Cryptography

Well, a gentle intro to cryptography. Actually, a fairly "hand-wavy" intro to crypto (we'll discuss why)

Special Thanks: to our friends at the Australian Defense Force Academy for providing the basis for these slides

Definition

- Cryptography is concerned with developing algorithms which may be used:
  - To conceal the content of some message from all except the sender and recipient (*privacy* or *secrecy*), and/or
  - Verify the correctness of a message to the recipient (*authentication* or *integrity*)
- The basis of many technological solutions to computer and communication security problems

Terminology

- Cryptography: The art or science encompassing the principles and methods of transforming an intelligible message into one that is unintelligible, and then retransforming that message back to its original form
- Plaintext: The original intelligible message
- Ciphertext: The transformed message
- Cipher: An algorithm for transforming an intelligible message into one that is unintelligible

Terminology (cont).

- Key: Some critical information used by the cipher, known only to the sender & receiver
  - Or perhaps only known to one or the other
- Encrypt: The process of converting plaintext to ciphertext using a cipher and a key
- Decrypt: The process of converting ciphertext back into plaintext using a cipher and a key

Still More Terminology…

- Cryptanalysis: The study of principles and methods of transforming an unintelligible message back into an intelligible message ***without knowledge of the key!***
- Cryptology: The field encompassing both cryptography and cryptanalysis

Concepts

- Encryption: The mathematical operation mapping plaintext to ciphertext using the specified key:

$$C = E_K(P)$$

- Decryption: The mathematical operation mapping ciphertext to plaintext using the specified key: $P = E_K^{-1}(C) = D_K(C)$

Concepts

- Cryptographic system: The family of transformations from which the cipher function E_K is chosen
  - It is a family of transformations since each key K effectively creates a different transformation
- Key is selected from a *keyspace K*
  - How the key is selected is important
    - It's not always uniform!

Concepts (cont.)

- More formally we can define the cryptographic system as a single parameter family of invertible transformations

  E_K for K in *K* maps P -> C

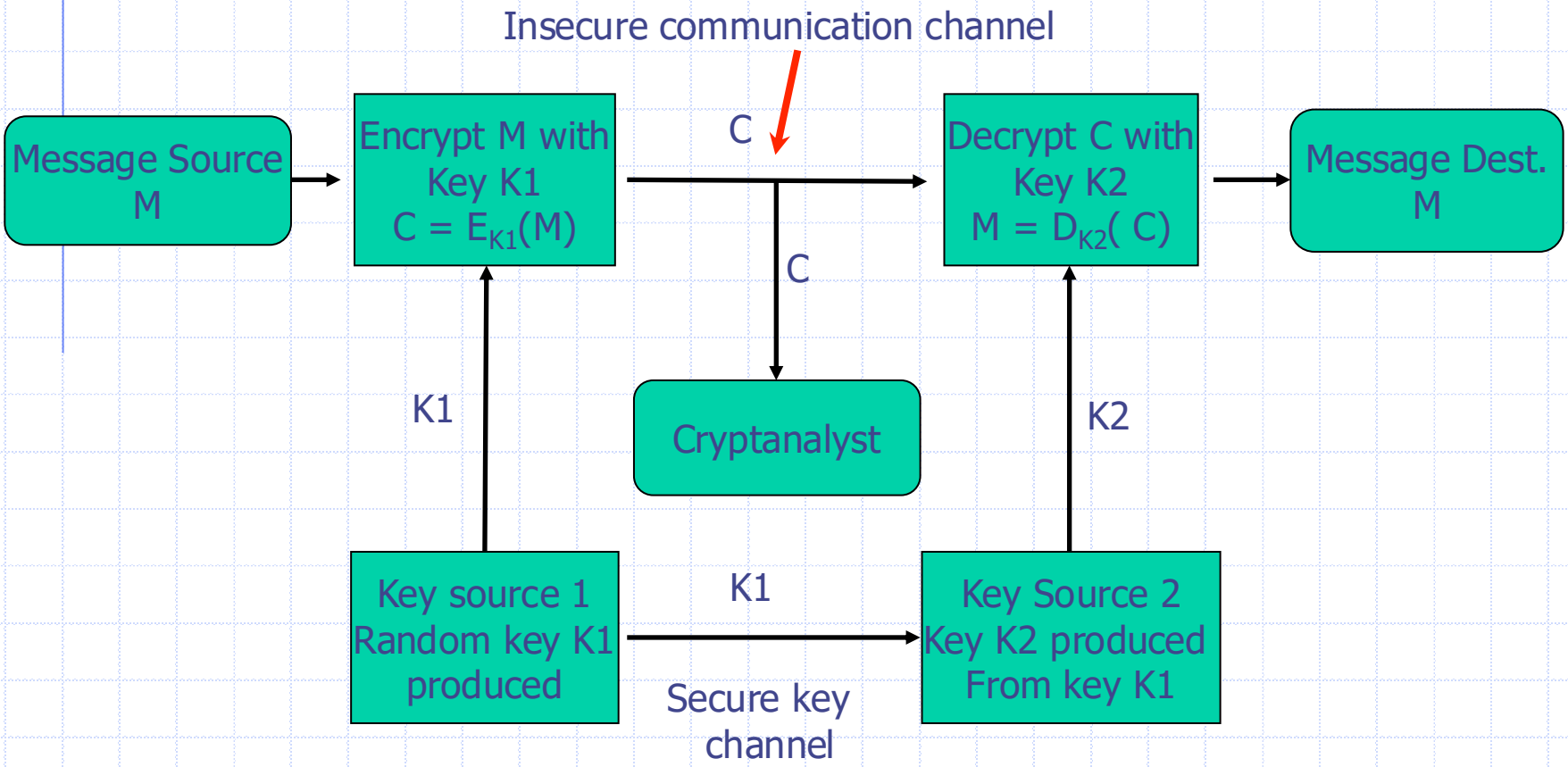
With unique inverse $P = E_K^{-1}$ for K in *K* maps C -> P

