UCS1505 INTRODUCTION TO CRYPTOGRAPHIC TECHNIQUES

Jonathan Katz, Yehuda Lindell,

“Introduction to Modern Cryptography”,

2nd Edition

(Chapman & Hall/CRC Cryptography and Network Security Series), 2014

Course Objectives

- To understand the classical and symmetric cryptographic techniques
- To study about message authentication and hash functions
- To learn number theory fundamentals needed by cryptographic algorithms
- To understand the various key distribution and management schemes
- To understand the concepts of Public key cryptography and digital signatures.

Course Outcomes

- Describe and implement classical and symmetric ciphers (K2)
- Describe the authentication schemes and hash algorithms (K2)
- Understand the number theoretic foundations of cryptography (K3)
- Compare and contrast various public key cryptographic techniques (K5)
- Illustrate various public key cryptographic techniques (K3).

Cryptography (historically)

"…the art of writing or solving codes…"

- Historically, cryptography focused exclusively on ensuring *private communication* between two parties sharing secret information in advance using "codes" (aka *private-key encryption*)

- Historically, cryptography was an *art*
  - Heuristic, unprincipled design and analysis
  - Schemes proposed, broken, repeat…

Modern cryptography

- Much broader scope and deals with
  - Data integrity, authentication, protocols,
  - The *public-key setting*
  - Group communication
  - More-complicated trust models
  - Foundations (e.g., number theory, quantum-resistance) to systems (e.g., electronic voting, blockchain, cryptocurrencies)

Modern cryptography

Design, analysis, and implementation of ***mathematical techniques*** *for* securing information*, systems, and distributed computations against adversarial attack*

- Cryptography is now much more of a *science*
  - Rigorous analysis, firm foundations, deeper understanding, rich theory

Cryptography (historically)

- Used primarily for military/government applications, plus a few niche applications in industry (e.g., banking)

Modern cryptography

- Cryptography is ubiquitous!
  - Password-based authentication, password hashing
  - Secure credit-card transactions over the internet
  - Encrypted WiFi
  - Disk encryption
  - Digitally signed software updates
  - Bitcoin

Basics

	Secrecy	Integrity
Private-key setting	Private-key encryption	Message authentication codes
Public-key setting	Public-key encryption	Digital signatures

- Building blocks
  - Pseudorandom (number) generators
  - Pseudorandom functions/block ciphers
  - Hash functions
  - Number theory

