Υπολογιστική Κρυπτογραφία
4η Σειρά Ασκήσεων

Μαραβίτσας Παναγιώτης
Α.Μ: 03116206

25 Φεβρουαρίου 2021

Περιεχόμενα

Άσκηση 1

Ψάχνουμε το πλήθος των m για τα οποία ισχύει:

$$m^e \equiv m \ \textbf{(mod N)}$$

Ισοδύναμα χρησιμοποιώντας το Κινέζικο Θεώρημα Υπολοίπων ψάχνουμε το πλήθος των διανυσμάτων $(m_p, m_q) \in \mathbb{Z}_p \times \mathbb{Z}_q$ έτσι ώστε να ικανοποιούνται οι σχέσεις:

$$\begin{cases} \mathrm{m}_p^e \equiv m_p \ (\mathrm{mod\ p}) \\ \mathrm{m}_q^e \equiv m_q \ (\mathrm{mod\ q}) \end{cases}$$

Για την πρώτη εξίσωση διακρίνουμε δύο περιπτώσεις:

Περίπτωση 1: $gcd(m_p, p) > 1$
Τότε, αφού ο p είναι πρώτος παίρνουμε τη μοναδική λύση:

$$m_p \equiv 0 \ (\mathrm{mod\ p})$$

Περίπτωση 2: $gcd(m_p, p) = 1$

$$m_p^e \equiv m_p \ (\mathrm{mod\ p}) \implies m_p^{e-1} \equiv 1 \ (\mathrm{mod\ p})$$

Έστω g ένας γεννήτορας της ομάδας $\mathbb{Z}_p^*$. Το πλήθος των λύσεων της παραπάνω εξίσωσης ισούται με το πλήθος των $k \in \mathbb{Z}_p$ για τα οποία ισχύει:

$$(g^k)^{e-1} \equiv 1 \ (\mathrm{mod\ p})$$

Τα k τα οποία ικανοποιούν αυτή τη σχέση είναι ακριβώς τα στοιχεία του συνόλου $\{\lambda \frac{p-1}{gcd(e-1,p-1)} | \lambda \in \mathbb{Z}\}$ όμως αφού $k \in \mathbb{Z}_p$ το πλήθος των λύσεων είναι ακριβώς $gcd(e-1, p-1)$.

Από τις παραπάνω περιπτώσεις προκύπτουν τελικά $1+gcd(e-1, p-1)$ το πλήθος m_p που ικανοποιούν την πρώτη σχέση. Με όμοιο τρόπο προκύπτουν $1 + gcd(e-1, q-1)$ το πλήθος m_q που ικανοποιούν τη δεύτερη σχέση.

Τελικά το πλήθος των σταθερών σημείων του RSA για δημόσιο κλειδί (pq, e) είναι $[1 + gcd(e-1, p-1)][1 + gcd(e-1, q-1)]$.

Άσκηση 2

Έστω ιδιωτικό κλειδί x, δημόσιο κλειδί $y = g^x$ και μήνυμα m. Τότε μία έγκυρη υπογραφή για το m είναι η:

$$(m, -\mathcal{H}(m) \text{ mod p-1}, g^{-\mathcal{H}(m)}y^{-1} \text{ mod p})$$

Πράγματι αν $a = -\mathcal{H}(m)$ mod p-1 και $b = g^{-\mathcal{H}(m)}y^{-1}$ mod p έχουμε :

$$yb \equiv g^a \text{ (mod p)} \implies yg^{-\mathcal{H}(m)}y^{-1} \equiv g^{-\mathcal{H}(m)} \text{ (mod p)} \implies g^{-\mathcal{H}(m)} \equiv g^{-\mathcal{H}(m)} \text{ (mod p)}$$

και

$$g^{\mathcal{H}(m)}yb \equiv 1 \text{ (mod p)} \implies g^{\mathcal{H}(m)}yg^{-\mathcal{H}(m)}y^{-1} \equiv 1 \text{ (mod p)} \implies 1 \equiv 1 \text{ (mod p)}$$

Άρα αυτό το σχήμα υπογραφών δεν προστατεύει από επίθεση καθολικής πλαστογράφησης.

