθέστε ότι υπάρχει αλγόριθμος A που μπορεί να αντιστρέψει το 1% των εισόδων της μορφής $y = x^e \bmod n$.

A. Δείξτε ότι για κάθε $a, b \in \mathbb{Z}_n^*$ ισχύει ότι $\text{RSA}_{n,e}(a) \text{RSA}_{n,e}(b) = \text{RSA}_{n,e}(ab)$.

B. Αποδείξτε ότι χρησιμοποιώντας τον αλγόριθμο A , μπορούμε να αντιστρέψουμε οποιαδήποτε είσοδο με μεγάλη πιθανότητα.

A.

Γνωρίζουμε ότι εξ' ορισμού ισχύει ότι $\text{RSA}_{x,e}(x) = x^e \bmod n$. Οπότε έχουμε:

$$\text{RSA}_{x,e}(a) \text{RSA}_{x,e}(b) = (a^e \bmod n) (b^e \bmod n) = a^e b^e \bmod n = (ab)^e \bmod n = \text{RSA}_{x,e}(ab)$$

Οπότε και αποδείξαμε το ζητούμενο.

B.

Ο αλγόριθμος A λαβαίνει για είσοδο ένα κρυπτογραφημένο μήνυμα v που προέκυψε ως $v = M^e \bmod n$ και επιστρέφει το M . Αν ο αλγόριθμος δεν μπορεί να βρει το M επιστρέφει ένα μήνυμα λάθους.

Θεωρούμε λοιπόν ότι έχουμε ένα υποσύνολο $V \subseteq Z_n^*$, με $|V| = 0.01|Z_n^*|$ για το οποίο ισχύει ότι $\forall v \in V$ μπορεί να εφαρμοστεί ο αλγόριθμος A. Με αυτά υπόψη, ο σκοπός μας είναι να βρούμε έναν τρόπο ώστε ακόμα και αν ένα στοιχείο δεν ανήκει στο V , να το μετατρέψουμε σε μια αναστρέψιμη μορφή που να ανήκει όμως στο V .

Έτσι επιλέγουμε $r \in Z_n^*$, και υπολογίζουμε $v' = vr^e \bmod n$, κατόπιν τρέχουμε τον αλγόριθμο A για το v' . Αν ο αλγόριθμος επιστρέψει μια έξοδο M' , και όχι μήνυμα λάθους, τότε μπορούμε να υπολογίσουμε το v ως $M' r^{-1}$, όπου r^{-1} είναι ο πολλαπλασιαστικός αντίστροφος του r .

Συνοψίζοντας τη διαδικασία μετατροπής έχουμε τις ακόλουθες σχέσεις:

$$v = M^e \bmod n \quad (1)$$

$$v' = vr^e \bmod n \quad (2)$$

$$M' = A(v') \quad (3)$$

$$v' = M'^e \bmod n \quad (4)$$

$$M = M' r^{-1} \quad (5)$$

Εύκολα αποδεικνύεται και η ορθότητα της μετατροπής:

$$\underline{M^e (5) (M' r^{-1})^e} = \underline{M'^e r^{-e} (4) v' r^{-e} (2) v r^e r^{-e} (1) M^e}$$

Βλέπουμε δηλαδή πως αρχίζοντας από το M^e καταλήγουμε πάλι σε αυτό.

