# CPSC 467: Cryptography and Computer Security

Michael J. Fischer

Lecture 5
September 13, 2017

# Modification Attack

# Man-in-the-middle attacks

An *active attacker* is one who can both read and alter messages en route to their destinations.

We refer to such an attacker as "Mallory", and we call such an attack a *man-in-the-middle* attack.

In a *modification attack*, Mallory can modify the contents of a message in specific semantically-meaningful ways even though (s)he has no idea what the message actually is.

# Modification attack against the Caesar cipher

Suppose Alice sends $c$ to Bob. Mallory intercepts it and changes $c$ to $(c + 5)$ mod 26.

Even though she doesn't know the key and cannot read $m$, she knows that she has changed $m$ to $(m + 5)$ mod 26.

Why? Let's do the calculations. (All arithmetic is modulo 26).

$$D_k(c') = D_k(c + 5) = c + 5 - k = D_k(c) + 5 = m + 5.$$

Depending on the application, this could be a devastating attack. Suppose Alice were a financial institution that was making a direct deposit of $m$ thousand dollars to Mallory's bank account at the Bob bank. By this attack, Mallory could get an extra 5 thousand dollars put into her account each month.

# A modification attack on English vowels

In our encoding scheme, vowels are represented by even numbers: $A = 0$, $E = 4$, $I = 8$, $O = 14$, and $U = 20$. If $m$ is a vowel, then $m' = (m + 5) \bmod 26$ is guaranteed not to be a vowel.

How could Mallory use this to his advantage?

# A general's orders

- Suppose Alice is a general sending an order to a field commander whether or not to attack.
- She uses the Caesar cipher to encrypt the order.
- A vowel means to attack; a consonent to hold the position.
- She feels very clever for having encoded the attack bit in such a non-obvious way.
- Mallory's $c + 5$ transformation changes every "attack" message to "don't attack" (and some "don't attack messages" to "attack").
- This effectively prevents Alice from attacking when it is to her advantage.

The fact that she was using a cryptosystem for which perfect secrecy is known did not protect her.

# Moral

The security of a system in practice depends critically on the kinds of attacks available to an attacker.

In this case, the cryptosystem that is provably perfectly secure against a passive eavesdropper using a ciphertext-only attack fails miserably against a known plaintext attack or against an active attacker.

# Computational Security

# A mathematical definition of computational security

We have looked at several different notions of confidentiality. For each, there is a corresponding security problem, namely, find a cryptosystem with the desired confidentiality properties.

These properties involve:

- The time complexity of encryption and decryption.
- The time complexity for a probabilistic adversary to violate confidentiality.
- The probability of a successful attack within an assumed time bound.

We proceed to complete our formal definition of computational security.

# Where do we assume randomness?

1. The message is drawn at random from some arbitrary probability distribution over the message space $\mathcal{M}$. Both $\mathcal{M}$ and the distribution are part of Eve's *a priori* knowledge.
2. The secret key is chosen uniformly at random from the key space $\mathcal{K}$.
3. Eve has access to an *independent* source of randomness which she may use while attempting to break the system. For example, Eve can choose an element $k' \in \mathcal{K}$ at random. With probability $p = 1/|\mathcal{K}|$, her element $k'$ is actually the correct key $k$.

# Independence

The three sources of randomness are assumed to be *statistically independent*.

Eve's random numbers do not depend on (nor give any information about) the message or key used by Alice.

Alice's key does not depend on the particular message or vice versa.

# Joint probability distribution

These multiple sources of randomness give rise to a *joint probability distribution* that assigns a well-defined probability to each triple $(m, k, z)$, where $m$ is a message, $k$ a key, and $z$ is the result of the random choices Eve makes during her computation.

The independence assumption implies that

$$\Pr[\mathrm{m}, \mathrm{k}, \mathrm{z}] = \Pr[\mathrm{m}] \times \Pr[\mathrm{k}] \times \Pr[\mathrm{z}]$$

where

- $\Pr[\mathrm{m}]$ is the probability that $m$ is the chosen message,
- $\Pr[\mathrm{k}]$ is the probability that $k$ is the chosen key,
- $\Pr[\mathrm{z}]$ is the probability that $z$ represents Eve's random choices.

