Introduction to Cryptography

What is Cryptography?

- Literally: “hidden writing”
- (Technical, the entire field is “cryptology”, but almost no one except the NSA and historians uses that word.)
- Today, we do much more with cryptography than just encryption, i.e., hiding a message’s contents

History of Cryptography

- Arguably, goes back 4,000 years
  - Some say there is quasi-cryptography in some Egyptian hieroglyphics and in the Old Testament
  - There are Mesopotamian cuneiform inscriptions with substitutions
  - The 5th century BCE Spartans used the *skytale*: wrapping papyrus around the shaft of a spear but writing across
  - An Indian treatise from the 4th century BCE speaks of cryptanalysis
- Julius Caesar

The Caesar Cipher

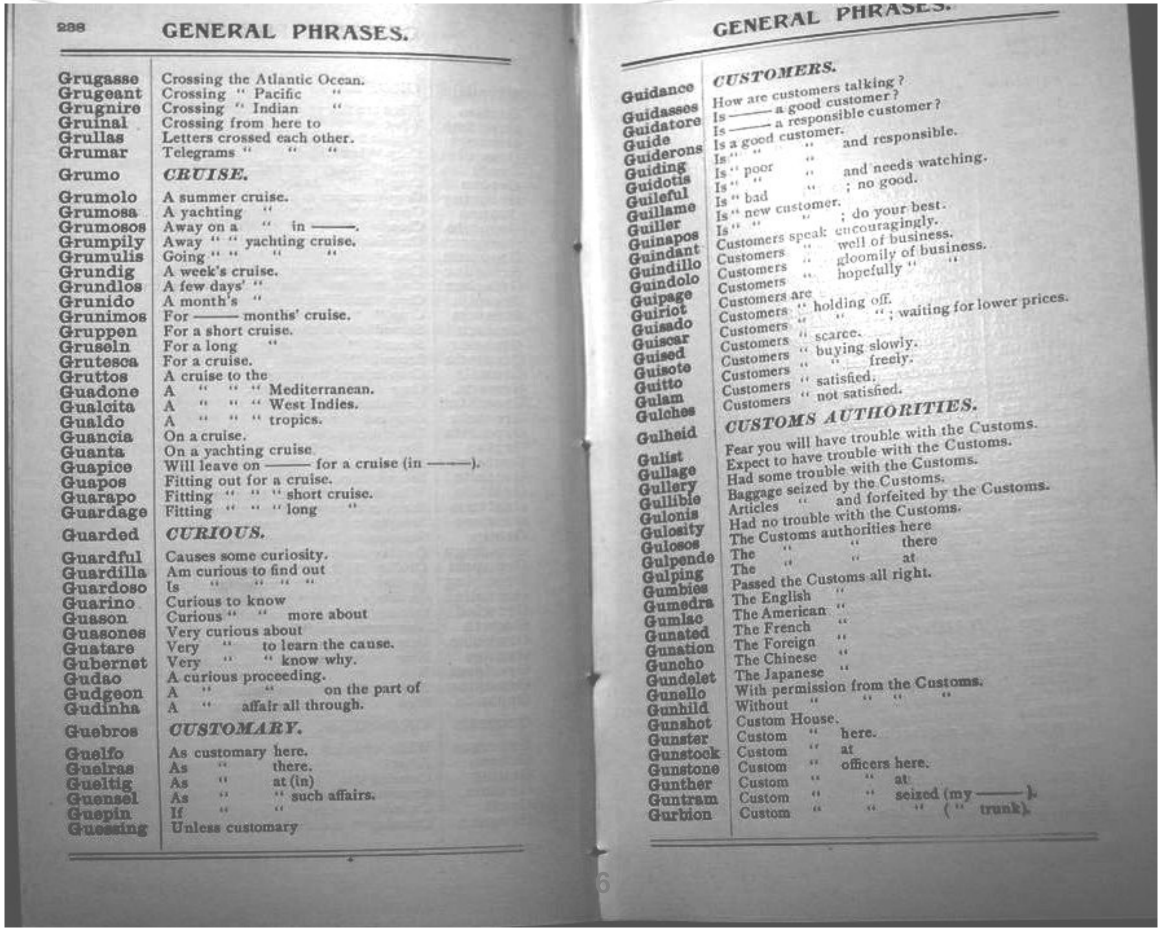
$$\begin{tikzcd}
A \arrow[d] & B \arrow[d] & C \arrow[d] & D \arrow[d] & \ldots & W \arrow[d] & X \arrow[d] & Y \arrow[d] & Z \arrow[d] \\
D & E & F & G & \ldots & Z & A & B & C
\end{tikzcd}$$

