Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Εθνικό Μετσόβιο Πολυτεχνείο
Ροή Λ, 9ο εξάμηνο

Υπολογιστική Κρυπτογραφία

Τέταρτη σειρά ασκήσεων

Σπουδαστής
Παπασκαρλάτος Αλέξανδρος (Α.Μ.: 03111097)

Ημερομηνία Υποβολής: 29 Φεβρουαρίου 2020

Άσκηση 1

Θεωρώ ένα σταθερό σημείο m (για δεδομένα e, N=p · q).
Είναι $m^e \equiv m \pmod{p \cdot q}$
Από CRT, ισχύει $m^e \equiv m \pmod{p}$
$m^e \equiv m \pmod{q}$

Εξετάζουμε το (mod p) . Δύο περιπτώσεις για το m:

1. $m \equiv 0 \pmod{p}$
   Τότε $m^e \equiv 0^e \equiv 0 \equiv m \pmod{p}$, δηλαδή το m είναι ένα σταθερό σημείο.
2. $m \in Z_p^*$
   Το Z_p^* είναι πολλαπλασιαστική ομάδα. Συνεπώς, το στοιχείο m έχει σίγουρα αντίστροφο, m^{-1}.
   Είναι $m^e \equiv m \pmod{p} \Leftrightarrow m^{e-1} \equiv 1 \pmod{p}$
   Από θεωρία ομάδων, γνωρίζουμε πως υπάρχουν πως υπάρχουν gcd(e-1, p-1) που ικανοποιούν την παραπάνω ισοτιμία (καθώς p-1 είναι η τάξη της ομάδας).
   Πράγματι, έστω d = gcd(e-1, p-1) και g ένας γεννήτορας του (mod p).
   Είναι $d \mid e-1 \quad \Rightarrow \exists\, r, \quad e-1 = r\,d$
   Τα στοιχεία που ικανοποιούν την παραπάνω ισοτιμία είναι ακριβώς τα 1, $g^{(p-1)/d}$, $g^{2(p-1)/d}$, …, $g^{(d-1)(p-1)/d}$, καθώς $(g^{k(p-1)/d})^{(e-1)} = g^{(p-1)\,k\,(e-1)/d} \equiv g^{(p-1)\,k\,r} \equiv 1$, δηλαδή υπάρχουν d τέτοια στοιχεία.

Από τις παραπάνω παρατηρήσεις, προκύπτει πως υπάρχουν συνολικά gcd(e-1, p-1) + 1 που ικανοποιούν αυτήν την ισοτιμία στο $Z_p = Z_p^* \cup \{0\}$

Όμοια, έχουμε gcd(e-1, q-1) + 1 στοιχεία στο Z_q .

Συνεπώς, μπορούμε να δημιουργήσουμε (gcd(e-1, q-1) + 1) (gcd(e-1, q-1) + 1) ζεύγη σημείων στο Z_p x Z_q . Κάθε τέτοιο ζεύγος αντιστοιχεί σε ακριβώς ένα σημείο στο Z_{pq} (από CRT) και δύο ζεύγη δεν μπορούν να αντιστοιχηθούν στο ίδιο σημείο.
Επομένως, προκύπτουν (gcd(e-1, q-1) + 1) (gcd(e-1, q-1) + 1) σημεία m στο Z_{pq} για τα οποία ισχύει η ισοτιμία $m^e \equiv m \pmod{p \cdot q}$

Άσκηση 2

Είναι ιδιωτικό κλειδί x και δημόσιο $y = g^x \bmod p$.

i. Ο υπογράφων επιλέγει $h \in \{0,...,p-2\}$ ώστε $H(m) + x + h \equiv 0 \pmod{p-1}$
ii. $sign(x, m) = (m, (x+h) \bmod p-1, g^h \bmod p)$
iii. Επαλήθευση (m, a, b) :

- $y\ b \equiv g^a \pmod p$ και
- $g^{H(m)}\ y\ b \equiv 1 \pmod p$

Προκειμένου να δείξουμε πως το σχήμα αυτό δεν είναι ασφαλές ενάντια σε καθολική πλαστογράφηση, αρκεί να δείξουμε πως ο επιτιθέμενος, μπορεί, για οποιοδήποτε δεδομένο μήνυμα m, να επιλέξει κατάλληλα a, b, ώστε η τριάδα (m, a, b) να περνάει επιτυχώς τη διαδικασία της επαλήθευσης, χωρίς ο επιτιθέμενος να γνωρίζει το ιδιωτικό κλειδί x.