Άσκηση 3

Από το θεώρημα των Goldwasser, Micali, Tong η ύπαρξη αλγορίθμου πολυωνυμικού χρόνου που υπολογίζει τη συνάρτηση loc ισοδυναμεί με ύπαρξη αλγορίθμου πολυωνυμικού χρόνου που "σπάει" το RSA. Η συνάρτηση loc ορίζεται ως εξής:

$$loc_{e,N}(x^e \bmod \mathrm{N}) = \begin{cases} 0, \ \mathrm{x} \leq \frac{N}{2} \\ 1, \ \text{αλλιώς} \end{cases}$$

Η συνάρτηση αυτή μπορεί να χρησιμοποιηθεί σε συνδυασμό με το malleability του RSA. Συγκεκριμένα αν $Enc_{e,N}(m_1) = c_1$ και $Enc_{e,N}(m_2) = c_2$:

$$Enc_{e,N}(m_1 \cdot m_2) = c_1 \cdot c_2$$

Χρησιμοποιώντας τα παραπάνω μπορεί να πραγματοποιηθεί δυαδική αναζήτηση του m στο $\mathbb{Z}_N$ χρησιμοποιώντας το γεγονός ότι:

- $loc_{e,N}(Enc_{e,N}(m)) = 0 \iff m \in [0, \frac{N}{2})$
- $loc_{e,N}(Enc_{e,N}(2m)) = 0 \iff m \in [0, \frac{N}{4}) \bigcup (\frac{N}{2}, \frac{3N}{4})$
- ...

Με αυτόν τον τρόπο, χρησιμοποιώντας ένα μαντείο loc μπορούμε να ανακτήσουμε το m σε $logN$ βήματα.

Ο κώδικας επισυνάπτεται.

Άσκηση 4

Έστω ότι ένα σχήμα δέσμευσης διαθέτει την ιδιότητα της τέλειας δέσμευσης. Τότε ακόμα και ένας unbounded αποστολέας δεν μπορεί να βρει δύο διαφορετικά μηνύματα που να επαληθεύουν κάποια δέσμευση, άρα σε κάθε δέσμευση αντιστοιχεί ακριβώς ένα μήνυμα που την επαληθεύει. Τότε όμως ένας unbounded παραλήπτης μπορει ακόμα και με χρήση brute force να βρει το μοναδικό μήνυμα που επαληθεύει τη δέσμευση άρα το σχήμα αυτό δεν έχει την ιδιότητα της τέλειας απόκρυψης.

Για την αντίθετη κατεύθυνση, έστω ένα σχήμα δέσμευσης που διαθέτει την ιδιότητα της τέλειας απόκρυψης. Τότε ένας unbounded παραλήπτης δεν μπορεί να βρει το μήνυμα με το οποίο προέκυψε η δέσμευση αφού υπάρχουν παραπάνω από ένα μηνύματα που την επαληθεύουν. Τότε όμως ένας unbounded αποστολέας μπορεί να βρει αυτά τα διαφορετικά μηνύματα και να αλλάξει το αρχικό μήνυμα κατα το "άνοιγμα" της δέσμευσης.

Επομένως ένα σχήμα δέσμευσης δεν μπορεί να διαθέτει ταυτόχρονα τις ιδιότητες της τέλειας δέσμευσης και της τέλειας απόκρυψης.

Άσκηση 5

Η δομή του κώδικα είναι η εξής:

1. Επιλέγονται τυχαίοι πρώτοι αριθμοί p,q τουλάχιστον 512 bits έτσι ώστε

$$p = 2q + 1$$

2. Επιλέγεται τυχαία κάποιος γεννήτορας της υποομάδας της $\mathbb{Z}_p^*$ τάξης q ως εξής:
   Επιλέγεται τυχαία κάποιο $g \in \mathbb{Z}_p^* - \{1, p-1\}$. Αν $g^{\frac{p-1}{2}} = -1$ τότε το g είναι γεννήτορας της Z_p^* και το g^2 έχει τάξη q, αλλιώς επαναλαμβάνουμε τη διαδικασία επιλέγοντας καινούριο g.

3. Επιλέγεται τυχαία κάποιο $x \in \mathbb{Z}_q^*$ και υπολογίζεται το $h = g^x \pmod{\text{p}}$

4. Για την υπογραφή επιλέγουμε $t \in_R \mathbb{Z}_q^*$ και υπολογίζουμε το $y = g^t \pmod{\text{p}}$

5. Υπολογίζουμε την τιμή $c = \mathcal{H}(y||m)$, όπου m το το hash του μεγάλου αρχείου, καθώς και την τιμή $s = t - cx \pmod{\text{q}}$

6. Για τον έλεγχο της υπογραφής, ο verifier ελέγχει αν $c = \mathcal{H}(g^s h^c||m)$

Ο κώδικας επισυνάπτεται.

