Cryptography

Due 02/14 at 11:59PM

General Overview

1660 (Part I)	1620 (Part II)
1. Grades 2. Ivy 3. Keys	(Everything from part 1) 1. Padding

All problems in each part are **separate/self-contained** → the assignment can be completed in any order.

Move to the next problem if you get stuck!

How the stencils work

- Each problem gets its own directory
- Stencil code in multiple language
- What lives in each stencil dir?
  - STENCIL.md: Super helpful stuff about this stencil
  - Makefile: If required => Run make to compile

To start: you should COPY the stencil files to the directory for that problem:

```
cs1660-user@container: ~/repo$ cp -Trv ivy/stencil/go ivy
```

Repo layout

```
<repo root>
 |- grades/          # <--- Problem directory for ivy
 |   |- stencil/     # <--- Stencil code for grades
 |      |- go/
 |      |   |- STENCIL.md  # Guide for using this stencil
 |      |   |- sol.go
 |      |   |- ...
 |      |- python/
 |      |   |- STENCIL.md
 |      |   |- ...
 |      |- ...
 |- ivy/             # <--- Problem directory for ivy
 |   |- stencil/     # <--- Stencil code for ivy
 |      |- ...
 |- keys/            # <--- Problem directory for keys
 |   |- ...
 |- ...
```

What you should submit

For each problem, your repo should have:

- Your script (called `sol`)
- (Anything else required by your stencil)
- README
  - Describe the attack, how you did it, what you might change
  - Anything else we should know (what you tried, feedback, issues, etc.)

At repo root: `COLLABORATORS{,.txt,.md}`

- List anyone you collaborated with, and on which part

1. Grades

- ➔ **GOAL:** Uncover information from an encrypted database
- ➔ YOU HAVE:
  a. Database encrypted with ECB mode
  b. Some statistics:
     a. 100000 students total
     b. 30 grades/student
     c. Distribution of all grades: 50% As, 30% Bs, ...

- What do we want to know?

1. Since you know the plaintext format of the database, how many possible unique ciphertext blocks exist? (Hint: you can answer this question based on only the information here—then you can check your answer with a script!)
2. What ciphertext block corresponds to an A grade? B? C? N?
3. There's a student who's famous at the university for being the only student to ever get both As and Cs but no Bs. Exactly how many As, Cs, and Ns has this student received?

Questions on grades

- If generate-database is slow, write file to /tmp (see Ed)

Ivy

- ➔ **GOAL:** Decrypt traffic from router
- ➔ **NEED:** Shared (symmetric) key ***k***
- ➔ **HAVE:**
  a. Ability to sniff network traffic

  b. An “encryption oracle” via personal router → chosen plaintext attack

  c. The key (***k***) encrypted → $E_k(k)$

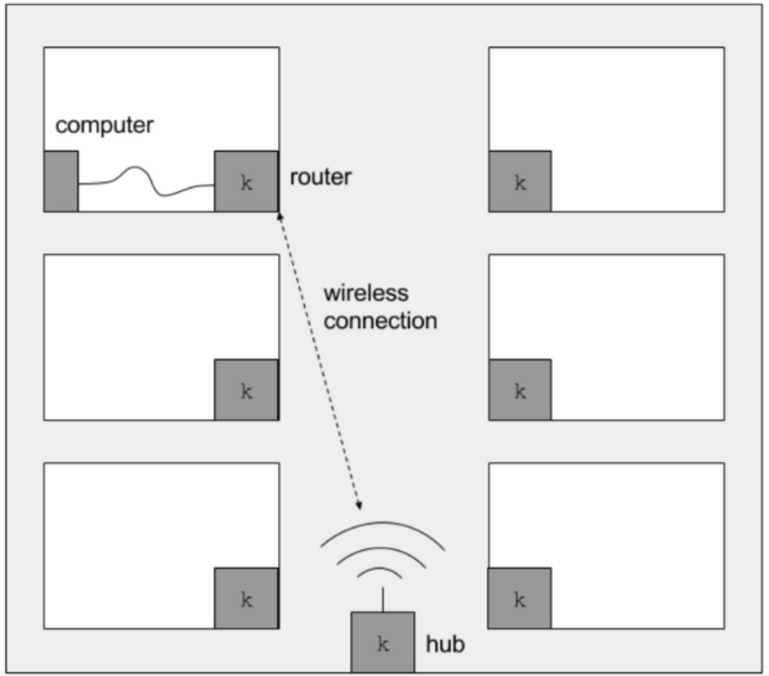


Figure 1: Network configuration for the Ivy problem.

Ivy demo

Ivy (cont.)

- R is a source of randomness. Each time R is queried, it generates a uniformly-distributed random number which is 16 bits long (that is, its outputs are uniformly distributed in $\{0, ..., 2^{16} - 1\}$).

