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

Υπολογιστική Κρυπτογραφία

Πέμπτη σειρά ασκήσεων

Σπουδαστής
Παπασκαρλάτος Αλέξανδρος (Α.Μ.: 03111097)

Ημερομηνία Υποβολής: 27 Ιουλίου 2020

Άσκηση 1

Η απάντηση προκύπτει άμεσα από τις διαφάνειες του μαθήματος.

Χωρίς pairings (σε 3 γύρους)

1. Ο A στέλνει το y_a στον B, ο B στέλνει το y_b στον C, ο C στέλνει το y_c στον A (κυκλικά).
2. Ο A υπολογίζει το $t_a = y_c^{xc} = g^{xc\ xa}$, ο B υπολογίζει το $t_b = y_a^{xB} = g^{xb\ xa}$ και ο C υπολογίζει το $t_c = y_b^{xc} = g^{xb\ xc}$
3. Ο A στέλνει το t_a στον B, ο B στέλνει το t_b στον C, ο C στέλνει το t_c στον A (πάλι κυκλικά).
4. Όλοι υπολογίζουν το κοινό κλειδί ως εξής:

• Ο A με $t_c^{xb} = g^{xb\ xc\ xa}$
• Ο B με $t_a^{xb} = g^{xc\ xa\ xb}$
• Ο C με $t_b^{xc} = g^{xa\ xb\ xc}$

Με pairings (σε ένα γύρο)

Υποθέτουμε δύο ομάδες G, G με τάξη ένα πρώτο q
και μία συμμετρική διγραμμική ζεύξη $e : G \times G \rightarrow G_T$.

• Όλοι οι συμμετέχοντες εκπέμπουν τα δημόσια κλειδιά τους $y_a = g^{xa}$, $y_b = g^{xb}$, $y_c = g^{xc}$.
• Με την βοήθεια της ζεύξης το κοινό κλειδί μπορεί να υπολογιστεί ως εξής:
    • $e(g^{xb}, g^{xc})^{xa} = e(g, g)^{xb\ xc\ xa}$
    • $e(g^{xa}, g^{xc})^{xb} = e(g, g)^{xa\ xc\ xb}$
    • $e(g^{xa}, g^{xb})^{xc} = e(g, g)^{xa\ xb\ xc}$

Άσκηση 2

$x^2 \equiv 119 \pmod{209}$

Είναι $209 = 11 \cdot 19$
Επειδή το 209 είναι γινόμενο δύο πρώτων και gcd(119, 209) = 1,
θα έχουμε 4 τετραγωνικές ρίζες (ή 0) .

Ισχύουν τα κάτωθι:

- $x^2 \equiv 119 \equiv 9 \pmod{11} \Rightarrow x \equiv \pm 3 \pmod{11}$
- $x^2 \equiv 119 \equiv 5 \pmod{19} \Rightarrow x \equiv \pm 9 \pmod{11}$

Προκύπτουν 4 αποδεκτά x:

- $x \equiv 3 \pmod{11}$, $x \equiv 9 \pmod{19}$ $\Rightarrow$ $x \equiv 47 \pmod{209}$
- $x \equiv 3 \pmod{11}$, $x \equiv -9 \equiv 10 \pmod{19}$ $\Rightarrow$ $x \equiv 124 \pmod{209}$
- $x \equiv -3 \equiv 8 \pmod{11}$, $x \equiv 9 \pmod{19}$ $\Rightarrow$ $x \equiv 85 \pmod{209} \equiv -124$
- $x \equiv -3 \equiv 8 \pmod{11}$, $x \equiv -9 \equiv 10 \pmod{19}$ $\Rightarrow$ $x \equiv 162 \pmod{209} \equiv -47$

Λόγω του μικρού μεγέθους των πρώτων παραγόντων, μπόρεσα να βρω τα αποτελέσματα με δοκιμές

πχ για $x \equiv 3 \pmod{11}$, $x \equiv 9 \pmod{19}$, είναι $11 \mid (x-3) \quad \wedge \quad (x = 19k + 9)$
άρα $11 \mid 19k + 6$
Δοκιμάζω τιμές για το $k \in \mathbb{N}$, ώστε να βρω μία που να ικανοποιεί την παραπάνω σχέση.
Ισχύει για k = 2, οπότε προκύπτει x = 47

Παρουσιάζω και έναν πιο γενικευμένο τρόπο, εύρεση του αποτελέσματος με CRT:

πχ για $x \equiv 3 \pmod{11}$, $x \equiv -9 \equiv 10 \pmod{19}$
Είναι $M_1 = 209 / 11 = 19$, $N_1 \equiv M_1^{-1} \equiv 19^{-1} \equiv 8^{-1} \equiv 7 \pmod{11}$
Είναι $M_2 = 209 / 19 = 11$, $N_2 = M_2^{-1} \equiv 11^{-1} \equiv 7 \pmod{19}$

Επομένως, $x \equiv b_1 M_1 N_1 + b_2 M_2 N_2 \equiv 3 \cdot 19 \cdot 7 + 10 \cdot 11 \cdot 7 \equiv 1169 \equiv 124 \pmod{209}$

Οι άλλες δύο ρίζες μπορούν να βρεθούν με ένα από τους παραπάνω τρόπους ή, ευκολότερα, ως οι αντίθετες των ήδη υπολογισμένων ριζών:

Ισχύει $x^2 \equiv (-x)^2 \pmod{n}$, οπότε οι 4 τετραγωνικές ρίζες μπορούν να χωριστούν σε δύο ζεύγη αντιθέτων στο (mod 209) : (47, 162) και (124, 85).

Άσκηση 3

1. Είναι $H_{g,g}(x,y) = g^x \cdot g^y \pmod p = g^{x+y} \pmod p$
Η $H_{g,g}$ δεν έχει αντίσταση δεύτερου ορίσματος, καθώς δεδομένου ενός ζεύγους (x,y), προκύπτει σύγκρουση με κάθε (x', y'), όπου $x' + y' \equiv x + y \pmod q$,
καθώς τότε: $H_{g,g}(x,y) = g^{x+y} \pmod p = g^{x'+y'} \pmod p = H_{g,g}(x',y')$
Τέτοια (x', y') είναι εύκολα υπολογίσιμα.
Πρόκειται για τα $((x+k) \bmod q, (y-k) \bmod q)\ \forall k$

Ωστόσο, η $H_{g,g}$ έχει αντίσταση πρώτου ορίσματος ανν το DLP είναι δυσεπίλυτο.
Block άτοπο: Έστω η $H_{g,g}$ δεν έχει αντίσταση πρώτου ορίσματος

- Δεδομένου του $H_{g,g}(x,y)$, μπορούμε να βρούμε (x', y') ώστε $H_{g,g}(x',y') = H_{g,g}(x,y)$
- Έστω δίνεται $f = g^z \pmod p$, με γνωστό g, άγνωστο z και ζητείται να βρεθεί το z
  Τότε, θεωρούμε $f = g^z$ ως το αποτέλεσμα της $H_{g,g}$ για κάποια άγνωστη είσοδο.
  Αντιστρέφουμε την $H_{g,g}$ και βρίσκουμε (x, y), ώστε $g^z = g^{x+y} \pmod p \Rightarrow z = x+y \pmod q$
  Συνεπώς, δεδομένου μόνο του $f = g^z$ και της αντιστρεψιμότητας της $H_{g,g}$, βρήκαμε το z, δηλαδή λύσαμε το πρόβλημα διακριτού λογαρίθμου.

