

UNIT 11C

The Internet: Encryption and Security

3. Volvelle designed by Gabriel de Collange for Johannes Trithemius' *Polygraphie* (No. 8)

Announcement

- Lab Exam 2 Today
- Final Exam Part 1 : Friday 9am
- Final Exam Part 2 : Friday 4:30pm

Overview

- Security issues
- Encryption and cryptanalysis
- Encryption in the digital age
 - Symmetric encryption
 - Asymmetric encryption
- Applications of encryption
- Encryption is not security!

Security issues

...

Data Privacy

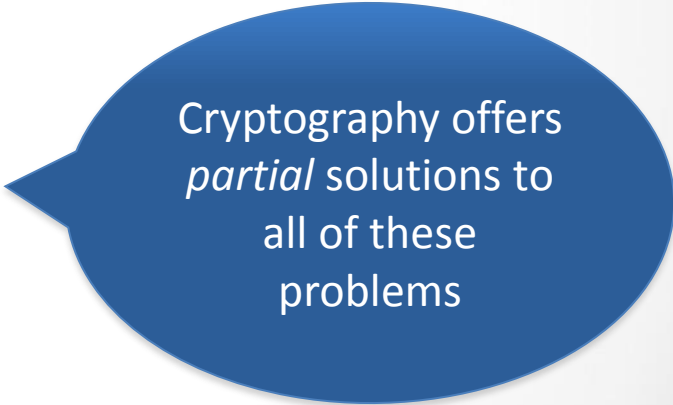
- Suppose you send an email to a friend.
 - Who has access to that email?
 - What if you encrypt the message so it is private?
 - Can someone who intercepts the email decrypt it?
- On the Internet and any computer network, any transmitted data can be intercepted and copied.

Networking is a security issue

- Why?
- If you want a really secure machine, lock it in an electromagnetically shielded room and don't connect it to any networks or other sources of data beyond your control.
- Not much fun, is it?

The Problem

- The Internet is **public**
 - Messages sent pass through many machines and media
- Anyone intercepting a message might
 - read it and/or
 - replace it with a different message
- The Internet is anonymous
 - IP addresses don't establish identity
- Anyone may send messages under a false identity



Cryptography offers *partial* solutions to all of these problems

A Shady Example

- I want to make a purchase online and click a link that takes me to

<http://www.sketchystore.com/checkout.jsp>

- What I see in my browser:

Enter your credit card number:

Enter your expiration date:

A Shady Example (cont' d)

- When I press SUBMIT, my browser sends this:

POST /purchase.jsp HTTP/1.1

Host: www.sketchystore.com

User-Agent: Mozilla/4.0

Content-Length: 48

Content-Type: application/x-www-form-urlencoded

userid=**rbd**&creditcard=**2837283726495601**&exp=**01/09**

A Shady Example (cont' d)

- If this information is sent unencrypted, who has access to my credit card number?
 - Other people who can connect to my wireless ethernet
 - Other people physically connected to my wired ethernet
 - ...
- Packets are passed from router to router.
 - *All* those routers have access to my data.

A caveat



cryptography is not security

A CRYPTO NERD'S
IMAGINATION:

HIS LAPTOP'S ENCRYPTED.
LET'S BUILD A MILLION-DOLLAR
CLUSTER TO CRACK IT.



WHAT WOULD
ACTUALLY HAPPEN:

HIS LAPTOP'S ENCRYPTED.
DRUG HIM AND HIT HIM WITH
THIS \$5 WRENCH UNTIL
HE TELLS US THE PASSWORD.



