Encryption

CC BY NC

CS @CU

The Cryptography Problem

We Need Strong Cryptography

- We need it to protect Internet and cellular traffic
- We need it to protect our devices
- We need it to protect our data
- We need it for authentication

Strong Cryptography is a Problem

- The FBI says it's "going dark" and can't monitor terrorist communications
- Local police say they can't search seized devices for evidence
- And what about the NSA?
- They've called for exceptional access (AKA a "golden key" or a "back door") to let them read encrypted content

CS@CU

Technical Basics

Cryptography Basics

- Encryption is a *function* of *plaintext* and a *key*; its result is *ciphertext*

$$E(K, P) \rightarrow C$$

- Decryption is a function of ciphertext and a key; its result is plaintext

$$D(K', C) \rightarrow P$$

- Encryption and decryption are *inverses*

$$D(K', E(K, P)) = P$$

Public Key Cryptography

- For conventional (AKA *symmetric*) ciphers, *K* and *K'* are the same or are trivially related to each other
- For *asymmetric* (AKA *public key*) ciphers, the encryption key *K* can be public and it is difficult or impossible to derive *K'* from *K*
  - Invented at GCHQ circa 1971; reinvented in the open sector circa 1975 by Diffie and Hellman
- Public key cryptography is at the heart of all Internet cryptography—when you use HTTPS to connect to, say, Google, your traffic is encrypted to Google's *public key*

Digital Signatures

- You can often encrypt with the decryption key; this is known as a *digital signature*
  - Only the party that has the private key can create this signature; that's your guarantee of authenticity
- Anyone who knows the corresponding public key—which can, after all, be public—can verify the signature

CS@CU

Today's Major Algorithms

- Symmetric cipher: AES (Advanced Encryption Standard)
- Public key cipher: RSA (aging), ECDH (Elliptic Curve Diffie-Hellman)
- Digital signature: RSA, ECDSA (Elliptic Curve Digital Signature Algorithm)

If used and implemented correctly, these algorithms are believed to be unbreakable

CS@CU

(Alphabets and Keys)

- >200 years ago: ciphertext could be (and often was) composed of all sorts of glyphs; keys were lists of mappings
- Post-telegraph era: ciphertext was letters and digits, symbols easily transmitted by telegraph; keys were generally letters
- Post-1974: plaintext and ciphertext are arbitrary bytes; keys are strings of bits
  - Today, keys for symmetric ciphers are 128-256 bits long
  - Key lengths for asymmetric ciphers vary, depending on the algorithm

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time

CS@CU

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis

CS@CU

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis
- Third thought: Alice and Bob agree on a list of keys to use

CS@CU

A Keylist for the Commercial Enigma

Datum	Innere Einstellung Walzenl.	Ringstellg.	Ausseneinstellung f.Admiralsverkehr 0000-1159	1200-2359
1.10.	I III II	H L D Z	R H K O	L Y E F
2.10.	I II III	I Y M A	H B L Q	R W A Z
3.10.	III I II	W V L T	Z X Q P	W H X X
4.10.	I III II	I T R U	U T E Y	R Q C P
5.10.	I II III	Q W K T	K N W V	X Y D Z
6.10.	III II I	G W Y I	T P F I	B A H L
7.10.	II III I	P H Z Y	K L T M	T V N W
8.10.	II I III	T X R S	R I K H	Y C R E
9.10.	III I II	D S T P	Y P S P	E A F X
10.10.	III II I	U M K G	S V Q V	A R F R
11.10.	II I III	H Z P Q	K O T K	V U Z K
12.10.	II III I	Y S P K	X R X U	X C A R
13.10.	II I III	T C S C	K A Z B	Z T Q F
14.10.	I III II	Y H Y H	N O F K	R L E E
15.10.	I II III	R X M N	K G U P	S Y D S

A portion of a key list for the commercial Enigma used by German forces in the Spanish civil war. For each day it shows, under *Innere Einstellung* (inner setting), the positions for the three removable rotors in Roman numerals and the alphabet ring settings for the three removable rotors and the reversing rotor. Under *Ausseneinstellung* (outer setting), it gives the rotor positions for the start of encipherment of each message sent during two periods of the day.

(From David Kahn, *Seizing the Enigma*)

CS@CU

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis
- Third thought: Alice and Bob agree on a list of keys to use
  - Messages must contain *metadata* saying which key was used

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis
- Third thought: Alice and Bob agree on a list of keys to use
  - Messages must contain *metadata* saying which key was used
- Fourth thought: what if you send more than one message per period?

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis
- Third thought: Alice and Bob agree on a list of keys to use
  - Messages must contain *metadata* saying which key was used
- Fourth thought: what if you send more than one message per period?
  - You need to choose—and send—some randomization for the pre-arranged key

Alice Wants to Send a Message to Bob—How?

