Όνομα: ΑΜ:

Κρυπτογραφία – Γραπτή εξέταση Ιουνίου 2019

Οδηγίες: Απαντήστε και τα 4 (τέσσερα) θέματα. Δεν επιτρέπονται σημειώσεις ή υπολογιστικές συσκευές. Τα θέματα είναι ισοδύναμα. Μέγιστη βαθμολογία είναι το 120. Στις απαντήσεις σας να είστε σαφείς, να εξηγείτε τη σκέψη σας, και να απαντάτε στο ζητούμενο. Πριν αρχίσετε να λύνετε, διαβάστε όλα τα θέματα. Για τους μεταπτυχιακούς υπάρχουν επιπλέον ερωτήματα με κατάλληλη ένδειξη.

Θέμα 1. Θεωρήστε ένα 3-out-of-n σχήμα διαμοίρασης μυστικού του Shamir στο $\mathbb{Z}_{17}$, όπου ο παίκτης i λαμβάνει μερίδιο $y_i = P(i)$ (όπου $P(x)$ το πολυώνυμο που επέλεξε ο διαμοιραστής). Οι παίκτες $2, 5, 10$ έλαβαν μερίδια $1, 1, 2$, αντίστοιχα.

(α΄) Υπολογίστε το μυστικό. Για την παρεμβολή Lagrange δίνεται ο τύπος των συντελεστών λ_i, όπου $\lambda_i := \prod_{j \neq i}(x-j)/(i-j)$. Δείξτε αναλυτικά τους υπολογισμούς σας.

[Μονάδες: 15 (Μεταπτυχιακοί: 10)]

(β΄) Είναι δυνατό να ενταχθούν στο σχήμα παίκτες με δείκτη 0 ή -1; Εξετάστε τις περιπτώσεις ξεχωριστά και εξηγήστε την απαντησή σας.

[Μονάδες: 15 (Μεταπτυχιακοί: 10)]

(γ΄) **Μόνο Μεταπτυχιακοί.** Έστω ότι ένας νέος παίκτης (με αριθμό-δείκτη το 16) θέλει να ενταχθεί στο σχήμα. Ο διαμοιραστής (dealer) δεν είναι διαθέσιμος. Με δεδομένο ότι οι παίκτες είναι τίμιοι, εξηγήστε πως οι παίκτες $2, 5, 10$ μπορούν να στείλουν στον νέο παίκτη αρκετές πληροφορίες ώστε να υπολογίσει το μεριδιό του. Στη λύση σας δεν επιτρέπεται κάποιος παίκτης να είναι σε θέση να παράξει το μυστικό. Θεωρήστε ότι είναι εφικτό οι παίκτες να στέλνουν προσωπικά μηνύματα ο ένας στον άλλο.

[Μονάδες: 10 (Μόνο μεταπτυχιακοί)]

Λύση (α΄) Το ζητούμενό μας είναι να υπολογίσουμε το $p(0)$ για πολυώνυμο p το οποίο θα ανασυνθέσουμε από τα διαθέσιμα μερίδια μέσω παρεμβολής Lagrange. Ελέγχουμε ότι το πλήθος των μεριδίων που έχουμε είναι ακριβώς αυτό που χρειαζόμαστε (3 out of n και γνωρίζουμε 3 μερίδια αντίστοιχα). Στις πράξεις εργαζόμαστε modulo 17.

Το $p(x)$ γνωρίζουμε ότι θα είναι ίσο με $p(x) = y_2 \cdot \lambda_2(x) + y_5 \cdot \lambda_5(x) + y_{10} \cdot \lambda_{10}(x)$ οπότε $p(x) = 1 \cdot \lambda_2(x) + 1 \cdot \lambda_5(x) + 2 \cdot \lambda_{10}(x)$.

Μένει να υπολογίσουμε τα λ_i από τον τύπο που μας δόθηκε και να προσδιορίσουμε την τιμή του πολυωνύμου για $x = 0$.

Για το λ_2 έχουμε: $\lambda_2(x) = \frac{x-5}{2-5} \cdot \frac{x-10}{2-10} = \frac{x-5}{-3} \cdot \frac{x-10}{-8} = \frac{(x-5)\cdot(x-10)}{24} = \frac{(x-5)\cdot(x-10)}{7}$. Για να αντιστρέψουμε το 7, μπορούμε να χρησιμοποιήσουμε αλγόριθμο Ευκλείδη ή να παρατηρήσουμε ότι $5 \cdot 7 = 35 = 1 + 2 \cdot 17$, οπότε $7^{-1} = 5 \mod 17$. Τελικά, $\lambda_2(0) = (0-5) \cdot (0-10) \cdot 5 = 50 \cdot 5 = -1 \cdot 5 = 12$.