Έστω μήνυμα m.
Θεωρούμε $b \equiv (g^{H(m)}\ y)^{-1} \pmod p$
Σημειώνουμε πως το b, δηλαδή ο αντίστροφος του $g^{H(m)}\ y$ στο (mod p) υπολογίζεται εύκολα με EGCD, καθώς το $g^{H(m)}\ y$ είναι γνωστό (ή καλύτερα, εύκολα υπολογίσιμο καθώς είναι γνωστοί οι όροι που το συνθέτουν) και το p είναι πρώτος.

Επίσης, θεωρούμε $a = p - 1 - H(m)$
Το a είναι εύκολα υπολογίσιμο καθώς το p είναι γνωστό και το H(m) εύκολα υπολογίσιμο δεδομένου του m.

Τότε για την επαλήθευση, έχουμε :

- $y\ b \equiv g^x\ (g^{H(m)}\ g^x)^{-1} \equiv g^{-H(m)} \equiv g^{p-1}\ g^{-H(m)} \equiv g^{p-1-H(m)} \equiv g^a \pmod p$ και
- $g^{H(m)}\ y\ b \equiv g^{H(m)}\ y\ (g^{H(m)}\ y)^{-1} \equiv 1 \pmod p$

Δηλαδή επαληθεύονται οι συνθήκες της επαλήθευσης.
Επομένως, το σχήμα επιτρέπει καθολική πλαστογράφηση.

Άσκηση 3

Δίνεται το παρακάτω πρωτόκολλο μεταξύ ενός prover P και ενός verifier V το οποίο έχει στόχο την απόδειξη γνώσης του μηνύματος που αντιστοιχεί σε ένα δεδομένο κρυπτοκείμενο RSA με δημόσιο κλειδί (e, n), δηλαδή $m \in Z_n^*$ τέτοιο ώστε $y = m^e \bmod n$. Επιπλέον θεωρήστε ότι e πρώτος.

• Ο P επιλέγει τυχαία ένα $t \in Z * n$ και στέλνει στον V το $h = t^e \bmod n$.
• Ο V επιλέγει ένα τυχαίο c, $c \in \{0, \ldots, e - 1\}$, και το στέλνει στον P.
• Ο P υπολογίζει το $r = t\, m^c \bmod n$ και το στέλνει στον V.
• Ο V αποδέχεται αν και μόνο αν $r^e \equiv h\, y^c \pmod n$.

Πληρότητα

$r^e \equiv (t \cdot m^c)^e \equiv t^e\, m^{e\,c} \equiv h\, y^c \pmod n$

Special Soundness

Έστω 2 επιτυχείς εκτελέσεις του πρωτοκόλλου (h, c, r) και (h, c', r') .

Είναι $\mathbf{r^e \equiv h\, y^c}$ και $\mathbf{r'^{\,e} \equiv h\, y^{c'}}$

Καταρχάς, δείχνουμε πως ίδια δέσμευση στο h, σημαίνει χρήση ίδιου t, δηλαδή πως η e-οστή ρίζα του h είναι μοναδική.

Block άτοπο: Έστω υπάρχουν $t, t' \in Z_n^*$ με $t \neq t'$ και $t^e \equiv t'^e$