Άτοπο, καθώς το DLP θεωρείται δυσεπίλυτο .
Επομένως, η $H_{g,g}$ έχει αντίσταση πρώτου ορίσματος.

2. Έστω πως θέλω να λύσω το DLP στη <g>, δηλαδή δεδομένου $f = g^z \pmod p$, θέλω να βρω το z. Ανν $f \equiv 1$, τότε $z \equiv 0$.

Διαφορετικά, επειδή ord(<g>) = q πρώτος, προκύπτει πως $ord_p(f) = q$, ορίζεται η $H_{g,f}$ σύμφωνα με την εκφώνηση. Κάνω τα εξής:
Θεωρώ την $H_{g,f}(x,y) = g^x f^y \pmod p = g^x g^{zy} \pmod p = g^{x+yz} \pmod p$
Βρίσκω σύγκρουση: $H_{g,f}(x,y) = H_{g,f}(x',y') \Rightarrow g^{x+yz} \pmod p = g^{x'+y'z} \pmod p$
$\Rightarrow x + yz \equiv x' + y'z \pmod q \quad \Rightarrow z\,(y-y') \equiv x'-x \pmod q$
$\Rightarrow z \equiv (x'-x)\,(y-y')^{-1} \pmod q$
Άρα z είναι εύκολα υπολογίσιμο δεδομένων των x, x', y, y' .
Σημειώνουμε πως παραπάνω θεωρήθηκε πως υπάρχει ο αντίστροφος του (y-y'). Αυτό ισχύει ανν $(y-y') \bmod q \in \mathbb{Z}_q^*$, δηλαδή ανν $(y-y') \not\equiv 0 \pmod q$
Αυτό ισχύει και το αποδεικνύουμε με άτοπο:
Block άτοπο: Έστω $(y-y') \equiv 0 \pmod q$

- Ισχύει $z\,(y-y') \equiv x'-x \pmod q \ \wedge\ (y-y') \equiv 0 \pmod q \quad \Rightarrow x' - x \equiv 0 \pmod q$
- $x, x', y, y' \in \mathbb{Z}_q \ \wedge\ x \equiv x' \pmod q \ \wedge\ y = y' \pmod q \quad \Rightarrow \quad (x = x' \ \wedge\ y = y')$

Άτοπο, καθώς $(x,y) \neq (x',y')$ από τον ορισμό της σύγκρουσης .
Επομένως, $(y-y') \not\equiv 0 \pmod q$.

3.

Θα ορίσουμε αλγόριθμο ο οποίος να λύνει το DLP στην <g> σύμφωνα με τα παραπάνω.
Έστω δίνεται $f = g^z$ και θέλω να βρω το z.

Θεωρώ τη συνάρτηση $H_{g,f}(x,y)$.
Κάνω απλή επίθεση γενεθλίων, δηλαδή διαλέγω τυχαία ζεύγη (x_i,y_i) και υπολογίζω τις τιμές $h_i = H_{g,f}(x,y)$. Από αρχή του περιστερώνα, επειδή $h_i \in$ <g> (πληθικότητας q) .
Μάλιστα, από θεωρία πιθανοτήτων, η πιθανότητα να έχουμε βρει σύγκρουση μετά από τυχαία επιλογή (x_1,y_1), (x_2,y_2), …, (x_k,y_k) είναι περίπου ½ όταν $k \simeq 1.17(\sqrt{q})$.
Οι συγκρίσεις στα h_i μπορούν να γίνουν αποδοτικά σε O(1) με χρήση hash table.
Συνεπώς, μπορούμε να βρούμε σύγκρουση σε $O(\sqrt{q})$ average complexity.
Στη συνέχεια, έχοντας βρει σύγκρουση στην $H_{g,f}$ υπολογίζουμε το z, ώστε $g^z = f$, όπως στο υποερώτημα (2), δηλαδή για σύγκρουση στα (x,y), (x',y'), είναι $z \equiv (x'-x)(y-y')^{-1} \pmod q$

Για τον brute force αλγόριθμο, οφείλουμε να δοκιμάσουμε ενδεχόμενες τιμές για το $z' \in \mathbb{Z}_q$ και να ελέγχουμε κάθε φορά αν $g^{z'} = f$. Συνεπώς, χρειαζόμαστε O(q) δοκιμές.

Άρα, ο αλγόριθμός μας είναι καλύτερος από τον brute force.

Αξίζει να σημειώσουμε πως στην πραγματικότητα, ο αλγόριθμός μας είναι ψεύδο-ριζικός, καθώς η είσοδός μας δεν έχει μέγεθος q, αλλά $\sim\lambda = \log(q)$, δηλαδή λ το πλήθος των bit των διαφόρων δεδομένων.
Η πολυπλοκότητα του brute force είναι $O(2^{\lambda})$, ενώ η πολυπλοκότητα του αλγορίθμου μας είναι $O(2^{\lambda/2})$, δηλαδή ασυμπτωτικά είναι ίδια.

Επίσης, αξίζει να αναφερθούμε στον γνωστό αλγόριθμο Baby step - Giant step (Shanks) ο οποίος παρουσιάζεται στις διαφάνειες του μαθήματος.

Ο αλγόριθμος Baby step - Giant step (Shanks) έχει χρονική πολυπλοκότητα $O(\sqrt{q}) = O(2^{\lambda/2})$.

Αξίζει να σημειώσουμε πως επειδή ord(G) = q , όπου q πρώτος, δεν μπορούμε να εφαρμόσουμε τον αλγόριθμο Pohlig-Hellman.