- First thought: Alice just encrypts the message with *K* as the key
  - Umm—how do Alice and Bob both know the same *K*?
- Second thought: Alice and Bob agree on *K* ahead of time
  - Umm—sending multiple (non-random) messages with the same key is a bad idea; it can enable easy cryptanalysis
- Third thought: Alice and Bob agree on a list of keys to use
  - Messages must contain *metadata* saying which key was used
- Fourth thought: what if you send more than one message per period?
  - You need to choose—and send—some randomization for the pre-arranged key

And now what happens if Alice needs to talk to Carol, or Bob needs to talk to David?

Which is Bob's Key?

- Alice wants to send a message to Bob, so she looks up his key on the web
- Did she find Bob's key—or Eve's?
- Answer: a *certificate*, a digitally signed message saying "this is Bob's key"
  - (Certificates were invented in 1978 by Loren Kohnfelder, an MIT undergrad, in his senior thesis)
- Who does the signing? A *certificate authority*, a mutually trusted party

CS@CU

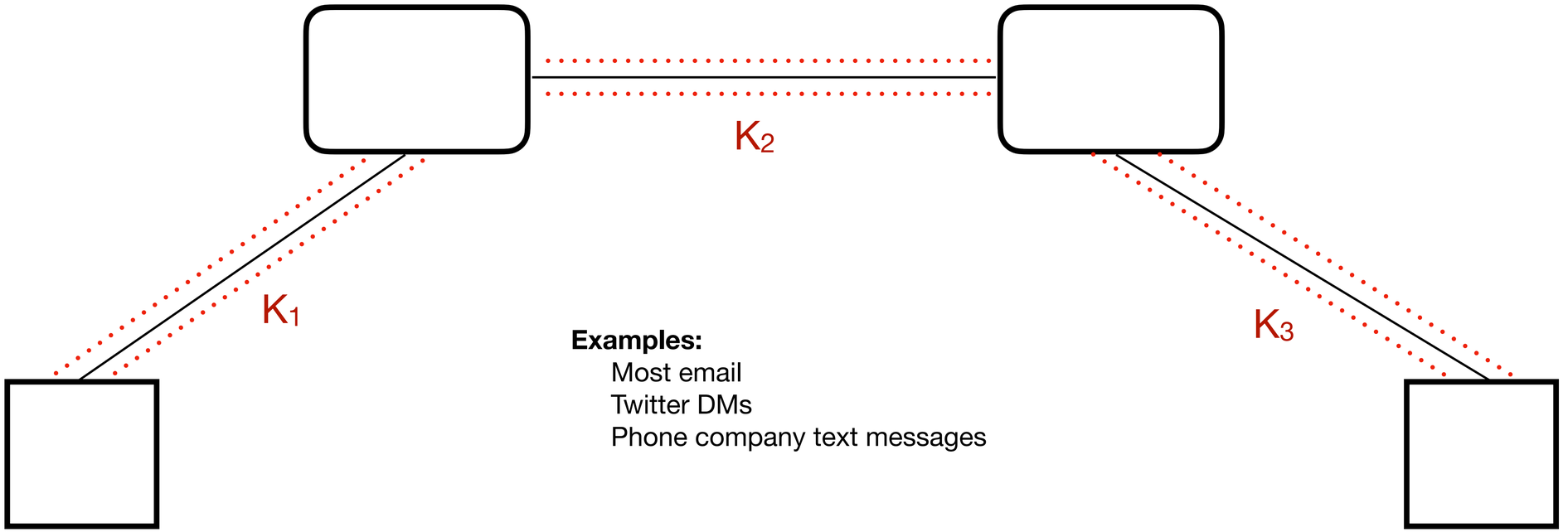
Normal Use

(Very oversimplified…)

- Alice generates a random *session key* K_s
- She encrypts message *m* using K_s
- Alice finds Bob's certificate and extracts from it his public key K_p
- She encrypts K_s using K_p
- She sends Bob the encrypted message and the encrypted session key
- This transmission needs *lots* of metadata…