- Είναι $t^e \equiv t'^e \pmod n \Leftrightarrow t^e\, t'^{-e} \equiv 1 \pmod n \Leftrightarrow (t\, t'^{-1})^e \equiv 1 \pmod n \Leftrightarrow \mathrm{ord}_n(t\, t'^{-1}) \mid e$
- $\mathrm{ord}_n(t\, t'^{-1}) \mid e \;\wedge$ e πρώτος $\Rightarrow \mathrm{ord}_n(t\, t'^{-1}) = 1$ ή $\mathrm{ord}_n(t\, t'^{-1}) = e$
- Είναι $|Z_n^*| = \varphi(n)$, όπου $Z_n^* = U(Z_{pq})$ οι σχετικά πρώτοι με το n
- Είναι $\mathrm{ord}_n(t\, t'^{-1}) \mid |Z_n^*| \Rightarrow \mathrm{ord}_n(t\, t'^{-1}) \mid \varphi(n)$
- Από ιδιότητες RSA, έχουμε $\gcd(e, \varphi(n)) = 1$
- Άρα, $\mathrm{ord}_n(t\, t'^{-1}) \mid \varphi(n) \;\wedge\; \gcd(e, \varphi(n)) = 1 \;\wedge\; e > 1$ (αφού e πρώτος) $\Rightarrow \mathrm{ord}_n(t\, t'^{-1}) \neq e$
- Άρα, $(\mathrm{ord}_n(t\, t'^{-1}) = 1$ ή $\mathrm{ord}_n(t\, t'^{-1}) = e) \;\wedge\; \mathrm{ord}_n(t\, t'^{-1}) \neq e \Rightarrow \mathrm{ord}_n(t\, t'^{-1}) = 1$
- $\mathrm{ord}_n(t\, t'^{-1}) = 1 \Rightarrow (t\, t'^{-1}) = 1 \Rightarrow t = t'$

Άτοπο, καθώς υποθέσαμε πως $t \neq t'$.

Επομένως, η e-οστή ρίζα του h είναι μοναδική.

Συνεπώς και στις δύο εκτελέσεις του πρωτοκόλλου, έχουμε ίδιο t.

Είναι $\mathbf{r \equiv t\, m^c}$ και $\mathbf{r' \equiv t\, m^{c'}}$

$r\, m^{-c} \equiv t$ $r'\, m^{-c'} \equiv t$

Οπότε, $r\, m^{-c} \equiv r'\, m^{-c'} \Leftrightarrow \mathbf{r\, r'^{-1} \equiv m^{c - c'}}$

Σημειώνουμε πως η r'^{-1} υπολογίζεται εύκολα με EGCD.

Είναι c, c' ϵ {0, …, e-1} και c ≠ c' .
Χωρίς βλάβη της γενικότητας, θεωρούμε c > c', οπότε (c - c') ϵ {0, …, e-1} .

Είναι e πρώτος $\wedge$ (c - c') < e $\Rightarrow$ gcd(e, (c - c')) = 1
Λοιπόν, χρησιμοποιώντας EGCD, βρίσκουμε a,b ϵ Z τ.ω. **$a e + b (c - c') = 1$**

Επομένως, ισχύουν **$m^e \equiv y$** και **$m^{c - c'} \equiv r r'^{-1}$** και **$a e + b (c - c') = 1$**
$m^{a e} \equiv y^a$ και $m^{b (c - c')} \equiv (r r'^{-1})^b$
Οπότε, $m^{a e} m^{b (c - c')} \equiv y^a (r r'^{-1})^b$
$m^{a e + b (c - c')} \equiv y^a (r r'^{-1})^b$
$m \equiv y^a (r r'^{-1})^b$, όπου y, r, r'^{-1}, a, b όλα γνωστά ή εύκολα υπολογίσιμα

Τελικά, αφού ο P ξέρει να απαντήσει 2 τέτοιες ερωτήσεις, σίγουρα ξέρει το m.

<u>HVZK</u>
Έστω simulator S και τίμιος V.

- Αρχικά, ο S δεσμεύεται κανονικά στο $h = t^e$
- Ο V επιλέγει τυχαίο c, c ∈ {0, . . . , e − 1}.
- Αν ο S μπορεί να απαντήσει, το πρωτόκολλο συνεχίζει κανονικά. Η πιθανότητα αυτής της περίπτωσης είναι αμελητέα.
- Διαφορετικά, κάνουμε rewind.
- Στη δεύτερη εκτέλεση ο S δεσμεύεται στο $h = y^{-c} = (y^c)^{-1}$ (εύκολα υπολογίσιμο).
- Ο V επιλέγει ίδιο c (ίδιο random tape).
- Ο S στέλνει r = 1
- Ο V θα δεχτεί αφού: $r^e \equiv 1^e \equiv 1 \equiv y^{-c} y^c \equiv h y^c \pmod n$

Τελικά, το πρωτόκολλο διαθέτει πληρότητα, ειδική ορθότητα και HVZK, επομένως είναι Σ-πρωτόκολλο.

