

Objective: The EdX MOOC platform aims to provide educational content to users of the platform. The platform includes other auxiliary objectives such as being able to update course content, provide inter-person interactivity, and reward users for course completion.

Principals and Authorized Behavior: The platform has three main roles: student, teacher, and administrator. Each have their own set of allowed functions:

- Student: a student must be able to request enrollment into a course, view content for an enrolled courses (to the extent allowed by course instructor(s)), submit digitally signed answers and feedback to (enrolled) course instructor, modify their own profile information (login username, password, email), and receive certificate of completions as determined by the instructor.
- Teacher: authorize student enrollment into their course, authorize other instructors to co-administer their course, modify course meta-data (name, description) and content, send signed feedback to student, view the submissions of student enrolled in their course(s), modify their own profile information (email, login username, password), add new course, and award certificate of completion to select student users.
- Administrator: add teacher user accounts.

Confidentiality/Integrity Details:: Specifically, by use of some security mechanism (perhaps, OAuth 2.0 or like), we intend that the platform forbids the student viewing or modify the content and progress of other users (except to view the content of their enrolled courses as allowed by their course instructors). The student should not be able to obtain completion certificates without proper authentication.

The teacher should not be able to view the submissions of students not currently in their course. They should not be able to update the content of the courses that they were not authorized to administer.

A. They were encrypted using the same pad. The two words are ADVERSARIAL AND MATHEMATICS.

```
CT1_str = 'd2 6b a5 0d 27 6a 34 2d 8e 53 0e'
CT2_str = 'de 6e a7 00 30 74 34 2b 8e 51 11'

CT1 = [int(byte, 16) for byte in CT1_str.split(' ')]
CT2 = [int(byte, 16) for byte in CT2_str.split(' ')]

def xor(xs, ys):
    '''Perform pairwise XOR operation on two lists'''
    return [x ^ y for x, y in zip(xs, ys)]

X = xor(CT1, CT2)

dictionary = open('dictionary', 'r').read().split()
words = set()
for w in dictionary:
    if len(w) == len(X):
        try:
            words.add(w.upper().encode())
        except:
            print 'Problem encoding', w

for PT1_str in words:
    PT1 = [ord(byte) for byte in PT1_str]
    PT2 = xor(PT1, X)
    PT2_str = "".join([chr(byte) for byte in PT2])
    if PT2_str in words:
        pad = xor(PT1, CT1)
        print('PT1 = %s, PT2 = %s, pad = %s' % (PT1_str, PT2_str, pad))
```

B.

6.857 Homework
Erica Du
Skanda Koppula
Jessica Wang

Problem Set 1

1-3 - Vulnerability/Mechanism Chains

February 15, 2015