Άσκηση 6

Completeness

Έχουμε ότι:

$$r^e \equiv (tm^c)^e \equiv t^e m^{ec} \equiv hy^c \pmod{\text{n}}$$

Άρα ο honest verifier $\mathcal{V}$ πάντα αποδέχεται με honest prover $\mathcal{P}$.

Soundness

Ο malicious prover $\mathcal{P}^*$ επιλέγει κάποιο $c' \in \mathbb{Z}_n^*$ και στέλνει στον honest verifier $\mathcal{V}$:

$$h = t^e y^{-c'} \bmod \text{n}$$

Αν ο $\mathcal{V}$ επιλέξει $c = c'$ τότε ο $\mathcal{P}$ θέτει $r = t$. Τότε:

$$hy^c \equiv t^e y^{-c} y^c \equiv t^e \equiv r^e \pmod{\text{n}}$$

Όμως αυτό συμβαίνει με πιθανότητα $\frac{1}{e}$ που είναι αμελητέα.

Special Soundness

Έστω δύο επιτυχείς εκτελέσεις του πρωτοκόλλου με transcripts $(h, c, r), (h, c', r')$.

$$\begin{cases} r^e \equiv hy^c \pmod{\text{n}} \\ r'^e \equiv hy^{c'} \pmod{\text{n}} \end{cases} \implies r^e y^{-c} \equiv r'^e y^{-c'} \pmod{\text{n}} \implies r^e m^{-ce} \equiv r'^e m^{-c'e} \pmod{\text{n}} \implies$$

$$(rm^{-c})^e \equiv (r'm^{-c'})^e \pmod{\text{n}} \implies rm^{-c} \equiv r'm^{-c'} \pmod{\text{n}} \implies rr'^{-1} \equiv m^{c-c'} \pmod{\text{n}}$$

Για να βρούμε το m πρέπει να λύσουμε το σύστημα:

$$\begin{cases} y \equiv m^e \pmod{\text{n}} \\ rr'^{-1} \equiv m^{c-c'} \pmod{\text{n}} \end{cases}$$

Έχουμε ότι $gcd(e, c - c') = 1$ αφού ο e είναι πρώτος και $c, c' \in \mathbb{Z}_e$. Επομένως:

$$\exists t_1, t_2 \in \mathbb{Z} : t_1 e + t_2 (c - c') = 1$$

Τα t_1, t_2 μπορούν να υπολογιστούν εύκολα μέσω του επεκτεταμένου αλγορίθμου του Ευκλείδη. Τελικά ο υπολογισμός του m είναι εύκολος αφού:

$$y^{t_1}(rr'^{-1})^{t_2} = m^{t_1 e} + m^{t_2(c-c')} = m^{t_1 e + t_2(c-c')} = m$$

Επομένως με δύο επιτυχείς εκτελέσεις του πρωτοκόλλου μπορούμε να εξάγουμε τον witness.

Honest Verifier Zero Knowledge

Έστω simulator $\mathcal{S}$ που δεν έχει το x και τίμιος verifier $\mathcal{V}$. Η επικοινωνία γίνεται ως εξής:

- Ο $\mathcal{S}$ δεσμεύεται στην τιμή $h = t^e y^{-c'} \bmod \text{n}, t \in_R \mathbb{Z}_n^*$.
- Ο $\mathcal{V}$ επιλέγει $c \in_R \mathbb{Z}_n^*$.
- Αν $c = c'$, τότε ο $\mathcal{S}$ μπορεί απαντήσει σωστά (αυτό συμβαίνει με αμελητέα πιθανοτητα $\frac{1}{e}$. Αλλιώς κάνει rewind.
- Ο $\mathcal{S}$ αυτή τη φορά δεσμεύεται στην τιμή $h = t^e y^{-c} \bmod \text{n}$.
- Ο $\mathcal{V}$ είναι honest και χρησιμοποιεί το ίδιο random tape άρα στέλνει στον $\mathcal{S}$ το challenge c.
- Ο $\mathcal{S}$ στέλνει στον $\mathcal{V}$ r=t.
- Ο $\mathcal{V}$ αποδέχεται αφού $hy^c \equiv t^e y^{-c} y^c \equiv t^e \equiv r^e \pmod{\text{n}}$.