Άσκηση 4

α. Ο Β μπορεί να κλίνει το $b_A \oplus b_B$ στην τιμή 0 πάντα, ανεξάρτητα από το σύστημα κρυπτογράφησης.
Πράγματι, αρκεί να δημοσιοποιήσει $c_B = c_A$.
Τότε, κατά την αποκρυπτογράφηση, θα προκύψει $b_B = b_A \Rightarrow b_A \oplus b_B = b_A \oplus b_A = 0$

β. Χρησιμοποιούμε εκθετικό ElGamal:
$Enc(r,m) = (g^r \bmod p, g^m p_k^r \bmod p)$, όπου $p_k = g^x$, x το ιδιωτικό κλειδί

Η Α καθώς παίζει τίμια, επιλέγει r, b και δημοσιοποιεί $(g^r \bmod p, g^b p_k^r \bmod p) = (c_1 , c_2)$

Ο Β μπορεί να κλίνει το $b_A \oplus b_B$ στην τιμή 1 δημοσιοποιώντας:
$(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p) = (g^{r(q-1)} \bmod p, g\,(g^b p_k^r)^{q-1})$

Διακρίνω 2 περιπτώσεις:

- $\underline{b_A = 0}$
  $b_A = 0 \Rightarrow g^b = 1 \Rightarrow c_2 = p_k^r \bmod p$
  $\Rightarrow Dec(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p) = g\, c_2^{q-1} / (c_1^{q-1})^x = g\, p_k^{r(q-1)} / g^{r(q-1)x} = g\, g^{xr(q-1)} / g^{xr(q-1)}$
  $= g = g^1 \Rightarrow b_B = 1$
  Δηλαδή η Τ, αποκρυπτογραφώντας το $(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p)$, κατέληξε πως $b_B = 1$.
  Οπότε προκύπτει $b_A \oplus b_B = 0 \oplus 1 = 1$

- $\underline{b_A = 1}$
  $b_A = 1 \Rightarrow g^b = g \Rightarrow c_2 = g\, p_k^r \bmod p$
  $\Rightarrow Dec(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p) = g\, c_2^{q-1} / (c_1^{q-1})^x = g\,(g\, p_k^r)^{(q-1)} / g^{r(q-1)x}$
  $= g\, g^{q-1}\, p_k^{r(q-1)} / g^{xr(q-1)} = g^q\, g^{xr(q-1)} / g^{xr(q-1)} =$
  $= g^q = 1 = g^0 \Rightarrow b_B = 0$
  Δηλαδή η Τ, αποκρυπτογραφώντας το $(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p)$, κατέληξε πως $b_B = 0$.
  Οπότε προκύπτει $b_A \oplus b_B = 1 \oplus 0 = 1$

Επομένως, καταλήγουμε πως αν ο Β δημοσιοποιήσει το $(c_1^{q-1} \bmod p, g\, c_2^{q-1} \bmod p)$, τότε θα προκύψει $b_A \oplus b_A = 1$ σε κάθε περίπτωση.

Άσκηση 5

Έστω n = pq, p, q πρώτοι, και $p \equiv q \equiv 3 \pmod 4$.
Έστω η συνάρτηση: $SQ(x) = \min\{x^2 \bmod n, n - x^2 \bmod n\}$, όπου $0 < x < n/2$

1. Διακρίνουμε δύο περιπτώσεις:
i. gcd(x, n) = 1
Είναι $x \in UZ_{pq}$, όπου UZ_{pq} πολλαπλασιαστική ομάδα, οπότε $x^2, n-x^2 \in UZ_{pq}$, άρα $SQ(x) \in UZ_{pq}$.

Η $y = x^2 \bmod n$ έχει είτε 0 ή 4 ρίζες, οι οποίες μπορούν να χωριστούν σε δύο ζεύγη αντιθέτων $(r_1, -r_1)$ και $(r_2, -r_2)$
Επειδή, n είναι γινόμενο περιττών, το n είναι περιττός, οπότε $\forall\ r \in \mathbb{Z}$, ισχύει $r \neq n/2$
Ισχύει η ισοδυναμία $r_1 < n/2 \Leftrightarrow n-r_1 > n/2$
Δηλαδή, ακριβώς μία εκ των r_1 , $-r_1$ ανήκει στο (0, n/2).
Όμοια για τα r_2 , $-r_2$.

Συμπεραίνουμε, πως για x με $SQ(x) = x^2 \bmod n$, υπάρχει ακριβώς ένα x' με $x' \neq x$ και $SQ(x') = x'^{\,2} \bmod n$, για το οποίο ισχύει SQ(x) = SQ(x').

Όμοια με τα παραπάνω, η $y = n - x^2 \bmod n \Leftrightarrow n - y = x^2 \bmod n$ έχει είτε 0 ή 4 ρίζες. Αν έχει ρίζες, ακριβώς οι 2 εκ των 4 βρίσκονται στο (0, n/2) .

Συμπεραίνουμε, πως για x με $SQ(x) = n - x^2 \bmod n$, υπάρχει ακριβώς ένα x' με $x' \neq x$ και $SQ(x') = n - x'^{\,2} \bmod n$, για το οποίο ισχύει SQ(x) = SQ(x').

Μένει να δείξω πως για x με $SQ(x) = x^2 \bmod n$, δεν υπάρχει y με $SQ(y) = n - y^2 \bmod n$, για το οποίο ισχύει SQ(x) = SQ(y).

Block άτοπο: Έστω υπάρχουν τέτοια x, y

- Είναι $x^2 \equiv n - y^2 \pmod n$ $\Rightarrow$ $x^2 \equiv n - y^2 \pmod p$ $\Rightarrow x^2 \equiv - y^2 \pmod p$
- Θέτω $a = - y^2$
- Από κριτήριο Euler, η ισοτιμία $a \equiv x^2 \pmod p$ έχει λύση αν $a^{(p-1)/2} \equiv 1 \pmod p$
- Είναι $\mathbf{p \equiv 3 \pmod 4}$ $\Rightarrow p = 4k + 3 \Rightarrow p-1 = 4k + 2 \Rightarrow (p-1)/2 = 2k + 1$, δηλαδή το (p-1)/2 είναι περιττός αριθμός
- $a^{(p-1)/2} \equiv (-y^2)^{(p-1)/2} \equiv (-1)^{(p-1)/2}\, y^{(p-1)} \equiv (-1)^{2k+1} \equiv -1 \pmod p$
- Άρα από κριτήριο Euler, η ισοτιμία $x^2 \equiv - y^2 \pmod p$ δεν έχει λύση

Άτοπο
Επομένως, δεν υπάρχουν τέτοια x, y.