Για το λ_5 έχουμε: $\lambda_5(x) = \frac{x-2}{5-2} \cdot \frac{x-10}{5-10} = \frac{x-2}{3} \cdot \frac{x-10}{-5} = \frac{(x-2)\cdot(x-10)}{-15} = \frac{(x-2)\cdot(x-10)}{2}$. Για την αντιστροφή, υπολογίζουμε ότι $2^{-1} = 9 \mod 17$. Τελικά $\lambda_5(0) = 20 \cdot 9 = 180 = 10 \mod 17$.

Για το λ_{10} έχουμε: $\lambda_{10}(x) = \frac{x-2}{10-2} \cdot \frac{x-5}{10-5} = \frac{x-2}{8} \cdot \frac{x-5}{5} = \frac{(x-2)\cdot(x-5)}{6} =$. Για την αντιστροφή, υπολογίζουμε ότι $6^{-1} = 3 \mod 17$. Τελικά $\lambda_{10}(0) = (-2) \cdot (-5) \cdot 3 = 30 = 13 \mod 17$.

Άρα $p(0) = 1 \cdot \lambda_2(0) + 1 \cdot \lambda_5(0) + 2 \cdot \lambda_{10}(0) = 1 \cdot 12 + 1 \cdot 10 + 2 \cdot 13 = 48 = 14 \mod 17$

(β΄) Εάν εντάξουμε παίκτη με δείκτη 0, θα λάβει ως μερίδιο το $p(0) = s$ δηλαδή το μυστικό που θέλουμε να διαμοιράσουμε, γεγονός το οποίο θα γνωρίζει ο αντίστοιχος παίκτης. Για το -1 δεν υπάρχει ζήτημα (το αρνητικό πρόσημο δεν αλλάζει κάτι, και σε κάθε περίπτωση $-1 \equiv 16 \mod 17$).

Θέμα 2. (α΄) Να δώσετε τον ορισμό της Στατιστικής Απόστασης ανάμεσα σε δύο κατανομές.

[Μονάδες: 10]

(β΄) Έστω κυκλική ομάδα $\mathbb{G} = \langle g \rangle$ τάξης q, όπου q πρώτος μήκους λ bits. Να υπολογίσετε τη στατιστική απόσταση ανάμεσα στις παρακάτω κατανομές:

$$\mathcal{D}_\lambda := \left\{ a, b \xleftarrow{\text{r}} \mathbb{Z}_q \colon (g^a, g^b, g^{ab}) \right\}$$
$$\mathcal{R}_\lambda := \left\{ a, b, c \xleftarrow{\text{r}} \mathbb{Z}_q \colon (g^a, g^b, g^c) \right\}$$

[Μονάδες: 10]

(γ΄) Να διατυπώσετε την υπόθεση DDH και να εξηγήσετε εάν είναι συμβατή με τον παραπάνω υπολογισμό σας.

[Μονάδες: 10]

Λύση (α΄) Σημειώσεις, ορισμός 1.6.1 [2_mathreview_handout.pdf]

(β΄) Σημειώσεις, παράγραφος 6.3.1 [5_diffie_handout.pdf]. Επίσης είναι στις δυνατότητές σας να τη λύσετε και ως άγνωστη άσκηση.

(γ΄) Σημειώσεις, (βλ παραπάνω και επιπλέον 1.7 Statistical Tests). Ο παραπάνω υπολογισμός αφορά τη στατιστική απόσταση των δύο κατανομών (η οποία είναι μεγάλη). Η υπόθεση DDH είναι υπολογιστική, δηλαδή περιορίζεται σε αντιπάλους με πολυωνυμική υπολογιστική ισχύ. Οπότε αν και οι κατανομές έχουν μεγάλη απόσταση, υποθέτουμε ότι είναι δύσκολο να τις διαχωρίσουμε υπολογιστικά, άρα δεν υπάρχει λογική σύγκρουση. Επιπλέον[1], αν η στατιστική απόσταση ήταν μικρή, δεν θα είχαμε λόγο να αναπτύξουμε κάποια υπόθεση. Θα μπορούσαμε να επικαλεστούμε το ότι η στατιστική απόσταση είναι κάτω φράγμα για την υπολογιστική από τις σημειώσεις.

