EE309 Advanced Programming Techniques for EE

Lecture 17: Classical cryptography

INSU YUN (윤인수)

School of Electrical Engineering, KAIST

[Slides from Introduction to Cryptography -- MATH/CMSC 456 at UMD]

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*)

Modern cryptography

- Much broader scope!
  - 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 (historically)

"…the art of writing or solving codes…"

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

Modern cryptography

- Cryptography is now much more of a *science*
  - Rigorous analysis, firm foundations, deeper understanding, rich theory
- The "crypto mindset" has permeated other areas of computer security
  - Threat modeling
  - Proofs of security

Rough course outline

	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

Motivation

- Allows us to “ease into things…,” introduce notation
- Shows why unprincipled approaches are dangerous
- Illustrates why things are more difficult than they may appear

Classical cryptography

- Until the 1970s, exclusively concerned with ensuring *secrecy* of communication

- I.e., *encryption*

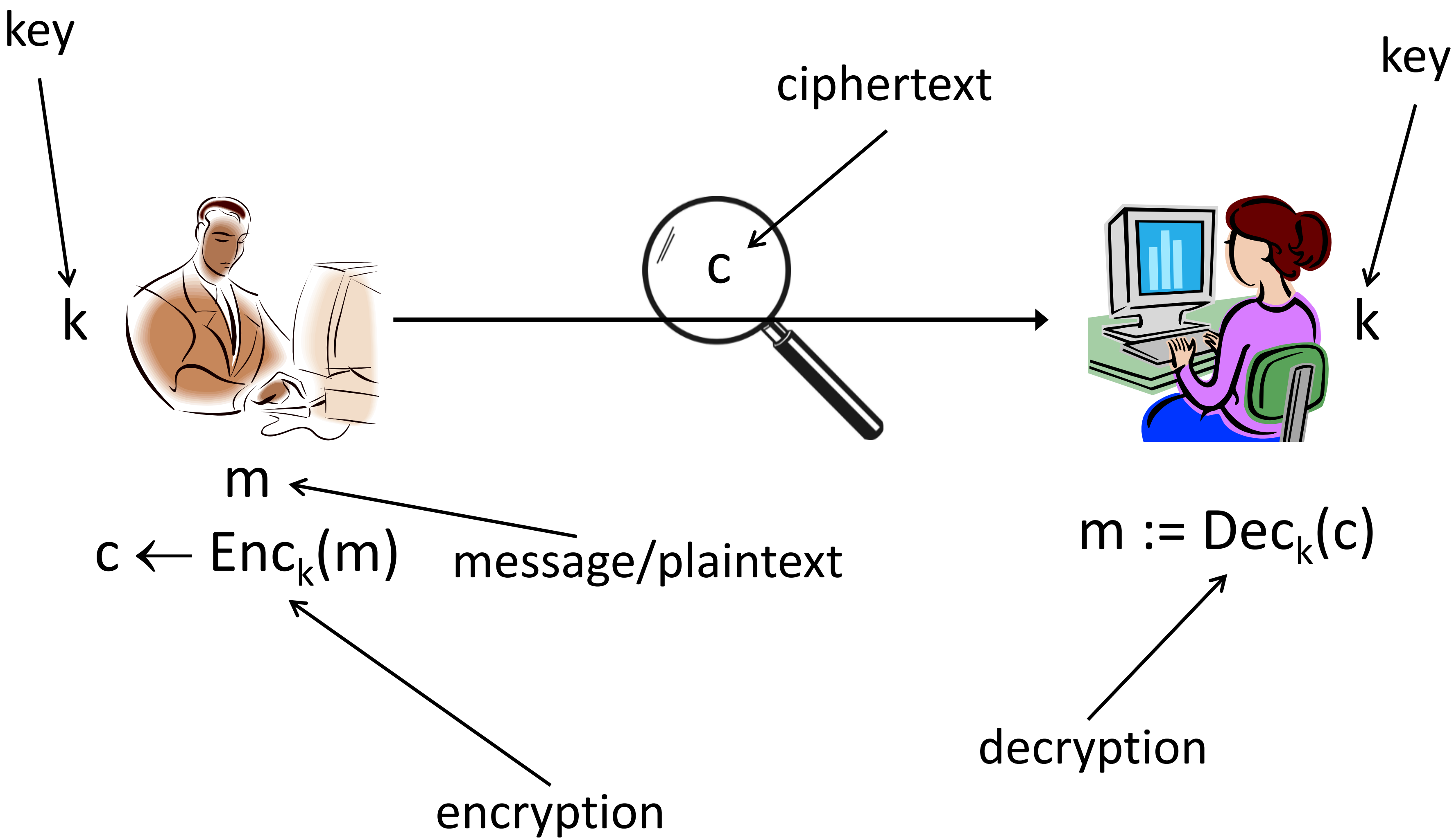
Classical cryptography

- 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

- 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

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

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

For all $m \in \mathcal{M}$ and k output by Gen,

$$\text{Dec}_k(\text{Enc}_k(m)) = m$$

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

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)
- Decry

```
helloworldz
ccccccccccc
-----------
jgnnqyqtnfb
```

Modular arithmetic

- $x = y \bmod N$ if and only if N divides $x-y$
- $[x \bmod N]$ = the remainder when x is divided by N
  - I.e., the unique value $y \in \{0, \ldots, N-1\}$ such that $x = y \bmod N$

- $25 = 35 \bmod 10$
- $25 \neq [35 \bmod 10]$
- $5 = [35 \bmod 10]$

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”

- Example of a “brute-force” or “exhaustive-search” attack

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

Example

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

