Project-crypto

Οδηγίες.

1. Η εργασία ΕΙΝΑΙ υποχρεωτική. Εφόσον ασχοληθείτε, πρέπει να την επιστρέψετε με τα ονόματα και τους αριθμούς μητρώου σας μέχρι την ημερομηνία του **προ - τελευταίου μάθηματος**. Μπορείτε να την ανεβάσετε στο elearning με ονόμα (ενός αρχειου zip) της μορφής : **project-aem1.**
2. Οι γλώσσες προγραμματισμού που μπορείτε να χρησιμοποιήσετε είναι είτε Python 2/3 είτε C/C++ (με compiler gcc).
3. Να σταλεί το **tex+pdf+κώδικας** σε ένα zip file. Όλες οι ασκήσεις να βρίσκονται σε ένα tex αρχείο και όχι ξεχωριστά σε πολλά. Σε ασκήσεις που απαιτούν μόνο κώδικα, εκτός του κώδικα, να δοθεί στο τεχ και μια σύντομη ανάλυση και να σχολιαστούν τα αποτελέσματα. Σε περίπτωση που δεν τα καταφέρετε με το tex, μπορείτε να χρησιμοποιήσετε libreoffice (όχι MS-word).
4. Αν τα λύσετε όλα σωστά θα έχετε **4 μονάδες από τις 10**.
5. Επίσης υπάρχουν (στο τέλος) δέκα ασκήσεις Θεωρίας Αριθμών.
6. Ο πίνακας 1 δίνει μια 5-bit κωδικοποίηση χαρακτήρων. Θα χρειαστεί στο θέμα 1(ii) και 6.
7. Για *tex template* μπορειτε να χρησιμοποιήσετε (αν θέλετε) αυτό.
8. Έχει σημασία και η στυλιστική παρουσίαση.
9. Απαγορεύεται αυστηρά ο πλαγιαρισμός. Εκτεταμένη και αποδεδειγμένη χρήση του, θα συνεπάγεται τον μηδενισμό όλης της εργασίας.
10. Υπάρχουν κάποιες ασκήσεις που η θεωρία τους δεν έχει αναπτυχθεί στο μάθημα. π.χ. το θέμα 3 και 9. Είναι εύκολες ασκήσεις, απλά θα χρειαστεί ίσως να ανατρέξετε σε λίγη θεωρία που δεν έχουμε πει.

Πίνακας 1

(0) A	00000	(12)M	01100	(24)Y	11000
(1) B	00001	(13)N	01101	(25)Z	11001
(2) C	00010	(14)O	01110	(26) .	11010
(3) D	00011	(15)P	01111	(27) !	11011
(4) E	00100	(16)Q	10000	(28) ?	11100
(5) F	00101	(17)R	10001	(29) (	11101
(6) G	00110	(18)S	10010	(30))	11110
(7) H	00111	(19)T	10011	(31) -	11111
(8) I	01000	(20)U	10100		
(9) J	01001	(21)V	10101		
(10)K	01010	(22W	10110		
(11)L	01011	(23)X	10111		

ΘΕΜΑΤΑ

Θέμα 1. (20%)

(i). Υλοποιήστε τον RC4. Χρησιμοποιώντας το κλειδί HOUSE κρυπτογραφήστε το μήνυμα (ξαναγράψτε το κείμενο χωρίς κενά).

WE ALL MAKE MISTAKES AND WE ALL PAY A PRICE

Η υλοποίηση σας πρέπει και να αποκρυπτογραφεί σωστά. Αν χρειαστει μπορείτε να επαναλάβετε το κλειδί ώστε τα bits του μηνύματος να είναι ίδιου πλήθους με τα bits του κλειδιού.

(ii). Υλοποιήστε τον OTP αφού αρχικά μετατρέψετε το μήνυμα σας σε bits με χρήση του πίνακα 1. Θα πρέπει η κρυπτογραφηση και η αποκρυπτογράφηση να δουλεύουν σωστά. Το μήνυμα δίνεται κανονικά, και εσωτερικά μετατρέπεται σε bits. Το κλειδί είναι διαλεγμένο τυχαία και έχει μήκος όσο το μήκος του μηνύματος σας. Το κρυπτογραφημένο μήνυμα δίνεται όχι σε bits, αλλά με λατινικούς χαρακτήρες.

Θέμα 2. (20%)

(i). Εξετάστε αν ισχύει το avalanche effect στον AES. Αναλυτικότερα, φτιάξτε αρκετά ζευγάρια (>30) μηνυμάτων (m1,m2) που να διαφέρουν σε ένα bit. Εξετάστε σε πόσα bit διαφέρουν τα αντίστοιχα κρυπτομηνύματα. Δοκιμάστε με δύο καταστάσεις λειτουργίας : ECB και CBC (η δεύτερη θέλει και IV) .