CS@CU

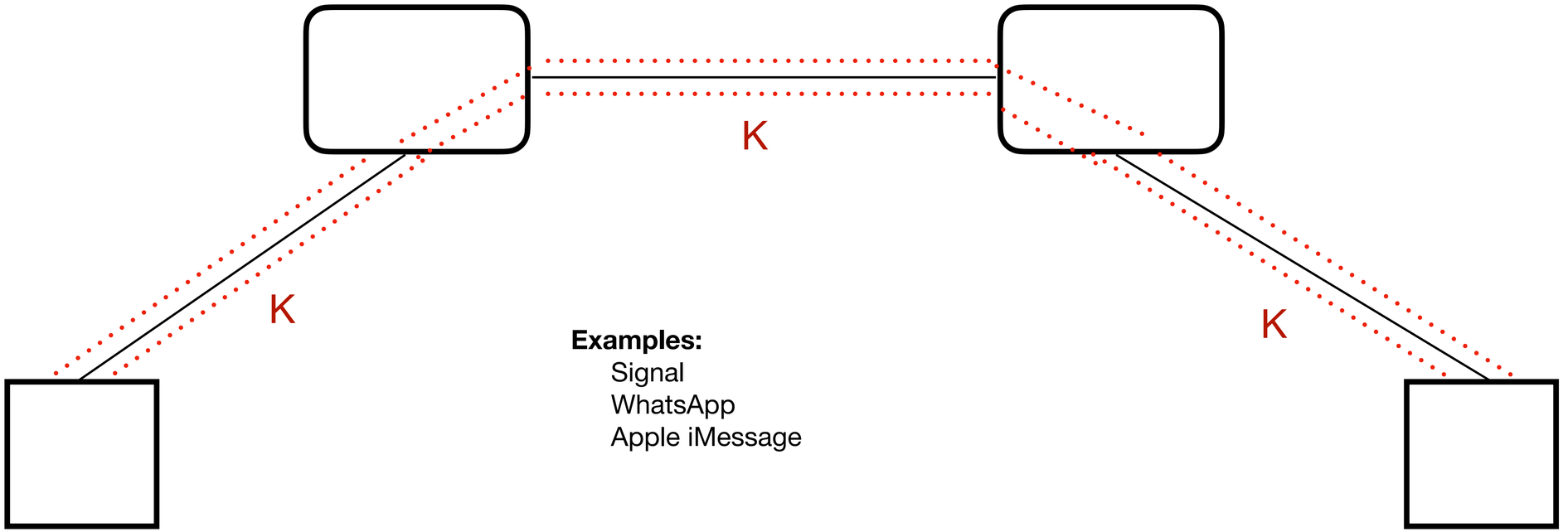
Hop-by-Hop Encryption



Examples:

Most email
Twitter DMs
Phone company text messages

End-to-End Encryption



Examples:

Signal
WhatsApp
Apple iMessage

Golden Keys?

Cryptography is Hard…

The very first academic paper on cryptographic protocols (Needham and Schroeder,1978) ended with a warning:

> Finally, protocols such as those developed here are prone to extremely subtle errors that are unlikely to be detected in normal operation. The need for techniques to verify the correctness of such protocols is great, and we encourage those interested in such problems to consider this area.

They were right—a simple flaw in their scheme went unnoticed for 18 years…

CS@CU

Historical Example: Enigma

- Picking non-random letters for the session key was a fatal flaw
- Encrypting the session key twice was a fatal flaw
- Sending the same, simple message every day was a fatal flaw
- Sending a message consisting of nothing but the letter “L” was a fatal flaw

