Lecture 2 – A Look Back

CS-466 Applied Cryptography
Adam O’Neill
A Taste of Notation

you can think of algorithms as Turing Machines

- randomized: has a random coin tape (or can make coin flips

\[ y \leftarrow A(...) \quad y \leftarrow A(...) \]

- run A with fresh randomness
- run A with coins fixed to \( \nu \)

- oracles:

\[ y \leftarrow A O_1(.),...,O_n(.) (...) \]

- \( O_1, O_2, \ldots, O_n \) access to oracles

If \( S \) is a finite set then \(|S|\) is cardinality of \( S \)

- means \( s \) is assigned a random element from \( S \)

\( \mathbb{0,1}^* \): set of all strings
\( \mathbb{0,1}^n \): set of strings of length \( n \)

- \(|S|\) length of string \( S \).

If \( s_1, s_2 \) are strings then \( s_1 || s_2 \) denotes an encoding from which \( s_1, s_2 \) are uniquely recoverable

- if lengths of \( s_1, s_2 \) are known then concatenation suffices
Probabilities

Games (will formalize a syntax later)

Ex. Game 1

\[ \begin{align*}
    a & \leftarrow X \backslash \{ \cdot \} \\
    b & \leftarrow Y \backslash \{ a \} \\
    c & \leftarrow a \oplus b
\end{align*} \]

\( \Pr [c=x] \) means the probability that \( c=x \) over the choice of randomness for the entire game

Random Variables

If \( X \) is a random variable it is over some finite set \( \mathcal{X} \) of the possible outcomes of \( X \). It puts a probability distribution on \( X \)

\( \Pr [\ldots] \) is true when \( X \) is sampled randomly according to its distribution
Defining Cryptographic Objects

Think of cryptographic objects the same way you think of a data type in programming.

To define a cryptographic object, you need to specify:
- **Syntax**: What are the constituent algorithms, what are their inputs and outputs?
- **Correctness**: What functionality these algorithms need to fulfill in the absence of an adversary, to be useful.
- **Security**: What an adversary is allowed to do in attacking a scheme
  1. What is the adversary’s goal
  2. How do they try to break the scheme

Typically specified by a game.
Symmetric-Key Encryption: Syntax

A symmetric-key encryption scheme with message-space $\mathcal{M}$ is a tuple of algorithms $SE = (K, E, D)$ defined as follows:

- $K$ outputs a key $K 
  K \leftarrow K$

- $E$ on input a key $K$ and a message $m$ outputs a ciphertext $c$
  $c \leftarrow E(K, m)$ also written as $c \leftarrow E_K(m)$

- $D$ on input a key $K$ and ciphertext $c$ outputs a message $m$ or special "error symbol" $\bot$

Deterministic why??

$m \leftarrow D(K, c)$ or $m \leftarrow D_K(c)$
Usage

We will explain how parties can initially agree on $K$ later!

$K \leftarrow \$K$

Alice wants to transmit a message $m$ to Bob

$C \leftarrow E_K(m)$

C

INTERNET

Bob

$K \leftarrow \$K$

$m' \leftarrow D_K(c')$

$K \leftarrow \$K$

Does $m' = m$?
Correctness

Bob

Fix a message $m$ in the message space. Fix a key $K$ that can be output by $k_2$ or "perfectly correct."

Consider the game

Sample space: Coin space of $E$

We say $SE$ is correct if $A_m A_K$

always!

Bob gets the message Alice sent when nobody tampers with the ciphertext doesn't really matter if $E$ is observing.

What are some equivalent ways of writing the correctness condition?
Weakest adversarial goal: Given some $C \leftarrow \mathcal{E}_K(M)$, recover $M$.

If this is easy, the encryption scheme is clearly insecure.

Adversary knows the scheme’s message space $\mathcal{M}$ and algorithms $\mathcal{K}, \mathcal{E}, \mathcal{D}$ but not the key $K$. (Kerckhoff's principle)

**IMPORTANT**

Usually the adversary will figure these out anyway. If security relies on secrecy of the algorithms and there is a compromise, it is much harder to replace the entire scheme than choose a new random key $K$. 
Today’s lecture concerns substitution ciphers, a particular type of symmetric-key encryption.

Prior to modern cryptography, basically all encryption schemes were of this form.

Yet we will see that such schemes are fundamentally flawed.
Scheme Setup

**Definition**

\[ \Sigma = \{A, B, \ldots, Z\} \cup \{\sqcup, \ldots, ?, !, \ldots\} \]

The message space must be a subset of \( \Sigma \)

the ciphertext space can be any symbols

The key generation algorithm must output a permutation on \( \Sigma \)

one-to-one mapping

The encryption algorithm applies \( \Pi \) "component-wise" to the characters in the input message, i.e. has the form (can be written as) in the following slide

To each character individually one-by-one
Definition of S.C. Algorithms

Algorithm $E_\pi(M)$
For $i = 1, \ldots, |M|$ do
\[ C[i] \leftarrow \pi(M[i]) \]
Return $C$

Algorithm $D_\pi(C)$
For $i = 1, \ldots, |C|$ do
\[ M[i] \leftarrow \pi^{-1}(C[i]) \]
Return $M$

Exercise prove correctness
Cryptanalysis

Suppose adversary has ciphertext:

```
COXBX TBX CVK CDGXR DI T GTI’R ADHX VOXI OX ROKQAU
IKC RNXPQATCX: VOXI OX PTI’C THHKBU DC, TIU VOXI OX
PTI.
```
## Frequency

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
<th>L</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>3</td>
<td>7</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>3</td>
<td>9</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>O</th>
<th>P</th>
<th>Q</th>
<th>R</th>
<th>S</th>
<th>T</th>
<th>U</th>
<th>V</th>
<th>W</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>8</td>
<td>3</td>
<td>2</td>
<td>4</td>
<td>0</td>
<td>8</td>
<td>3</td>
<td>4</td>
<td>0</td>
<td>13</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>
Any ideas?