Υποδ. Υπάρχουν έτοιμες βιβλιοθήκες για τον AES στην C/C++ και python. Πχ. μια επιλογη για C/C++ είναι : https://github.com/BrianGladman/AES Ενώ για python 2 (και 3) η βιβλιοθήκη pycrpto έχει υλοποίηση του AES.

(ii). Το ίδιο με έναν άλλο αλγόριθμο τμήματος (π.χ. Blowfish ή Serpend).

Θέμα 3. (10%) (Vigenere)

Να αποκρυπτογραφήσετε το κείμενο που βρίσκεται στο text file vigenere.txt (όταν το αποκρυπτογραφήσετε, προσπαθήστε να βάλετε τα σωστά σημεία στίξης).

Θέμα 4. (10%)

Τα γράμματα του Αγγλικού αλφαβήτου έχουν αριθμηθεί όπως παρακάτω.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0

Ένα μήνυμα με **n** γράμματα έχει κρυπτογραφηθεί με ένα κλειδί που αποτελείται από **n** γράμματα. Ο αριθμός του κάθε γράμματος προστίθεται στον αντίστοιχο αριθμό του κλειδιού και το αποτέλεσμα ανάγεται ***mod 26*** και αντικαθίσταται από τα γράμματα του πίνακα. Αν το κλειδί περιέχει μόνο τα γράμματα **K,E,Y,** αποκρυπτογραφήστε το κείμενο, **χωρίς** χρήση κώδικα. Το μήνυμα είναι :

AJZBPMDLHYDBTSMFDXTQJ

Θέμα 5. (10%) (Dictionary attack)

Το αρχείο test_zip.zip (υπάρχει στο zip) είναι "κλειδωμένο" με κωδικό. Ο στοχός αυτής της άσκησης είναι να τον βρείτε κάνοντας χρήση βιβλιοθήκων που ανοίγουν τέτοιου τύπου αρχεία (π.χ. στην python έχουμε την zipfile). Σε αυτήν την άσκηση θα χρειαστείτε ένα λεξικό το οποίο και σας επισυνάπτω με όνομα *english.txt*

Θέμα 6. (15%)

Σας δίνω μια φράση (χωρίς κενούς χαρακτήρες) η οποία έχει κρυπτογραφηθεί με χρήση ενός LFSR-10 bit (αφού πρώτα χρησιμοποίησα την 5-bit κωδικοποίηση για να το μετατρέψω σε μια ακολουθία από bits). Η feedback function που χρησιμοποίησα είναι

$$x^{10} + x^9 + x^7 + x^6 + 1$$

Επίσης δίνεται η κρυπτογράφηση του ab : ENC(ab) = .s
Αποκρυπτογραφήστε το κείμενο που υπάρχει στο *readme.txt*

Θέμα 7. (10%)

Έστω ένα μηνυμα m : 16-bits. Θεωρούμε την κρυπτογράφηση :

$$c = m \oplus (m << 6) \oplus (m << 10)$$

Όπου $m << a$
σημαίνει κύλιση προς τα αριστερά κατά α-bits.
Βρείτε τον τύπο αποκρυπτογράφησης.

Θέμα 8. (30%)

(i) Υπολογίστε τον gcd(126048,5050) και βρείτε τους συντελεστές Bezout. (χωρίς υπολογιστή)

(ii) Υπολογίστε το αντίστροφο στοιχείο του 809 στο $\mathbf{Z}_{1001}$.

(iii) Υπολογίστε το υπόλοιπο του 2^{100} με το 101 (χωρίς την χρήση του παρακάτω αλγορίθμου).

(iv) Υλοποιήστε τον παρακάτω αλγόριθμο **fast(),** σε όποια γλώσσα προγραμματισμού θέλετε (ο αλγόριθμος αυτός είναι παραλλάγη αυτού που έχω στις σημειώσεις)

Input. a, g, N

Output. $a^g \bmod N$

1. $g = (g_n g_{n-1} \ldots g_0)_2$
2. $x \leftarrow a,\ \delta \leftarrow 1$
3. for i=0 to n do
   if $g_i = 1$ then $\delta \leftarrow \delta x \bmod N$; end if
   $x \leftarrow x^2 \bmod N$
   end do
   return δ

Κατόπιν υπολογίστε τις δυνάμεις $2^{1234567} \bmod 12345,\ 130^{7654321} \bmod 567$

Θέμα 9. (25%) (Εντροπία)

Έστω η p(x,y) δίνεται από τον πίνακα.