Τελικά, καταλήξαμε πως για κάθε $x \in (0,n/2)$ με $x \in UZ_{pq}$ και SQ(x), υπάρχει ακριβώς ένα $x' \in (0,n/2)$ με $x' \neq x$ και SQ(x') = SQ(x) .

ii. gcd(x, n) > 1

Ισχύει gcd(x, n) = p $\oplus$ gcd(x,n) = q

ΧΒΤΓ, θεωρούμε πως gcd(x, n) = p $\Rightarrow$ x = k · p, για κάποιο k ϵ $\mathbb{Z}$

Είναι 0 < x < n/2 $\Rightarrow$ 0 < k · p < pq / 2 $\Rightarrow$ 0 < k < q/2

Είναι kp $\notin$ UZ_{pq} και $(kp)^2$ $\notin$ UZ_{pq} και $(n - (kp)^2) = (p\,(q-k^2p))$ $\notin$ UZ_{pq} , άρα SQ(x) $\notin$ UZ_{pq}

- Έστω $SQ(x) = x^2 \bmod n$
  - Ισχύει $x^2 \equiv (k\,p)^2 \equiv 0 \pmod p$
  - και $x^2 \equiv (k\,p)^2 \pmod q$
  - Είναι k < q/2 $\wedge$ q πρώτος $\Rightarrow$ gcd(k,q) = 1
  - gcd(k,q) = 1 $\wedge$ gcd(p,k) = 1 $\Rightarrow$ $(k\,p)^2$ ϵ $\mathbb{Z}_q^*$
  - Η $x^2 \equiv (k\,p)^2 \pmod q$ έχει 2 ρίζες στο $\mathbb{Z}_q^*$, την kp και την -kp
  - Η kp (mod q) αντιστοιχεί στην kp (mod n) = kp , από CRT καθώς
    - kp ≡ 0 (mod p)
    - kp ≡ kp (mod q)

    Δηλαδή επαληθεύονται οι CRT συνιστώσες.
  - Η -kp (mod q) αντιστοιχεί στην -kp (mod n) = n - kp , από CRT καθώς
    - n - kp ≡ pq - kp ≡ p (q - k) ≡ 0 (mod p)
    - n - kp ≡ pq - kp ≡ - kp (mod q)

    Δηλαδή επαληθεύονται οι CRT συνιστώσες.
  - Όμως, 0 < kp < n /2 $\Rightarrow$ n > n - kp > n/2, άρα η n - kp δεν ανήκει στο (0, n/2)

  Καταλήγω πως αν ισχύει gcd(x, n) = p και $SQ(x) = x^2$, δεν υπάρχει x' ϵ (0, n/2) με x' ≠ x και $SQ(x') = x'^2 \bmod n$

- Όμοια με τα παραπάνω, θεωρώντας x = kp ϵ (0,n/2) με $SQ(x) = n - x^2 \bmod n$, καταλήγω πως δεν υπάρχει x' ϵ (0,n/2) με x' ≠ x και $SQ(x') = x'^2 \bmod n$

- Όμοια με το block άτοπο της προηγούμενης σελίδας (αλλά στο mod q), καταλήγω πως το $(kp)^2$ δεν μπορεί να ισούται με τον αντίθετο κανενός τετραγώνου στο mod n.

Με τα παραπάνω βήματα, αφού δεν έβλαψα τη γενικότητα, καταλήγω στα ίδια συμπεράσματα για τα y ϵ (0,n/2) με gcd(x,n) = q .

Τελικά, καταλήξαμε πως για κάθε x ϵ (0,n/2) με x $\notin$ UZ_{pq} και SQ(x),

δεν υπάρχει κανένα x' ϵ (0,n/2) με x' ≠ x και SQ(x') = SQ(x) .

Συνολικά

Τελικά, καταλήγουμε πως η SQ είναι (το πολύ 2)-προς-1.

Επίσης, $\forall$ x ϵ (0,n/2), ($x^2 \bmod n < n/2$ $\vee$ $n - x^2 \bmod n < n/2$) $\Rightarrow$ SQ(x) ϵ [1, (n-1)/2]

καθώς η ισοτιμία $x^2 = 0$ έχει μοναδική λύση το x = 0 $\notin$ (0, n/2)

Οπότε, η SQ είναι (το πολύ 2)-προς-1 στο {1, …, (n-1)/2 }

2.

Block άτοπο: Έστω η SQ δεν είναι ελεύθερη συγκρούσεων

- Μπορώ να βρω x, y ϵ (0, n/2) με x ≠ y και SQ(x) = SQ(y)
- Επειδή x ≠ y και SQ(x) = SQ(y), από υποερώτημα (1), προκύπτει x,y ϵ UZ_{pq}
- Ισχύει πως x ≠ -y καθώς δε γίνεται να ισχύει y ϵ (0, n/2) $\wedge$ -y ϵ (0, n/2)
- Συνεπώς, από το υποερώτημα (1), υπάρχουν δύο περιπτώσεις:
    a. $SQ(x) = x^2 \bmod n \wedge SQ(y) = y^2 \bmod n$
    b. $SQ(x) = n - x^2 \bmod n \wedge SQ(y) = n - y^2 \bmod n$

  Και στις 2 περιπτώσεις, ισχύει $x^2 \equiv y^2 \pmod n$, ΧΒΤΓ θεωρώ x > y
- $x^2 \equiv y^2 \pmod n \Rightarrow x^2 \equiv y^2 \pmod p \Rightarrow p \mid (x^2 - y^2) \Rightarrow p \mid (x-y)(x+y)$
- p πρώτος $\wedge$ $p \mid (x-y)(x+y) \Rightarrow p \mid (x-y) \vee p \mid (x+y)$
- Είναι $0 < y < x < n/2 \Rightarrow 0 < (x-y) < (x+y) < n = pq$
- $(p \mid (x-y) \vee p \mid (x+y)) \Rightarrow (\gcd((x-y), n) = p \vee \gcd((x+y), n) = p)$
- Συνεπώς, δεδομένων των x, y ϵ (0, n/2) με x ≠ y και SQ(x) = SQ(y), βρήκαμε το p.

Άτοπο, καθώς η εύρεση του p θεωρείται υπολογιστικά ανέφικτη από εκφώνηση.
Επομένως, η SQ είναι ελεύθερη συγκρούσεων.

3.

Ομολογώ πως είναι δύσκολο να φτιάξω ζητούμενη SQ' χωρίς να κάνω παραδοχές που δεν έχω αποδείξει.

1ος τρόπος

Επεκτείνω την SQ: θεωρώ την f: $[0, (n-1)/2] \rightarrow [0, (n-1)/2]$

με $f(x) = \begin{cases} SQ(x), & 0 < x \le (n-1)/2 \\ 0, & x = 0 \end{cases}$

Η f είναι ελεύθερη συγκρούσεων εφόσον η SQ είναι ελεύθερη συγκρούσεων.

Θεωρώ την $f'(x) = f(x) \bmod 2^{511}$, με $f': \{0,1\}^{1022} \rightarrow \{0,1\}^{511}$, δηλαδή κρατάω μόνο τα μισά least significant bits.
Σημειώνω πως δεν μπορώ να ορίσω πεδίο ορισμού της f' το $\{0,1\}^{1024}$, καθώς n ϵ $\{0,1\}^{1024}$ και $x < n/2 \Rightarrow x \epsilon \{0,1\}^{1023}$. Όμως, και πάλι δεν μπορώ να θεωρήσω αξιόπιστο πεδίο ορισμού το $\{0,1\}^{1023}$, καθώς $x < (n-1)/2 \le 2^{1023} - 1$. Συνεπώς, για να είμαι αξιόπιστα στο πεδίο ορισμού, περιορίζω το x να είναι μικρότερο του $2^{1022} \Rightarrow x \epsilon \{0,1\}^{1022}$

Εδώ κάνω την αυθαίρετη παραδοχή πως η f'(x) δεν εμφανίζει συγκρούσεις.