Encryption and cryptanalysis

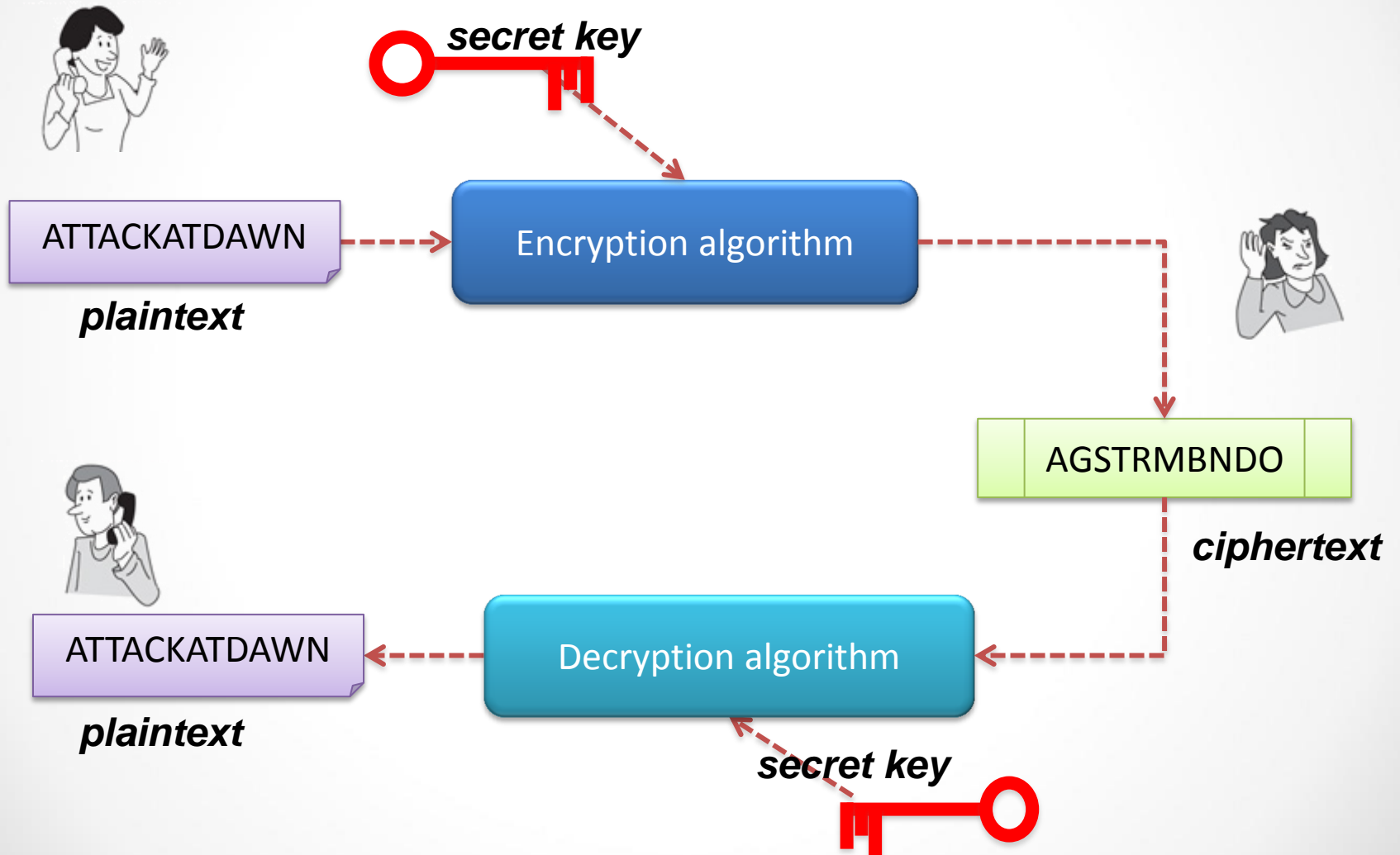


basic concepts

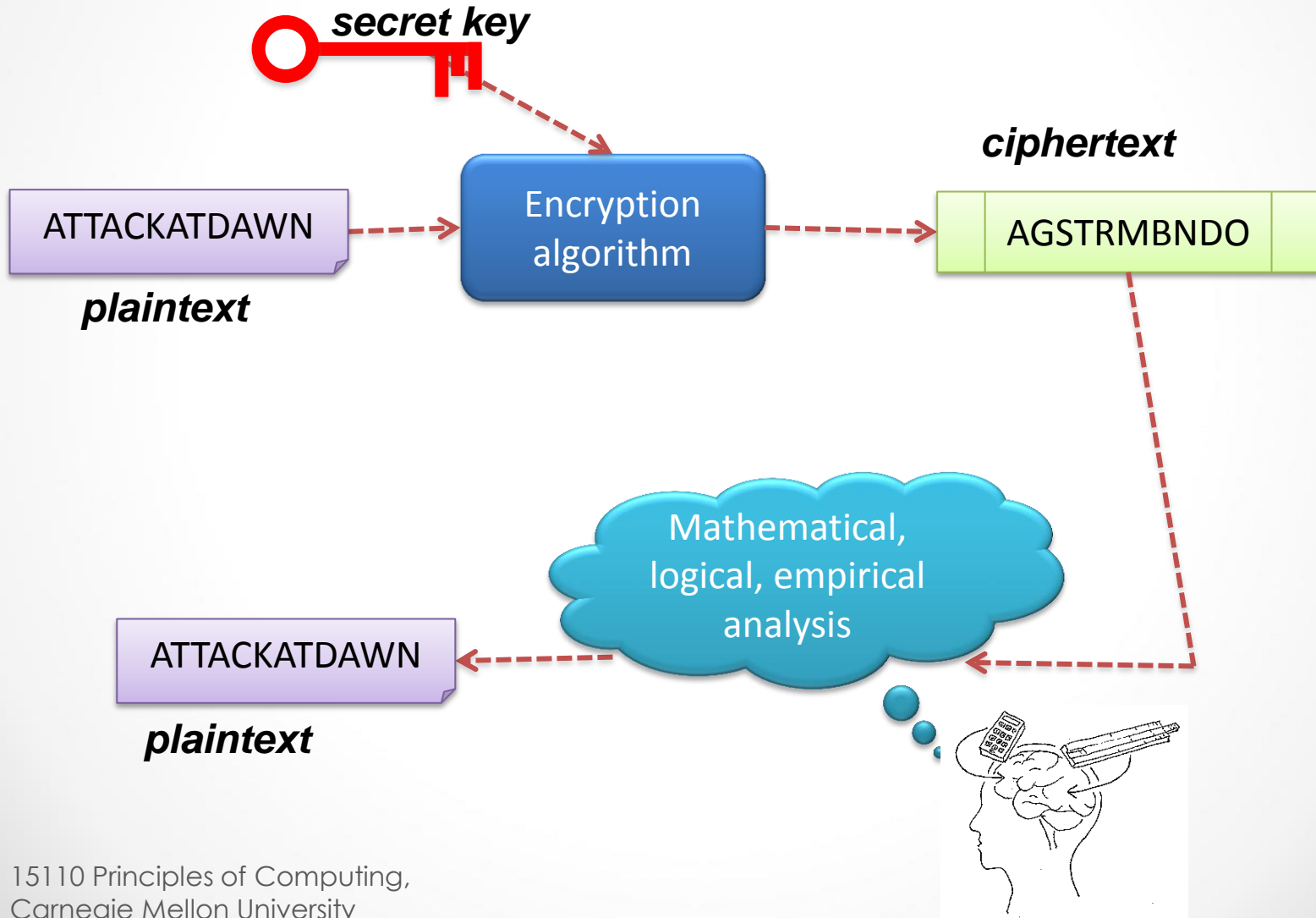
Encryption

- We encrypt (encode) our data so others can't understand it (easily) except for the person who is supposed to receive it.
- We call the data to encode **plaintext** and the encoded data the **ciphertext**.
- **Encoding** and **decoding** are *inverse functions* of each other.

Encryption/Decryption



Cryptanalysis



Encryption techniques

• • •

substitution and transposition

Two basic ways of altering text to encrypt/decrypt

- Substitute one letter for another using some kind of rule

Substitution cipher

- Scramble the order of the letters using some kind of rule

Transposition cipher

Substitution Ciphers

- Simple encryption scheme using a substitution cipher:

Shift every letter forward by 1:

$A \rightarrow B, B \rightarrow C, \dots, Z \rightarrow A$

MESSAGE \rightarrow NFTTBHF

- Can you decrypt **TFDSFU**?

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Substitution Ciphers