- Replace each letter with one three later in the alphabet
- The offset need not be 3
- 3 is the *key*

Codes and Ciphers

- Roughly speaking, ciphers operate at the *syntactic* level: letters, groups of letters, or bits
- Codes operate at the *semantic* level: words, phrases, sentences
- Codes have more or less died out; they're not as secure and they're too hard to automate—and you need automation to encrypt today's traffic volumes

An Old Commercial Code



288 GENERAL PHRASES.

Grugasse	Crossing the Atlantic Ocean.
Grugeant	Crossing " Pacific "
Grugnire	Crossing " Indian "
Gruinal	Crossing from here to
Grullas	Letters crossed each other.
Grumar	Telegrams " " "
Grumo	*CRUISE.*
Grumolo	A summer cruise.
Grumosa	A yachting "
Grumosos	Away on a " in ———.
Grumpily	Away " " yachting cruise.
Grumulis	Going " " " "
Grundig	A week's cruise.
Grundlos	A few days' "
Grunido	A month's "
Grunimos	For ——— months' cruise.
Gruppen	For a short cruise.
Gruseln	For a long "
Grutesca	For a cruise.
Gruttos	A cruise to the
Guadone	A " " " Mediterranean.
Gualcita	A " " " West Indies.
Gualdo	A " " " tropics.
Guancia	On a cruise.
Guanta	On a yachting cruise
Guapice	Will leave on ——— for a cruise (in ———).
Guapos	Fitting out for a cruise.
Guarapo	Fitting " " " short cruise.
Guardage	Fitting " " " long "
Guarded	*CURIOUS.*
Guardful	Causes some curiosity.
Guardilla	Am curious to find out
Guardoso	Is " " " "
Guarino	Curious to know
Guason	Curious " " more about
Guasones	Very curious about
Guatare	Very " to learn the cause.
Gubernet	Very " " know why.
Gudao	A curious proceeding.
Gudgeon	A " " on the part of
Gudinha	A " affair all through.
Guebros	*CUSTOMARY.*
Guelfo	As customary here.
Guelras	As " there.
Gueltig	As " at (in)
Guensel	As " " such affairs.
Guepin	If " "
Guessing	Unless customary

GENERAL PHRASES.

Guidance	*CUSTOMERS.*
Guidasses	How are customers talking?
Guidatore	Is ——— a good customer?
Guide	Is ——— a responsible customer?
Guiderons	Is a good customer.
Guiding	Is " " " and responsible.
Guidotis	Is " poor "
Guileful	Is " " " and needs watching.
Guillame	Is " bad " ; no good.
Guiller	Is " new customer.
Guinapos	Is " " " ; do your best.
Guindant	Customers speak encouragingly.
Guindillo	Customers " well of business.
Guindolo	Customers " gloomily of business.
Guipage	Customers " hopefully " "
Guiriot	Customers are
Guisado	Customers " holding off.
Guiscar	Customers " " " ; waiting for lower prices.
Guised	Customers " scarce.
Guisote	Customers " buying slowly.
Guitto	Customers " " freely.
Gulam	Customers " satisfied.
Gulches	Customers " not satisfied.
Gulheid	*CUSTOMS AUTHORITIES.*
Gulist	Fear you will have trouble with the Customs.
Gullage	Expect to have trouble with the Customs.
Gullery	Had some trouble with the Customs.
Gullible	Baggage seized by the Customs.
Gulonis	Articles " and forfeited by the Customs.
Gulosity	Had no trouble with the Customs.
Gulosos	The Customs authorities here
Gulpende	The " " there
Gulping	The " " at
Gumbies	Passed the Customs all right.
Gumedra	The English "
Gumlac	The American "
Gunated	The French "
Gunation	The Foreign "
Guncho	The Chinese "
Gundelet	The Japanese "
Gunello	With permission from the Customs.
Gunhild	Without " " " "
Gunshot	Custom House.
Gunster	Custom " here.
Gunstock	Custom " at
Gunstone	Custom " officers here.
Gunther	Custom " " at
Guntram	Custom " " seized (my ———).
Gurbion	Custom " " " (" trunk).