# Eve's success probability

The joint distribution gives rise to an overall success probability for Eve (once we decide on what it means for an attack to succeed).

We want Eve's success probability to be "small".

Here, "small" is measured relative to a *security parameter* $s$, which you can think of as the key length.

# Negligible function

## Definition

A function $f$ is *negligible* if for every polynomial $p(\cdot)$ there exists an $N$ such that for all integers $n > N$ it holds that $f(n) < \frac{1}{p(n)}$.

We require that the success probability be a negligible function of the security parameter $s$.

# Computational security

Putting this all together, we get a general notion of computational security.

## Definition

A cryptosystem is *computationally secure* relative to a notion of compromise if, for all probabilistic polynomial-time algorithms $\mathcal{A}$, when given as input the security parameter $s$ and all of the information available to Eve, the algorithm succeeds in compromising the cryptosystem with success probability that is negligible in $s$.

# Practical security considerations

In practice, the important tradeoff is between the amount of time that Alice and Bob are willing to spend to use the cryptosystem versus what a determined adversary might be willing to spend to break the system.

Asymptotic complexity results will not tell us how to set the security parameter for a system, but they may inform us about how much security improvement we can expect as the key length increases.

# Information-Theoretic Security

# A mathematical definition of information-theoretic security

In order to talk precisely about computational security we needed formal definitions of time complexity and success probability.

Similarly, in order to talk precisely about information-theoretic security we need formal definitions of probability theory and statistical independence.

# Probability distributions and events

We give a quick overview of probability theory.

A *discrete probability distribution* $p$ assigns a real number $p_\omega \in [0, 1]$ to each element $\omega$ of a probability space $\Omega$ such that

$$\sum_{\omega \in \Omega} p_\omega = 1.$$

An *event* $E$ is a subset of $\Omega$. The probability of $E$ is

$$\Pr[\mathrm{E}] = \sum_{\omega \in \mathrm{E}} \mathrm{p}_\omega.$$

# Random variables

A *random variable* is a function $X : \Omega \to \mathcal{X}$, where $\mathcal{X}$ is a set.

We think of $X$ as describing a random choice according to distribution $p$.

Let $x \in \mathcal{X}$. Event $X = x$ means that the outcome of choice $X$ is $x$.

Formally, the event $X = x$ is the set $\{\omega \in \Omega \mid X(\omega) = x\}$.
Its probability is therefore

$$\Pr[\mathrm{x} = \mathrm{X}] = \sum_{\omega:\mathrm{X}(\omega)=\mathrm{x}} \mathrm{p}_\omega.$$

We sometimes ambiguously write $x$ to denote the event $X = x$.

# Experiments

Sometimes $m$ denotes the random variable that describes the experiment of Alice choosing a message $m \in \mathcal{M}$ according to the assumed message distribution.

Other times, $m$ denotes a particular message in set $\mathcal{M}$.

Hopefully, which meaning is intended will be clear from context.

# Conditional probability

Let $E$ and $F$ be events and assume $\Pr[\mathrm{F}] \neq 0$. The conditional probability of $E$ given $F$ is defined by

$$\Pr[\mathrm{E} \mid \mathrm{F}] = \frac{\Pr[\mathrm{E} \cap \mathrm{F}]}{\Pr[\mathrm{F}]}.$$

Intuitively, it is the probability that $E$ holds given that $F$ is known to hold.

Example:

| Ω | p |
|---|---|
| 1 | .2 |
| 2 | .2 |
| 3 | .3 |
| 4 | .1 |
| 5 | .2 |

$E = \{1, 2, 3\}$, $F = \{2, 3, 4\}$.