Byte-wise shift cipher

- Work with an alphabet of *bytes* rather than (English, lowercase) *letters*
  - Works natively for arbitrary data!

- Use XOR instead of modular addition
  - Essential properties still hold

ASCII

- Characters (often) represented in ASCII
  - 1 byte/char = 2 hex digits/char

Hex	Dec	Char		Hex	Dec	Char	Hex	Dec	Char	Hex	Dec	Char
0x00	0	NULL	null	0x20	32	Space	0x40	64	@	0x60	96	`
0x01	1	SOH	Start of heading	0x21	33	!	0x41	65	A	0x61	97	a
0x02	2	STX	Start of text	0x22	34	"	0x42	66	B	0x62	98	b
0x03	3	ETX	End of text	0x23	35	#	0x43	67	C	0x63	99	c
0x04	4	EOT	End of transmission	0x24	36	$	0x44	68	D	0x64	100	d
0x05	5	ENQ	Enquiry	0x25	37	%	0x45	69	E	0x65	101	e
0x06	6	ACK	Acknowledge	0x26	38	&	0x46	70	F	0x66	102	f
0x07	7	BELL	Bell	0x27	39	'	0x47	71	G	0x67	103	g
0x08	8	BS	Backspace	0x28	40	(	0x48	72	H	0x68	104	h
0x09	9	TAB	Horizontal tab	0x29	41	)	0x49	73	I	0x69	105	i
0x0A	10	LF	New line	0x2A	42	*	0x4A	74	J	0x6A	106	j
0x0B	11	VT	Vertical tab	0x2B	43	+	0x4B	75	K	0x6B	107	k
0x0C	12	FF	Form Feed	0x2C	44	,	0x4C	76	L	0x6C	108	l
0x0D	13	CR	Carriage return	0x2D	45	-	0x4D	77	M	0x6D	109	m
0x0E	14	SO	Shift out	0x2E	46	.	0x4E	78	N	0x6E	110	n
0x0F	15	SI	Shift in	0x2F	47	/	0x4F	79	O	0x6F	111	o
0x10	16	DLE	Data link escape	0x30	48	0	0x50	80	P	0x70	112	p
0x11	17	DC1	Device control 1	0x31	49	1	0x51	81	Q	0x71	113	q
0x12	18	DC2	Device control 2	0x32	50	2	0x52	82	R	0x72	114	r
0x13	19	DC3	Device control 3	0x33	51	3	0x53	83	S	0x73	115	s
0x14	20	DC4	Device control 4	0x34	52	4	0x54	84	T	0x74	116	t
0x15	21	NAK	Negative ack	0x35	53	5	0x55	85	U	0x75	117	u
0x16	22	SYN	Synchronous idle	0x36	54	6	0x56	86	V	0x76	118	v
0x17	23	ETB	End transmission block	0x37	55	7	0x57	87	W	0x77	119	w
0x18	24	CAN	Cancel	0x38	56	8	0x58	88	X	0x78	120	x
0x19	25	EM	End of medium	0x39	57	9	0x59	89	Y	0x79	121	y
0x1A	26	SUB	Substitute	0x3A	58	:	0x5A	90	Z	0x7A	122	z
0x1B	27	FSC	Escape	0x3B	59	;	0x5B	91	[	0x7B	123	{
0x1C	28	FS	File separator	0x3C	60	<	0x5C	92	\	0x7C	124	|
0x1D	29	GS	Group separator	0x3D	61	=	0x5D	93	]	0x7D	125	}
0x1E	30	RS	Record separator	0x3E	62	>	0x5E	94	^	0x7E	126	~
0x1F	31	US	Unit separator	0x3F	63	?	0x5F	95	_	0x7F	127	DEL