The Renaissance

- Cryptography, especially codes, was heavily used for diplomatic correspondence in Europe
- Many countries had "Black Chambers"—organizations that would unseal letters, copy and cryptanalyze the contents, and close them again with forged seals
- (King Philip II of Spain complained to the pope that Henry IV of France must be using black magic to read his messages. The pope's own cryptanalysts were also reading them…)

Cryptanalysis

- As noted, evidence for cryptanalysis >2,300 years ago in India
- The first written description of a systematic approach to cryptanalysis is by Al-Kindi in Baghdad in the 9th century CE
- Sustained use in practice in Europe during the Renaissance
- No systematic improvements until Kasiski (Prussia) in 1863
- But the real birth of modern systems—and modern attacks—came with World War I

Why World War I?

- Large volume of communications
  - Hand encryption systems couldn't really cope with the volume
  - Much of the communication was by radio, making it easy to intercept
- Technological improvements (and technological mindsets)

Three Major Themes

- Mechanization of encryption
    - The famous Enigma machine (and other, lesser-known but equally important machines) was developed shortly after the war
- Cryptanalysis started to become mathematical (Friedman (US), circa 1920)
    - Developed far more by Rejewski's team (Poland) in the 1930s; their work was expanded at Bletchley Park
- The use of metadata, first as an adjunct to cryptanalysis, and then as something useful in its own right

Metadata

- Traffic analysis
  - Who speaks to whom
  - Message size
  - Message volume and timing
- Other information (including "indicators") in the message headers
  - *How you used encryption had become crucial!*

Cryptography is Very Hard

- The design of ciphers is very hard
- *Cryptographic protocol* design is very hard
- Using ciphers correctly is very hard

