Introduction to Cryptography

Lecture 2

Vahid Amin-Ghafari

Vahidaming@ustc.edu.cn

Introduction and Classical Cryptography

- Crypto is amazing!
  - Can do things that seem impossible…
- Crypto is *important* and *pervasive*
  - It impacts each of us every day
- Crypto is fun!
  - Deep theory interacting with practice
  - Attackers' mindset, fun assignments

Textbook

- **Required** textbook: “Cryptography Theory and Practice, Fourth Edition” *Stinson and Paterson* (Chinese version)

- **Required** textbook: “Introduction to Modern Cryptography, 3rd edition,” *Katz and Lindell*

TEXTBOOKS IN MATHEMATICS

Cryptography

Theory and Practice

FOURTH EDITION

Douglas R. Stinson
Maura B. Paterson

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

CHAPMAN & HALL/CRC
CRYPTOGRAPHY AND NETWORK SECURITY

Jonathan Katz
Yehuda Lindell

Introduction to MODERN CRYPTOGRAPHY

Third Edition

CRC Press
Taylor & Francis Group
A CHAPMAN & HALL BOOK

How to reach me

- Best way to contact me is by email: Vahidaming@ustc.edu.cn
- Please put “cryptography course” in subject line

- Please email me in advance if you plan to come to office hours

- Questions?

- Please ask questions throughout!

Course goals

- Understand the *theoretical foundations* for *real-world cryptography*
- When you encounter crypto in your career:
  - Understand the key terms
  - Understand the security guarantees needed/provided
  - Know how to use crypto
  - Understand what goes on “under the hood”
- “Crypto mindset”

Course non-goals

- Designing your own crypto schemes
  - This is hard!
- Implementing crypto for real-world use
  - This is hard!

- Course goal:
  realize when to consult an expert!