Άσκηση 4

Υλοποίησα το ζητούμενο πρόγραμμα σε python3.

Κατασκεύασα ένα αρχείο κώδικα που υλοποιεί την επίθεση αποκρυπτογράφησης κρυπτοκειμένου RSA χρησιμοποιώντας ένα oracle (υπό τη μορφή συνάρτησης που χρησιμοποεί το ιδιωτικό κλειδί) για τη συνάρτηση loc.

Συγκεκριμένα, κατασκεύασα (από το μηδέν) σχήμα κρυπτογράφησης RSA με n τον RSA-120 semiprime.
Στη συνέχεια, επιλέγω τυχαίο έγκυρο κλειδί και δημιουργώ 100 τυχαία μηνύματα και τα αντίστοιχα κρυπτοκείμενα τα οποία αποκρυπτογραφούνται μέσω της παραπάνω επίθεσης.

Περισσότερες λεπτομέρειες βρίσκονται στα σχόλια του κώδικα.

Άσκηση 5

Έστω το παρακάτω πρωτόκολλο μηδενικής γνώσης. Οι δημόσιες παράμετροι είναι⟨p, m, g, h⟩ και ο prover γνωρίζει ένα x τέτοιο ώστε $g^x = h \bmod p$.

• Ο prover επιλέγει τυχαία ένα $t \in Z_m^*$ και στέλνει στον verifier το $y = g^t \bmod p$.
• Ο verifier επιλέγει τυχαία $c \in Z_m^*$ και το στέλνει στον prover.
• Ο prover υπολογίζει το $s = t + c + x$ και το στέλνει στον verifier.
• Ο verifier αποδέχεται αν και μόνο αν $g^s = y\, g^c\, h \bmod p$.

Το παραπάνω πρωτόκολλο έχει την ιδιότητα HVZK.
Πράγματι, έστω simulator S και τίμιος V.

- Ο S δεσμεύεται στο $y = h^{-1}$. Σημειώνουμε πως ο αντίστροφος του h υπολογίζεται εύκολα με EGCD, εφόσον ο p είναι πρώτος.
- Ο V επιλέγει ένα $c \epsilon Z_m^*$.
- Ο S στέλνει s = c
- Ο V θα δεχτεί αφού: $g^s = g^c = h^{-1} g^c h = y\, g^c\, h$

Σημειώνουμε δε, πως λόγω της ίδιας διαδικασίας (καθώς δε χρησιμοποιούμε rewinds), βλέπουμε πως ένας κακόβουλος P* μπορεί να ξεγελάσει τίμιο V με πιθανότητα 1, δηλαδή το πρωτόκολλο δε διαθέτει ορθότητα.

Πάντως, το πρωτόκολλο διαθέτει HVZK, αλλά δεν αρκεί ως απόδειξη πως ο prover γνωρίζει ένα x τέτοιο ώστε $g^x = h \bmod p$.

Άσκηση 6

Έστω ένα σχήμα δέσμευσης όπου η Alice δεσμεύεται σε μια τιμή χωρίς όμως να θέλει να την αποκαλύψει στον Bob.
Λοιπόν, έστω $x \in X$ η τιμή της Alice και $y = commit(x, r)$ η τιμή που στέλνει στον Bob, όπου το $r \in R$ είναι αλάτι.

Έστω πως το παραπάνω σύστημα διαθέτει τέλεια δέσμευση, δηλαδή, εφόσον η Alice έχει δεσμευτεί σε ένα y, δεν μπορεί να αλλάξει το x και να προκύψει ίδιο y, δηλαδή είναι,
για $x \neq x' \Rightarrow commit(x, r) \neq commit(x', r')$

Σημειώνουμε πως εφόσον έχουμε **τέλεια** δέσμευση, η Alice δεν μπορεί να βρεί τέτοιο x' ακόμα και αν δεν είναι υπολογιστικά φραγμένη. Δεν υπάρχει κατάλληλο x' που να δίνει το ίδιο y.