Historical Example: The World War II Enigma Machine

Photo: public domain

Historical Example: The World War II Enigma Machine

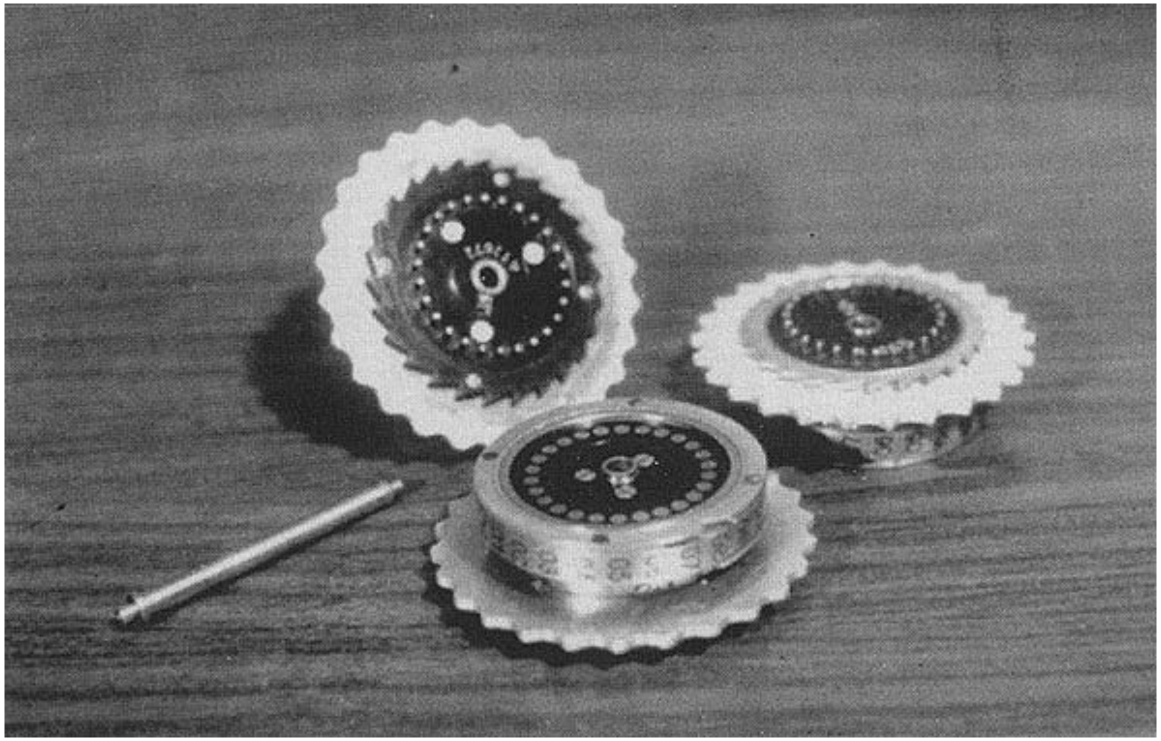


Photo: public domain

You select the proper rotors

Historical Example: The World War II Enigma Machine

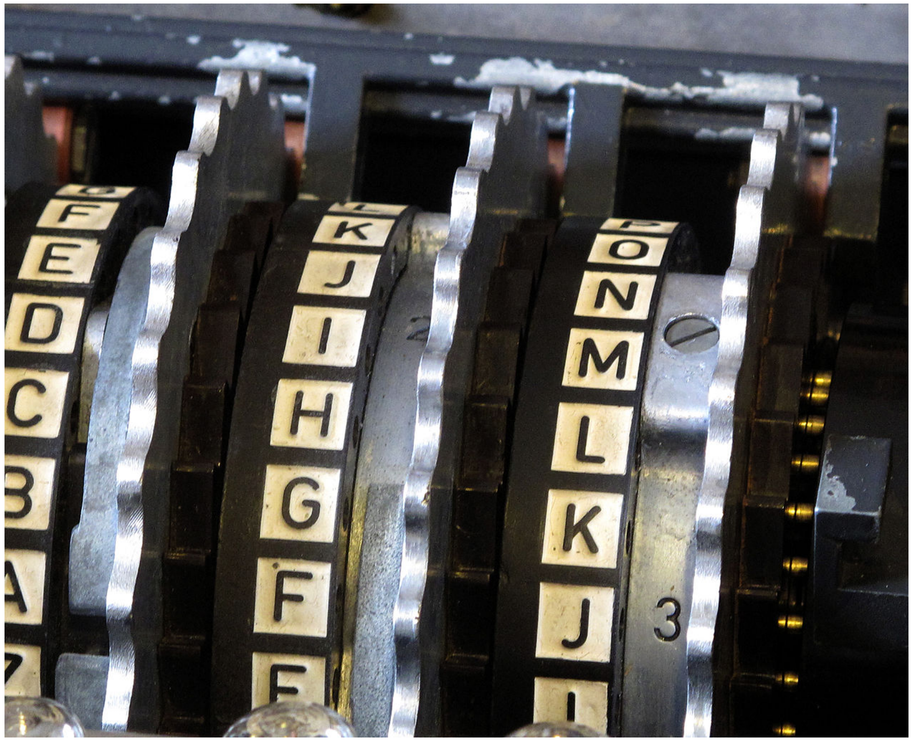


Photo: public domain

Adjust the rotors to their “ground setting”. (More complex than I’m explaining now.)