- Simple encryption scheme using a substitution cipher:

Shift every letter forward by 1:

$A \rightarrow B, B \rightarrow C, \dots, Z \rightarrow A$

MESSAGE \rightarrow NFTTBHF

- Can you decrypt **TFDSFU**? **SECRET**

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Caesar Cipher

- Shift forward n letters; n is the *secret key*.

For example, shift forward 3 letters
(This is a Caesar cipher using a **key** of 3):

A → D, B → E, ..., Z → C

MESSAGE → PHVVDJH

ABCDEFGHIJKLMNOPQRSTUVWXYZ


- How can we crack this encrypted message if we don't know the key?

DEEDUSEKBTFEIIRBOTUSETUJXYI

Caesar Cipher (cont' d)

DEEDUSEKBTFEIIYRBOTUSETUJXYI
EFFEVTFLCUGFJJZSCPUVTFUVKYZJ
FGGFWUGMDVHGKKATDQVWUGVWLZAK
GHHGXVHNEWIHLLBUERWXVHWMABL
HIIHYWIOFXJIMMCVFSXYWIXYNBCM
IJJIZXJPGYKJNNDWGTYZXJYZOCDN
JKKJAYKQHZLK00EXHUZAYKZAPDEO
KLLKBZLRIAMLPPFYIVABZLABQEFP
LMMLCAMSJBNMQQGZJWBCAMBCRFGQ
MNNMDBNTKCONRRHAKXCDBNCDGHR
NOONECOULDPOSSIBLYDECODETHIS
OPPOFDPVMEQPTTJCMZEFDPEFUIJT
PQQPGEQWNFRQUUKDNAFGEQFGVJKU