[1] Ο συλλογισμός αυτός δεν χρειάζεται στην απαντησή σας

Όνομα: ΑΜ:

Θέμα 3. (α΄) Έστω $\langle \mathcal{P}, \mathcal{V} \rangle$ ένα ζεύγος αλληλεπιδρόντων προγραμμάτων. Ορίζουμε ως $\text{out}^{\mathcal{P}}_{\mathcal{P},\mathcal{V}}(x, w, z)$ να είναι η έξοδος του $\mathcal{P}$ όταν οι $\mathcal{P}$ και $\mathcal{V}$ εκτελούνται με τη δημόσια είσοδο x και τις ιδιωτικές εισόδους w και z (ο $\mathcal{P}$ δέχεται τα x, w και ο $\mathcal{V}$ τα x, z). Το $\text{out}^{\mathcal{V}}_{\mathcal{P},\mathcal{V}}$ ορίζεται όμοια για το $\mathcal{V}$.

Να ορίσετε πότε ένα διαδραστικό πρωτόκολλο $\langle \mathcal{P}, \mathcal{V} \rangle$ είναι μια ***απόδειξη μηδενικής γνώσης (zero-knowledge proof)*** για μία γλώσσα $\mathcal{L}_R = \{x | \exists w : R(x, w) = 1\}$ (όπου R πολυωνυμικός αλγόριθμος με έξοδο στο $\{0, 1\}$) δίνοντας τις τρεις ιδιότητες :

- Πληρότητα (completeness)
- Εγκυρότητα (soundness)
- Μηδενική γνώση (zero-knowledge)

Για κάθε ιδιότητα να δώσετε ορισμό ακολουθώντας τους παραπάνω συμβολισμούς. Επίσης να εξηγήσετε τι σημαίνει η ιδιότητα μηδενικής γνώσης για τίμιους επαληθευτές (honest verifier zero-knowledge).

[Μονάδες: 15 (Μεταπτυχιακοί: 10)]

(β΄) Δίνεται μια παραλλαγή του πρωτοκόλλου των Chaum και Pedersen, με το οποίο θέλουμε να να αποδείξουμε την ισότητα δύο διακριτών λογαρίθμων, δηλαδή οτι κάποια πρόταση της μορφής $\langle \langle \mathbb{G}, g_1, g_2, q \rangle, h_1, h_2 \rangle$ ανήκει στη γλώσσα EQDLOG $:= \{\langle \langle \mathbb{G}, g_1, g_2, q \rangle, h_1, h_2 \rangle : h_1 = g_1^w \cap h_2 = g_2^w$ για κάποιο $w \in \mathbb{Z}_q\}$ χωρίς να φανερώσουμε το $w = \log_{g_i} h_i$.

i. Ο $\mathcal{P}$ διαλέγει $t \xleftarrow{\text{r}} \mathbb{Z}_q$ και στέλνει τα $y_1 = g_1^t, y_2 = g_2^t$ στον $\mathcal{V}$.

ii. Ο $\mathcal{V}$ διαλέγει μια πρόκληση $c \xleftarrow{\text{r}} \mathbb{Z}_q$ και την στέλνει στον $\mathcal{P}$.

iii. Ο $\mathcal{P}$ υπολογίζει το $s = t + w + c \bmod q$ και στέλνει το s στον $\mathcal{V}$.

iv. Ο $\mathcal{V}$ ελέγχει και αποδέχεται αν και μόνο αν $g_1^s = y_1 h_1 g_1^c$ και $g_2^s = y_2 h_2 g_2^c$.

Να εξετάσετε το παραπάνω πρωτόκολο ως προς την πληρότητα, εγκυρότητα, και μηδενική γνώση τίμιου επαληθευτή (να εξεταστούν και οι τρεις ιδιότητες).

[Μονάδες: 15 (Μεταπτυχιακοί: 10)]

(γ΄) **Μόνο Μεταπτυχιακοί.** Στο μάθημα έχουμε δείξει ότι το πρωτόκολο του Schnorr επιτυγχάνει μηδενική γνώση μόνο απέναντι σε τίμιους επαληθευτές. Η παρακάτω σκιαγράφηση απόδειξης φέρεται να περιγράφει ένα Simulator μέσω του οποίου επιτυγχάνουμε μηδενική γνώση απέναντι σε οποιοδήποτε επαληθευτή $\mathcal{V}^*$.