$\Pr[\mathrm{E}] = .2 + .2 + .3 = .7$

$\Pr[\mathrm{F}] = .2 + .3 + .1 = .6$

$\Pr[\mathrm{E} \cap \mathrm{F}] = .2 + .3 = .5$

$\Pr[\mathrm{E} \mid \mathrm{F}] = .2/.6 + .3/.6 = 5/6.$

# Statistical independence

Formally, events $E$ and $F$ are *statistically independent* if $\Pr[\mathrm{E} \mid \mathrm{F}] = \Pr[\mathrm{E}]$.

An equivalent definition is that $\Pr[\mathrm{E} \cap \mathrm{F}] = \Pr[\mathrm{E}] \cdot \Pr[\mathrm{F}]$.

This is easily seen by dividing both sides by $\Pr[\mathrm{F}]$ and applying the definition of $\Pr[\mathrm{E} \mid \mathrm{F}]$.

(This assumes $\Pr[\mathrm{F}] \neq 0$.)

# Information-theoretic security

Putting this all together, we get a general notion of information-theoretic security.

## Definition

A cryptosystem is *information-theoretically secure* if $\Pr[\mathrm{m}] = \Pr[\mathrm{m} \mid \mathrm{c}]$.

In words, *c* gives no information about *m*.

This is equivalent to saying that *m* and *c* are *statistically independent*.

We also call this *perfect secrecy*.

# Example: Caesar cipher on 1-letter messages

Simplify the Casear cipher by restricting to a 3-letter alphabet.

$\mathcal{M} = \mathcal{C} = \mathcal{K} = \{0, 1, 2\}$
$E_k(m) = (m + k) \bmod 3$
$D_k(m) = (m - k) \bmod 3.$

## Theorem

*The simplified Caesar cipher achieves perfect secrecy.*

# Joint message-key distribution

A priori message probabilities:

| $m$ | $p_m$ |
|---|---|
| 0 | 1/2 |
| 1 | 1/3 |
| 2 | 1/6 |

Each key has probability 1/3.

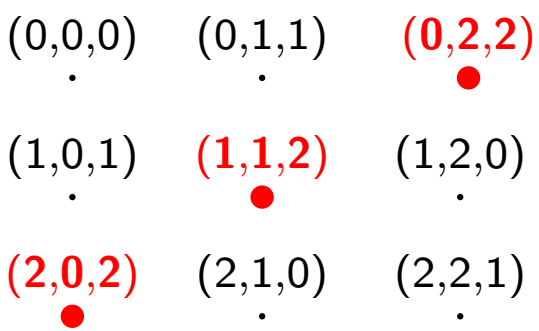
Joint probability distribution:

| | | $k$ | | |
|---|---|---|---|---|
| | | 0 | 1 | 2 |
| $m$ | 0 | 1/6 | 1/6 | 1/6 |
| | 1 | 1/9 | 1/9 | 1/9 |
| | 2 | 1/18 | 1/18 | 1/18 |

# Conditional probability distribution

$\Pr[\mathrm{m} = 1] = 1/3$.
Eve sees $c = 2$.
She wishes to compute $\Pr[\mathrm{m} = 1 \mid \mathrm{c} = 2]$.

First, find the sample space $\Omega$.
Points in $\Omega$ are triples $(m, k, c)$, where $c = E_k(m)$.:




Points for which $c = 2$ are shown in bold red.

# Proof of perfect secrecy

$\Pr[c = 2]$ is the sum of the probabilities of the bold face points, i.e., $1/6 + 1/9 + 1/18 = 6/18 = 1/3$.

| | | $k$ = 0 | 1 | 2 |
|---|---|---|---|---|
| $m$ | 0 | 1/6 | 1/6 | **1/6** |
| | 1 | 1/9 | **1/9** | 1/9 |
| | 2 | **1/18** | 1/18 | 1/18 |

The only point for which $m = 1$ is $(1, 1, 2)$ (the center point).
It's probability is $1/9$, so $\Pr[m = 1 \wedge c = 2] = 1/9$.
By definition of conditional probability,