Έστω τώρα P η πιθανότητα να πετύχουμε το v' να ανήκει στο V με την I επανάληψη. Θεωρώντας την αντίστοιχη τυχαία μεταβλητή X , θα έχουμε:

$$P(X=I) = 0.99^{I-1} 0.01$$

Η παραπάνω σχέση προκύπτει θεωρώντας ότι στις $I-1$ προηγούμενες περιπτώσεις είχαμε αποτύχει και επιτυγχάνουμε στην I -οστή. Άρα μετά από k επαναλήψεις η πιθανότητα επιτυχίας είναι:

$$P(X \leq k) = \sum_{i=1}^k P(X=i) = \sum_{i=1}^k 0.99^{i-1} 0.01 = 0.01 \sum_{i=1}^k 0.99^{i-1} = 0.01 \sum_{i=0}^{k-1} 0.99^i$$

Χρησιμοποιώντας τον τύπο για το άθροισμα όρων γεωμετρικής προόδου έχουμε:

$$P(X \leq k) = 0.01 \frac{0.99^k - 1}{0.99 - 1} = 1 - 0.99^k$$

Στη συνέχεια μπορούμε να θέσουμε ένα όριο προκειμένου να εξασφαλίσουμε μεγάλη πιθανότητα επιτυχίας. Ας θεωρήσουμε λοιπόν ότι θέλουμε να έχουμε 99% πιθανότητα επιτυχίας. Τότε έχουμε:

$$1 - 0.99^k \leq 0.99 \Rightarrow k \leq \log_{0.99} 0.01 \leq \frac{\log 0.01}{\log 0.99} \approx 458,21$$

Επομένως με 459 επαναλήψεις μπορούμε να πετύχουμε με πιθανότητα 99% να βρούμε ένα πολλαπλάσιο του n που να ανήκει στο V και επομένως να αντιστρέφεται από τον αλγόριθμο A . Φυσικά για να λάβουμε το n πρέπει να πολλαπλασιάσουμε με r^{-1} (πολλαπλασιαστικός αντίστροφος), αφού βρούμε το αποκρυπτογραφημένο μήνυμα.

Άσκηση 7^η

Υποθέτουμε πως βλέπουμε την κρυπτογράφηση RSA των μηνυμάτων $m, m+1, m+2$ με $e=3$. Πως μπορούμε να βρούμε το m σε πολυωνυμικό χρόνο;

Αρχικά υπολογίζουμε ποια είναι τα κρυπτογραφημένα μηνύματα:

$$K1=m^3 \bmod n$$

$$K2=(m+1)^3 \bmod n$$

$$K3=(m+2)^3 \bmod n$$

Οπότε αναπτύσσοντας τις ταυτότητες έχουμε

$$K1=m^3 \bmod n$$

$$K2=m^3+3m^2+3m+1 \bmod n$$

$$K3=m^3+6m^2+12m+8 \bmod n$$

Θεωρούμε λοιπόν ότι γνωρίζουμε τα $K1, K2, K3$ και θέλουμε να βρούμε έναν τρόπο για να υπολογίσουμε το m . Μια σκέψη είναι να βρούμε έναν κατάλληλο συνδυασμό των $K1, K2, K3$ που θα δώσει έναν εύκολο τρόπο υπολογισμού του m .

Έστω λοιπόν ο συνδυασμός $K=K3+K1-2K2=6m+6 \bmod n = 6(m+1) \bmod n$.

Από την τελευταία εξίσωση μπορούμε αντικαθιστώντας $x=m+1$ να λύσουμε ως προς x , οπότε και έχουμε:

$$K3+K1-2K2=6x \bmod n \quad (1)$$

Μπορούμε λοιπόν από την (1) να υπολογίσουμε το x , άρα και το m . Επίσης εφόσον η Modular-Linear-Equation-Solver τρέχει σε πολυωνυμικό χρόνο, άρα και η λύση μπορεί να υπολογιστεί σε πολυωνυμικό χρόνο.

Άσκηση 8^η

Έστω $n=pq$, όπου p, q περιττοί πρώτοι αριθμοί, διαφορετικοί μεταξύ τους. Έστω

$\lambda(n) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)}$. Υποθέτουμε πως τροποποιούμε το πρωτόκολλο RSA έτσι ώστε

$ed=1 \bmod \lambda(n)$.