> Το πρωτόκολο του Schnor επιτυγχάνει τέλεια μηδενική γνώση απέναντι σε οποιοδήποτε επαληθευτή $\mathcal{V}^*$. Προς τούτο, θα δώσουμε κατάληλο simulator πιθανοτικού πολυωνυμικού χρόνου $S_{\mathcal{V}*}$. Ο simulator $S_{\mathcal{V}*}$ εργάζεται ως εξής: επιλέγουμε τυχαία $c^*, s \leftarrow \mathbb{Z}_q$ και θέτουμε $y = g^s / h^{c^*}$. Εκκινούμε τον V^* με τυχαιότητα ρ και στέλνουμε το y. Εάν ο V^* επιστρέψει c^*, ολοκληρώνουμε επιτυχώς, διαφορετικά επαναλαμβάνουμε με νέα ρ, c^*, s. Για να ολοκληρώσουμε την απόδειξη, παρατηρούμε ότι όποτε ο S τερματίζει επιτυχημένα, η τριάδα y, c^*, s έχει την ίδια κατανομή με μια τριάδα $\bar{y}, \bar{c}, \bar{s}$ από μια πραγματική εκτέλεση.
>
> Συγκεκριμένα, το $y = g^s / h^{c*}$ είναι ομοιόμορφα κατανεμημένο στο $\langle g \rangle$ αφού το g^s είναι ομοιόμορφα κατανεμημένο στο $\mathbb{Z}_q$. Ομοίως το $\bar{y}$ ενός τίμιου Prover είναι της μορφής g^t και άρα επίσης ομοιόμορφα κατανεμημένο. Για τα $c^*, \bar{c}$ παρατηρούμε ότι $c^* = f_{\mathcal{V}^*}(y, \rho)$ και αντίστοιχα $\bar{c} = f_{\mathcal{V}^*}(\bar{y}, \rho)$, για κάποια συνάρτηση $f_{\mathcal{V}^*}$ που υπολογίζει ο $\mathcal{V}^*$ για δεδομένο ρ. Αφού οι κατανομές των $y, \bar{y}$ είναι ίδιες, ίδιες θα είναι και αυτές των $f_{\mathcal{V}^*}(y), f_{\mathcal{V}^*}(\bar{y})$ δηλαδή των $c^*, \bar{c}$. Τέλος, επειδή γνωρίζουμε ότι για δεδομένα y, c υπάρχει μοναδικό s με το οποίο ένας verifier αποδέχεται, θα πρέπει να συμπίπτουν και οι κατανομές των $s, \bar{s}$, αφού το κάθε ένα είναι μοναδικό ως προς τα y, c^* και $\bar{y}, \bar{c}$ αντίστοιχα.

Εντοπίστε και εξηγήστε το λάθος στην παραπάνω απόδειξη.

[Μονάδες: 10 (Μόνο μεταπτυχιακοί)]

Λύση

1. Σημειώσεις.

   Παρατήρηση: Η εκφώνηση ζητούσε απλή εγκυρότητα (soundness), όχι ειδική εγκυρότητα (special soundness) ή εγκυρότητα γνώσης (knowledge soundness). Για συμβατότητα με προηγούμενες εκδόσεις των σημειώσεων, σωστές απαντήσεις με τις ισχυρότερες έννοιες δεν βαθμολογούνται αρνητικά.