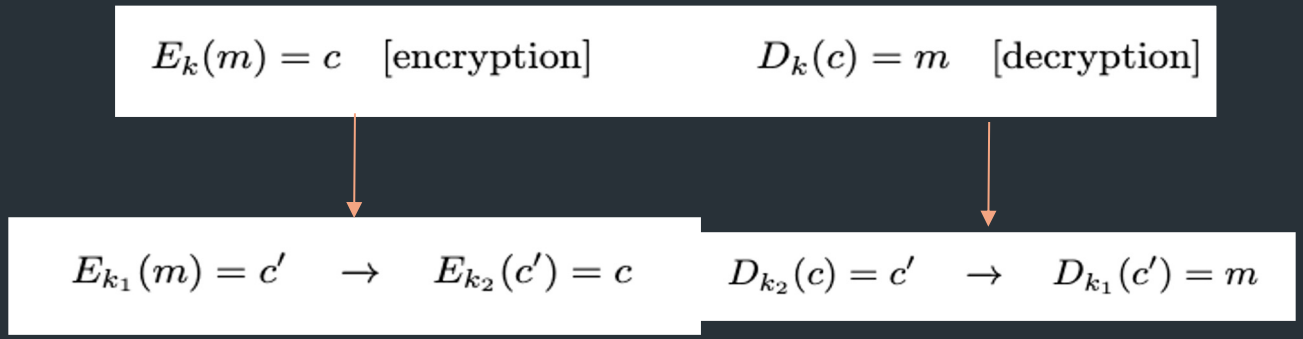
```
ENCRYPT(k, m)
1  iv = R()          // Generate initialization vector
2  s = iv ++ k       // Concatenate iv and k
3  r = G(s, |m|)     // Generate |m| random bytes
4  c = m ⊕ r         // XOR m and r to get ciphertext c
5  return (iv, c)
```

```
DECRYPT(k, iv, c)
1  s = iv ++ k       // Concatenate iv and k
2  r = G(s, |c|)     // Generate |c| random bytes
3  m = c ⊕ r         // XOR c and r to get plaintext m
4  return m
```

$$
\begin{aligned}
c \oplus r &= (m \oplus r) \oplus r \\
&= m \oplus (r \oplus r) \\
&= m \oplus 0 \qquad \text{[a number XORed with itself = 0]} \\
&= m
\end{aligned}
$$

Keys

- ➔ GOAL: Break Academy's block cipher encryption scheme
- ➔ NEED: Both encryption keys
- ➔ HAVE:
  a. `Set of (plaintext, ciphertext) pairs from encryption scheme`
  b. `Cipher + stencil code`



Questions on Keys

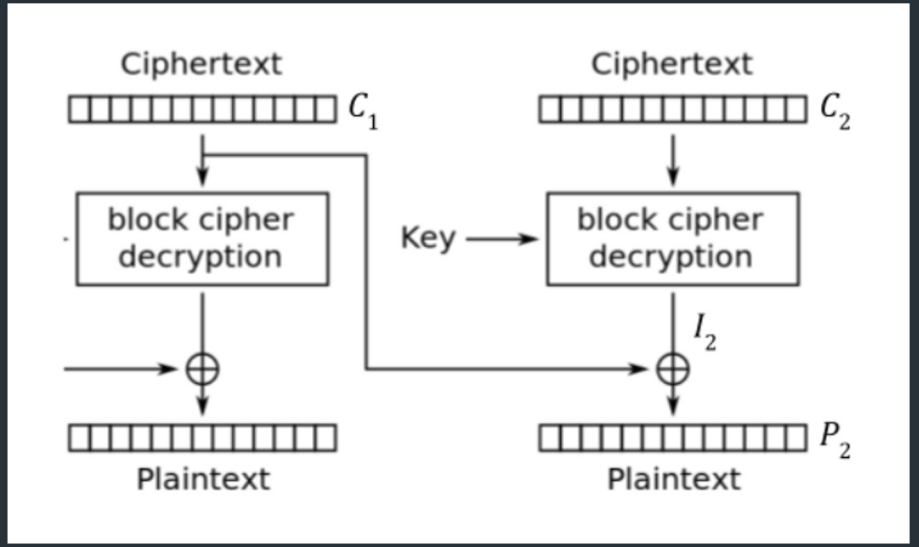
- Java stencil: you must install Java first
  - See STENCIL.md for instructions

- Performance?
  - No specific requirements, so long as you carry out the attack we're asking you to perform
  - Autograder will time out at 40min—which should be much more than enough
  - You can discuss performance, time/space complexity in your README!
- What's the cipher? Irrelevant to problem (but look up TEA cipher)

Padding

- ➔ **GOAL:** Reveal the grades of student with ID: 12345
- ➔ **HAVE:**
  i. Server as a binary
  ii. Encryption scheme → CBC mode
  iii. Whether the padding is correct or not

- ➔ **NEED:** Ability to find correct ciphertext given any plaintext:
  i. Determine <u>current</u> intermediate state I_n
  ii. Find correct <u>previous</u> ciphertext C_{n-1} to generate P_n



Transcript

- ➔ **GOAL:** Forge your transcript
- ➔ **NEED:** (?)
- ➔ **HAVE:**
  a. Challenge/response protocol binary (“challenge”)
  b. JSON-encoded copy of the website’s public key (“server.pub”)
  c. Signature method (“encrypt”) via binary
  d. Knowledge that the **same** RSA key pair is used for signatures & verification of the website’s integrity