Admin and The Shift Cipher

lecture 01

Welcome!

- Crypto is amazing
  - Can do things that initially seem impossible
- Crypto is important
  - It impacts us every day
- Crypto is fun!
  - Deep theory
  - Attackers' mindset

Necessary administrative stuff

- Course webpage:
  https://www.cs.umd.edu/users/gasarch/COURSES/456/F18/index.html
  - Prerequisites/information posted there
  - Syllabus posted there
  - HWs posted there
  - Announcements posted there
  - Midterm already scheduled- Oct 29 in class.

Necessary administrative stuff

- Canvas/ELMS or Gradescope (still working that out)
  - Used only to submit homework electronically-Must be Typed
  - Let me know if unable to access
- Piazza
  - Useful for discussions/questions
  - Preferable to email if you think others will have the same question

TAs

- Nathan Grammel
- Jeremy Klein
- Dan McVicker
- Jacob Prinz
- Jake Yamada

What You Need For This Class

- Mathematical prerequisites
  - Discrete math, probability, modular arithmetic
- Requires mathematical maturity
  - Proofs, abstraction

What You Need For This Class

- CS prerequisites
  - Binary, hex, pseudocode, algorithms, big-O notation
- Programming assignments
  - Hard part should not be the programming, but the thought behind it
  - Flexibility in choice of language

How to Get the Most Out of This Class

1. Read textbook and/or slides before class
   Note: On Slide Website it says on some line
   WHAT IS BELOW IS STILL A WORK IN PROGRESS.
   Should not read slides that are below that line.

2. Ask questions on Piazza and/or bring questions to class

3. This course will be taped so can catch up or review. Caution:

   3.1 If cut class and DO watch videos in sync, fine.
   3.2 If cut class and INTEND to watch videos insync, not fine.

HWs/exams

- HWs most weeks.
- Due Monday on before class begins.
- Sick Cat Policy: Can post Wed before class without penalty
- WARNING: YOU have already been given an extension, HW solutions will be posted on Wed, so NO extensions past that.
- We will keep track of your lateness NOT for grade, but for letters.
- In-class midterm and final

Textbook

Required textbook: "Introduction to Modern Cryptography, 2nd Edition," Katz and Lindell

Can buy on Amazon used.

Don't tell Katz I said so.

Laptops/electronics

- No laptops/electronics policy
  - Distracting to you
  - Distracting to others
- If you feel you need an exception, talk to me

How to contact Prof or TAs

- Prof email: gasarch@cs.umd.edu
- Please put "CMSC456" in subject line
- Prof Office hours MW 12-2, 3:30-5:00 or by Appt.
- Prof around a lot outside of office hours, feel free to drop in, but he will feel free to say *Sorry, I'm busy*.
- TA's - email and office hours on syllabus.

Course goals

- Understand real-world crypto via a rigorous approach
- When you encounter crypto in your career
  - Understand the key terms
  - Understand the security guarantees provided
  - Know how to use crypto
  - Understand what goes on "under the hood"
- "Crypto mindset"

Course non-goals

- Designing your own crypto-schemes
- Implementing your own crypto for real-world use
- Course goal:

  Realize when to consult an expert!

A Personal Note

This is a theory course much of what we do has direct application!

A Personal Note

This is a theory course much of what we do has direct application!

I do not mind that, but I am not used to that.

A Personal Note

This is a theory course much of what we do has direct application!

I do not mind that, but I am not used to that.

Last spring I taught

A Personal Note

This is a theory course much of what we do has direct application!

I do not mind that, but I am not used to that.

Last spring I taught

CMSC 452: Elementary Theory of Computation

taught what computer CAN"T do. Indirect applications.

And also

A Personal Note

This is a theory course much of what we do has direct application!

I do not mind that, but I am not used to that.

Last spring I taught

CMSC 452: Elementary Theory of Computation

taught what computer CAN"T do. Indirect applications.

And also

CMSC 858R: Ramsey Theory and its "Applications"
There were applications

A Personal Note

This is a theory course much of what we do has direct application!

I do not mind that, but I am not used to that.