2. - Πληρότητα. Ελέγχουμε εάν ο επαληθευτής τελικά θα αποδεχτεί κατά τη συνομιλία του με ένα τίμιο prover σε μια αληθή πρόταση. Αρχικά ελέγχει: $g_1^s = y_1 h_1 g_1^c$. Για ένα τίμιο prover, ισχύει ότι $y_1 = g_1^t$ και επίσης $s = t + w + c \bmod q$. Αντικαθιστώντας έχουμε: $g_1^{t+w+c} = g_1^t h_1 g_1^c$ ή ισοδύναμα: $g_1^w = h_1$ που ισχύει. Αντίστοιχα και η δεύτερη εξίσωση. Άρα, τελικά ο verifier αποδέχεται.
   - Δεν ισχύει η εγκυρότητα. Θα δείξουμε ότι υπάρχει στρατηγική ώστε ένας prover να κερδίζει πάντα ακόμα και αν μια πρόταση είναι ψευδής (δηλαδή όχι απλά δεν γνωρίζει κατάλληλο μάρτυρα, αλλά δεν υπάρχει καν μάρτυρας). Έστω π.χ. η πρόταση $P = \langle\langle \mathbb{G}, g_1, g_2, q\rangle, g_1^1, g_2^{2019}\rangle$. Ένας τέτοιος prover θέτει $y_i = h_i^{-1} \cdot g_i^t$ για τυχαίο $t \leftarrow \mathbb{Z}_q$, και τελικά απαντά $s = c + t$. Είναι εύκολο να δούμε ότι και οι δύο εξισώσεις επαληθεύονται αφού $g_i^s = g_i^{c+t} = g_i^c g_i^t = g_i^c g_i^t h_i^{-1} h_i = g_i^c y_i h_i$.

     Παρατήρηση: Υπήρξαν προσπάθειες να αποδείξετε ότι ισχύει η ειδική εγκυρότητα. Με τους συνηθισμένους υπολογισμούς, καταλήγουμε σε παραστάσεις όπου τα h_i απαλείφονται, και άρα αδυνατούμε να εξάγουμε μάρτυρα. Αυτή η απαλοιφή, αν και δεν αποτελεί απόδειξη ότι δεν έχουμε ειδική εγκυρότητα[2], είναι ισχυρή ένδειξη ότι κάπου πάσχει η παραλλαγή που κάναμε: επειδή το h_i δεν υψώνεται σε κάποιο απρόβλεπτο εκθέτη, ο prover μπορεί να το ακυρώσει με κατάλληλο y_i.

     Παρατήρηση # 2: Ακραίος cheating prover θα έθετε $y_i = h_i^{-1}$, $s = c$.
   - Μηδενική Γνώση Τίμιου Επαληθευτή. Θα δώσουμε την περιγραφή ενός simulator που θα παράγει συζητήσεις που έχουν την ίδια κατανομή με αυτές ανάμεσα σε τίμιο prover και verifier. Ο simulator αρχικά επιλέγει τυχαία c, s από το $\mathbb{Z}_q$. Παρατηρούμε ότι έχουν την ίδια κατανομή με αυτά μιας πραγματικής συζήτησης: το c άμεσα, το s επειδή σε μια πραγματική συζήτηση είναι ίσο με $c \cdot w + t$ όπου $t \xleftarrow{r} \mathbb{Z}_q$, άρα ακολουθεί και αυτό την ομοιόμορφη. Τέλος, ο simulator θέτει $y_i = g_i^s / (h_i \cdot g_i^c)$. Για να ολοκληρώσουμε την αποδειξη, ισχυριζόμαστε ότι δεδομένου του ότι η συζήτηση οδηγεί σε αποδοχή, και για τις συγκεκριμένες τιμές των c, s τα y_i του simulator είναι μοναδικά, και άρα έχουν την ίδια κατανομή με αυτά μιας συζήτησης τίμιων παικτών.

     Παρατήρηση: Μια άλλη λύση θα ήταν να βασίσουμε τον simulator στον cheating prover του προηγούμενου ερωτήματος. Σε αυτή την περίπτωση ισχύει ότι οι συζητήσεις οδηγούν σε αποδοχή αλλά δεν είναι προφανές ότι έχουν την ίδια κατανομή, οπότε χρειάζεται επιπλέον απόδειξη. Αυτό στην περίπτωση του πρώτου cheating prover είναι αποδείξιμο, αλλά στην περίπτωση του ακραίου, όχι.

[2] Και με τη σειρά της, η απουσία ειδικής εγκυρότητας δεν είναι απαραίτητα απόδειξη για την απουσία εγκυρότητας.

Όνομα: ΑΜ:

Θέμα 4. Εργαζόμαστε με το σύστημα κρυπτογράφησης ElGamal.