Classical Cryptography

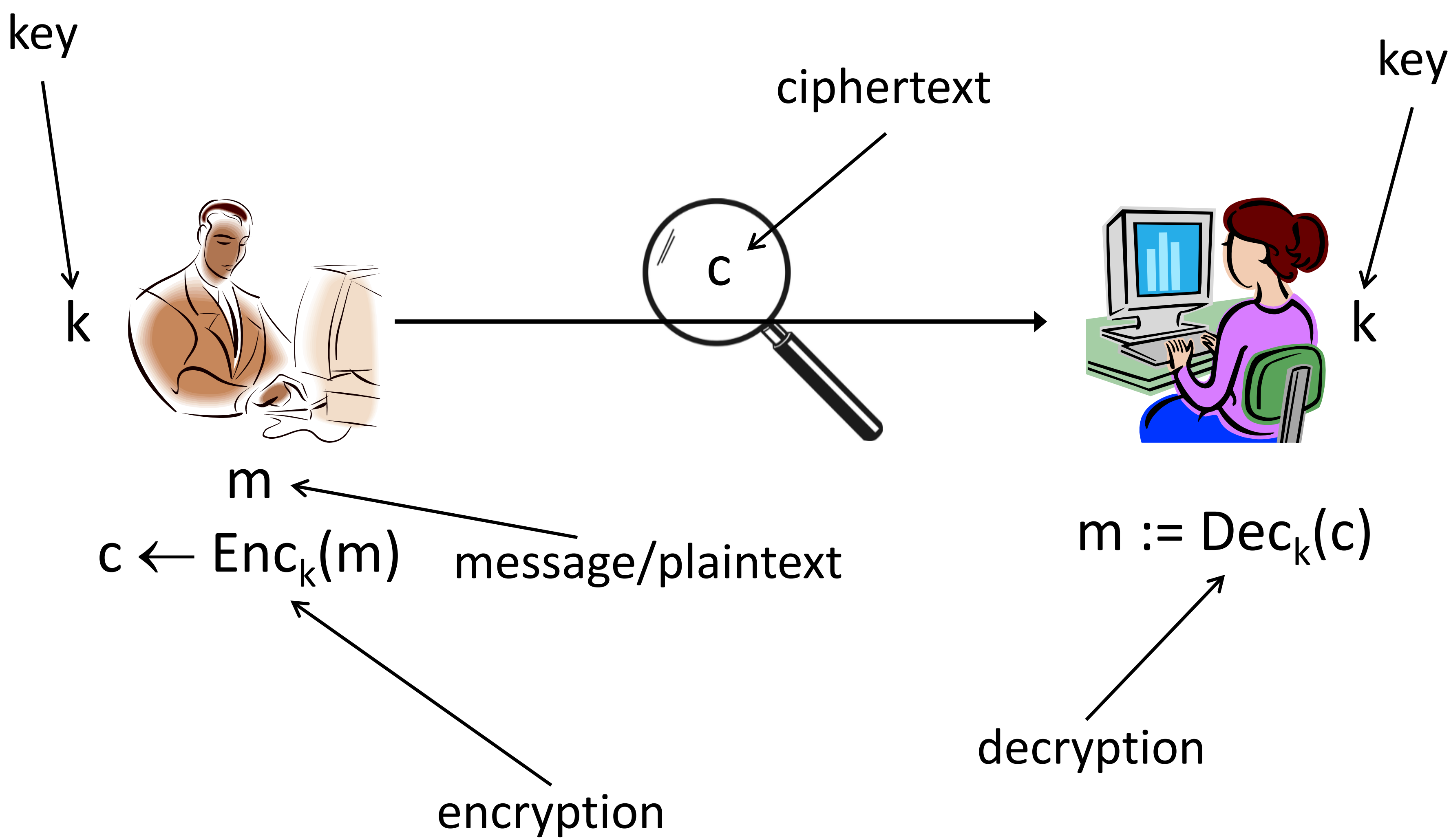
Classical cryptography

- Until the 1970s, relied exclusively on secret information (a *key*) shared in advance between the communicating parties