Y\X	**0**	**1**	**2**
0	1/7	1/7	1/7
1	0	1/7	1/7
2	2/7	0	0

(i) Να υπολογιστεί η *εντροπία* H(X) και H(Y)

(ii) Να υπολογιστεί η *από κοινού εντροπία* H(X,Y)

(iii) Να υπολογιστεί η *δεσμευμένη εντροπία* H(X|Y),H(Y|X)

(iv) Να υπολογιστεί η *αμοιβαία πληροφορία* I(X,Y)

(v) Να υπολογιστεί το *μέτρο συσχέτισης* ρ των X,Y

(vi) Υλοποίηστε 4 συναρτήσεις που να δέχονται ως είσοδο την κατανομή των (X,Y) και να επιστρέφουν

[H(X),H(Y)], H(X,Y), [H(X|Y),H(Y|X)] και I(X,Y).

Θέμα 10. (10%) (textbook RSA)

Δίνεται το δημόσιο κλειδί (N,e)=(11413,19). Βρείτε το ιδιωτικό κλειδί και κατόπιν αποκρυπτογραφήστε το μήνυμα

C=(3203,909,3143,5255,5343,3203,909,9958,5278,5343,9958,5278,4674,909,9958,792,909,4132,3143,9958,3203,5343,792,3143,4443)

(το κείμενο υπάρχει στο αρχείο readme.txt)

Υποθέστε, ότι τα γράμματα στο αρχικό μήνυμα m, αναπαρίστανται από τις ASCII τιμές τους (δουλέψτε block by block το C).
Υποδ. *Παραγοντοποιήστε το N, κατόπιν υπολογίστε το φ(N)…θα χρειαστεί και η συνάρτηση fast().*

Θέμα 11. (20%) (textbook RSA)

Αν

(N,e)=(194749497518847283, 50736902528669041)

και το κρυπτογραφημένο κείμενο δίνεται στο αρχείο *readme.txt*, έχει προκύψει από το textbook-**RSA** (block by block) και κατόπιν κωδικοποιήθηκε.

Εφαρμόστε την επίθεση του *Wiener* για να βρείτε το κλειδί **d**.
Υποθέτουμε ότι στο αρχικό κείμενο **m** κάθε χαρακτήρας έχει αντικατασταθεί από την ASCII τιμή του. Για τον υπολογισμό των δυνάμεων $x_i^d \bmod N$, χρησιμοποιείστε την συνάρτηση **fast().** Τέλος, βρείτε το αρχικό μήνυμα **m**.

Θέμα 12. (20%) (Πιστοποίηση πρώτων)

(i). Να κατασκευάσετε με την μέθοδο Fermat έναν πρώτο αριθμό με 2048 bit
(ii). Να κατασκευάσετε με την μέθοδο Miller-Rabin έναν πρώτο αριθμό με 1300 bits.
(iii). Να κατασκευάσετε έναν safe prime 3000 bits.
(Αν p πρώτος αριθμός τέτοιος ώστε ο 2p+1 να είναι επίσης πρώτος, τότε ο 2p+1 ονομάζεται *safe prime* ενώ ο p *Sophie Germain prime*. Τέτοιου τύπου πρώτους αριθμούς χρησιμοποιούμε για παράδειγμα, στην ψηφιακή υπογραφή DSA και στο Naccache-Stern knapsack cryptosystem.)

Θέμα 13. (10%) (CRT)

Λύστε το σύστημα των γραμμικών ισοδυναμιών (Κινέζικο Θεώρημα Υπολοίπων)

$$x \equiv 9 \pmod{17}$$
$$x \equiv 9 \pmod{12}$$
$$x \equiv 13 \pmod{19}$$

Βρείτε την λύση που ικανοποιεί 0<x<1000.

Θέμα 14. (10%)

Έστω $f(x) = x^2 + x - 1354363$. Διαλέξτε 1000 τυχαίους ακέραιους αριθμούς x, στο διάστημα $[1,10^5]$ και εφαρμόστε το Τεστ των Miller-Rabin στους αριθμούς $|f(x)|$.
(α) Τι παρατηρείτε
(β) Υπάρχει πολυώνυμο του **Z**[x], που για κάθε ακέραια τιμή να μας δίνει πρώτο αριθμό;
(δικαιολογήστε την απάντηση σας, είτε δίνοντας κάποια αναφόρα, είτε αποδεικνύοντας κάποιο θεώρημα).

Θέμα 15. (15%) (Rabin)

Έστω p=5, q=11 και το κρυπτογραφημένο μήνυμα c=14 έχει προκύψει από την εφαρμογή της TDF του Rabin σε ένα μήνυμα m. Βρείτε το αρχικό μήνυμα m αν γνωρίζετε ότι το m<20.