- Usually assume the cryptographic system is public, and only the key is secret information
  - Why?

Rough Classification

- Symmetric-key encryption algorithms
- Public-key encryption algorithms
- Digital signature algorithms
- Hash functions
- Cipher Classes
  - Block ciphers
  - Stream ciphers

Symmetric-Key Encryption System



Symmetric-Key Encryption Algorithms

- A Symmetric-key encryption algorithm is one where the sender and the recipient share a common, or closely related, key
  - Managing this key is nontrivial
  - Plus there is the question: how does the key come to be shared?

Symmetric-Key Encryption Algorithms

- Historically, symmetric-key algorithms were developed first
  - They are generally good at efficiently encrypting large amounts of data
    - Currently, consumer cores can achieve, using AES-256 cipher, over 1 GB/s
    - Specialized IP cores can achieve throughputs greater than 4 Tbps (yes, that's a T)
    - Bottom line: real-time encryption can be achieved with minimal latency

Types of Cryptanalytic Attacks

- These are attacks on the algorithms
  - There are other types of attacks on crypto in general
- Ciphertext-only attack
- Known-plaintext attack
- Chosen-plaintext attack
- Chosen-ciphertext attack

Ciphertext-only attack

- Attacker only knows algorithm and some ciphertext (but not plaintext associated with it)
  - use statistical attacks only
    - Probability distributions describing characteristics of plaintext message
  - plus publicly available knowledge

Ciphertext-only attack

- Attacker only knows algorithm and some ciphertext (but not plaintext associated with it)
  - must be able to identify when have plaintext
  - Note: This is the most difficult of the attack classes.
    - For this reason, this is not the attack you want to use to measure the efficacy of an encryption scheme.

Known-plaintext attack

- Know (or strongly suspect) some plaintext-ciphertext pairs
- More information gives attacker more leverage. Easier than ciphertext only attack, but still not a good measure of cipher strength

Known-plaintext attack

- How does an adversary obtain plaintext-ciphertext pairs?
  - Secret data might not remain secret forever
    - Example: Encrypted message suspected of being contents of official diplomatic statement that is later released
    - Example: If message gives location of an attack (known after attack)
    - Example: Message is text of contract later made public

Known-plaintext attack

- How does an adversary obtain plaintext-ciphertext pairs?
  - Fixed header/preamble analysis
    - Protocol messages start with a fixed known preamble
  - Constant headers of a file format
    - E.g., PDF, Word, .zip
  - Standardized reports (have predictable phrases)
    - E.g., weather report
    - In WWII, the British broke the Enigma ciphers because the Germans transmitted weather reports in a standardized format

Cryptanalytic Attacks

- Chosen-Plaintext Attack: The attacker can exercise partial control over what the honest party encrypts
  - Can select plaintext and obtain corresponding ciphertext

Chosen-Plaintext Attacks

- How does adversary get the pairs here?
  - Adversary may obtain a device that performs encryptions
    - E.g., attacker types on a terminal which encrypts what is typed with a key shared with a remote terminal (key and remote terminal unknown).
    - Sometimes called "lunchtime" or "midnight" attack
  - Note: Encryption scheme should remain secure when used with same key to encrypte data for another user

Chosen-Plaintext Attacks

- How does adversary get the pairs here?
  - Public key encryption: encryption key is public and known
  - Encrypted services/Web APIS
    - If a web service accepts user input and sends back an encrypted version of it
  - Interaction with encryption software (if it uses the same key as target)
  - Exploiting non-randomized encryption
    - In which case same inputs encrypted with same key always give same outputs

Chosen Plaintext Attacks in Real World

- British coerced Germans to send ciphertexts corresponding to locations of mines
  - During WWII, British placed mines in certain locations knowing that when the mines were found the Germans would encrypt the locations and send them back to headquarters.
  - These encrypted messages used by Bletchley Park to help break the German encryption scheme

Chosen Plaintext Attacks in Real World

- Japan and Midway Island during WWII
  - In May 1942 navy cryptanalysts intercepted encrypted messages from Japanese which they were able to partially decode. These indicated a planned attack on a location AF, where AF was a ciphertext fragment navy was unable to decrypt.