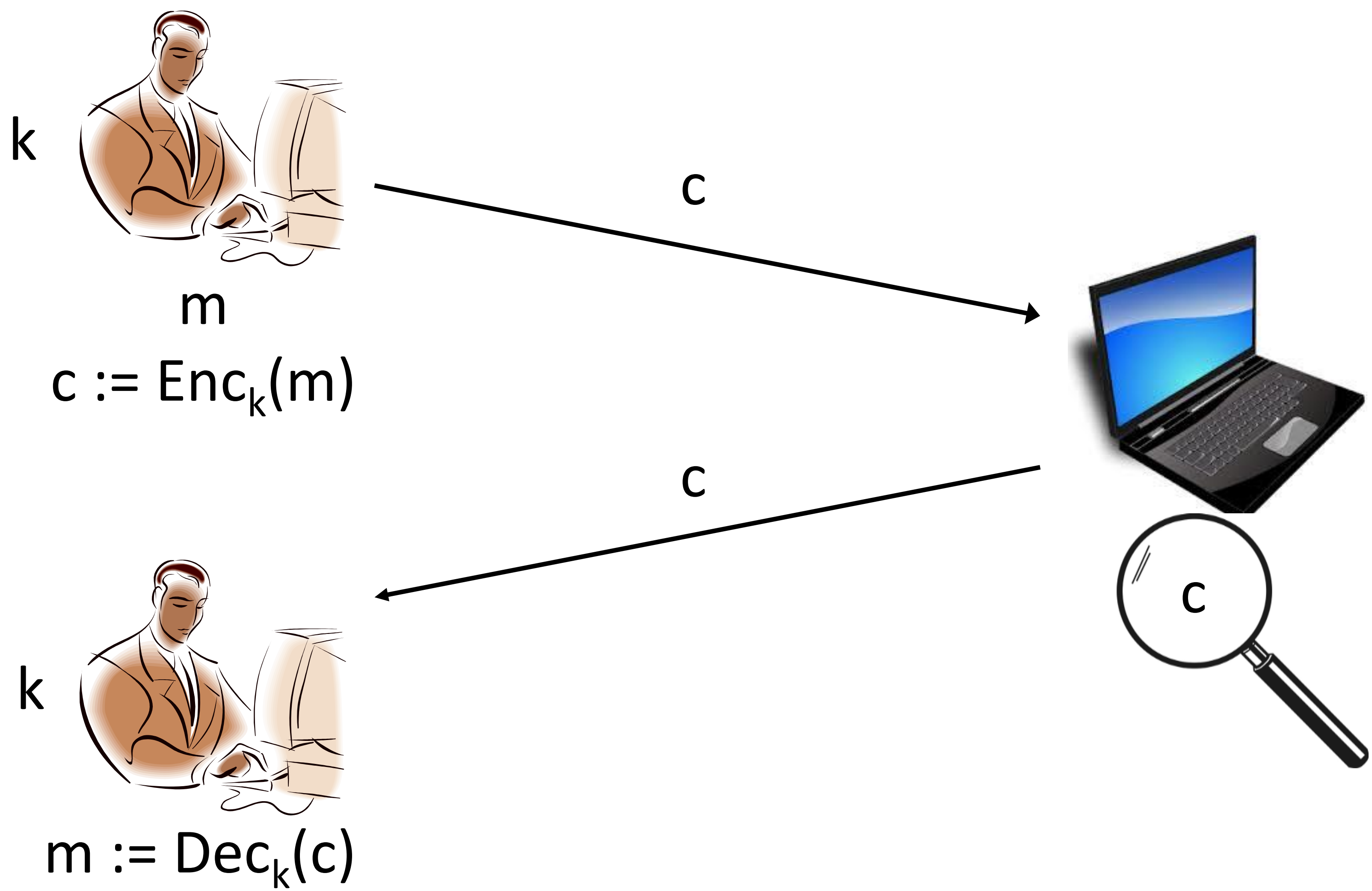
Private-key cryptography

– aka secret-key / shared-key / symmetric-key cryptography

Private-key encryption



Private-key encryption



Private-key encryption

- A *private-key encryption scheme* is defined by a message space $\mathcal{M}$ and algorithms (Gen, Enc, Dec):
  - Gen (key-generation algorithm): outputs $k \in \mathcal{K}$
  - Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$ as input; outputs ciphertext c

    $$c \leftarrow \mathrm{Enc}_k(m)$$

  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m or "error"

    $$m := \mathrm{Dec}_k(c)$$

Correctness requirement:
For all $m \in \mathcal{M}$ and k output by Gen,
$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = m$$

Kerckhoffs's principle

- *The encryption scheme* is not secret
  - The attacker knows the encryption scheme
  - The only secret is the *key*
  - The key must be chosen at random; kept secret
- Arguments in favor of this principle
  - Easier to keep *key* secret than *algorithm*
  - Easier to change *key* than to change *algorithm*
  - Standardization
    - Ease of deployment
    - Public scrutiny

Caesar's cipher.

- Julius Caesar encrypted by shifting the letters of the alphabet ***3*** places forward
- Immediate problem with this cipher is that the encryption method is fixed

The shift cipher

- Consider encrypting English text
- Associate ‘a’ with 0; ‘b’ with 1; …; ‘z’ with 25
- $k \in \mathcal{K} = \{0, \ldots, 25\}$
- To encrypt using key k, shift every letter of the plaintext by k positions (with wraparound)
- Decryption just does the reverse

The shift cipher, formally

- $\mathcal{M}$ = {strings over lowercase English alphabet}
- Gen: choose uniform $k \in \{0, \ldots, 25\}$
- $\text{Enc}_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $$c_i := [m_i + k \bmod 26]$$
- $\text{Dec}_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where
  $$m_i := [c_i - k \bmod 26]$$
- Can verify that correctness holds…

Is the shift cipher secure?