Είναι προφανές ότι τα 2 transcripts της κανονικής και της simulated εκτέλεσης του πρωτοκόλλου που είναι $(t \in \mathbb{Z}_n^*, c \in \mathbb{Z}_e^*; t^e, c, tm^c)$ και $(t \in \mathbb{Z}_n^*; t^e y^{-c}, c, t)$ αντίστοιχα έχουν την ίδια κατανομή. Συγκεκριμένα:

- Τα στοιχεία $t^e, t^e y^{-c}$ είναι ομοιόμορφα επιλεγμένα στοιχεία του $\mathbb{Z}_n^*$ οπότε εμφανίζονται με πιθανότητα $\frac{1}{n}$.
- Το c και στα δύο transcripts είναι ομοιόμορφα επιλεγμένο στοιχείο του $\mathbb{Z}_e^*$ και εμφανίζεται με πιθανότητα $\frac{1}{e}$.
- Τέλος τα στοιχεία tm^c, t είναι ομοιόμορφα επιλεγμένα στοιχεία του $\mathbb{Z}_n^*$ και εμφανίζονται με πιθανότητα $\frac{1}{n}$.

Τελικά το πρωτόκολλο αυτό είναι Σ-πρωτόκολλο.

Άσκηση 7

Έστω malicious prover $\mathcal{P}^*$ που δεν γνωρίζει το x και honest verifier $\mathcal{V}$. Τότε ο $\mathcal{P}^*$ μπορεί να πείσει τον $\mathcal{V}$ να κάνει accept με τον ακόλουθο τρόπο:

- Ο $\mathcal{P}^*$ στέλνει στον $\mathcal{V}$ το $y = h^{-1}$.
- Ο $\mathcal{V}$ επιλέγει τυχαία κάποιο $c \in \mathbb{Z}_m^*$ και το στέλνει στον $\mathcal{P}^*$.
- Ο $\mathcal{P}^*$ στέλνει στον $\mathcal{V}$ το $s = c$.
- Ο $\mathcal{V}$ επιβεβαιώνει ότι $yg^c h \equiv h^{-1} g^c h \equiv g^c \equiv g^s \pmod{\text{p}}$ οπότε αποδέχεται.

Τελικά το πρωτόκολλο δεν έχει την ιδιότητα του soundness, επομένως δεν είναι πρωτόκολλο μηδενικής γνώσης για τίμιους επαληθευτές.

Άσκηση 8

1.

Έστω συνάρτηση σύνοψης $\mathcal{H} : \{0,1\}^* \to \{0,1\}^n$ η οποία είναι collision resistant και ασφαλής για PoW. Κατασκευάζουμε τη συνάρτηση $\mathcal{H}' : \{0,1\}^* \to \{0,1\}^{2n}$ η οποία ορίζεται ως:

$$\mathcal{H}'(x) = \mathcal{H}(x)||\mathcal{H}(x)$$

Η $\mathcal{H}'(x)$ δεν είναι ασφαλής για PoW αφού όλα τα outputs που παράγει ανήκουν στο σύνολο $Y = \{(x||x)|x \in \{0,1\}^n\}$ το οποίο είναι σημαντικά μικρότερο του $\{0,1\}^{2n}$, αφού $|Y| = 2^n$ ενώ το πλήθος των δυαδικών συμβολοσειρών μήκους 2n είναι 2^{2n}. Επίσης, έστω ότι υπάρχει πολυωνυμικός αλγόριθμος $\mathcal{A}$ ο οποίος μπορεί να βρει collisions για την $\mathcal{H}'$. Τότε μπορεί να βρει x_1, x_2 με $x_1 \neq x_2$:

$$\mathcal{H}'(x_1) = \mathcal{H}'(x_2) \implies \mathcal{H}(x_1)||\mathcal{H}(x_1) = \mathcal{H}(x_2)||\mathcal{H}(x_2) \implies \mathcal{H}(x_1) = \mathcal{H}(x_2)$$

Έτσι λοιπόν μπορούμε να βρούμε σύγκρουση στην $\mathcal{H}$. Άρα η $\mathcal{H}'$ είναι collision resistant.

2.

Δεν διευκρινίζεται από την εκφώνηση τι ιδιότητες έχει η $\mathcal{H}$ άρα υποθέτουμε ότι είναι collision resistant και ασφαλής για PoW, σύμφωνα με το συμβολισμό της υπόδειξης του ερωτήματος 1.