QRRQHFRX0GSRVVLE0BGHFRGHWKLV
RSSRIGSYPHTSWWMFPCHIGSHIXLMW
STTSJHTZQIUTXXNGQDIJHTIJYMNX
TUUTKIUARJVUYYOHREJKIUJKZNOY
UVVULJVBSKWVZZPISFKLJVKLAOPZ
VWVMKWCTLXWAAQJTGLMKWLMBPQA
WXXWNLXDUMYXBBRKHUHMNLXMNCQRB
YXXXOMYEVNZYCCSLVINOMYNODRSC
YZZYPNZFWOAZDDTMWJOPNZOPESTD
ZAAZQOAGXPBAEEUNXKPQOAPQFTUE
ABBARPBYQCBFFVOYLQRPBQRGUVF
BCCBSQCIZRDCGGWPZMRSQCRSHVWG
CDDCTRDJASEDHXQANSTRDSTIWXH

- How long would it take a computer to try all 25 shifts?

Vigenère Cipher

- **Shift different amount for each letter.** Use a *key word*; each letter in the key determines how many shifts we do for the corresponding letter in the message.
- Example: key word “cmu”: shift by 2, 12, 20
- Message “pittsburgh”
 cmucmucmuc
encrypted: runvevwda j
- Try it yourself at
http://www.simonsingh.net/The_Black_Chamber/v_square.html

ABCDEFGHIJKLMNOPQRSTUVWXYZ

| | | |
|----------|------------------------------------|------------|
| A | ABCDEFGHIJKLMNOPQRSTUVWXYZ | no shift |
| B | BCDEFGHIJKLMNOPQRSTUVWXYZA | shift by 1 |
| C | CDEFGHIJKLMNOPQRSTUVWXYZAB | shift by 2 |
| D | D EFGHIJKLMNOPQRSTUVWXYZABC | shift by 3 |
| E | EFGHIJKLMNOPQRSTUVWXYZABCD | etc. |
| F | FGHIJKLMNOPQRSTUVWXYZABCDE | |

...

- Message:

ATTACKATDAWN

- Pick a secret key

DECAFDECAFDE

- Encrypted:

D

1st letter in the message is shifted by 3, 2nd letter is shifted by 4, ...

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A ABCDEFGHIJKLMNOPQRSTUVWXYZ

B BCDEFGHIJKLMNOPQRSTUVWXYZA

C CDEFGHIJKLMNOPQRSTUVWXYZAB

D DEFGHIJKLMNOPQRSTUVWXYZABC

E EFGHIJKLMNOPQRSTUVWXYZ**X**YZABCD

F FGHIJKLMNOPQRSTUVWXYZABCDE

...

- Message:
- Pick a secret key
- Encrypted:

ATTACKATDAWN

DECAFDECAFDE

DX

1st letter in the message is shifted by 3, 2nd letter is shifted by 4, ...

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A ABCDEFGHIJKLMNOPQRSTUVWXYZ

B BCDEFGHIJKLMNOPQRSTUVWXYZA

C CDEFGHIJKLMNOPQRSTUVWXYZ**A**B

D DEFGHIJKLMNOPQRSTUVWXYZABC

E EFGHIJKLMNOPQRSTUVWXYZABCD

F FGHIJKLMNOPQRSTUVWXYZABCDE

...

- Message:
- Pick a secret key
- Encrypted:

ATTACKATDAWN

DECAFDECAFDE

DXV

1st letter in the message is shifted by 3, 2nd letter is shifted by 4, ...

ABCDEFGHIJKLMNOPQRSTUVWXYZ

A | ABCDEFGHIJKLMNOPQRSTUVWXYZ

B | BCDEFGHIJKLMNOPQRSTUVWXYZA

C | CDEFGHIJKLMNOPQRSTUVWXYZAB

D | DEFGHIJKLMNOPQRSTUVWXYZABC

E | EFGHIJKLMNOPQRSTUVWXYZABCD

F | FGHIJKLMNOPQRSTUVWXYZABCDE

...