- No -- only 26 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - Only one possibility will “make sense”
  - (What assumptions are we making here?)

- Example of a “brute-force” or “exhaustive-search” attack
- An attack that involves trying every possible key.
- *sufficient key-space principle:*

Any secure encryption scheme must have a key space that is sufficiently large to make an exhaustive-search attack infeasible.

Determination of feasibility depends on resources , Time

Example

- Ciphertext `uryybjbeyq`
- Try every possible key…
  - `tqxxaiadxp`
  - `spwwzhzcwo`
  - …
  - `helloworld`

mono-alphabetic substitution cipher

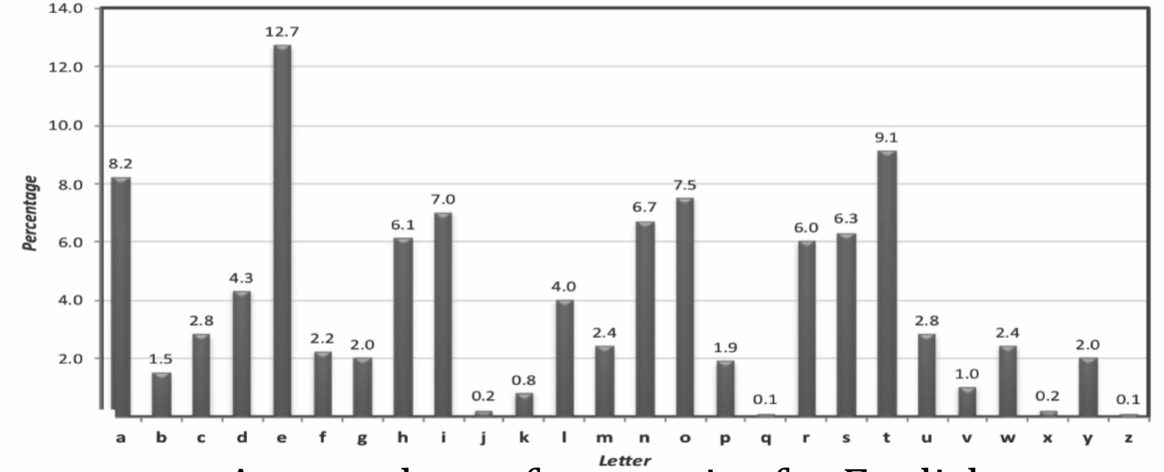
- In the mono-alphabetic substitution cipher the key also defines a map on the alphabet, but the map is now allowed to be arbitrary subject only to the constraint that it be one-to-one

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	E	U	A	D	N	B	K	V	M	R	O	C	Q	F	S	Y	H	W	G	L	Z	I	J	P	T

tellhimaboutme

Assuming the English alphabet is being used, the key space is of size $26! = 26 \cdot 25 \cdot 24 \cdots 2 \cdot 1$, or approximately 2^{88}, and a brute-force attack is infeasible but not secure.

- mono-alphabetic substitution cipher can then be attacked by utilizing statistical properties of the English language.



Average letter frequencies for English-language text.

