Εργασία #1

Οδηγίες.

1. Η 1η εργασία είναι προαιρετική. Εφόσον ασχοληθείτε, πρέπει να την επιστρέψετε με τα ονόματα και τους αριθμούς μητρώου σας μέχρι **10 Απριλίου**. Μπορείτε να τη στείλετε στο e-mail : *drazioti at csd.auth.gr* με θέμα **project-1-aem1-aem2**
2. Η κάθε ομάδα να αποτελείται από **2** άτομα το πολύ.
   Φυσικά, μπορείτε να δουλέψετε και ατομικά.
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 αρχεία tex.
4. Αν τα λύσετε όλα σωστά θα έχετε **+1 μονάδα** στο βαθμό της τελικής εξέτασης με την **προϋπόθεση** να γράψετε **τουλάχιστον 5**.
5. Μην αναζητήσετε έτοιμο κώδικα στο internet! Αλλά αν αυτο συμβεί τουλάχιστον να είστε έντιμοι και να δώσετε τα απαραίτητα credits!

ΘΕΜΑΤΑ

Θέμα 1. (10%) Να απαντήσετε σύντομα στις παρακάτω ερωτήσεις:
(i) Διατυπώστε την αρχή του Kerchoff και γράψτε ποιος ο λόγος που διατυπώθηκε.
(ii) Δώστε τον ορισμό της τέλειας ασφάλειας ενός κρυπτοσυστήματος. Υπάρχουν συστήματα με τέλεια ασφάλεια;
(iii) Περιγράψτε μία αποκάλυψη του E.Snowden.
(iv) Το OTP παραμένει ασφαλές αν χρησιμοποιήσουμε το ίδιο κλειδί δύο φορές;
(v) Περιγράψτε την κατάσταση λειτουργίας GCM σε ένα συμμετρικό κρυπτοσύστημα τμήματος.

[στα θέματα 2 και 9 χρησιμοποιήστε την παρακάτω 5-bit κωδικοποίηση]
[υπάρχουν βοηθητικές συναρτήσεις στο https://github.com/AristotleUniversity/python_scripts/blob/master/lfsr_project.py]

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) ?	11011
(5) F	00101	(17)R	10001	(29) (	11100
(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		

Θέμα 2. (10%) Υλοποιήστε τον RC4. Χρησιμοποιώντας το κλειδί MATRIX κρυπτογραφήστε το μήνυμα (αγνοήστε τα κενά).

Never send a human to do a machine s job

Θέμα 3. (15%) (Vigenere) Να αποκρυπτογραφήσετε το κείμενο που βρίσκεται στο text file vigenere.txt
Να γραφεί αναλυτικά η μεθοδολογία.
(όταν το αποκρυπτογραφήσετε, προσπαθήστε να βάλετε τα σωστά σημεία στίξης).

Θέμα 4. (10%) Το παρακάτω κείμενο κρυπτογραφήθηκε με το σύστημα της μετατόπισης (thema4.txt)

ΟΚΗΘΜΦΔΖΘΓΟΘΧΥΚΧΣΦΘΜΦΜΧΓΟΣΨΧΚΠΦΧΘΖΚΠ

Να βρεθεί το αρχικό κείμενο.

Θέμα 5. (10%) Κάνοντας χρήση της βιβλιοθήκης pycrpto (https://pypi.python.org/pypi/pycrypto) εξετάστε αν ισχύει το avalanche effect στον AES. Αναλυτικότερα, φτιάξτε αρκετά ζευγάρια (>30) μηνυμάτων (m1,m2) που να διαφέρουν σε ένα bit. Μελετήστε σε πόσα bit διαφέρουν τα αντίστοιχα κρυπτομηνύματα.
Δοκιμάστε με δύο καταστάσεις λειτουργίας ECB,CBC (η δεύτερη θέλει και IV)
Αναλυτικότερα

```
#το key πρέπει να είναι 16 byte. Διαλέξτε όποιο κλειδί θέλετε.
from Crypto.Cipher import AES
key='something'
obj=AES.new(key,AES.MODE_ECB)
# κρυπτογράφηση #
message="something" # το message να είναι 16 bytes.
ciphertext1=obj.encrypt(message)
ciphertext2=ciphertext1.encode('hex') [python 2]
```

[σε python 3]

```
import binascii
ciiphertext2=binascii.hexlify(c1)
```

```
ciphertext=bin(int(ciphertext2, 16))[2:] # το κρυπτογραφημένο μήνυμα μετατρέπεται σε δυαδικά ψηφία.
```

...

Θέμα 6. (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. Απαντήσεις που θα έχουν brute force **δεν θα χρεωθούν** τις μονάδες της άσκησης.