- Message: **ATTACKATDAWN**
- Pick a secret key **DECAFDECAFDE**
- Encrypted: **DXVAHNEVDFZR**

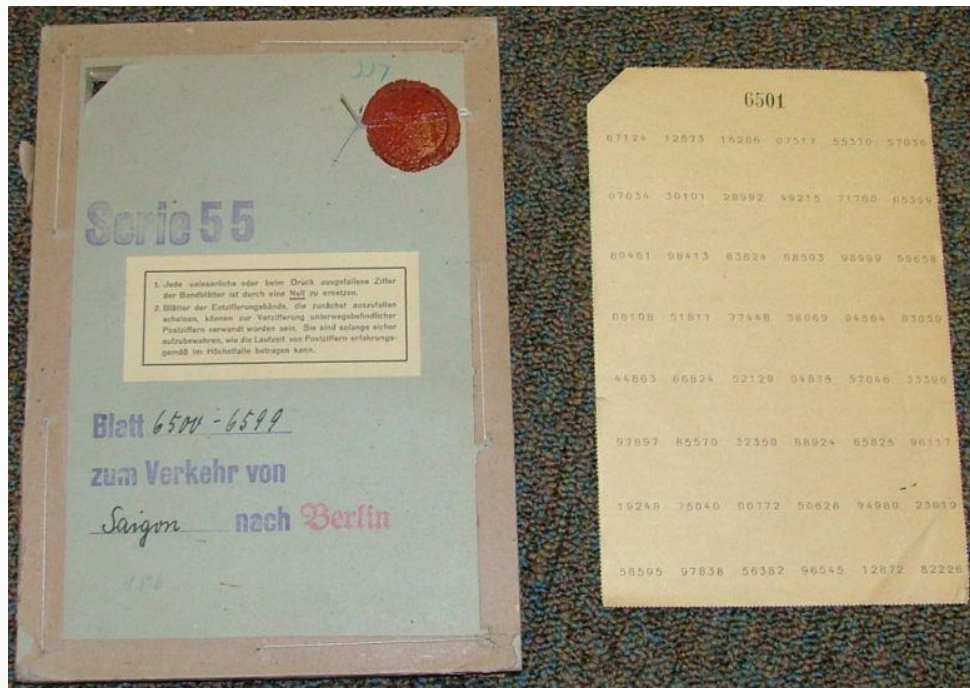
1st letter in the message is shifted by 3, 2nd letter is shifted by 4, ...

Vernam Cipher

- Vigenère cipher was broken by Charles Babbage in the mid 1800s by exploiting the repeated key
 - The length of the key determines the cycle in which the cipher is repeated.
- Vernam cipher:
Make the key the same length as the message; Babbage's analysis doesn't work.

One-time Pads

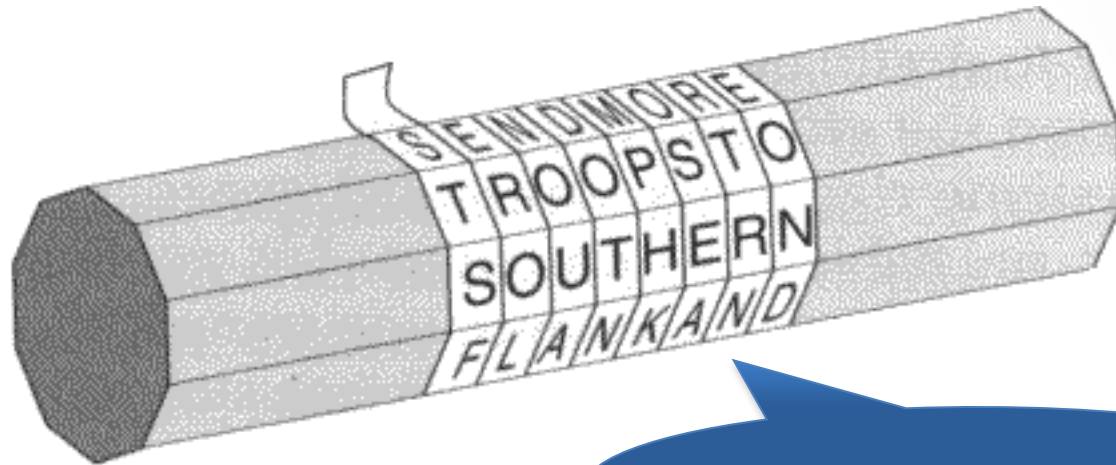
- Vernam cipher is commonly referred to as a one-time pad.



Alice and Bob have identical “pads” (shared keys)

- If random keys are used one-time pads are unbreakable in theory.