Θέμα 16. (15%) (PGP)

Με χρήση του προγράμματος GPG δημιουργήστε
ένα PGP-πιστοποιητικό και κάνοντας χρήση του δικού μου δημοσίου κλειδιού
goo.gl/2Aek39 ή $gpg --keyserver pgp.ocf.berkeley.edu/ --recv-keys 0xEB1185F82713D6DF
στείλτε μου ένα κρυπτογραφημένο μήνυμα με χρήση του ηλ.ταχυδρομείου. Μην ξεχάσετε να μου στείλετε και το δημόσιο κλειδί σας σε ξεχωριστό μήνυμα (εννοείται χωρίς κρυπτογράφηση). Σε περίπτωση που είστε ομάδα, κάθε άτομο της ομάδας να κάνει την εργασία ξεχωριστά. Ως απάντηση γραψτε μου την **ημερομηνία αποστολής** του μυνήματος σας και το **hash** που θα σας στείλω ως απάντηση στο κρυπτογραφημένο μηνυμά σας.

Θέμα 17. (15%)

Υλοποιήστε την TDF : CRT-RSA.
Θα υλοποιήσετε **τρει**ς αλγορίθμους.
- Ο πρώτος πρέπει να παράγει κλειδιά όπως ο κλασικός RSA, δηλ. *(p,q,e,d,N).* Αλλά επιπλέον και τα dp,dq,qinv.
- Ο δεύτερος αλγόριθμος (κρυπτογράφηση) θα είναι ο ίδιος όπως ο κλασικός RSA και ο
- τρίτος (αποκρυπτογράφηση) θα χρησιμοποιεί το CRT

(δείτε τις σημειώσεις).

Θέμα 18. (15%)

Μπορείτε να ανοίξετε το secure.zip?

Hint. Ότι χρειάζεστε είναι στην διαφάνεια *course-1.pdf* στο elearning.

Θέμα 19. (10%) (openSSL)

Το openssl είναι ένα open source πρόγραμμα που υλοποιεί πολλές κρυπτογραφικες ρουτίνες. Στην άσκηση αυτή πρέπει να καταβάσετε (σε κάποια μορφή) το πιστοποιητικό του server :
https://cryptology.csd.auth.gr:8080
Κατόπιν να εξάγεται το RSA modulus σε ακέραιο αριθμό με χρήση του OPENSSL.
Πόσα bits είναι;
Εφαρμόστε την επίθεση Wiener (θα πρέπει να αποτύχει!)

Θέμα 20. (10%) (Βιβλιογραφική)

(i). Να αναλύσετε πως η χρήση του ίδιου κλειδιού στον OTP επιτρέπει να βρούμε μηνύματα που είναι γραμμένα σε κάποια φυσική γλώσσα.

(ii). Να αναλύσετε πως δουλεύει το RSA-OAEP

(iii). Να μελετήσετε βιβλιογραφικα τις στρατηγικές
first sign then encrypt & first encrypt then sign
και να καταλήξετε σε κάποια συμπεράσματα
(πλεονεκτήματα/μειονεκτήματα)

Οι απαντήσεις να είναι αναλυτικές (όχι συνοπτικές) και να υπάρχει και η βιβλιογραφία που χρησιμοποιήσατε.

Οι παρακάτω ασκήσεις βρίσκονται στο text book στο elearning,
public_key_crypto.pdf (θα αναρτηθεί σύντομα στο elearning)

Aem mod 3 = 0

Θέμα 21. (10%) Ασκηση 3.1
Θέμα 22. (10%) Άσκηση 3.4
Θέμα 23. (10%) Άσκηση 3.6
Θέμα 24. (10%) Άσκηση 3.17
Θέμα 25. (10%) Άσκηση 3.24
Θέμα 26. (10%) Άσκηση 3.25
Θέμα 27. (10%) Άσκηση 3.26
Θέμα 28. (10%) Άσκηση 3.36
Θέμα 29. (10%) Άσκηση 3.49
Θέμα 30. (10%) Άσκηση 3.99

Aem mod 3 = 1

Θέμα 21. (10%) Ασκηση 3.1
Θέμα 22. (10%) Άσκηση 3.4
Θέμα 23. (10%) Άσκηση 3.7
Θέμα 24. (10%) Άσκηση 3.16
Θέμα 25. (10%) Άσκηση 3.24
Θέμα 26. (10%) Άσκηση 3.30
Θέμα 27. (10%) Άσκηση 3.39
Θέμα 28. (10%) Άσκηση 3.42
Θέμα 29. (10%) Άσκηση 3.59
Θέμα 30. (10%) Άσκηση 3.100

Aem mod 3 = 2

Θέμα 21. (10%) Ασκηση 3.1
Θέμα 22. (10%) Άσκηση 3.4
Θέμα 23. (10%) Άσκηση 3.8
Θέμα 24. (10%) Άσκηση 3.15
Θέμα 25. (10%) Άσκηση 3.24
Θέμα 26. (10%) Άσκηση 3.26
Θέμα 27. (10%) Άσκηση 3.28
Θέμα 28. (10%) Άσκηση 3.40
Θέμα 29. (10%) Άσκηση 3.42
Θέμα 30. (10%) Άσκηση 3.100

Καλή Διασκέδαση!