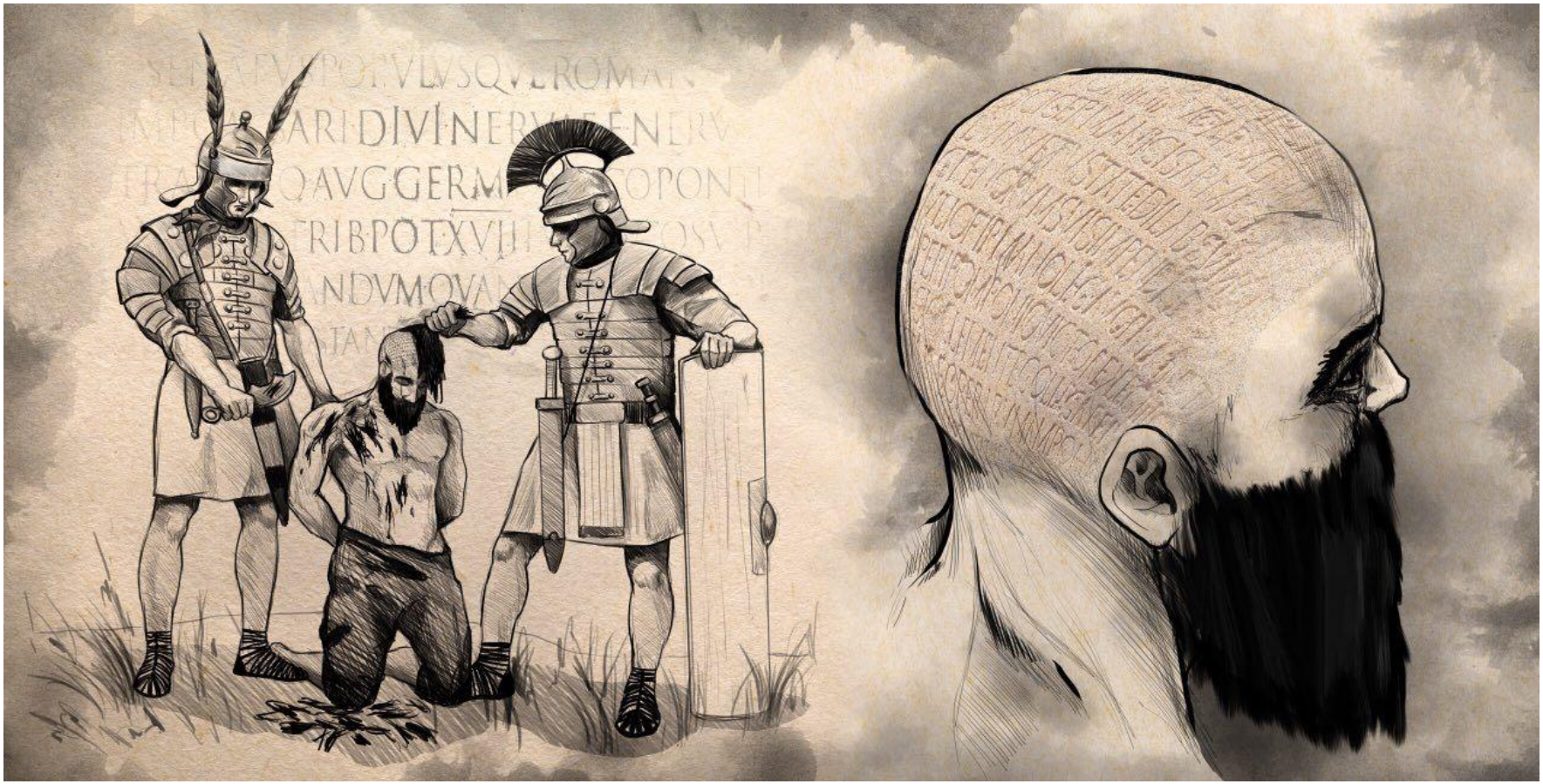
Disposition of Period and Assessment

- **Class hour：(48 hours)**
- **Assessment： (20 points)**
  - **process assessment（40%）**
    - **Class attendance 5% (only 1 session absence)**
    - **Oral questions 10%**
    - **Homework assignment 5%+X**
    - **You should choose a topic in information security area and get my approve (20%-X, (Homework + Presentation=25%)):**
      - ✓**A presentation for 15 minutes as a recorded file or voiced. Try to innovate in this field.**
      - ✓**A report at least 10 page size: 12 pt, line spacing 1**
  - **(final + midterm) exam 65%, 5% extra points**
  - **Students will be fired with more than 6 absences.**
  - **Assistant = 5%**
- I can't verify your problems.
- Rules are the same for all students.

Cryptography

"…the art of writing or solving codes…"

- Shave the head, tattoo a secret message

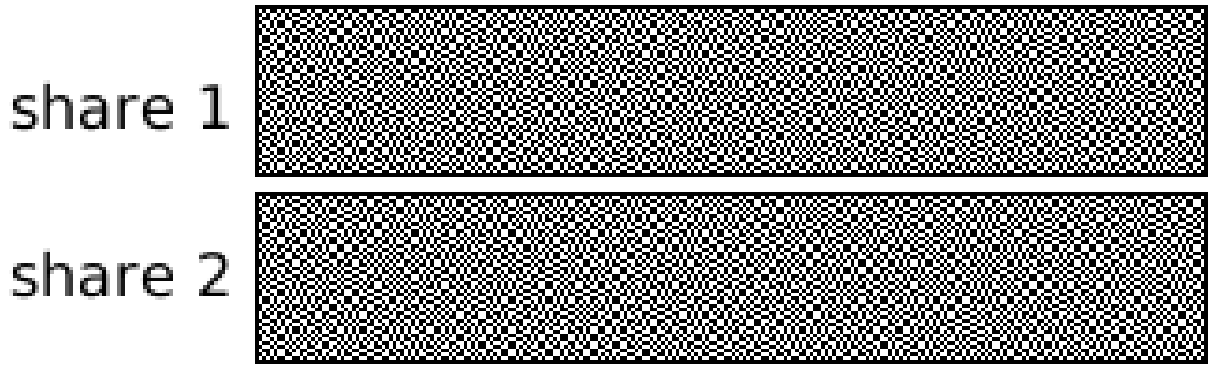


Modern cryptography

- 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, privacy-preserving ML, blockchain, cryptocurrency)

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

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
  - …

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
- Illustrates why things are more difficult than they may appear
- Motivates a more harsh (rigorous) approach