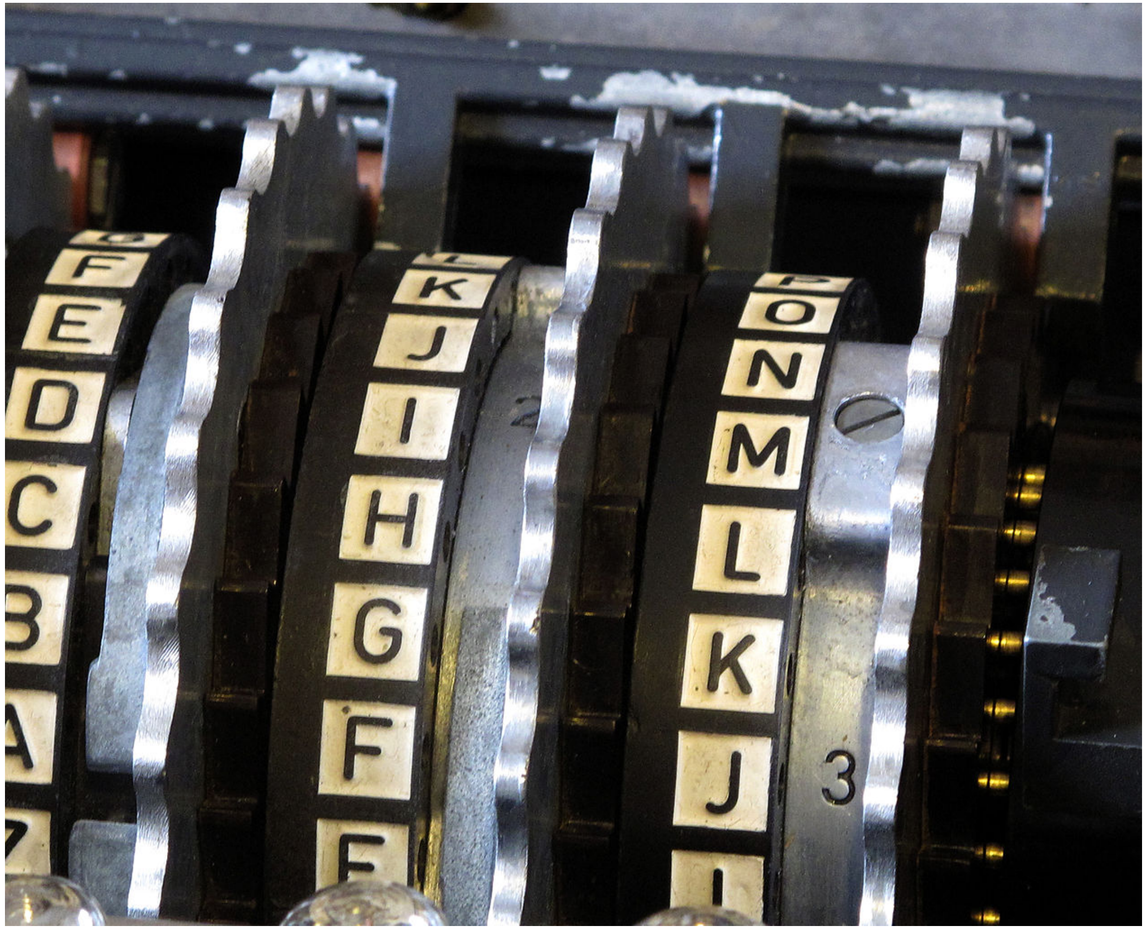


Photo: public domain

Modern Examples

- Incorrectly padding a short message to match the encryption algorithm's requirements has resulted in security flaws
- Not authenticating every encrypted message has resulted in flaws. (That was a flaw found in Apple's iMessage protocol.)
- Omitting sequence numbers from encrypted messages has resulted in flaws
- The existence of older, "exportable" algorithms in the key and algorithm negotiation protocol has resulted in flaws
- Trying to provide an "additional encryption key" for the government has resulted in flaws

Additional Encryption Key

- Someone had a bright idea: add an *additional encryption key* to the certificate
- When Alice gets Bob's certificate, she'll learn his public key and the government's, and can encrypt K_s to both
- The certificate authority signed Bob's name and Bob's public key, but—in this scheme—did not sign the additional encryption key
- Result: an attacker can edit the certificate to substitute their own encryption key, so the message is encrypted to them (instead of the government) as well as to Bob

Cryptography is hard…

Procedural Issues

CS@CU

Authentication and Authorization

- Who is authorized to request decryption?
- Law enforcement? Which agencies? From which countries? Their customs agents? Their intelligence agencies?
- How does the designated unlocking agent for a phone (probably the vendor) authenticate the requester? How is this done internationally?

How Are Golden Keys Protected?

- Golden keys would be used tens of times per day or more—can you protect them from foreign intelligence agencies?
- What about protection of the keys that requesters to authenticate their requests?
- If requests are very frequent, are they carefully vetted?

Exceptional Access for Communications

- If some agency somewhere wants to read a communication, how did they acquire it? Legally? From where?
- This is especially serious for international access requests—is this traffic *you* think they should be allowed to read?
  - All countries criminalize some things, e.g., child pornography
  - Most criminalize drug trafficking
  - But what if it's a dissident's traffic? One of your business people?

Exceptional Access for Devices

- An easier problem than for communications, because you can use possession of the device as an authentication factor
- But: was the device taken “legitimately”? According to what countries’ standards?
- What about travelers crossing borders into which don’t respect the rule of law?
- Remember that phones are very commonly used as authenticators to other services

International Issues

- If one major country has the ability to bypass encryption, every other country will want the ability, too
- Are golden keys locked to some country? How?
  - What about border-crossings or international communications?
  - What about imported phones?
- Does the vendor's country have veto power over another country's unlock requests? How well will that play with other countries?
  - Even knowledge of unlock requests can be sensitive

CS@CU

Conclusions

- The exceptional access problem is not one problem but many
- Some of the issues are political, not technical; this can make them harder to solve
- But the technical difficulties are daunting enough!
- Strong encryption is essential for security; the debate here is not privacy versus security, it's security versus security

Exceptional Access?

- Most cryptographers think that exceptional access is a really bad idea
- Why?
- Because we haven't “nerded harder”?
- No—it's because we think it's inherently unsafe
- Maybe we can get the protocol right—but it will be a low-assurance solution; we won't know that it's correct

CS@CU

Bird of the Day

Red-breasted nuthatch, Central Park, March 6, 2021

CS@CU