Source: http://benborowiec.com/2011/07/23/better-ascii-table/

Byte-wise shift cipher

- $\mathcal{M}$ = {strings of bytes}
- Gen: choose uniform $k \in \mathcal{K}$ = {0x00, …, 0xFF}
  - 256 possible keys
- $\text{Enc}_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $$c_i := m_i \oplus k$$
- $\text{Dec}_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where
  $$m_i := c_i \oplus k$$

- Verify that correctness holds…

Is this scheme secure?

- No -- only 256 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - If ciphertext is long enough, only one plaintext will “make sense”

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?

- (Believed secure for many years…)

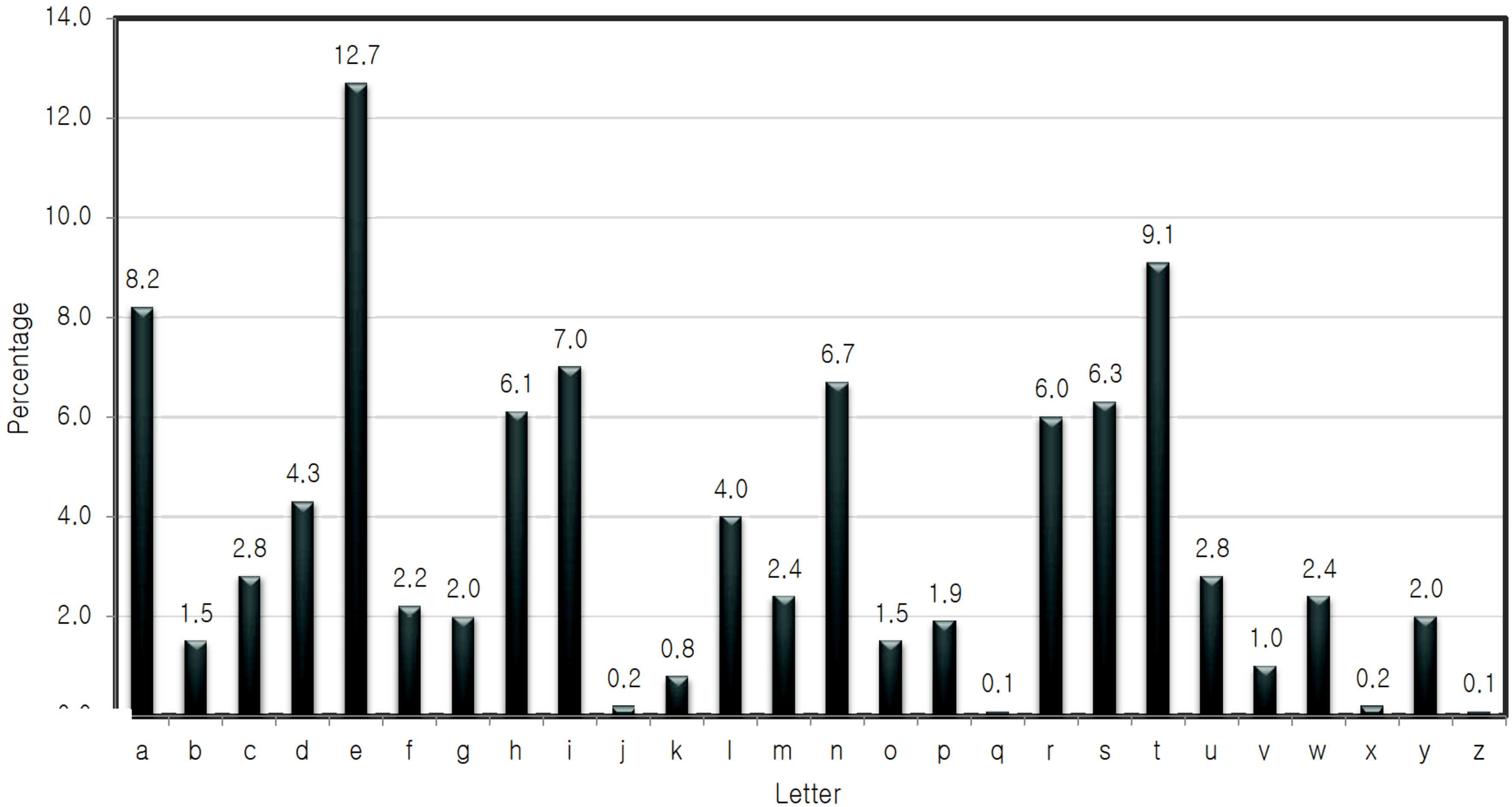
Attacking the Vigenère cipher

- (Assume a 14-character key)
- Observation: every 14th character is "encrypted" using the same shift

```
veqpjiredozxoeualpcmsdjqu
iqndnossoscdcusoakjqmxpqr
hyycjqoqqodhjcciowieii
```

- Looki… (almost) like looking at ciphertext encrypted with the shift cipher
  – Though a direct brute-force attack doesn't work…
  – Why not?

Using plaintext letter frequencies



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

- Better attacks for Vigenère cipher exist, but do not discuss this in our lecture

So far…

- “Heuristic” constructions; construct, break, repeat, …
- Can we *prove* that some encryption scheme is secure?
- First need to *define* what we mean by “secure” in the first place…

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

Importance of definitions

- Definitions are *essential* for the design, analysis, and sound usage of crypto

Importance of definitions -- design

- 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

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

Importance of clear assumptions

- Allow researchers to (attempt to) *validate* assumptions by studying them
- 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

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

Limitations?

- Cryptography remains partly an *art* as well
- Given a proof of security based on some assumption, we still need to *instantiate* the assumption
  - Validity of various assumptions is an active area of research

Limitations?

- Proofs given an iron-clad guarantee of security
  - …relative to the definition and the assumptions!

- Provably secure schemes can be broken!
  - If the definition does not correspond to the real-world threat model
    - I.e., if attacker can go “outside the security model”
    - This happens a lot in practice
  - If the assumption is invalid
  - If the implementation is flawed
    - This happens a lot in practice

Nevertheless…

- This does not detract from the importance of having formal definitions in place
- This does not detract from the importance of proofs of security