Last spring I taught

CMSC 452: Elementary Theory of Computation

taught what computer CAN"T do. Indirect applications.

And also

CMSC 858R: Ramsey Theory and its "Applications"
There were applications to other parts of pure mathematics.

Classical VS Modern cryptography

Classical: (1900 BCE?–1975)

1. More of an art. Not much Mathematics.
2. WW II: They used people good at crossword puzzles.
3. Turing and others brought math into it, but not much math compared compared to Modern

Modern: (1976-today)

1. Lots of Math. Lots of Rigor.
2. The notion of Provably Secure important.

Note: The cutoff of 1975–1976 is approximate.

Rough course outline

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

Classical Cryptography

lecture 01

Motivation

- Allows us to "ease into things. . . ,"
- Shows why unprincipled approaches are dangerous
  (unprincipled means not-rigorous, not immoral)
- Illustrates why things are more difficult than they may appear

Alice, Bob, and Eve

- Alice sends a message to Bob in code.
- Eve overhears it.
- We want Eve to not be able to decode it.

This can mean one of two things:

- Eve does not have enough information to decode it. So even if Eve had unlimited computing power she could not decode.
- Assuming Eve can't Factor quickly (or some other function) then Eve cannot break the code.

The First Step in Any Cipher-Spaces

I want to encode

Cryptography is an important part of security

Spaces give away information! For example, SHIFT-BY-1 yields:

Dszuphsbqiz jt bo jnqpsubou qbsu pg tfdvsjuz

Without any fancy math Eve knows that the second and third word are two letters long. Thats information she can use!

What to do?

The First Step in Any Cipher-Blocks of Five

I want to encode

Cryptography is an important part of security

Break it up into blocks of 5:

Cryto graph yisan impor tantp artof secur ity

However you code it, spaces will not give anything away.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.
   Get rid of all punctuation.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.
   Get rid of all punctuation.
3. What to do about numbers?

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.
   Get rid of all punctuation.
3. What to do about numbers?
   Just like letters- alphabet is 36 characters

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.
   Get rid of all punctuation.
3. What to do about numbers?
   Just like letters- alphabet is 36 characters
   More generally, set your mod equal to your alphabet size.

The First Step in Any Cipher-Other Issues

I want to encode

Are my TAs for CMSC/MATH 456 awesome? YES!

1. Capital and small letters leak information.
   Map everything to Capitals.
2. Punctuation leaks information.
   Get rid of all punctuation.
3. What to do about numbers?
   Just like letters- alphabet is 36 characters
   More generally, set your mod equal to your alphabet size.

Note: In this class we will use 26-letter English only.

The Shift Cipher

lecture 01

The Shift Cipher

- Consider encrypting English text
- associate 'a' with 0; 'b' with 2; . . . ; 'z' with 25
- $k \in \mathcal{K} = \{0, \ldots, 25\}$ (or could think of $k \in \{a, \ldots, z\}$)
- To encrypt using key k, shift every letter of the plaintext by k positions (with wraparound)
- Decryption just does the reverse

```
 hello world
+22222 22222
=jgnnq yqtnf
```

Modular arithmetic

- $x \equiv y \pmod{N}$ 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 $x \equiv y \pmod{N}$.
- $25 \equiv 35 \pmod{10}$
- $25 \neq [35 \bmod 10]$
- $5 = [35 \bmod 10]$

The Shift Cipher, Formally

- $\mathcal{M} = \{\text{all texts in lowercase English alphabet}\}$
  All arithmetic mod 26.
- Choose uniform $k \in \{0, \ldots, 25\}$
- Encode $(m_1 \ldots m_t)$ as $(m_1 + k, \ldots m_t + k)$
- Decode $(c_1 \ldots c_t)$ as $(c_1 - k, \ldots c_t - k)$
- Can verify that correctness holds.

Is the Shift Cipher Secure?