- Δείξτε ότι η κρυπτογράφηση και η αποκρυπτογράφηση είναι αντίστροφες συναρτήσεις στο τροποποιημένο πρωτόκολλο.
- Αν $p=37, q=79$ και $d=7$, υπολογίστε το e του τροποποιημένου πρωτοκόλλου, καθώς και του κανονικού πρωτοκόλλου RSA.

A.

Θεωρούμε ότι $C=M^e \bmod n$ είναι το κρυπτογραφημένο μήνυμα και $S(C)=C^d \bmod n$ είναι η συνάρτηση αποκρυπτογράφησης. Θέλουμε λοιπόν να δείξουμε ότι:

$$M^{ed} \bmod n = M \bmod n$$

$$\text{Γνωρίζουμε πως } ed = 1 \bmod \lambda(n) \Rightarrow ed = 1 + k \frac{(p-1)(q-1)}{\gcd(p-1, q-1)} \quad (1)$$

Αρχικά θεωρούμε ότι $M \neq 0 \bmod p$

Οπότε από την (1) έχουμε

$$M^{ed} \bmod p = M^1 \left(M^{p-1} \right)^{k \frac{q-1}{\gcd(p-1, q-1)}} \bmod p \quad (2)$$

Όμως από το θεώρημα του Fermat γνωρίζουμε πως $a^{p-1} = 1 \bmod p$

$$\text{Οπότε (2)} \Rightarrow M^{ed} \bmod p = M \bmod p \quad (3)$$

Αν τώρα $M = 0 \bmod p$, τότε προφανώς $M^{ed} \bmod p = M \bmod p$.

Ακολουθώντας αντίστοιχη ανάλυση μπορούμε να αποδείξουμε και ότι $M^{ed} \bmod q = M \bmod q$ (4)

Από τις (3), (4) και το Corollary 32.29 του κινέζικου θεωρήματος έχουμε ότι:

$$M^{ed} \bmod(pq) = M \bmod(pq) \Rightarrow M^{ed} \bmod n = M \bmod n$$

Οπότε αποδείξαμε ότι οι διαδικασίες κρυπτογράφησης και αποκρυπτογράφησης είναι συμπληρωματικές.

B.

Αρχικά θα υπολογίσουμε το e του κανονικού πρωτοκόλλου RSA.

Όπως γνωρίζουμε το e υπολογίζεται ως ο πολλαπλασιαστικός αντίστροφος του d ως προς $\phi(n)$. Δηλαδή:

$$7e = 1 \bmod \phi(n), \text{ όπου } \phi(n) = (p-1)(q-1) \quad (4)$$

Εφόσον $p=37$, $q=79$, $d=7$ έχουμε ότι $\phi(n)=2808$, οπότε χρησιμοποιώντας την Modular-Linear-Equation-Solver, έχουμε ότι $e=2407$.

Για το τροποποιημένο πρωτόκολλο έχουμε ότι:

$$7e = 1 \bmod \lambda(n), \text{ όπου } \lambda(n) = \frac{(p-1)(q-1)}{\gcd(p-1, q-1)} \quad (5)$$

Οπότε αντικαθιστώντας έχουμε ότι $\lambda(n)=468$. Στη συνέχεια τρέχουμε πάλι την Modular-Linear-Equation-Solver, οπότε και προκύπτει ότι $e=67$.

Παράρτημα

Πολλές φορές στην επίλυση των ασκήσεων χρησιμοποιήσαμε τις συναρτήσεις Modular-Linear-Equation-Solver, και Modular-Exponentiation όπως περιγράφονται στο “Introduction to Algorithms”. Για αυτό και τελικά αποφάσισα πως θα ήταν πολύ βολικό να τις υλοποιήσω σε matlab.

Η συνάρτηση `modularSolver` υλοποιεί την `Modular-Linear-Equation-Solver` και η συνάρτηση `modularExp` υλοποιεί την `Modular-Exponentiation`. Στη συνέχεια ακολουθούν οι κώδικες.