Transposition ciphers



an ancient Greek method

STSF...EROL...NOUA...DOTN...MPHK...OSEA...RTRN...EOND...

image:<http://crypto.interactive-maths.com/simple-transposition-ciphers.html>

Encryption in computing

...

fast computation makes encryption usable by all of us

Encryption in computing

- **One-time pads impractical** on the net (why?)
- Basic assumption:
 - the encryption/decryption *algorithm* is known;
 - only the **key is secret**
- Very complicated encryptions can be computed fast:
 - typically, elaborate combinations of substitution and transposition

HTTPS

- Security protocol for the Web, the peoples' encryption
- Purpose:
 - confidentiality (prevent eavesdropping)
 - message integrity and authentication (prevent “man in the middle” attacks that could alter the messages being sent)
- Techniques:
 - asymmetric encryption (“public key” encryption) to exchange secret key
 - certificate authority to obtain public keys
 - symmetric encryption to exchange actual messages

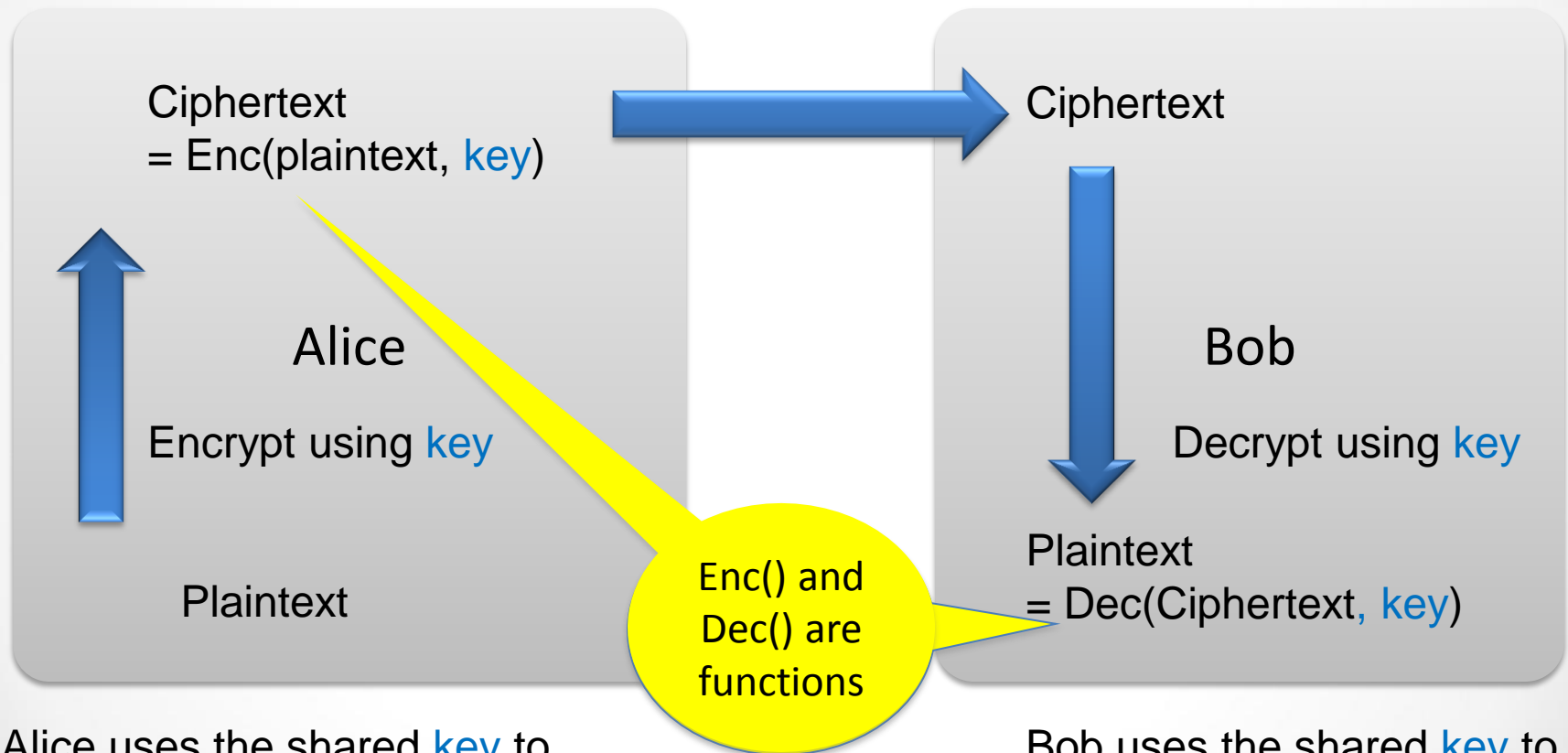
Symmetric vs. Asymmetric encryption

- **Symmetric** (shared-key) encryption: commonly used for long messages
 - Often a complicated mix of substitution and transposition encipherment
 - Reasonably fast to compute
 - Requires a shared secret key usually communicated using (slower) *asymmetric encryption*
- **Asymmetric** encryption: different keys are used to encrypt and to decrypt