Historical Example: The World War II Enigma Machine

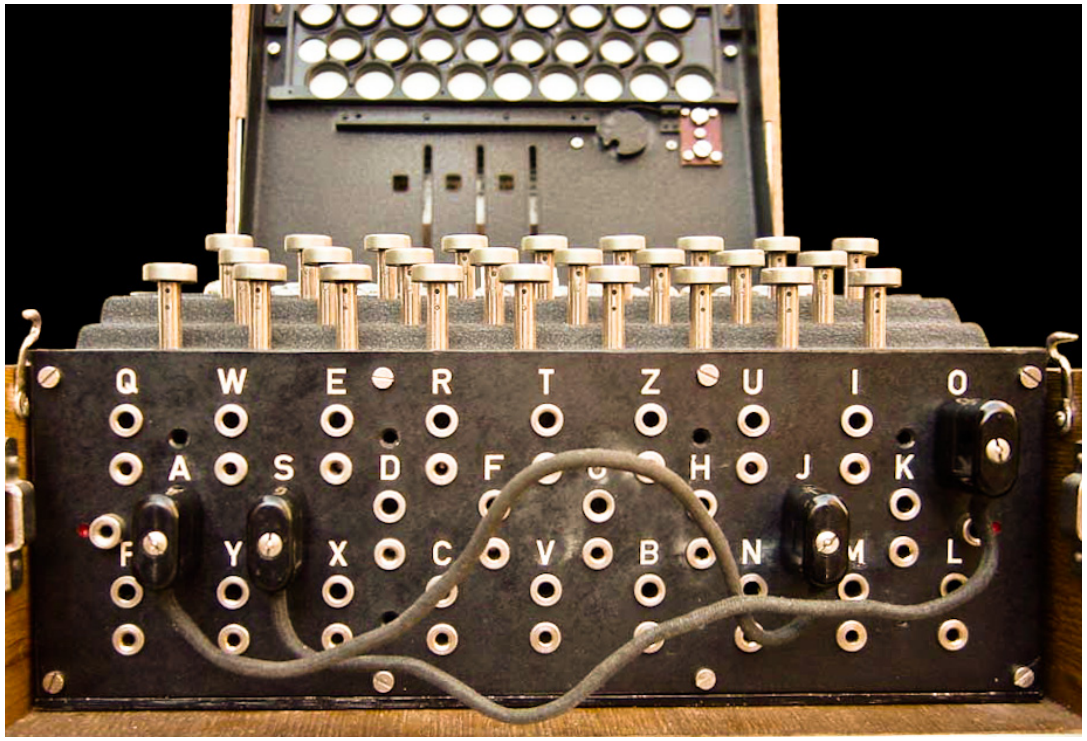


Photo: Bob Lord, via WikiMedia Commons

Set the plugboard

Note: in today's terminology, the rotor selection, initial rotor settings, and plugboard setting were the daily *key*, which was listed in a book.