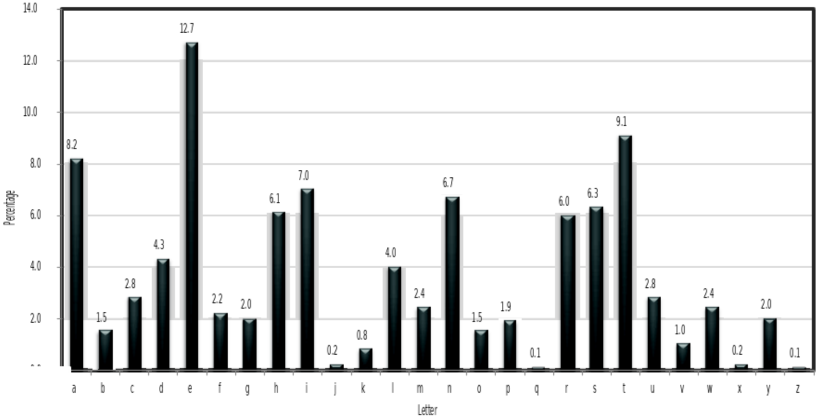
- No – only 26 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - Only one possibility will "make sense"
- Example of a "brute-force" or "exhaustive-search" attack

Example

- Ciphertext `uryyb jbeyq`
- Try every possible key...
  - tqxxa iadxp
  - spwwz hzcwo
  - ...
  - hello world

Question: We can tell that hello world is correct but how can a computer do that. Can we mechanize the process of picking out the right one?

Letter Frequencies



Use Letter Freqs to Test "Looks Like English"

Let T be a long text of normal English.
Let $\vec{f}$ be the freq vector of English. The components are all between 0 and 1 and add up to 1.
We assume freq vector of T is approx $\vec{f}$.

- One can compute that

$$\vec{f} \cdot \vec{f} \approx 0.065$$

- Let $s \in \{1, \ldots, 25\}$. Let T_s be the text shifted by s. Let $\vec{g}$ be the freq vector for T_s. One can compute that

$$\vec{f} \cdot \vec{g} \leq\approx 0.038$$

Is English

We describe a way to tell if a text Is English that we will use throughout this course.
Let $\vec{f}$ be the freq vector for English.

1. Input(T) a text
2. Compute $\vec{g}$, the freq vector for T
3. Compute $\vec{g} \cdot \vec{f}$. If ≈ 0.065 then output YES, else NO

Cracking Shift Cipher

- Given T a long text that you KNOW was coded by shift.
- For $s = 0$ to 25
  - Create T_s which is T shifted by s.
  - If Is English(T_s)=YES then output T_s and stop. Else try next value of i.

Note: No Near Misses. There will not be two values of s that are both close to 0.065.

Pedagoical Note: Would normally have written Key instead of Note but the word Key is important in crypto so I can't use it to say something is important. Oh Well.

A Note on Cracking Shift Cipher

In the last slide we tried *all* shifts in order.
Can do better:

- Given T a long text that you KNOW was coded by shift.
- Find frequencies of all letters, form vector $\vec{f}$
- Sort vector. So most common letter is σ_1, next is σ_2, etc.
- For $i = 0$ to 25
  - Create T_s which is T shifted as if σ_i maps to e.
  - Compute $\vec{g}$, the freq vector for T_s
  - Compute $\vec{g} \cdot \vec{f}$. If ≈ 0.065 then stop: T_s is your text. Else try next value of s.

Note: Quite likely to succeed in the first try, or at least very early.

What if only transmit one letter?

Odd Situation: What if message is only one letter long?
Discuss: Can Eve crack a one-letter message?

What if only transmit one letter?

Odd Situation: What if message is only one letter long?
Discuss: Can Eve crack a one-letter message?
No (We will formalize this later.)

What if only transmit one letter?

Odd Situation: What if message is only one letter long?
Discuss: Can Eve crack a one-letter message?
No (We will formalize this later.)
Discuss: Can Eve learn from two 1-letter messages?

What if only transmit one letter?

Odd Situation: What if message is only one letter long?
Discuss: Can Eve crack a one-letter message?
No (We will formalize this later.)
Discuss: Can Eve learn from two 1-letter messages?
Yes
Scenario:
In clear: Is Jacob a double agent working for the Klingons?
The answer comes via a shift cipher: A (which is either Y or N)

In clear: Is Jacob a double agent working for the Romulans?
The answer comes via a shift cipher: A (which is either Y or N)

What if only transmit one letter?