$O_X$ in ciphertext $\Rightarrow \pi^{-1}(0) \in \{B, H, M, W\}$

Guess $\pi^{-1}(0) = H$ since 0 has pretty high frequency
Guess $\pi^{-1}(C) = T$ since there is no \textit{in ciphertext} so WHERE is unlikely.

Could be: THERE, THESE, WHERE,....

THE E E T E T E T
COXBX TBX CVK CDGX R DI T GTI'R ADHX VOXI OX ROKQAU
OX PTI'C THKBU DC, TIU VOXI
There are T E A A ' E H E H E H
Coxbx tbx cvk cdgxr di t gti'r adhx voxI ox rokqaU
T E ATE: HE HE A 'T A R T, A HE
IKC RNXPQATCX: VOXI OX PTI'C THHKBU DC, TIU VOXI
HE A .
OX PTI.

<table>
<thead>
<tr>
<th>π⁻¹(Τ)</th>
<th>R</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>Τ</td>
<td>N</td>
<td>O</td>
</tr>
</tbody>
</table>

Τ is a single-letter word so π⁻¹(Τ) ∈ {A, Ι}
We know π⁻¹(B) ∈ {R, S}
So TBX could be: ARE, ASE, IRE, ISE
We guess ARE
There are two times in a man's life when he should not speculate: when he can't afford it, and when he can. Ox PTI.
Example 1 (Cryptography). Stanford’s Statistics Department has a drop-in consulting service. One day, a psychologist from the state prison system showed up with a collection of coded messages. Figure 1 shows part of a typical example.
The General Setting

\[ f : \{\text{code space}\} \longrightarrow \{\text{usual alphabet}\}. \]

To get the statistics, Marc downloaded a standard text (e.g., War and Peace) and recorded the first-order transitions: the proportion of consecutive text symbols from \( x \) to \( y \). This gives a matrix \( M(x, y) \) of transitions. One may then associate a plausibility to \( f \) via

\[
\text{Pl}(f) = \prod_i M(f(s_i), f(s_{i+1}))
\]

\( M[x, y] \)

Likelihood (according to the training set) that \( y \) follows \( x \)

Can generalize to \( n \)-th order transitions, i.e., "n-grams"

\( \exists \) in substitution ciphers like Enigma

\( \approx \) in "souped up" substitution ciphers
The Metropolis Algorithm

- Start with a preliminary guess, say $f$.  
- Compute $P_l(f)$.  
- Change to $f_*$ by making a random transposition of the values $f$ assigns to two symbols.  
- Compute $P_l(f_*)$; if this is larger than $P_l(f)$, accept $f_*$.  
- If not, flip a $P_l(f_*)/P_l(f)$ coin; if it comes up heads, accept $f_*$.  
- If the coin toss comes up tails, stay at $f$.  

$\begin{align*}
  f(\text{h}) &= q \\
  f(\text{t}) &= c \\
\end{align*}$

- Don't want to get stuck in local maximum.
to bat-rb. con todo mi respeto. i was sitting down playing chess with danny de emf and boxer de el centro was sitting next to us. boxer was making loud and loud voices so i tell him por favor can you kick back homie cause im playing chess a minute later the vato starts back up again so this time i tell him con respecto homie can you kick back. the vato stop for a minute and he starts up again so i tell him check this out shut the f**k up cause im tired of your voice and if you got a problem with it we can go to celda and handle it. i really felt disrespected thats why i told him. anyways after i tell him that the next thing I know that vato slashes me and leaves. dy the time i figure im hit i try to get away but the c.o. is walking in my direction and he gets me right dy a celda. so i go to the hole. when im in the hole my home boys hit doxer so now "b" is also in the hole. while im in the hole im getting schoold wrong and
Take Away?

One might conclude substitution ciphers are flawed, but only for long enough plaintexts. By thinking more adversarially, we can come up with an example that shows that such ciphers are flawed even with 1-bit plaintexts.
Voting

Everyone shares a key $K$

1. $c_1 \leftarrow E_K(v_1)$
2. $c_2 \leftarrow E_K(v_2)$
3. $\vdots$
4. $c_n \leftarrow E_K(v_n)$

Voters encrypt 011 for "no" or "yes" and $v_i$ has vote $v_i$

At $EVE$ sits here and watches the ciphertexts

Claim: If $Eve$ learns the election outcome then she can decrypt all ciphertexts.

If $Eve$ is one of the voters she knows who voted the same as her (even for many choices of votes)
Suppose there is a missile silo and a general transmits fire / don't fire a missile. Many transmissions fix / don't fire.

\[ \leftarrow E_k(x) \]

Army General

Eve

If after the first ciphertext Eve sees what happens to the missile then she knows what will happen next time she sees a ciphertext.