Historical Example: The World War II Enigma Machine

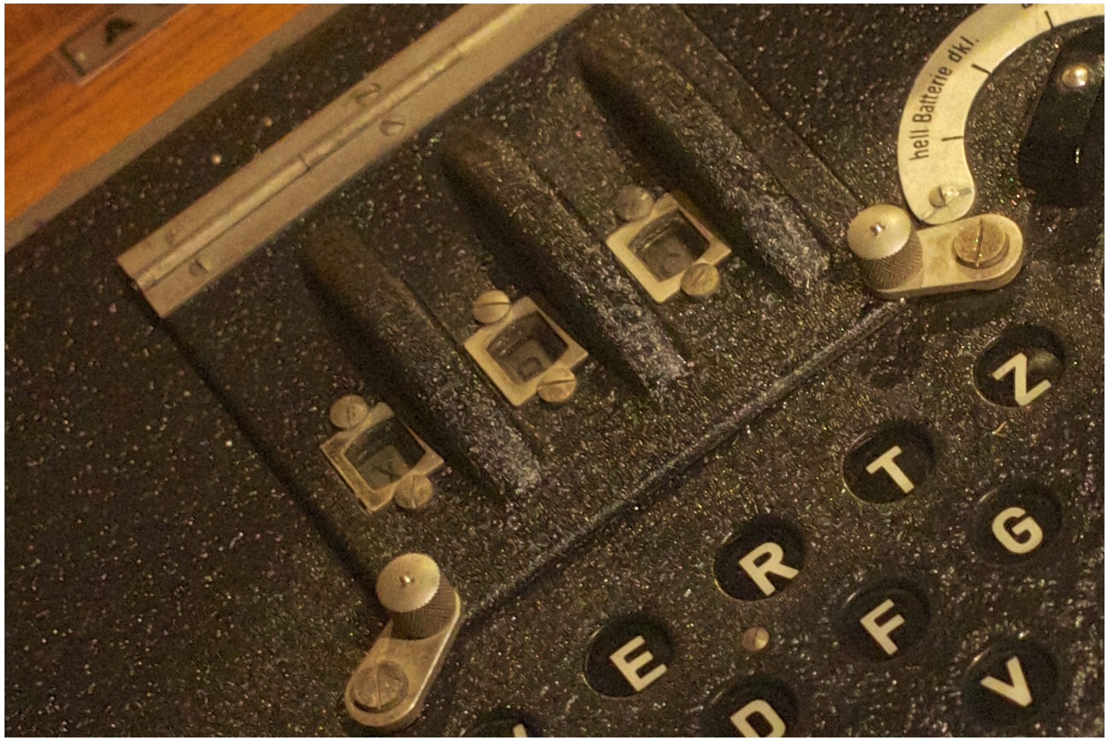


Photo: Paul Hudson, via Flickr

- Pick three random letters and encrypt them twice, and send those six letters as the start of the encrypted message
- (In modern terminology, these letters are the *session key*)
- Reset the rotors to those three letters

What Could Go Wrong?

- Sending the same, simple message every day was a fatal flaw
- Picking non-random letters was a fatal flaw
- Sending a message consisting of nothing but the letter "L" was a fatal flaw—this is partly usage, and partly a design weakness in the Enigma
- Encrypting the three letters *twice* was a fatal flaw

The Three Letters

- Imagine that "`XJM`" was encrypted to "`AMRDTJ`"
- The cryptanalysts realized that `A` and `D` represented the same letter, `M` and `T` were the same, and `R` and `J` were the same
- This gave away valuable clues to the rotor wiring and the rotor order!

Cryptography is very hard…

Modern Cryptography

- Algorithms
- Keys
- Protocols
- Hash functions (not covered in this lecture)
- Digital signatures (also not covered)
- Really arcane math…

Modern Cryptography

- An *algorithm* and a *key* convert *plaintext* into *ciphertext*
- A different algorithm and the key convert ciphertext into plaintext
- Kerckhoff's Maxim (1883): "The system must not require secrecy, and it does not matter if it falls into the hands of the enemy."
  - Don't trust secret designs!

The Data Encryption Standard

- In 1974, NBS (now NIST) adopted the *Data Encryption Standard (DES)* for sensitive but unclassified data
  - Some sources claim the need was realized after the Soviets intercepted calls by US private sector grain deal negotiators
- DES is a *block cipher* and encrypts 64 bits (8 bytes) at a time
- The key length is 56 bits—72,057,594,037,927,936 possible keys

NSA Limited the Key Length

- The original design (from IBM) used 112-bit keys
- The commercial design used 64-bit keys
- This posed a dilemma for the NSA—protect US communications better, with longer keys, or ease cryptanalysis with shorter keys?
- They ended up asking for 48 bits
- The compromise with IBM was 56 bits, 256× (i.e., 2^8×) stronger than 48 bits but 256× weaker than 64 bits

Exhaustive Search

- Any cipher can be cracked by trying each possible key
- For a Caesar cipher, there are only 26 possible keys—clearly too few!
  - In Julius Caesar's day, there were only 23 possible keys…
- If a chip can try one key every microsecond and you have 1 million chips, full search of DES takes ~72,000 seconds—less than a day, on a machine estimated to cost ~$10 million (ballpark figure)
- Moore's Law means the design gets cheaper over time

Using Classical Cryptography

- Alice and Bob want to exchange secret messages
- First, Alice and Bob have to have a *shared key*, which they need to keep secret
- Alice and Bob can then exchange secret messages
- If Carol wants to talk securely with Alice, they need their own key
- If Carol also wants to talk to Bob, they need a key, too
- With *n* people, you need approximately $n^2/2$ keys
- That's a lot of keys to issue and protect
    - How do these people exchange keys in the first place?