(α΄) Καλείστε να αξιολογήσετε τις παρακάτω πιθανές υλοποιήσεις της συνάρτησης GGen η οποία παράγει ομάδες για χρήση με το κρυπτοσύστημα ElGamal. Οι υλοποιήσεις χρησιμοποιούν τις συναρτήσεις `find_prime(n)` η οποία επιστρέφει ένα τυχαίο πρώτο αριθμο μήκους n bits, και `is_prime(p)` η οποία επιστρέφει 1 εάν ο p είναι πρώτος και 0 διαφορετικά. Οι απαιτήσεις μας για την GGen είναι να δέχεται ένα όρισμα λ, και να μας επιστρέφει μια τριάδα (p, q, g) που αναπαριστά μία ομάδα τάξης $q \approx 2^\lambda$, η οποία ορίζεται ως υποομάδα του $\mathbb{Z}_p^*$ και παράγεται από το g. Εάν το κρίνετε χρήσιμο, υποθέστε ότι $\lambda > 10$. Οι συναρτήσεις επιτρέπεται να αποτύχουν με κάποια πιθανότητα, στην οποία περίπτωση επιστρέφουν την ειδική τιμή (0,0,0).

function GGen-1(λ)
$q \leftarrow$ `find_prime`(λ)
$p \leftarrow 2q + 1$
if `is_prime(p)` **then**
$f \xleftarrow{r} \mathbb{Z}_p^*$
$g \leftarrow (20f)^{19} \mod p$
if $g \neq 1$ **then**
Return (p,q,g)
end if
else
Return (0,0,0)
end if
end function
	function GGen-2(λ)
	$q \leftarrow$ `find_prime`(λ)
	$p \leftarrow 2q + 1$
	if `is_prime(p)` **then**
	$f \xleftarrow{r} \mathbb{Z}_p^*$
	$g \leftarrow f^{20 \cdot 19} \mod p$
	if $g \neq 1$ **then**
	Return (p,q,g)
	end if
	else
	Return (0,0,0)
	end if
	end function
		function GGen-3(λ)
		$q \leftarrow$ `find_prime`(λ)
		$p \leftarrow 2q + 1$
		if `is_prime(p)` **then**
		$f \xleftarrow{r} \mathbb{Z}_p^*$
		$g \leftarrow ((2f)^0)^{19} \mod p$
		if $g \neq 1$ **then**
		Return (p,q,g)
		end if
		else
		Return (0,0,0)
		end if
		end function
			function GGen-4(λ)
			$q \leftarrow$ `find_prime`(λ)
			$p \leftarrow 2q + 1$
			if `is_prime(p)` **then**
			$f \xleftarrow{r} \mathbb{Z}_p^*$
			$g \leftarrow f^{2019} \mod p$
			if $g \neq 1$ **then**
			Return (p,q,g)
			end if
			else
			Return (0,0,0)
			end if
			end function

Να επιλέξετε την καταλληλότερη από τις παραπάνω υλοποιήσεις, με πρώτο κριτήριο την ορθότητα, δεύτερο την ασφάλεια της ομάδας που επιλέγεται και ανάμεσα σε εξίσου σωστές και ασφαλείς υλοποιήσεις την αποδοτικότητα (χρόνος εκτέλεσης & επαναλήψεις λόγω αποτυχίας). Τεκμηριώστε την απαντησή σας (ως προς την υλοποίηση που επιλέξατε και ως προς αυτές που απορρίψατε).

[Μονάδες: 10]

(β΄) Θέλουμε να εξετάσουμε την ασφάλεια του συστήματος ElGamal σε συνθήκες δυσκολότερες από αυτές του πειράματος IND-CPA. Συγκεκριμένα, θέλουμε να δώσουμε την εξής δυνατότητα στον αντίπαλο: πριν μαντέψει τον δείκτη του μηνύματος που βρίσκεται στο κρυπτογράφημα c, στέλνει στο πείραμα ένα κρυπτογράφημα c^* της επιλογής του. Το πείραμα επιστρέφει την αποκρυπτογράφηση του, εκτός αν ισχύει ότι $c^* = c$ στην οποία περίπτωση στέλνει $\perp$.

Διατυπωστε πλήρως το πείραμα και τον τον ορισμό ασφάλειας που προκύπτει από τις παραπάνω αλλαγές.

[Μονάδες: 10]

(γ΄) Εξετάστε αν το σύστημα ElGamal είναι ασφαλές ως προς τον παραπάνω ορισμό.