Try this

- JGRMQOYGHMVBJWRWQFPWHGFFDQGFPFZRKBEEBJIZQQOCIBZKLFAFGQVFZFW
WEOGWOPFGFHWOLPHLRLOLFDMFGQWBLWBWQOLKFWBYLBLYLFSFLJGRMQBOL
WJVFPFWQVHQWFFPQOQVFPQOCFPOGFWFJIGFQVHLHLROQVFGWJVFPFOLFHGQ
VQVFILEOGQILHQFQGIQVVOSFAFGBWQVHQWIJVWJVFPFWHGFIWIHZZRQGBABHZ
QOCGFHX

Improved attack on the shift cipher

- Let p_i, with $0 \le p_i \le 1$, denote the frequency of the ith letter in normal English text

$$\sum_{i=0}^{25} p_i^2 \approx 0.065$$

- Let qi denote the frequency of the ith letter of the alphabet in this ciphertext; i.e., q_i is simply the number of occurrences of the ith letter of the alphabet in the ciphertext divided by the length of the ciphertext.
- If the key is k, then p_i should be roughly equal to q_{i+k} for all i because the ith letter is mapped to the $(i + k)$th letter.

$$I_j \stackrel{\text{def}}{=} \sum_{i=0}^{25} p_i \cdot q_{i+j}$$

Improved attack on the shift cipher

- For each value of $j \in \{0,\ldots,25\}$, then we expect to find that $i_k \approx 0.065$ (where k is the actual key),
- Whereas i_j for j != k will be different from 0.065.
- This leads to a key-recovery attack that is easy to automate
- Compute I_j for all j, and then output the value k for which I_k is closest to 0.065.

Poly-alphabetic shift cipher - The Vigenère cipher

- The key is now a *string*, not just a character
- To encrypt, shift each character in the plaintext by the amount dictated by the next character of the key
  - Wrap around in the key as needed
- Decryption just reverses the process

```
tellhimaboutme
cafecafecafeca
--------------
veqpjiredozxoe
```

The Vigenère cipher

- Size of key space?
  - If keys are 14-character strings over the English alphabet, then key space has size $26^{14} \approx 2^{66}$
  - If variable length keys, even more…
  - Brute-force search infeasible
- Is the Vigenère cipher secure?