Key Management is Hard

- If the Alice-Bob key is ever compromised, all of their past and future traffic is vulnerable
- This means they need to change their key periodically
- The same is true for all of the other pairs of users
- Japanese key distribution difficulties during World War II were very useful to US cryptanalysts—and utterly vital to the US victory in the Battle of Midway

Public Key Cryptography

- In 1976, Diffie and Hellman proposed a radically different form of cryptography: *public key cryptography*
  - We now know that GCHQ had invented much of the same concept—but only five years ahead of Diffie and Hellman
  - Diffie and Hellman had one idea, digital signatures, that even GCHQ and NSA didn't have
- However, Diffie and Hellman only invented the *concept*; they did not know an actual way to do it
- In 1978, Rivest, Shamir, and Adleman developed the RSA algorithm and made it all work

Public Key Cryptography

- In all previous designs, the *decryption key* is the same as the *encryption key*
- In public key systems, there are *two different keys* that are related
- Furthermore, you cannot calculate the decryption key if all you know is the encryption key
- This was revolutionary—and it has enabled today's e-commerce

Using Public Key Cryptography

- Alice generates a *key pair*—an encryption key k and a decryption key k^{-1}—and then publishes k
- Bob does the same thing
- To send Alice a message, Bob looks up her encryption key on the Internet, encrypts the message, and sends it to Alice
- Only Alice knows k^{-1} and hence only she can decrypt the message

Easier Key Management!

- Carol can generate her own key pair, too
- So can anyone else
- For *n* people, we now need only *n* key pairs
- Alice can generate and publish a new key pair any time she wants to—she doesn't have to meet with Bob to exchange keys securely

How Does This Work?

- I'm not going to tell you…
  - It's not secret, it's just math that isn't relevant for this class
  - However, it created very practical uses for a branch of mathematics that had been thought to be of purely academic interest…
- The RSA algorithm is being replaced by a new scheme called *elliptic curve cryptography*, which uses more complex math
- If they ever build quantum computers, we'll need newer algorithms…

Key Sizes

- RSA keys (public and private) are much longer than conventional keys—today, about 2048 bits
  - Public key systems are often called *asymmetric*; conventional systems, from Caesar ciphers to today's, are called *symmetric*
- Symmetric and asymmetric key lengths are not easily comparable—the design and math are very different
- Today's standard cipher (AES) normally uses 128-bit keys, which corresponds to ~3072-bit RSA keys

Problem Solved?

- Well, no—using public key crypto is difficult
- For one thing, public key operations are computationally expensive
- For another, they're mathematically tricky
  - (Don't ask!)
- And we don't want to use any one key too much
- Besides—how do we know this public key we found on the Internet really belongs to Alice?