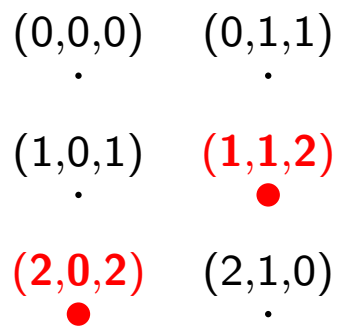
$$\Pr[m = 1 \mid c = 2] = \frac{\Pr[m = 1 \wedge c = 2]}{\Pr[c = 2]} = \frac{1/9}{1/3} = \frac{1}{3} = \Pr[m = 1].$$

Similarly, $\Pr[m = m_0 \mid c = c_0] = \Pr[m = m_0]$ for all $m_0$ and $c_0$.
Hence, simplified Caesar cipher is information-theoretically secure.

# A minor change

Suppose we reduce the key space to $\mathcal{K} = \{0, 1\}$.
The a priori message distribution stays the same, but the joint probability distribution changes as does the sample space.

| $m$ \ $k$ | 0 | 1 |
|---|---|---|
| 0 | 1/4 | 1/4 |
| 1 | 1/6 | 1/6 |
| 2 | 1/12 | 1/12 |




Now, $\Pr[\mathrm{c} = 2] = 1/6 + 1/12 = 3/12 = 1/4$, and $\Pr[\mathrm{m} = 1 \wedge \mathrm{c} = 2] = 1/6$. Hence,

$$\Pr[\mathrm{m} = 1 \mid \mathrm{c} = 2] = \frac{1/6}{1/4} = \frac{2}{3} \neq \frac{1}{3} = \Pr[\mathrm{m} = 1].$$

# Perfect secrecy lost

The probability that $m = 1$ given $c = 2$ is double what it was.

Once Eve sees $c = 2$ there are only two possibilities for $m$:

1. $m = 1$ (and $k = 1$)
2. $m = 2$ (and $k = 0$).

No longer possible that $m = 0$!

Eve narrows the possibilities for $m$ to the set $M = \{1, 2\} \subseteq \mathcal{M}$. Her probabilistic knowledge of $m$ changes from the initial distribution $(1/2, 1/3, 1/6)$ to the new distribution $(0, 2/3, 1/3)$. She has learned at lot about $m$, even without finding it exactly.

A seemingly minor change turns a cryptosystem with perfect secrecy into one that leaks a considerable amount of information!

# Symmetric Cryptosystem Families

# Letter-by-letter encryption

A *stream cipher* is any cryptosystem that operates in an online fashion:

- The message is encrypted one letter at a time.
- After each message letter is read, one or more ciphertext letters are emitted as output.

Polyalphabetic substitution ciphers such as Caesar, Vigenère, Enigma machines, and even the one-time pad, are all examples of stream ciphers.

# Structure of stream cipher

A stream cipher can be built from two components:

1. a cipher that is used to encrypt a given character;
2. a keystream generator that produces a different key to be used for each successive letter.

A commonly-used cipher is the simple XOR cryptosystem, also used in the one-time pad.

Rather than using a long random string for the keystream, we instead use a pseudorandom keystream generated on the fly using a state machine.

Like a one-time pad, a different master key (seed) must be used for each message; otherwise the system is easily broken.

# Encrypting several letters at a time

A *block cipher* is a cryptosystem that operates on fixed-length blocks of letters.

- The whole message is assumed to be initially available.
- For a block length $b$, the message is padded, if necessary, so its total length is a multiple of $b$.
- The message is split into a sequence of blocks and encrypted one block at a time.
- For each block, one or more ciphertext blocks are emitted as output.

The polygraphic ciphers discussed in lecture 2 (such as the Hill and Playfair ciphers), are examples of block ciphers.

# Duality between stream and block ciphers

A block cipher can be viewed as a stream cipher on a sequence of blocks, where each block is treated as a "letter" in an expanded alphabet.

What kind of cipher is the Caesar cipher?