Keyspace

- *Keyspace* is jargon for the number of possible secret keys, for a particular encryption/decryption algorithm
- Number of bits per key determines *size of keyspace*
 - important because we want to make *brute force attacks* infeasible
 - brute force attack: run the (known) decryption algorithm repeatedly with **every possible key** until a sensible plaintext appears
- Typical key sizes: several hundred bits

Symmetric (Shared Key) Encryption



Alice uses the shared **key** to encrypt the plaintext to produce the ciphertext

Bob uses the shared **key** to decrypt the ciphertext to recover the plaintext

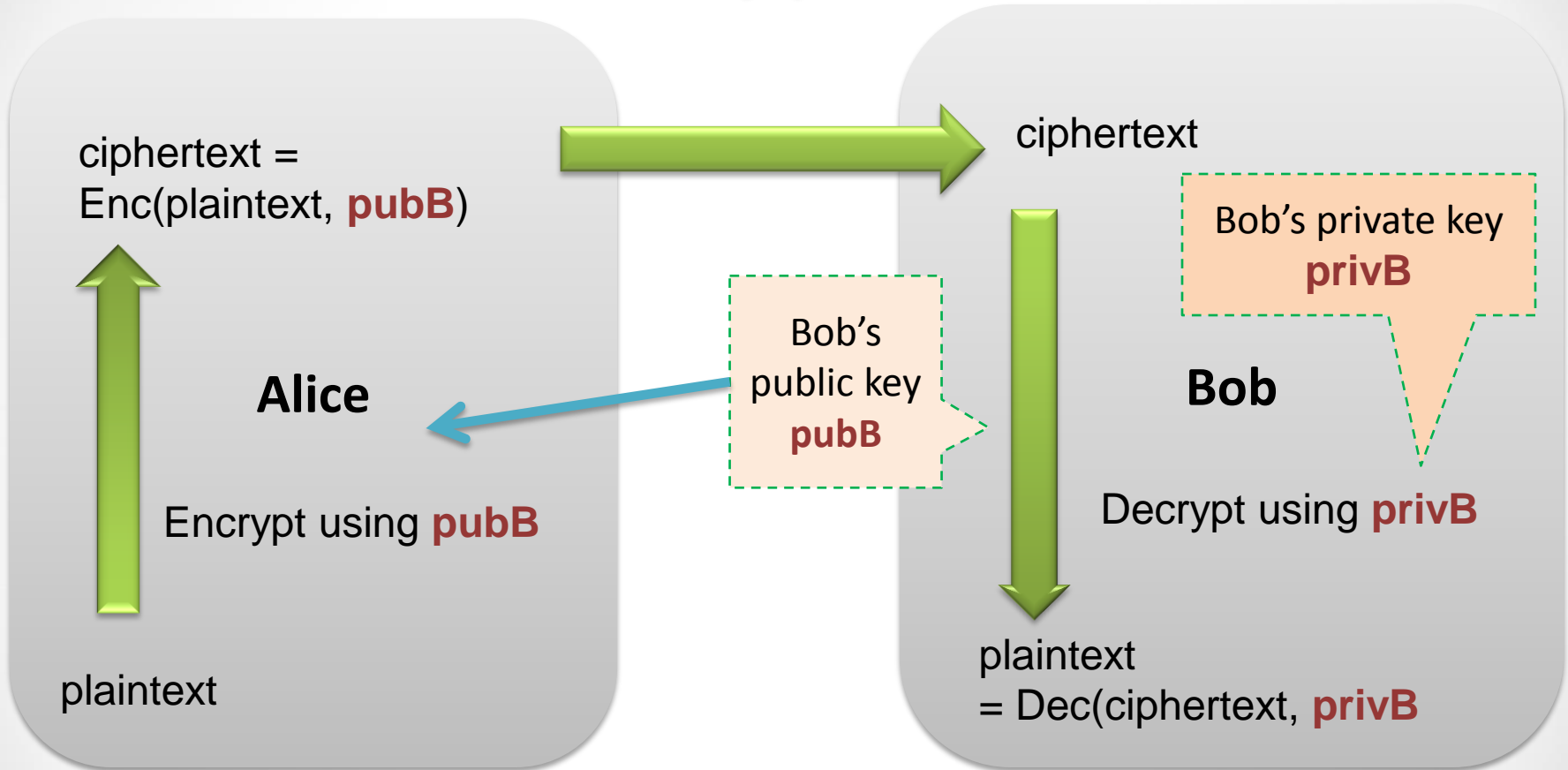
Establishing Shared Keys

- Problem: how can Alice and Bob secretly agree on a key, using a public communication system?
- Solution: asymmetric encryption based on *number theory*
 - Alice has one secret, Bob has a different secret; working together they establish a shared secret
 - Examples: Diffie-Hellman key exchange, RSA public key encryption