Στη συνέχεια, υλοποιώ την επέκταση συνάρτησης σύνοψης Merkle - Damgard με συνάρτηση βάσης την $f'(x) : \{0,1\}^{1022} \rightarrow \{0,1\}^{511}$
Στο τέλος, κάνω padding με μηδενικά, ώστε να περάσω στο μέγεθος 1024.

Δημιουργώ έτσι SQ' η οποία είναι ελεύθερη συγκρούσεων αν η f'(x) είναι ελεύθερη συγκρούσεων.

2ος τρόπος

Θεωρώ τη συνάρτηση $f : \{0,1\} \times [1, (n-1)/2] \rightarrow \{0,1\}^{1023}$

Θεωρώ $f(0, x) = SQ(x)$,

$f(1, x) = SQ((n+1)/2 - x)$

Σημειώνουμε πως $p \equiv q \equiv 3 \pmod 4$, άρα $p = 4k + 3$ και $q = 4m + 3$

Επομένως, $n = pk = (4k + 3)(4m + 3) \Rightarrow n \equiv 9 \equiv 1 \pmod 4 \Rightarrow n = 4d + 1 \Rightarrow (n+1)/2 = 2d + 1$,
άρα το $(n+1)/2$ είναι περιττό, οπότε για $\forall\ x \in \mathbb{Z}$, ισχύει $x \neq (n+1)/2 - x$

Θεωρώ $h_0(x) = f(0, x)$

$h_1(x) = f(1, x)$

Οι h_1 , h_2 είναι ελεύθερες συγκρούσεων εφόσον η SQ είναι ελεύθερη συγκρούσεων.

Θεωρώ $a \in [2, (n-1)/2]$. Πρόκειται για τιμή επιλεγμένη τυχαία, σταθερή.

Έστω SQ' με είσοδο ένα string $s \in \{0,1\}^*$

Ορίζω $SQ'(s) = 0 \,||\, f^s(a)$

Όπου για $b \in \{0,1\}$, $w \in \{0,1\}^+$, $f^{b\,||\,w}(x) = f^w (f(b,x))$ και $f^b (x) = f(b, x)$

Δηλαδή παίρνω ένα ένα τα bit του string s και για κάθε τέτοιο bit b_{i-1}, υπολογίζω την
$x_i = f(b_{i-1}, x_{i-1}) = h_{b\,i-1} (x_{i-1})$, με $x_0 = a$
Ουσιαστικά, παίρνω αρχικό σπόρο την a και στη συνέχεια εφαρμόζω την f στο σπόρο, όπως ορίζουν τα bit του s.

Στο τέλος, κάνω padding ενός bit, για να φτάσω τα 1024 bit.

Για παράδειγμα για s = 0110101, $SQ'(s) = 0 \,||\, h_1(h_0(h_1(h_0(h_1(h_1(h_0(a)))))))$

Άσκηση 6

1. Ισχύει $enc^t(c) = c \Rightarrow enc(\ enc^{t-1}(c)\ ) = c \Rightarrow enc^{t-1}(c) = m$
Άρα $m = enc^{t-1}(c)$

2. Υπάρχει RSA-βρόχος για κάθε $c \in \mathbb{Z}_n$.
Αν ακολουθήσουμε το μονοπάτι για αρκετή ώρα, θα εμφανίζει επαναλήψεις από αρχή του περιστερώνα, καθώς το σύνολο όλων των κρυπτοκειμένων είναι πεπερασμένο (έχει άνω όριο το 2^{λ} , όπου λ το πλήθος των bits του n) .

Θα δείξουμε πως το πρώτο στοιχείο που επαναλαμβάνεται είναι το $c = enc^0(c)$

Block άτοπο : Έστω συναντάμε την πρώτη επανάληψη στα στοιχεία i,j με $0 < i < j$

- $enc^i(c) = enc^j(c)$, με $0 < i < j$
- $enc^i(c) = enc(\ enc^{i-1}(c)\ ) \ \wedge \ enc^j(c) = enc(\ enc^{j-1}(c)\ ) \Rightarrow \quad enc^{i-1}(c) = enc^{j-1}(c)$
- Συνεπώς, συναντάμε επανάληψη στα i-1, j-1

Άτοπο, καθώς υποθέσαμε πως συναντάμε την πρώτη επανάληψη στα i, j .
Επομένως, το πρώτο στοιχείο που επαναλαμβάνεται είναι το $c = enc^0(c)$

Οπότε έχουμε έναν RSA-βρόχο.

Είναι $c = m^e \bmod n$,
$enc(c) = c^e \bmod n$, $enc^{i+1}(c) = enc(\ enc^i(c)\ ) = (enc^i(c)\ )^e \bmod n$
Επαγωγικά προκύπτει πως $enc^i(c) = c^{e\wedge i} \bmod n$

3. Για t το μήκος του RSA-βρόχου, έχουμε:
$enc^t(c) = c \ \Leftrightarrow c^{e\wedge t} \equiv c \pmod n \ \Leftrightarrow \ e^t \equiv 1 \pmod{\varphi(n)}$
Καθώς το t είναι ο ελάχιστος (μη μηδενικός) αριθμός ώστε να ισχύει $enc^t(c) = c$, προκύπτει πως είναι ο ελάχιστος αριθμός για να ισχύει $e^t \equiv 1 \pmod{\varphi(n)}$, άρα το e έχει τάξη t στην ομάδα $UZ_{\varphi(n)}$, δηλαδή την ομάδα σχετικά πρώτων με το φ(n).
Η $UZ_{\varphi(n)}$ έχει $\varphi(\varphi(n))$ στοιχεία, άρα ένα άνω όριο για το t είναι το $\varphi(\varphi(n))$.
Το άνω όριο αυτό δεν είναι tight. Όπως προείπα, ένα tight όριο είναι το $ord_{\varphi(n)}(e)$ που όμως δεν είναι συνάρτηση μόνο του n (όπως ζητείται στην εκφώνηση) .