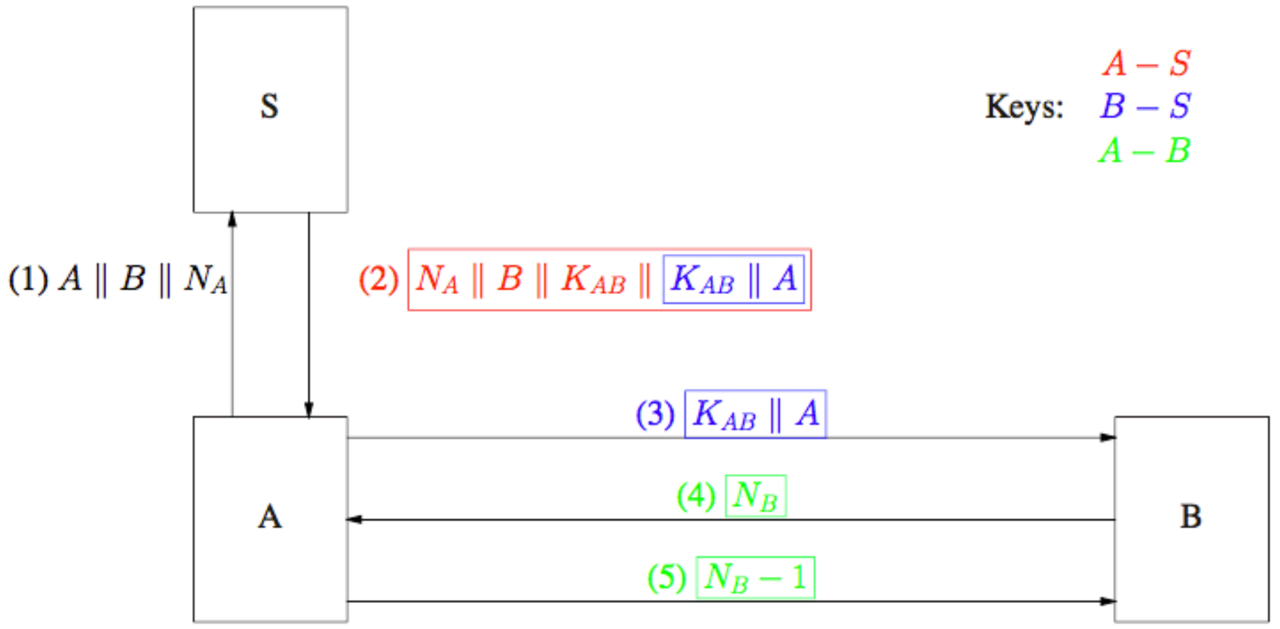
Cryptographic Protocols

- To really use cryptography properly, we need a *protocol*
- A cryptographic protocol is a stylized set of messages between the parties
- The usual goal is for the two parties who wish to communicate to end up with a shared *session key* and to be assured of the other's identity

Needham-Schroeder (1978)

- This is the oldest cryptographic protocol in the open literature
- I’ll show the symmetric version—there’s an asymmetric version, but it requires more complex concepts
- In addition to Alice and Bob, there is a trusted *key server S*
- Alice and Bob each share a key (K_{AS} and K_{BS}) with S
- They want to end up with a new session key K_{AB}

The Needham-Schroeder Protocol


S
Keys: $A - S$ $B - S$ $A - B$
(1) $A \parallel B \parallel N_A$
(2) $\boxed{N_A \parallel B \parallel K_{AB} \parallel \boxed{K_{AB} \parallel A}}$
(3) $\boxed{K_{AB} \parallel A}$
(4) $\boxed{N_B}$
(5) $\boxed{N_B - 1}$
A
B

The Messages

1. Alice sends S her identity, plus a random *nonce*
2. S's response is encrypted in K_{AS}, which guarantees authenticity
   1. N_A guarantees freshness
   2. K_{AB} is the new session key
   3. There's a sealed package for Bob encrypted with K_{BS}
3. Alice sends the sealed package to Bob
4. Bob sends a nonce to Alice encrypted with K_{AB}
5. Alice proves she could read the nonce

You Are Not Expected to Understand This

- Yes, it’s complicated
- Modern protocols are often more complicated—setting up a TLS connection takes six messages, plus several concepts I haven’t discussed
- It’s also not quite correct….

Errors in Needham-Schroeder

- Needham-Schroeder was published in 1978
- In 1981, Denning and Sacco found some flaws and proposed a fix
- In 1987, Needham and Schroeder found a flaw in the fix
- In 1996, Lowe found a new, previously unsuspected flaw in all of the older versions
- All of these flaws were obvious in retrospect—but one of them took 18 years and automated assistance to find

Don’t Try This at Home!

- The basic algorithms are hard
- The protocols are hard
- Users make mistakes
- There’s a lot more I haven’t told you about, stuff that’s utterly vital in the real world
  - In particular, while we still need trusted third parties, we don’t need to trust them quite as much as I’ve shown

Today's Status

- We have high confidence that the basic primitive algorithms are correct
- Our protocols are very complex—errors have been found in many, including some designed by sophisticated companies
- There are often bugs in the code—in one study, 85% of mobile apps got *straightforward* cryptography wrong
- User errors are a major problem

Daily Bird

Evening grosbeak, Riverside Park, February 3, 2019