[Μονάδες: 10]

Λύση (α΄) Παρατηρούμε ότι οι 4 υλοποιήσεις διαφέρουν μόνο στον υπολογισμό του g από το f, όπου το f σε κάθε περίπτωση είναι ένα τυχαίο στοιχείο του $\mathbb{Z}_p^*$. Αμεσα μπορούμε να παρατηρήσουμε ότι στην υλοποίηση 3, το $2f$ υψώνεται στη μηδενική οπότε το g είναι πάντα 1, και δεν παίρνουμε ποτέ κάτι χρήσιμο. Συνεπώς η συγκεκριμένη υλοποίηση είναι άχρηστη, τουλάχιστον ως προς την ορθότητα και την αποδοτικότητα.

Από τη θεωρία, ξέρουμε ότι το ElGamal χρειάζεται μια ομάδα με τάξη q, όπου q ένας πρώτος[3] αριθμός. Άρα, θέλουμε ο γεννήτορας g που θα χρησιμοποιήσουμε να έχει τάξη πρώτο.

Αρχικά, ελέγχουμε την τάξη του f. Ως τυχαίο στοιχείο της ομάδας $\mathbb{Z}_p^*$, η τάξη του θα είναι είτε η τάξη της ομάδας είτε κάποιος διαιρέτης της. Η τάξη του $\mathbb{Z}_p^*$ είναι $p-1$ δηλαδή $2q$. Άρα, πιθανές τάξεις του f είναι οι $2q, q, 2, 1$. Τάξη 1 έχει μόνο η μονάδα (για την οποία υπάρχει έλεγχος ως προς το g), οπότε το ζητούμενο κατά τον υπολογισμό του g από το f είναι η τάξη του g να είναι τελικά q (που είναι πρώτος) ή 1 (που απορρίπτεται από τον έλεγχο). Αυτό το πετυχαίνουμε με ύψωση σε άρτιο εκθέση (Υλοποίηση 2). Στις άλλες υλοποιήσεις (1 και 4), υψώνουμε στην 19 και 2019 αντίστοιχα, το οποίο δεν αλλάζει την τάξη του αποτελέσματος (αφού και το 19 και το 2019 είναι πρώτα ως προς την τάξη του f). Ο πολλαπλασιασμός με το 20 αντίστοιχα δεν αλλάζει κάτι (αφού το f επιλέχθηκε ως τυχαίο στοιχείο, και το $20f$ έχει την ίδια κατανομή μια και το 20 είναι αντιστρέψιμο στο $\mathbb{Z}_p^*$).

Άρα: Επιλέγουμε την υλοποίηση 2. Απορρίπτουμε την 3 λόγω ορθότητας (δεν παράγει ποτέ χρήσιμη έξοδο), και τις 1,4 λόγω ασφάλειας (με πιθανότητα κοντά στο $\frac{1}{2}$ παράγουν g με τάξη $2q$ το οποίο επιτρέπει επιθέσεις στον ElGamal).

Παρατήρηση: Κάποιος θα μπορούσε να θεωρήσει ότι το σφάλμα στις 1,4 έχει να κάνει **και** με την ορθότητα (αφού η εκφώνηση αναφέρει «ομάδα τάξης $q \approx 2^\lambda$» ενώ η διερεύνησή μας έδειξε ότι οι 1,4 επιτρέπουν και ομάδες τάξης $2q$). Δεν είναι λάθος.

Παρατήρηση #2: Ορισμένοι έκαναν διερεύνηση του κόστους των υπολογισμών. Στις συγκεκριμένες υλοποιήσεις υπερίσχυαν τα κριτήρια της ορθότητας και της ασφάλειας, οπότε μια ιδανική λύση δεν χρειαζόταν τέτοια διερεύνηση.

Όπου έγινε τέτοια διερεύνηση, αξιολογήθηκε θετικά μόνο όταν ήταν (σε γενικές γραμμές) σωστή. Συγκεκριμμένα[4], το $(a \cdot b)^n$ κοστίζει ένα πολλαπλασιασμό παραπάνω απο το a^n, και όχι περίπου τους διπλάσιους (δεν χρειάζεται να υποθέσουμε κάποιο optimization, αρκεί η σειρά των πράξεων). Επίσης, η ύψωση σε δύναμη είναι ασφαλές να υποθέσουμε[5] ότι γίνεται με αλγόριθμο αντίστοιχο του square and multiply με κόστος περίπου $\lceil \log_2 n \rceil + \mathcal{H}(n)$ πολλαπλασιασμούς, όπου $\mathcal{H}(n)$ το hamming weight του n. Συνεπώς, δεν είναι προφανές ότι η ύψωση στην 20η είναι ακριβότερη από ότι στην 19η (με απλό square and multiply είναι ταχύτερη!).