Odd Situation: What if message is only one letter long?
Discuss: Can Eve crack a one-letter message?
No (We will formalize this later.)
Discuss: Can Eve learn from two 1-letter messages?
Yes
Scenario:
In clear: Is Jacob a double agent working for the Klingons?
The answer comes via a shift cipher: A (which is either Y or N)

In clear: Is Jacob a double agent working for the Romulans?
The answer comes via a shift cipher: A (which is either Y or N)

Eve knows Jacob is working for either both or neither.

Eve Can Tell if Two Message Are Same or Not

Issue: If Eve sees two message, will know if they are the same or different.

Does this leak information: Discuss

Eve Can Tell if Two Message Are Same or Not

Issue: If Eve sees two message, will know if they are the same or different.

Does this leak information: Discuss

What to do about this? Discuss

Eve Can Tell if Two Message Are Same or Not

Issue: If Eve sees two message, will know if they are the same or different.

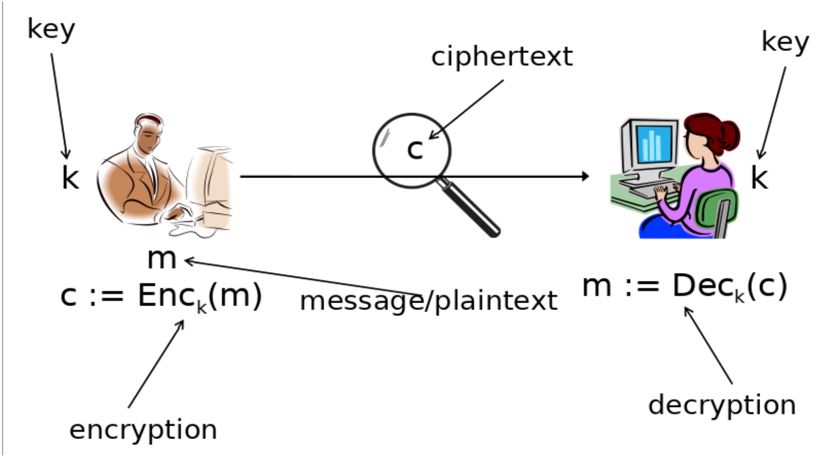
Does this leak information: Discuss

What to do about this? Discuss

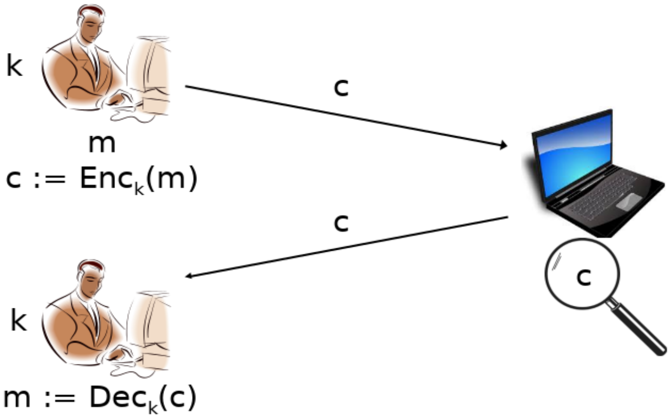
For Now Nothing Will come back to this issue after a few more ciphers.

For Now A lesson in how even definiting security and leak must be done carefully and rigorously.

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 $k \in \mathcal{K}$
    (For SHIFT this is $k \in \{0, \ldots, 25\}$. Should 0 be included?)
  - Enc (encryption algorithm): takes key k and message $m \in \mathcal{M}$ as input; outputs ciphertext c

    $$c \leftarrow Enc_k(m)$$

    (For SHIFT this is $\text{Enc}(m_1, \ldots, m_n) = (m_1 + k, \ldots, m_n + k)$.)
  - Dec (decryption algorithm): takes key k and ciphertext c as input; outputs m or "error"

    $$m := Dec_k(c)$$

    (For SHIFT this is $\text{Dec}(c_1, \ldots, c_n) = (c_1 - k, \ldots, c_n - k)$.)