Classical cryptography

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

- I.e., *encryption*

Classical cryptography

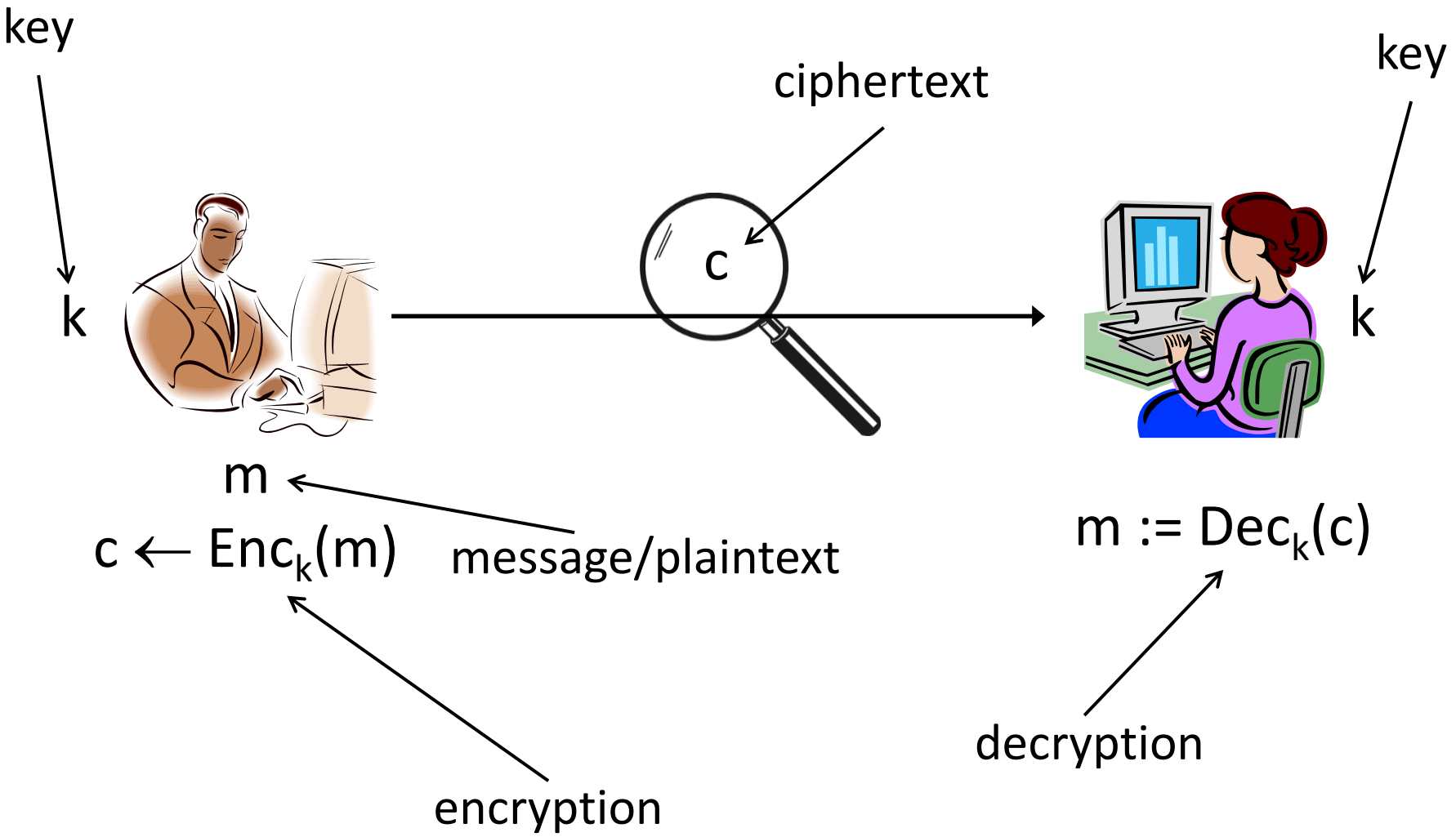
- Until the 1970s, relied exclusively on secret information (a *key*) shared in advance between the communicating parties