(β΄) Παραθέτουμε τον ορισμό από τις σημειώσεις, μαζί με τις αντίστοιχες προσθήκες:

Θεωρούμε ότι ένα κρυπτοσύστημα $\langle \mathcal{G}, \mathcal{E}, \mathcal{D} \rangle$ είναι ασφαλές ως προς IND-CCA-lite εάν για κάθε PPT αντίπαλο $\mathcal{A}$,

$$\text{Prob}[\text{Game}^{\mathcal{A}}_{\text{IND-CCA-lite}}(1^\lambda) = 1] \leq \frac{1}{2} + \text{negl}(\lambda).$$

Και το αντίστοιχο πείραμα είναι:

i. $\langle pk, sk \rangle \leftarrow \mathcal{G}(1^\lambda)$
ii. $\langle \text{aux}, M_0, M_1 \rangle \leftarrow \mathcal{A}(\text{play}, pk)$ for $M_0 \neq M_1$
iii. $b \xleftarrow{\text{r}} \{0,1\}$
iv. $c \leftarrow \mathcal{E}(pk, M_b)$
v. $c^* \leftarrow \mathcal{A}(\text{dec}, \text{aux}, c)$
vi. If $(c^* \neq c)$ then $r \leftarrow \mathcal{D}(sk, c^*)$ else $r \leftarrow \perp$.
vii. $b^* \leftarrow \mathcal{A}(\text{guess}, \text{aux}, c, r)$
viii. If $b = b^*$ output 1; otherwise 0.

Παρατήρηση: Ο ορισμός που δώσαμε είναι μια απλούστερη (και λιγότερο ισχυρή) παραλλαγή του IND-CCA. Στην παραπάνω απάντηση στο βήμα *v.* δεν βλέπουμε τον $\mathcal{A}$ να σώζει εκ νέου το state του σε κάποιο νέο aux, αφού με βάση τις εισόδους του είναι δυνατό να το αναπαράγει. Θα

[3] Βλ. ασκήσεις για ElGamal σε ομάδες σύνθετης τάξης, και άλλα παραδείγματα στις σημειώσεις
[4] Χωρίς προ-υπολογισμούς ή υποθέσεις για τα a, b, n.
[5] Αλλιώς διάφορες κατασκευές θα ήταν απαγορευτικά αργές.

μπορούσαμε να προσθέσουμε μια έξοδο aux′ δίπλα στο c^* την οποία θα έπαιρνε ως είσοδο στο βήμα *vii* αντί του aux.

(γ΄) Όχι, δεν είναι ασφαλές. Ο αντίπαλος μπορεί να αλλάξει ελαφρώς το c έτσι ώστε από τη μία να μην πάρει $\perp$ και από την άλλη να καταλάβει ποιό μήνυμα κρυπτογραφήθηκε. Ενδεικτικά, αν $c = \langle U, V\rangle$, μπορεί να στείλει $c' = \langle U \cdot g, V \cdot h\rangle$ το οποίο θα περιέχει το ίδιο μήνυμα. Επίσης μπορεί να στείλει $c^{-1} = \langle U^{-1}, V^{-1}\rangle$ και να αντιστρέψει την απάντηση, να στείλει $c = \langle U, V \cdot a\rangle$ και να διαιρέσει την απάντηση με a (για οποιοδήποτε $a \in \mathbb{G}$) ή κάποιο συνδιασμό των παραπάνω.

Παρατήρηση: Υπήρξαν απόπειρες στη βάση του «Στέλνουμε το $c^* = E(m_0)$ και ελέγχουμε αν το πείραμα επιστρέψει $\perp$» Δεν κερδίζουμε κάτι από αυτό: το πείραμα θα επιστρέψει $\perp$ μόνο αν έτυχε το randomness που χρησιμοποιήσαμε στο c^* να είναι το ίδιο με αυτό που χρησιμοποιήθηκε στο c. Η πιθανότητα για κάτι τέτοιο είναι αμελητέα (άσκηση), και επίσης, αν μια τέτοια επίθεση είχε νόημα, θα επιτύγχανε και στο πείραμα IND-CPA (δεύτερη άσκηση).

Καλή Επιτυχία