    $\forall k$ output by Gen $\forall m \in \mathcal{M}, Dec_k(Enc_k(m)) = m$

(For SHIFT this is $(m + k) - k = m$)

Kerckhoffs's principle

We made the comment We KNOW that SHIFT was used. More generally we use this principle.

- *The encryption scheme* is not secret
  - Eve knows the encryption scheme
  - The only secret is the key
  - The key must be chosen at random; kept secret
- Some 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 validation

Byte-wise Shift Cipher

lecture 01

Byte-wise Shift Cipher

- Instead of $a, b, c, d, \ldots, z$ have (for example) 0000, 0001,...,1111.
- Works for an alphabet of *bytes* rather than (English, lowercase) *letters*
  - Data in a computer is stored this way anyway. So works natively for arbitrary data!
- Use XOR instead of modular addition. Fast!
- Decode and Encode are both XOR.
  - Essential properties still hold

Hexadecimal (base 16)

Hex	Bits ("nibble")	Decimal	Hex	Bits ("nibble")	Decimal
0	0000	0	8	1000	8
1	0001	1	9	1001	9
2	0010	2	A	1010	10
3	0011	3	B	1011	11
4	0100	4	C	1100	12
5	0101	5	D	1101	13
6	0110	6	E	1110	14
7	0111	7	F	1111	15

Hexadecimal (base 16)

Notation: 0x before a string of $\{0, 1, \ldots, 9, A, B, C, D, E, F\}$ means that the string will be 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 with TWO hex-digits.
- Potentially 256 characters via $\{0, \ldots, 9, A, \ldots, F\} \times \{0, \ldots, 9, A, \ldots, F\}$
- Only use 128 characters via $\{0, \ldots 8\} \times \{0, \ldots, 9, A, \ldots, F\}$

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/

ASCII

- ‘1’ $=$ 0x31 $=$ 0011 0001
- ‘F’ $=$ 0x46 $=$ 0100 0110

Useful observations

- Only 128 valid ASCII chars (128 bytes invalid)
- 0x20-0x7E printable
- 0x41-0x7A includes upper/lowercase letters
  - Uppercase letters begin with 0x4 or 0x5
  - Lowercase letters begin with 0x6 or 0x7

Byte-wise shift cipher

- $\mathcal{M}$ = {strings of bytes}
- *Gen*: choose uniform byte $k \in \mathcal{K} = \{0,\ldots,255\}$
- $Enc_k(m_1 \ldots m_t)$: output $c_1 \ldots c_t$, where $c_i := m_i \oplus k$
- $Dec_k(c_1 \ldots c_t)$: output $m_1 \ldots m_t$, where $m_i := c_i \oplus k$
- Verify that correctness holds…

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Question: Should it worry Alice and Bob that the key itself was transmitted? Discuss

Example

Key is 11001110.
Alice wants to send $00011010, 11100011, 00000000$
She sends

$$00011010 \oplus 11001110, 11100011 \oplus 11001110, 00000000 \oplus 11001110$$

$$= 11010100, 00101101, 11001110$$

Question: Should it worry Alice and Bob that the key itself was transmitted? Discuss
No. Eve has no way of knowing that.

Is this cipher secure?

- No – only 256 possible keys!
  - Given a ciphertext, try decrypting with every possible key
  - If ciphertext is long enough, only one plaintext will "look like English" (use the vector method of the last set of slides).
- Can further optimize
  - First nibble of plaintext likely 0x4, 0x5, 0x6, 0x7 (assuming letters only)
  - Can reduce exhaustive search to 26 keys (how?)
  - Talk to your friends or blood enemies about this.

Sufficient key space principle

- The key space must be large enough to make exhaustive-search attacks impractical
  - How large do you think that is?
- Note: this makes some assumptions...
  - English-language plaintext
  - Ciphertext sufficiently long so only one valid plaintext