Private-key cryptography

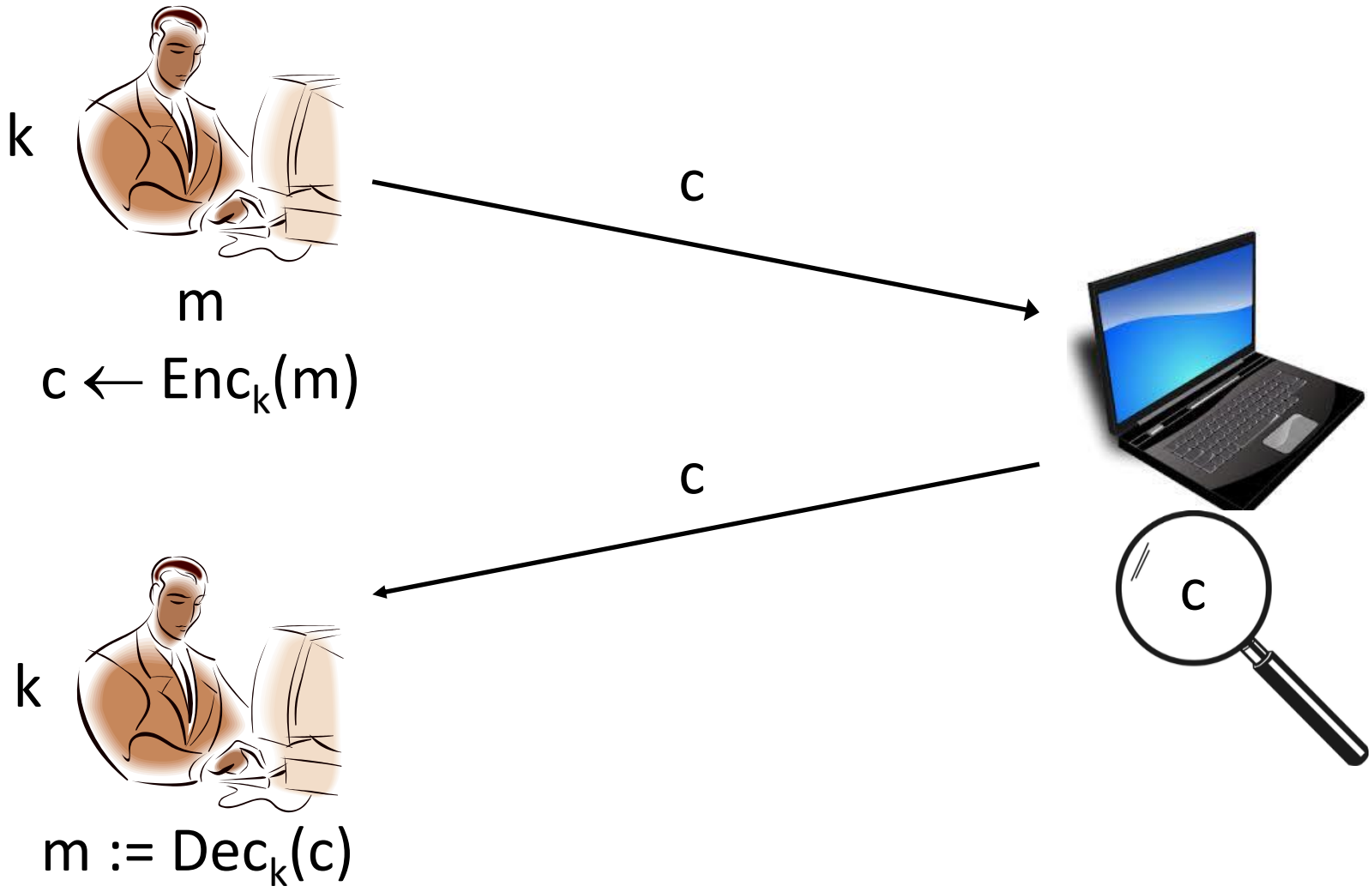
  - aka secret-key / shared-key / symmetric-key cryptography

Authentication and Key Agreement (**AKA**)

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

    $c \leftarrow Enc_k(m)$
  - 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,
$Dec_k(Enc_k(m)) = m$

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 (compatibility between different users)
    - Public scrutiny (examining look)

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

- 25 = 35 mod 10
- 25 ≠ [35 mod 10]
- 5 = [35 mod 10]

The shift cipher, formally

- $\mathcal{M}$ = {strings over lowercase English alphabet}
- Gen: choose uniform $k \in \{0, \ldots, 25\}$
- $Enc_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where
  $c_i := [m_i + k \bmod 26]$
- $Dec_k(c_1 \ldots c_t)$: 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
  - ery
  -
  -
- Ex ive-
se

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

Hexadecimal (base 16)

Hex	Bits ("nibble")	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7

Hex	Bits ("nibble")	Decimal
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

Hexadecimal (base 16)

- 0x10
  - 0x10 = 16*1 + 0 = 16
  - 0x10 = 0001 0000
- 0xAF
  - 0xAF = 16*A + F = 16*10 + 15 = 175
  - 0xAF = 1010 1111

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/

Useful observations

- Only 128 valid ASCII chars (128 bytes invalid)
- Only 0x20-0x7E printable
- 0x41-0x7a includes all upper/lowercase letters
  - Uppercase letters begin with 0x4 or 0x5
  - Lowercase letters begin with 0x6 or 0x7