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
BCDEFGHIJKLMNOPQRSTUVWXYZA
CDEFGHIJKLMNOPQRSTUVWXYZAB
DEFGHIJKLMNOPQRSTUVWXYZABC
EFGHIJKLMNOPQRSTUVWXYZABCD
FGHIJKLMNOPQRSTUVWXYZABCDE
GHIJKLMNOPQRSTUVWXYZABCDEF
HIJKLMNOPQRSTUVWXYZABCDEFG
IJKLMNOPQRSTUVWXYZABCDEFGH
JKLMNOPQRSTUVWXYZABCDEFGHI
KLMNOPQRSTUVWXYZABCDEFGHIJ
LMNOPQRSTUVWXYZABCDEFGHIJK
MNOPQRSTUVWXYZABCDEFGHIJKL
NOPQRSTUVWXYZABCDEFGHIJKLM
OPQRSTUVWXYZABCDEFGHIJKLMN
PQRSTUVWXYZABCDEFGHIJKLMNO
QRSTUVWXYZABCDEFGHIJKLMNOP
RSTUVWXYZABCDEFGHIJKLMNOPQ
STUVWXYZABCDEFGHIJKLMNOPQR
TUVWXYZABCDEFGHIJKLMNOPQRS
UVWXYZABCDEFGHIJKLMNOPQRST
VWXYZABCDEFGHIJKLMNOPQRSTU
WXYZABCDEFGHIJKLMNOPQRSTUV
XYZABCDEFGHIJKLMNOPQRSTUVW
YZABCDEFGHIJKLMNOPQRSTUVWX
ZABCDEFGHIJKLMNOPQRSTUVWXY
```

Attacking the Vigenère cipher

- Look at every 14th character of the ciphertext, starting with the first
  - Call this a “stream”
- Let α be the most common character appearing in this stream
- Most likely, α corresponds to the most common plaintext character (i.e., ‘e’)
  - Guess that the first character of the key is α - ’e’
- Repeat for all other positions

Finding the key length

- The previous attack assumes we know the key length
  - What if we don't?
- Note: can always try the previous attack for all possible key lengths
  - # of key lengths << # keys

Finding the key length

- When using the correct key length, the ciphertext frequencies $\{q_i\}$ of a stream will be *shifted versions* of the $\{p_i\}$
  - So $\Sigma\, q_i^2 \approx \Sigma\, p_i^2 \approx 0.065$
- When using an incorrect key length, expect (heuristically) that ciphertext letters are uniform
  - So $\Sigma\, q_i^2 \approx \Sigma\, (1/26)^2 = 1/26 = 0.038$

$$S_\tau \approx \sum_{i=0}^{25} \left(\frac{1}{26}\right)^2 \approx 0.038$$

- In fact, good enough to find the key length N that maximizes $\Sigma\, q_i^2$

Historically…

- Cryptography was an *art*
  - Heuristic design and analysis

- This isn’t very satisfying
  - How do we know when a scheme is secure?

Modern cryptography

- In the late ‘70s and early ‘80s, cryptography began to develop into more of a *science*
- Based on three principles that underpin most crypto work today

Core principles of modern crypto

- Formal definitions
  - Precise, mathematical model and definition of what security means

- Assumptions
  - Clearly stated and unambiguous

- Proofs of security
  - Move away from design-break-patch

Principles of Modern Cryptography

- Schemes are now developed and analyzed in a more systematic manner, with the ultimate goal being to give a rigorous proof that a given construction is secure.
- In order to articulate such proofs, one first need formal definitions that pin down exactly what "secure" means

Principle 1 – Formal Definitions Importance of definitions

- Definitions are *essential* for the design, analysis, and sound usage of crypto
- Developing a precise definition forces the designer to think about what they really want
  - What is essential and (sometimes more important) what is not
    - Often reveals subtleties of the problem

Importance of definitions -- design

If you don't understand what you want to achieve, how can you possibly know when (or if) you have achieved it?

Importance of definitions -- analysis

- Definitions enable meaningful analysis, evaluation, and comparison of schemes
  - Does a scheme satisfy the definition?
  - What definition does it satisfy?
    - Note: there may be multiple meaningful definitions!
    - One scheme may be less efficient than another, yet satisfy a stronger security definition

Importance of definitions -- usage

- Definitions allow others to understand the security guarantees provided by a scheme
- Enables schemes to be used as *components* of a larger system (modularity)
- Enables one scheme to be substituted for another if they satisfy the same definition

Two components of security definition

- A security definition has two components:
  - a security guarantee (or, from the attacker's point of view, what constitutes a successful attack) and a threat model.
- The security guarantee defines what the scheme is intended to prevent the attacker from doing.
- Threat model describes the power of the adversary, i.e., what actions the attacker is assumed able to carry out.

Secure encryption scheme guarantee

- It should be impossible for an attacker to recover the key
- It should be impossible for an attacker to recover the plaintext from the ciphertext
- It should be impossible for an attacker to recover any character of the plaintext from the ciphertext
- Regardless of any information an attacker already has, a ciphertext should leak no additional information about the underlying plaintext.

Threat model

- Ciphertext-only attack
- Known-plaintext attack
- Chosen-plaintext attack
- Chosen-ciphertext attack

Principle 2 – Precise Assumptions

- With few exceptions, cryptography currently requires *computational assumptions*
  - At least until we prove $P \neq NP$ (and even that would not be enough)

- Principle: any such assumptions should be made explicit and mathematically precise

Importance of clear assumptions

- Allow researchers to (attempt to) *validate* assumptions by studying them, should be examined and tested
- Allow meaningful *comparison* between schemes based on different assumptions
  - Useful to understand minimal assumptions needed
- Practical implications if assumptions are wrong

- Enable proofs of security

Principle 3 – Proofs of Security

- Provide a rigorous proof that a construction satisfies a given definition under certain specified assumptions
  - Provides an iron-clad guarantee (relative to your definition and assumptions!)

- Proofs are crucial in cryptography, where there is a malicious attacker trying to “break” the scheme

Test your Understanding

- Using the English-language shift cipher (as described in the book), which of the following plaintexts could correspond to ciphertext AZC?
  1. can
  2. bad
  3. dog
  4. run

Summary

- Classical Ciphers
- Modern cryptography
- Symmetric Ciphers