4. Τα παραπάνω σημαίνουν πως με t κρυπτογραφήσεις, μπορούμε να αποκρυπτογραφήσουμε ένα κρυπτοκείμενο.
Άρα αρκούν $O(\varphi(\varphi(n))$ κρυπτογραφήσεις για να σπάσουμε το RSA.
Σημειώνω πως βρίσκοντας το t, μπορούμε να βρούμε και το ιδιωτικό κλειδί d, καθώς:
$e^t \equiv 1 \pmod{\varphi(n)} \Leftrightarrow e\ e^{t-1} \equiv 1 \pmod{\varphi(n)} \Leftrightarrow e^{t-1} \equiv e^{-1} \equiv d \pmod{\varphi(n)}$

Άσκηση 7

$c = commit(m, r) = (g^r, g^m h^r) = (c_1 , c_2)$

Δέσμευση
Ο αποστολέας δεσμεύεται στο $(c_1 , c_2) = (g^r , g^m h^r)$

Είναι $ord_p(g) = q$ και $r \epsilon \mathbb{Z}_q$
Άρα, αν ο αποστολέας δεσμευτεί στο $c_1 = g^r$, δεν μπορεί να βρει $r' \epsilon \mathbb{Z}_q$ με $r' \neq r$, ώστε $c_1 = g^{r'}$
Αυτό συμβαίνει ακόμα και αν ο αποστολέας είναι παντοδύναμος.
Κυριολεκτικά δεν υπάρχει τέτοιο r'.

Συνεπώς, έχουμε τέλεια δέσμευση στο r.

$c_2 \equiv g^m h^r \pmod p \Leftrightarrow g^m \equiv c_2 h^{-r} \pmod p$

Άρα, με αμετάβλητα τα c_2 , h, r, p, προκύπτει πως το $m \epsilon \mathbb{Z}_q$ είναι μοναδικό.
Από προηγούμενο βήμα, είδαμε πως έχουμε τέλεια δέσμευση στο r και οι υπόλοιπες συνιστώσες είναι γνωστές στον παραλήπτη (μετά το commit).

Συνεπώς, έχουμε τέλεια δέσμευση στο m.

Τελικά, το δοθέν σχήμα δέσμευσης διαθέτει **τέλεια δέσμευση**, καθώς δεν υπάρχουν m, m', r, r' $\epsilon \mathbb{Z}_q$ με $(m', r') \neq (m, r)$ και $commit(m,r) = commit(m',r')$

Απόκρυψη

Καταρχάς, αναφέρουμε πως, όπως έχουμε δείξει σε προηγούμενη σειρά ασκήσεων, ένα σχήμα δέσμευσης δεν μπορεί να διαθέτει ταυτόχρονα τέλεια δέσμευση και τέλεια απόκρυψη.
Συνεπώς, το σχήμα δε διαθέτει τέλεια απόκρυψη.
Πράγματι, εάν ο παραλήπτης είναι μη φραγμένος, τότε μπορεί να λύσει το DLP στην G (πχ με brute force), οπότε να βρει το ιδιωτικό κλειδί x. Στη συνέχεια, μπορεί να υπολογίσει το $c_2 / c_1^x = g^m$ και τέλος να βρει το m αφού μπορεί να λύσει το DLP στη G.

Θα δείξουμε πως αν μπορούμε να σπάσουμε το σχήμα δέσμευσης, τότε μπορούμε να λύσουμε το CDHP.

Έστω πως, δεδομένου του $(c_1 , c_2) = (g^r, g^m h^r)$ μπορούμε να βρούμε το m, δηλαδή, ας υποθέσουμε πως έχουμε ένα oracle που ορίζεται συναρτήσει των p, q, $h = g^x$ και παίρνει ως είσοδο το (c_1 , c_2) και εξάγει το m.
Έστω έχουμε g^a, g^b και θέλουμε να υπολογίσουμε το g^{ab} (Computational DHP) .
Υποθέτουμε πως a είναι το άγνωστο ιδιωτικό κλειδί και $h=g^a$ το γνωστό δημόσιο.
Περνάω στο oracle είσοδο το (g^b, w), όπου $w \in \mathbb{Z}_p$ τυχαίο.
Το oracle δίνει έξοδο m και ισχύει $w = g^m h^b = g^m g^{a b}$.
Εφόσον, πήραμε ως έξοδο το m, υπολογίζουμε το g^m και στη συνέχεια το g^{-m} (με EGCD) .
Συνεπώς, μπορούμε να υπολογίσουμε το $g^{-m} w = g^{-m} g^m g^{a b} = g^{a b}$
Επομένως, δεδομένων των g^a, g^b, υπολογίσαμε το $g^{a b}$, δηλαδή λύσαμε το CDHP.

Επιπλέον, δεν μπορούμε να σπάσουμε το σχήμα δέσμευσης αν το DLP είναι δυσεπίλυτο.
Πράγματι, βλέπουμε πως ακόμα και αν έχουμε το ιδιωτικό κλειδί (ή ακόμα και το αλάτι r), δεδομένων των $(c_1 , c_2) = (g^r, g^m h^r)$, μπορούμε να πάρουμε μόνο το g^m (όπως στον στάνταρ εκθετικό El Gamal), οπότε και πάλι πρέπει να βρούμε το m με δεδομένο μόνο το $y = g^m$, δηλαδή να λύσουμε το DLP.

Τελικά, βλέπουμε πως το δοθέν σχήμα δέσμευσης διαθέτει **υπολογιστική απόκρυψη**, εάν το CDHP είναι δύσκολο στη G ή αν το DLP είναι δύσκολο στη G (δεδομένων τυχόντων περιορισμών για το m) .

Άσκηση 8

1. Έστω r το μέγεθος του block του AES_k .
Έστω είσοδος $x = x_1 \,||\, x_2$ με $|x_1| = r$
και αντίστοιχη έξοδος $y = y_1 \,||\, y_2$ με $|y_1| = r$

Τότε για είσοδο $x' = x_1 \,||\, x_2'$, θα λάβουμε έξοδο $y' = y_1 \,||\, y_2'$
Δηλαδή, εάν τα πρώτα r bit της εισόδου μείνουν αμετάβλητα, το ίδιο θα συμβεί για τα πρώτα r bit της εξόδου, όπου r το μέγεθος του block.

Συνεπώς, σχεδιάζουμε τον κάτωθι αλγόριθμο:

- Θεωρούμε ένα αυθαίρετο μήκος d.
- Θέτουμε στο oracle είσοδο $x_1 = a_1 \,||\, b_1$ με $|a_1| = d$ και λαμβάνουμε έξοδο y.
- Θέτουμε είσοδο $x_1' = a_1 \,||\, b_1'$, δηλαδή κρατάμε τα πρώτα d bit αμετάβλητα και αλλάζουμε τα επόμενα. Παίρνουμε έξοδο y'
  Καλό θα ήταν να επιβάλουμε $b_1[0] \neq b_1'[0]$, δηλαδή το πρώτο bit του b_1 να είναι διαφορετικό από το πρώτο bit του b_1' (ώστε να παραμείνουν αμετάβλητα μόνο το prefix μεγέθους d).
- Εξετάζουμε αν $\exists\ m \leq d$, τ.ω. το prefix μεγέθους m του y να είναι ίδιο με το prefix μεγέθους m του y', δηλαδή αν τα πρώτα m bit της εξόδου παρέμειναν αμετάβλητα.
  - Εάν δεν υπάρχει τέτοιο m, αυτό σημαίνει πως το d που επιλέξαμε είναι μικρότερο από το μέγεθος του μπλοκ, οπότε διπλασιάζουμε το d και επαναλαμβάνουμε από την αρχή.
  - Εάν υπάρχει τέτοιο m, τότε, για σιγουριά, επαναλαμβάνουμε τη διαδικασία από το προηγούμενο βήμα, θέτοντας $x_1'' = a_1 \,||\, b_1''$, $x_1''' = a_1 \,||\, b_1'''$, $x_1'''' = a_1 \,||\, b_1''''$, ώστε να εξαλείψουμε την πιθανότητα να έμειναν τα πρώτα m bit αμετάβλητα από σύμπτωση.
    Σε όλες τις περιπτώσεις, οφείλει να υπάρχει m τ.ω. το prefix μεγέθους m του y να είναι ίδιο με το prefix μεγέθους m του y' και του y'' και y''' και y''''. Αν δεν ισχύει κάτι τέτοιο, διπλασιάζουμε το d και επαναλαμβάνουμε τη διαδικασία από την αρχή.
- Φτάνουμε σε αυτό το βήμα αν υπάρχει τέτοιο m.
  Επιλέγω το m_{max} . Το m_{max} αντιστοιχεί σε n block, δηλαδή $m_{max} = n \cdot r$.
- Σε αυτό το σημείο, μπορούμε να βρούμε το r :
  - είτε δοκιμάζοντας όλα τα υποπολλαπλάσια του m_{max} , έστω r' και βρίσκοντας το ελάχιστο υποπολλαπλάσιο για το οποίο τα πρώτα r' bit της εξόδου παραμείνουν αμετάβλητα, εφόσον τα πρώτα r' bit της εισόδου παραμείνουν αμετάβλητα,
  - ή κάνοντας δυαδική αναζήτηση για το r. Δηλαδή είναι $1 \leq r \leq m_{max}$, οπότε, κάνω δυαδική αναζήτηση στο $[1, m_{max}]$ θέτοντας κάθε φορά d = (high + low)/2 και εξετάζοντας σε κάθε βήμα αν υπάρχει m για το οποίο τα πρώτα m bit της εξόδου παραμένουν αμετάβλητα αν παραμείνουν αμετάβλητα τα πρώτα m bit της εξόδου.

2. Βρίσκουμε το μέγεθος του block, r, όπως στο υποερώτημα (1) .
Στη συνέχεια, θέτουμε είσοδο $x = x_1 || x_1$, με $|x_1| = r$.
Αν λάβουμε έξοδο της μορφής $y = y_1 || y_1$, δηλαδή αν τα πρώτα r bit ταυτίζονται με τα επόμενα r bit, τότε το AES_k χρησιμοποιεί ECB mode.
Τυπικά, υπάρχει μια αμελητέα πιθανότητα, το AES_k να μη χρησιμοποιεί ECB mode και το αποτέλεσμα να ήταν μια απίθανη σύμπτωση. Εάν θέλουμε να είμαστε ακόμα πιο σίγουροι, μπορούμε να δοκιμάσουμε και άλλες εισόδους (συνολικού μήκους 2r η καθεμία):
$x_2 || x_2$, $x_3 || x_3$ κοκ.
Εάν το oracle χρησιμοποιεί ECB mode, σε κάθε έξοδο, τα πρώτα r bit θα ταυτίζονται με τα δεύτερα r bit.

3. Θεωρούμε πως μας δίνεται ένα κρυπτοκείμενο c και πρόσβαση στο encryption oracle AES_k (ECB) . Δεν κάνουμε άλλες παραδοχές (πχ παραδοχές που να επιτρέπουν plaintext injection attack) .

Μπορούμε να κάνουμε brute force σε κάθε δυνατό block.
Συγκεκριμένα, χωρίζουμε το κρυπτοκείμενο σε υπο-string μεγέθους r.
Στη συνέχεια, δοκιμάζουμε κάθε δυνατό string μεγέθους r και το κρυπτογραφούμε με το oracle.
Ελέγχουμε εάν η εκάστοτε έξοδος ταυτίζεται με κάποιο υπο-string του κρυπτοκειμένου.
Για καλύτερη ταχύτητα σύγκρισης, μπορούμε να έχουμε ταξινομήσει τα υπο-string ή να τα έχουμε περάσει σε ένα hash table.
Η μέθοδος αυτή έχει χρονική πολυπλοκότητα $O(2^r + L)$ όπου L το μέγεθος του κρυπτοκειμένου c (αν θεωρήσουμε πως οι διάφορες λειτουργίες στο hash table είναι O(1)) και γραμμική χωρική πολυπλοκότητα.

Άσκηση 9

Σημειώνουμε πως για να είναι το δεδομένα μας καλώς ορισμένα, πρέπει $q_1 \leq q_2 \leq q_3$.
Αυτό συμβαίνει, γιατί η f_i με σύνολο ορισμού το G_i (τάξης q_i) είναι αντιστρέψιμη, άρα είναι 1-1.
Επομένως, το G_{i+1} έχει τουλάχιστον όσα στοιχεία έχει το G_i .

1. Για να αποκρυπτογραφήσουμε το (b_1 , b_2 , a_3 , b_3) ακολουθούμε την εξής διαδικασία

- $f_3(a_2) = Dec_{g3,\ sk3}(a_3 , b_3) = b_3 / a_3^{sk3}$ (G_3)
- $a_2 = f_3^{-1}(f_3(a_2))$
- $f_2(a_1) = Dec_{\ g2,\ sk2}(a_2 , b_2) = b_2 / a_2^{sk2}$ (G_2)
- $a_1 = f_2^{-1}(f_2(a_1))$
- $m = Dec_{g1,\ sk1}(a_1 , b_1) = b_1 / a_1^{sk1}$ (G_1)

2. Γνωρίζουμε πως το απλό ElGamal διαθέτει ασφάλεια IND-CPA, αν το DDHP είναι δύσκολο (όπου πράγματι θεωρείται δύσκολο από εκφώνηση).
Διαισθητικά, αντιλαμβανόμαστε, πως το 3-ElGamal, το οποίο χρησιμοποιεί τρεις (ανεξάρτητους) ElGamal, θα διαθέτει επίσης IND-CPA. Ας το δείξουμε και τυπικά:

- Έστω πως το 3-ElGamal δε διαθέτει ασφάλεια IND-CPA.
- Υπάρχει A, ο οποίος μπορεί να νικήσει στο παιχνίδι CPA με μη αμελητέα πιθανότητα.
- Κατασκευάζω B ο οποίος μπορεί να κερδίσει το παιχνίδι DDHP με μη αμελητέα πιθανότητα:
    - Ο B λαμβάνει τριάδα στοιχείων: g^a , g^b , g^c
    - Ο B φέρεται ως encryption oracle στο παιχνίδι CPA και χρησιμοποιεί τον A
    - Στο CPA-game είναι, $g_1 = g$, $pk_1 = g^a$. Τα g_2 , g_3 , pk_2 , pk_3 είναι αδιάφορα, οπότε τα επιλέγω τυχαία.
    - Ο B απαντά στις κρυπτογραφήσεις του A. Σημειώνουμε πως ο B μπορεί να κρυπτογραφήσει ένα plain text, γνωρίζοντας μόνο τα δημόσια κλειδιά.
    - Όταν ο A είναι έτοιμος για την πρόκληση, στέλνει δύο μηνύματα M_0, M_1:
        - Ο B διαλέγει ομοιόμορφα τυχαίο bit ϵ {0,1}
        - Ο B, στο πρώτο βήμα της κρυπτογράφησης, κρυπτογραφεί το $M_{\mathbf{bit}}$ με g^b και πολλαπλασιάζει με g^c. Δηλαδή θέτει $(a_1 , b_1) = (g^b , M_{bit}\ g^c)$
        - Ο B ακολουθεί τίμια τα υπόλοιπα βήματα της κρυπτογράφησης και στέλνει το τελικό αποτέλεσμα (b_1 , b_2 , a_3 , b_3)
        - Ο A υπολογίζει την τιμή του bit* και τη στέλνει στον B
    - Αν bit = bit*, ο B απαντά πως $g^c = g^{a\ b}$. Διαφορετικά, απαντά πως $g^c \neq g^{a\ b}$.

Ανάλυση

- Εάν η είσοδος είναι τριάδα DH, δηλαδή $g^c = g^{a\,b}$, τότε $g^c = (g^a)^b = pk_1^{\,b}$.
  Συνεπώς, ο Α έλαβε ένα έγκυρο κρυπτοκείμενο 3-ElGamal, οπότε και μάντεψε το bit με non-negl(λ) πλεονέκτημα, δηλαδή η πιθανότητα να μάντεψε σωστά είναι ½ + non-negl(λ)

- Εάν η είσοδος είναι τυχαία τριάδα, δηλαδή $g^c \neq g^{ab}$, τότε $g^c \neq pk_1^{\,b}$.
  Συνεπώς, ο Α δεν έλαβε έγκυρο κρυπτοκείμενο 3-ElGamal, οπότε μάντεψε το bit τυχαία με πιθανότητα επιτυχίας ½ .

- Έστω τα γεγονότα {DH, non-DH, yes, no}
  όπου DH : δόθηκε τριάδα DH
  non-DH : δόθηκε τυχαία τριάδα
  yes : ο Β απάντησε πως δόθηκε τριάδα DH
  no : ο Β απάντησε πως δόθηκε τυχαία τριάδα
  succ : ο Β απάντησε σωστά
  Τότε, $Pr(succ) = Pr(DH) \cdot Pr(yes \mid DH) + Pr(non\text{-}DH) \cdot Pr(no \mid non\text{-}DH)$
  $= ½ (½ + non\text{-}negl(λ)) + ½ (½) = ½ + non\text{-}negl$
- Συνεπώς, ο Β απαντά σωστά με μη αμελητέο πλεονέκτημα, άρα κερδίζει το παιχνίδι DDHP

Τελικά, το 3-ElGamal έχει την ιδιότητα IND-CPA, δεδομένου πως το DDHP είναι δύσκολο, δηλαδή δεδομένου πως ισχύει η υπόθεση DDH .

3. Το 3-ElGamal δεν είναι ουσιαστικά πιο ασφαλές από το ElGamal. Εάν μπορώ να σπάσω το ElGamal με δυσκολία Δ, τότε μπορώ να σπάσω το 3-ElGamal με δυσκολία 3Δ, καθώς συνίσταται στο να σπάσω τρεις ανεξάρτητες κρυπτογραφήσεις ElGamal.
Στην πράξη, το 3-ElGamal ίσως προσδίδει λίγη ασφάλεια με την έννοια πως εάν ένας επιτιθέμενος έχει ήδη σπάσει το ElGamal σε μία εκ των ομάδων G_1 , G_2 , G_3 , (είτε λόγω πρότερης δουλειάς στην ομάδα ή λόγω διαρροής ενός εκ των κλειδιών), τότε και πάλι χρειάζεται να σπάσει άλλους δύο ElGamal.

Άσκηση 10

1. Δεν αποτελεί Σ-πρωτόκολλο, καθώς με μία επιτυχή εκτέλεση του πρωτοκόλλου, αποκαλύπτεται ο witness στον Verifier.
Πράγματι, ο V στο τέλος γνωρίζει τα t, c, s, όπου $s = t + c\,x$
Άρα, ο V υπολογίζει το x ως εξής: $s = t + c\,x \Leftrightarrow x = c^{-1}(s - t)$.

Λοιπόν, το πρωτόκολλο δε διαθέτει HVZK, καθώς ένας Simulator S (ο οποίος δε γνωρίζει το x), δεν μπορεί να πείσει τον V (ακόμα και με rewind), καθώς μια επιτυχής εκτέλεση του πρωτοκόλλου θα αποκάλυπτε το witness x, τον οποίο δε γνωρίζει ο S.

Συνεπώς, το πρωτόκολλο δεν είναι Σ-πρωτόκολλο.

2. Δεν αποτελεί Σ-πρωτόκολλο, καθώς ένας κακόβουλος Prover P* μπορεί να επιτύχει στο πρωτόκολλο με πιθανότητα Pr = 1.
Πράγματι, στον τρίτο γύρο, ο Prover καλείται να υπολογίσει την ποσότητα $\sigma = g^{t - c\,x}$ δεδομένων των t, c, $h = g^x$.
Είναι $\sigma = g^{t - c\,x} = g^t\, g^{-c\,x} = g^t\,(g^x)^{-c} = g^t\,(h^c)^{-1}$,
όπου τα g^t , $(h^c)^{-1}$ υπολογίζονται εύκολα (χωρίς γνώση του x)

Λοιπόν, το πρωτόκολλο δε διαθέτει ειδική ορθότητα. Για να έχει ένα πρωτόκολλο ειδική ορθότητα, θα πρέπει δύο επιτυχείς εκτελέσεις του πρωτοκόλλου με την ίδια δέσμευση, αλλά διαφορετική πρόκληση, να αποκαλύπτουν τον witness.
Όμως, εδώ βλέπουμε, πως ένας P*, μπορεί να επιτύχει σε δύο εκτελέσεις του πρωτοκόλλου με την ίδια δέσμευση και διαφορετική πρόκληση, ενώ δε γνωρίζει τον witness x. Συνεπώς, αποκλείεται να αποκαλύπτεται ο witness από αυτές τις δύο επιτυχείς εκτελέσεις.

Συνεπώς, το πρωτόκολλο δεν είναι Σ-πρωτόκολλο.