One type of asymmetric encryption: RSA

- Common encryption technique for transmitting symmetric keys on the Internet (https, ssl/tls)
 - Named after its inventors: Rivest, Shamir and Adleman
 - Used in https (you know when you're using it because you see the URL in the address bar begins with http**s**://)

Asymmetric Public Key Encryption



Alice uses Bob's **public key** to **encrypt** the plaintext to produce the ciphertext

Bob uses his **private key** to **decrypt** the ciphertext to recover the plaintext

How RSA works

- First, we must be able to represent any message as a single number (it may already be a number as is usual for a symmetric key)

- For example:

A T T A C K A T D A W N
012020010311012004012314

Public and Private Keys

used for encryption

- Every receiver has a **public key** (e, n) and a **private key** (d, n) .

used for decryption

- The transmitter encrypts a (numerical) message M into ciphertext C using the receiver's public key:

$$M^e \text{ modulo } n \rightarrow C \text{ (ciphertext)}$$

- The receiver decodes the encrypted message C to get the original message M using the private key (which no one else knows).

$$C^d \text{ modulo } n \rightarrow M \text{ (plaintext)}$$

RSA Example

- Alice's Public Key: $(3, 33)$ $(e = 3, n = 33)$
- Alice's Private Key: $(7, 33)$ $(d = 7, n = 33)$
 - Usually these are really huge numbers with many hundreds of digits!
- Bob wants to send the message **4**
 - Bob encrypts the message using e and n :
 $4^3 \text{ modulo } 33 \rightarrow 31$... Bob sends **31**
- Alice receives the encoded message **31**
 - Alice decrypts the message using d and n :
 $31^7 \text{ modulo } 33 \rightarrow 4$

Generating n , e and d

- p and q are (big) random primes. $p = 3, q = 11$
- $n = p \times q$ $n = 3 \times 11 = 33$
- $\varphi = (p - 1)(q - 1)$ $\varphi = 2 \times 10 = 20$
- e is small and relatively prime to φ $e = 3$
- d , such that:
 $e \times d \bmod \varphi = 1$ $3 \times d \bmod 20 = 1$
 $d = 7$

Usually the primes are huge numbers--hundreds of digits long.

Cracking RSA

- Everyone knows (e, n) . Only Alice knows d .
- If we know e and n , can we figure out d ?
 - If so, we can read secret messages to Alice.
- We **can** determine d from e and n .
 - Factor n into p and q .
$$n = p \times q$$
$$\varphi = (p - 1)(q - 1)$$
$$e \times d = 1 \pmod{\varphi}$$
 - We know e (which is public), so we can solve for d .
- But **only** if we can factor n

RSA is safe (for now)

- Suppose someone can factor my 5-digit n in 1 ms,
- At this rate, to factor a 10-digit number would take 2 minutes.
- ... to factor a 15-digit number would take 4 months.
- ... 20-digit number ... 30,000 years.
- ... 25-digit number... 3 billion years.
- We're safe with RSA! (at least, from factoring with digital computers)

Certificate Authorities

- **How do we know we have the right public key for someone?**
- *Certificate Authorities* sign digital certificates indicating authenticity of a sender who they have checked out in the real world.
- Senders provide copies of their certificates along with their message or software.
- But can we trust the certificate authorities? (only some)

Encryption is not security!

• • •

It's just a set of techniques

How (in)secure is the Internet?

- The NSA has a budget of \$11B
- Corporations and criminals also spy on us
- What can go wrong?
 - Insecure pseudo-random number generators
 - Untrustworthy certificate authorities
 - Malware
 - “Social engineering” attacks like phishing
 - Deliberately built-in insecurity in crypto products
 - Physical tapping of Internet routers

Security is an unsolved problem

Your cyber systems continue to function and serve you not due to the expertise of your security staff but solely due to the sufferance of your opponents.

– former NSA Information Assurance Director Brian Snow
(quoted by Bruce Schneier,

https://www.schneier.com/blog/archives/2013/03/phishing_has_go.html)

Summary

- Cryptography is cool mathematics and protocol design
- But cryptography is not security, only a set of techniques
- Security is a broader issue involving
 - Other technology
 - Social and legal factors

“Only amateurs attack machines; professionals target people”

Bruce Schneier