Για οποιοδήποτε x μπορούμε να βρούμε κατάλληλο r ώστε το $\mathcal{G}(x||r)$ να ανήκει είτε στο σύνολο $\{(x||0)|x \in \{0,1\}^n\}$ (για r της μορφής $r = (y||0)$) είτε στο σύνολο $\{(x||1)|x \in \{0,1\}^n\}$ (για r της μορφής $r = (y||1)$) Τα σύνολα αυτά έχουν μέγεθος 2^n, μισό απ' ότι το σύνολο τιμών της $\mathcal{G}$, οπότε δεν είναι σημαντικά μικρότερα από αυτό. Έχοντας έλεγχο του τελευταίου bit του output της $\mathcal{G}$ δεν μπορούμε να πετύχουμε κάτι καλύτερο ως προς το μέγεθος του συνόλου Y. Επομένως δεδομένου ότι η $\mathcal{H}$ είναι ασφαλής για PoW τότε και η $\mathcal{G}$ είναι ασφαλής για PoW.

Έστω ότι η $\mathcal{G}$ δεν έχει αντίσταση πρώτου ορίσματος. Τότε υπάρχει πολυωνυμικός αλγόριθμος $\mathcal{A}$ που για είσοδο y υπολογίζει κάποιο x έτσι ώστε $\mathcal{G}(x) = y$. Μπορούμε λοιπόν να κατασκευάσουμε αλγόριθμο $\mathcal{B}$ ο οποίος με είσοδο y υπολογίζει κάποιο x έτσι ώστε $\mathcal{H}(x) = y$ ως εξής:

- Ο $\mathcal{B}$ με είσοδο y δίνει στον $\mathcal{A}$ είσοδο $y||0$.
- Ο $\mathcal{A}$ επιστρέφει κάποιο x ώστε $\mathcal{G}(x) = y||0 \implies \mathcal{H}(x)||LSB(x) = y||0$.
- Ο $\mathcal{B}$ δίνει έξοδο x.

Αυτό βέβαια είναι άτοπο αφου η $\mathcal{H}$ είναι collision resistant και συνεπώς έχει αντίσταση πρώτου ορίσματος. Επομένως η $\mathcal{G}$ έχει αντίσταση πρώτου ορίσματος.

Άσκηση 9

1.

Μπορεί δύο miners που ακολουθούν πιστά το πρωτόκολλο να τελειώσουν ταυτόχρονα τα blocks τους. Αφού οι miners είναι honest έχουν ξεκινήσει από το ίδιο block και τα hash τους δείχνουν σε αυτό. Αν και η πιθανότητα να γίνει αυτό είναι μικρή, αυτό είναι ένα σενάριο στο οποίο δημιουργούνται δύο διαφορετικές αλυσίδες στο δίκτυο του bitcoin.

2.

Ακόμα κι αν δημιουργηθούν δύο διαφορετικές αλυσίδες σε κάποια χρονική στιγμή, η πιθανότητα να συνεχίσουν να μεγαλώνουν και οι δύο αλυσίδες είναι εξαιρετικά μικρή. Αυτό συμβαίνει επειδή οι επόμενοι miners συνεχίζουν να προσθέτουν blocks στο μεγαλύτερο κλάδο του δικτύου. Η πιθανότητα να τελειώσουν πάλι ταυτόχρονα δύο miners τα blocks των 2 αλυσίδων είναι εξαιρετικά μικρή, έτσι καθώς δημιουργούνται κανούρια blocks η πιθανότητα να συνεχίσουν να μεγαλώνουν και οι δύο αλυσίδες μειώνεται εκθετικά. Σύντομα κάποια αλυσίδα θα γίνει μεγαλύτερη από την άλλη με μεγάλη πιθανότητα και θα είναι αυτή απ' όπου θα συνεχιστεί το πρωτόκολλο από αυτό το σημείο και μετά.

3.

Προκειμένου η Μίνα να πραγματοποιήσει αυτή την αλλαγή πρέπει να δημιουργήσει εκ νέου το Merkle Tree, συνεπώς πρέπει να βρει εκ νέου κάποιο κατάλληλο nonce. Όσο όμως η Μίνα εκτελεί το PoW για να βρει το nonce, οι άλλοι miners έχουν ξεκινήσει ήδη να δουλεύουν για το επόμενο block, του οποίο το hash θα δείχνει στο πρώτο block που δημιουργήθηκε και όχι στης Μίνας. Έτσι δεν θα βγάλει επιπλέον κέρδη.