AJZBPMDLHYDBTSMFDXTQJ

Θέμα 7. (10%) Το αρχείο `test_zip.zip` (υπάρχει στον zip) είναι κλειδωμένο με κωδικό. Ο στοχός αυτής της άσκησης είναι να τον βρείτε κάνοντας χρήση της βιβλιοθήκης `zipfile` της python. Σε αυτήν την άσκηση θα χρειαστείτε ένα λεξικό το οποίο και σας επισυνάπτω στον zip με το όνομα english.txt.
Η εντολή που θα ελέγχει κάθε φορά τον υποψήφιο κωδικό από το english.txt είναι :

```
>> zFile=zipfile.ZipFile("test_zip.zip") # αυτό θα γίνεί μία φορα στην αρχή του κώδικα σας, και
>> zFile.extractall(pwd=password) # αυτή θα εκτελείτε σε μία conditional συνθήκη εώς ότου βρει τον κωδικό
```

Όταν ο κωδικός σας δεν είναι σωστός, οπότε και θα έχετε ένα μήνυμα λάθους, το οποίο πρέπει με κάποιο τρόπο να αγνοηθεί και το πρόγραμμα να συνεχίζει στην επόμενη γραμμή του english.txt (το english.txt δεν είναι λεξικό με την έννοια ότι περιέχει όλους του δυνατους συναδυασμους λέξεων,συμβόλων, κλ.π. Απλά περιέχει κάποιες *συνηθισμένες* λέξεις).

Θέμα 8. (10%) Στόχος αυτής της άσκησης είναι να αποκτήσετε πρόσβαση στον server sage.csd.auth.gr (όχι root!). Έχει έρθει στα χέρια σας (με κάποιο τρόπο...) ένα τμήμα του αρχείου `/etc/shadow` (το επισυνάπτω ως `password.txt`). Επίσης γνωρίζετε ότι ο κωδικός είναι εξαψήφιος (αποτελείται μόνο από αριθμούς). Εφόσον τον βρείτε κάντε *ssh* στον server χρησιμοποιώντας το account που βρήκατε. Τέλος, γράψτε το όνομα σας και το αεμ στο αρχείο `my_name` (βρίσκεται στο home folder του user. Χρησιμοποιείστε nano my_name για να εισάγετε αύξοντα αριθμο το **όνομα και το αεμ σας** και κατοπιν ctrl+x και Y για να το σώσετε). **Ο server θα είναι ανοιχτός για log in μόνο μία ημέρα, 5 Απρίλιου.**
Επίσης υποθέστε ότι έχετε αύξοντα αριθμό 4 και έχετε ΑΕΜ 2014. Τότε εκτελέστε τα παρακάτω

```
$ echo "4.το επιθετο σας με λατινικα.2014" >> 0.txt
$ cat 0.txt challenge >2014.txt
$ sha1sum 2014.txt
```

(κρατήστε το hash ως αποδεικτικό-που θα είναι στην απάντηση του θέματος μαζί με το password που θα βρείτε καθώς και τον κώδικα).
Τέλος,

```
$rm -rf 2014.txt 0.txt
```

(Υποδ. - Θα χρησιμοποιηθεί η συνάρτηση

```
import crypt
crypt.crypt("password","$6$salt$")
```

Όπου το salt θα το δείτε από το αρχείο `password.txt`

- Το (encrypted) password στην άσκηση αυτή το έχετε.Πρέπει να θέσετε password =" αυτο που σας δίνω "
- Δεν χρειάζεστε λεξικό (αλλά nested loops που να βρίσκουν όλες τις εξάδες ακεραιών μεταξύ 0 και 9.
- Για ssh από windows θα χρειαστείτε τον ssh-client putty)

Θέμα 9. (15%)

(i) (5%) 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,

$$. \; x^{10} + x^9 + x^7 + x^6 + 1$$

Το κείμενο υπάρχει στο zip file με το όνομα lfsr1.txt.

(ii) (15%) Στο zip file του project βρίσκεται επίσης και το αρχείο με όνομα lfsr2.txt

Αυτό έχει προκύψει από τον συνδυασμό των παρακάτω δύο lfsr's.

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

$$x^{16} + x^8 + x^7 + x^3 + x^2 + 1$$

Η κρυπτογράφηση του κειμένου έγινε ως εξής.

1. Το άρχικο κείμενο μετατράπηκε σε bit σύμφωνα με τον Πίνακα 1.
2. Οι έξοδοι των δύο lfsr's (με κάποια seed άγνωστα σε εσάς) έγιναν xor.
3. Θεωρώ ένα keystream (όπως προέκυψε από το βήμα 2) μήκους όσο το μήκος του μηνύματος (σε bit),
4. Το μήνυμα γίνεται xor με το keystream και μετατρέπεται σε λέξη σύμφωνα με τον πινακα 1.

Βρείτε το αρχικό κείμενο.

Καλή Διασκέδαση!