Όμως, σε αυτήν την περίπτωση ο Bob, μπορεί να δοκιμάσει όλα τα ζεύγη τιμών για το (x, r) από το X x R και να υπολογίσει το αντίστοιχο commit(x, r) μέχρις ότου να βρει το y που του έστειλε η Alice, στην οποία περίπτωση έχει βρει το x.
Συνεπώς, το σχήμα δε διαθέτει τέλεια απόκρυψη καθώς ο Bob μπορεί να βρει την αρχική τιμή της Alice.

Σημειώνουμε πως εφόσον ασχολούμαστε με **τέλεια** απόκρυψη, ο Bob δε θεωρείται υπολογιστικά φραγμένος, οπότε μπορεί να υπολογίσει όλα τα (μάλλον απαράδεκτα πολλά) ζεύγη (x, r).

Λοιπόν, καταλήγουμε πως εάν ένα σχήμα διαθέτει τέλεια δέσμευση, τότε δε διαθέτει τέλεια απόκρυψη και, με αντιθετοαναστροφή, προκύπτει πως, αν διαθέτει τέλεια απόκρυψη, τότε δε διαθέτει τέλεια δέσμευση.

Δηλαδή, ένα σχήμα δέσμευσης δεν μπορεί να διαθέτει ταυτόχρονα τις ιδιότητες τέλειας δέσμευσης και τέλειας απόκρυψης.

Άσκηση 7

Υλοποίησα το ζητούμενο πρόγραμμα σε python3.

Κατασκεύασα ένα αρχείο κώδικα που υλοποιεί το σχήμα υπογραφών Schnorr, βασισμένο σε Schnorr group.

Περισσότερες λεπτομέρειες βρίσκονται στα σχόλια του κώδικα.

Άσκηση 9

1.
Θεωρούμε πως το σύνολο $Y \subset \{0, 1\}^n$ είναι το $Y = \{ a \,||\, b \mid a = 0^k, b \in \{0,1\}^{n-k} \}$ για κάποιο δοθέν k.
Θεωρώ μια hash function $H(z) \in \{0,1\}^{n-k}$ collision resistant.
Θεωρώ την $H'(z) \in \{0,1\}^n$ με $H'(z) = 0^k \,||\, H(z)$.
Η H(z) είναι collision resistant. Πράγματι:

Block άτοπο: Έστω η H' δεν είναι collision resistant.

- Μπορούμε να βρούμε εύκολα σύγκρουση για την H' , δηλαδή μπορούμε να βρούμε x, y με $x \neq y$ και $H'(x) = H'(y)$.
- $H'(x) = H'(y) \Rightarrow 0^k \,||\, H(x) = 0^k \,||\, H(y) \Rightarrow H(x) = H(y)$
- Δηλαδή βρήκαμε σύγκρουση για την H

Άτοπο, καθώς η H είναι collision resistant.
Επομένως, η H' είναι collision resistant.

Ωστόσο, η H' είναι τετριμμένα χείριστη για PoW, καθώς $\forall\, z,\ H(z) \in Y$

2. Δίνεται η συνάρτηση $G(z) = H(z) \,||\, LSB(z)$.
Ζητείται νδο η G είναι ασφαλής για PoW αλλά δεν έχει αντίσταση πρώτου ορίσματος.

Ένα πρόβλημα είναι πως δε μας δίνεται καμία πληροφορία για την H.

Παρεμπιπτόντως, αν η G δεν έχει αντίσταση πρώτου ορίσματος, τότε ούτε η H έχει αντίσταση πρώτου ορίσματος. Πράγματι:

Block άτοπο: Έστω η G δεν έχει αντίσταση πρώτου ορίσματος και η H έχει

- Έστω μας δίνεται τιμή H(z), αλλά όχι το z.
- Θεωρούμε τις ποσότητες $g = H(z) \,||\, 0$ και $g' = H(z) \,||\, 1$
- Μία από τις δύο, θα αντιστοιχεί σίγουρα στην $G(z) = H(z) \,||\, LSB(z)$.
- Επομένως, δοκιμάζοντας να αντιστρέψουμε και τις δύο (καθώς η G δεν έχει αντίσταση πρώτου ορίσματος), τουλάχιστον η μία εκ των δύο θα δώσει έγκυρο z' τ.ω. $G(z') = G(z)$.
- Όμως, τότε $H(z') = H(z)$, δηλαδή αντιστρέψαμε την H αφού βρήκαμε έγκυρο z'.

