Lecture 3 – Perfect Security and One-Time Pad

COSC-260 Codes and Ciphers

Adam O’Neill
Motivation
Motivation

We saw that substitution ciphers cannot hope to achieve even the weakest privacy notion.
Motivation

We saw that substitution ciphers cannot hope to achieve even the weakest privacy notion.

But what’s the strongest privacy notion one can hope for? Is there a scheme achieving it?
Shannon’s Work
Shannon’s Work

Shannon (1948) addressed this in his landmark work, *A Mathematical Theory of Communication.*
Shannon’s Work

Shannon (1948) addressed this in his landmark work, *A Mathematical Theory of Communication.*

This can be viewed as the birth of modern cryptography.
Perfect Security

**Definition 0.1.** A cryptosystem \((\mathcal{K}, \mathcal{E}, \mathcal{D})\) is *perfectly secure* if for all distributions \(\mathcal{D}\) on messages and every message \(g\) and every ciphertext \(c\)

\[
\Pr[g = m \mid \mathcal{E}(K, m) = c] = \Pr[m = g]
\]

where the probability is over \(K \leftarrow \mathcal{K}\) and \(m \leftarrow \mathcal{D}\).
**Definition 0.2.** A cryptosystem \((K, E, D)\) is **Shannon secure** if for all messages \(m_0, m_1\) and ciphertexts \(c\)

\[
\Pr[E(K, m_0) = c] = \Pr[E(K, m_1) = c]
\]

where the probability is over \(K \leftarrow \mathcal{K}\).
The Equivalence
Why is This Useful?

Perfect security guarantees what we want, but Shannon security is easier to work with.
One-Time Pad
Voting Example
Key Re-Use

Suppose the key is used twice. What can the adversary learn?
Optimality
Where To?
Where To?

We have a scheme achieving perfect security and a proof that it’s optimal.
Where To?

We have a scheme achieving perfect security and a proof that it’s optimal.

But key-length is completely impractical.
We have a scheme achieving perfect security and a proof that it’s optimal.

But key-length is completely impractical.

The main key idea of modern cryptography is that it is sufficient to consider efficient adversaries and allow “negligible” success.
Modern Cryptography: A Computational Science

In other words, security of a practical system must rely not on the impossibility but on the computational difficulty of breaking it.
Modern Cryptography:
A Computational Science
Modern Cryptography: A Computational Science

We might prove, e.g., no attack running in time (or resources) at most $2^{160}$ succeeds with probability greater than $2^{20}$. 
Modern Cryptography: A Computational Science

We might prove, e.g., no attack running in time (or resources) at most $2^{160}$ succeeds with probability greater than $2^{20}$.

I.e., attacks could exist as long as it is prohibitive (in time/space, $$$) to mount them.
Algorithm Complexity
Algorithm Complexity

We measure the running-time of algorithms in the `bit-length` of their inputs. Not absolute value!
Algorithm Complexity

We measure the running-time of algorithms in the bit-length of their inputs. Not absolute value!

Efficient algorithms have code size, time and space use, etc. which is, e.g., polynomial in the input length.
Factoring Example
Recall

Lower-level primitives

Higher-level primitives

construction

reduction
Quantitative Reductions
Where To?

Our first lower-level primitive, blockciphers.
Next time...