Άτοπο, καθώς η H έχει αντίσταση πρώτου ορίσματος.
Επομένως, αν η G δεν έχει αντίσταση πρώτου ορίσματος, τότε ούτε η H έχει..

Τώρα, υποθέτουμε πως η H είναι καλή για PoW στο $Y \subset \{0,1\}^n$. Δεν ξέρω αν αυτό θεωρείται δεδομένο από την εκφώνηση, νομίζω αυτό εννοεί; Έστω $Y = \{ a \,||\, b \mid a = 0^k, b \in \{0,1\}^{n-k} \}$, για δεδομένο k.

Τότε, η G είναι ασφαλής για PoW σε ένα $Y' \subset \{0,1\}^{n+1}$, όπου $Y' = \{ a \,||\, b \mid a = 0^k, b \in \{0,1\}^{n+1-k} \}$ καθώς ο μετασχηματισμός $|| \, LSB(z)$ ουσιαστικά δεν επηρεάζει κάτι.

Άσκηση 10

1. Δύο miners λαμβάνουν την ίδια αλυσίδα από blocks και ένα πλήθος από νέες συναλλαγές τις οποίες θέλουν να οργανώσουν σε ένα νέο block.
Ο καθείς από αυτούς, υπολογίζει κατάλληλο nonce ώστε το block του να είναι έγκυρο.
Και οι δύο κάνουν broadcast το block τους στο δίκτυο ως συνέχεια της αλυσίδας.
Έτσι, δημιουργούνται δύο αλυσίδες (οι οποίες διαφέρουν στο τελευταίο block) εξίσου έγκυρες.

2. Φυσιολογικά, προκειμένου να δημιουργηθούν δύο αλυσίδες τοιουτοτρόπως, οι δύο miners πρέπει να υπολογίσουν το νέο τους block σχεδόν ταυτόχρονα κάτι που δεν είναι πολύ πιθανό. Όταν ο ένας εκ των δύο δημιουργήσει το νέο block, το κάνει broadcast στο δίκτυο, οπότε ο άλλος είναι σε θέση να το δει και να ακυρώσει τη δουλειά στο δικό του. Φυσικά, μπορεί ο δεύτερος να μη λάβει το block του πρώτου εγκαίρως (ή ακόμα και να μη νοιαστεί), οπότε να ολοκληρώσει τη δουλειά στο δικό του.
Ωστόσο, ακόμα και σε αυτήν την περίπτωση, όπου δημιουργούνται δύο διαφορετικές ισομηκείς αλυσίδες, στη συνέχεια, κάποιος miner θα λάβει μία εξ αυτών και θα υπολογίσει πρώτος ένα νέο block. Τότε αυτή η αλυσίδα θα έχει μεγαλύτερο μήκος από την άλλη, οπότε θα θεωρηθεί η σωστή σύμφωνα με το consensus του bitcoin.
Υπάρχει και πάλι η πιθανότητα κάποιος άλλος miner να καταπιαστεί με την άλλη αλυσίδα, οπότε να έχουμε νέα ισοπαλία, αλλά όσο αυτή η πιθανότητα είναι πολύ πιο μικρή (καθώς τώρα πρέπει και οι δύο πρώτοι να τελείωσαν τη δουλειά τους ταυτόχρονα, αλλά και οι δεύτεροι). Δηλαδή η πιθανότητα ισοπαλίας ολοένα και μικραίνει και, σίγουρα, κάποια στιγμή η μία αλυσίδα γίνεται μακρύτερη από την άλλη, άρα θεωρείται έγκυρη.

3. Εάν η Μίνα αλλάξει το coinbase transaction, τότε αλλάζει και το hash του συνολικού block. Επομένως, το μετασχηματισμένο block δε θεωρείται έγκυρο, καθώς το hash του δε συμφωνεί με τους περιορισμούς που απαιτεί το bitcoin system προκειμένου να θεωρηθεί έγκυρο ένα block.
Συνεπώς, προκειμένου το block να θεωρείται έγκυρο, η Μίνα οφείλει να βρει κατάλληλο nonce και να υπολογίσει το hash του νέου block, δηλαδή να κάνει όλη τη δουλειά από την αρχή (proof of work) για το block.