Chosen Plaintext Attacks in Real World

- Japan and Midway Island during WWII
  - For other reasons, navy believed attack location was Midway Island. Washington planners could not be convinced that this was the case
  - Navy cryptanalysts devised plan: Have US forces at Midway send a fake message that their freshwater supplies were low. Japanese intercepted this message and immediately sent out an encrypted message to superiors that AF is low on water

Cryptanalytic Attacks

- Chosen-ciphertext Attack: In addition to what attacker has with a chosen plaintext attack, she can also select ciphertext(s) and obtain corresponding plaintext(s) (with some limits)
- Ciphers that are secure against such attacks are said to be chosen-ciphertext attack secure (CCA-secure)
- But what does being "secure" against such an attack mean?

CCA Indistiguishability Experiment

1) A random key K of length n is generated
2) Adversary A is given oracle access to E_K and D_K, which respectively encrypt and decrypt with K.
3) A outputs a pair of messages m_1, m_2 of same length

CCA Indistiguishability Experiment

4) A random bit $b \in \{0,1\}$ is chosen, and then a ciphertext $E_K(m_b)$ is computed and given to A. Called the *challenge ciphertext*
5) A can use either oracle, but is not allowed to query the decryption oracle on the challenge ciphertext. Eventually A outputs a bit b'
6) Output of experiment is 1 if b'=b, 0 else. If output is 1, we say that A *succeeds*

CCA Indistiguishability Experiment

- A private-key (as opposed to public key) encryption scheme has indistinguishable encryptions under a chosen-ciphertext attack, (i.e., is CCA-Secure) if for all probabilititic polynomial time adversaries A the probability that A succeeds is ≤ ½ + negligible(n)
- There are technical definitions for what a probabilistic polynomal time algorithm is and what "negligible" means
- Note that no deterministic cipher can be CCA-Secure (why?)

CCA Indistiguishability Experiment

- A probabilistic algorithm is one which can access randomness (e.g., toss a coin) at each step of execution
- Negligible(n): A function f from the natural numbers to the nonnegative real numbers is negligible if for every positive polynomial p there is an N such that for all integers n > N, it holds that $f(n) < 1/p(n)$
  - Ex. 2^{-n}, $n^{-\log n}$

Probabilistic Encryption

- If no deterministic cipher can be CCA-Secure, what kind of cipher can be? A probabilistic cipher!
- Requirements: Encrypting message m with key K results in potentially different ciphertext each time. Decrypting any of those ciphertexts with key K results in message m
  - Is this even possible?

Probabilistic Encryption

Example of probabilistic encryption in the random oracle model:

- *x* - plaintext
- *f* - trapdoor permutation (deterministic encryption algorithm)
- *h* - random oracle (typically implemented using a publicly specified hash function)
- *r* - random string

$$\mathrm{Enc}(x) = (f(r), x \oplus h(r))$$

$$\mathrm{Dec}(y, z) = h(f^{-1}(y)) \oplus z$$

Thanks Wikipedia!

Chosen Ciphertext Attack in Real World

- Midway example: US forces could have sent cipher texts to Japanese and monitored their behavior (e.g., movement of forces)
- One common example: encryption algorithm often used as part of a higher level protocol. E.g., used as part of an authentication protocol, where one party sends a ciphertext to another, who decrypts it.
  - In this case honest party is the decryption oracle!

Chosen Ciphertext Attack in Real World

- Padding-Oracle Attacks
  - This is a chosen-ciphertext attack on a widely used encryption scheme
  - Worse, attack only requires ability of the attacker to determine whether or not a modified ciphertext decrypts correctly
    - Usually easy to obtain: Ex. A server that asks for retransmission or terminates if it receives a ciphertext that does not decrypt correctly
  - Attack has been shown to work on various deployed protocols!

Cryptanalytic Attacks

- A good cipher *must* resist all four attacks!
  - Typically, chosen ciphertext attack is the one that we measure against, since this gives the adversary the most weapons.

What is a Secure Cipher?

- So, the assumptions
  - We have an encryption scheme. Often written (K,E,D), where K is the key space, E is the family of ciphers (recall one cipher for each key in K), and D is the family of decryption algorithms
  - Two parties share a key $k \in K$, which is unknown to the adversary
- Given this, we seek to ascertain "what does it mean for the cipher to be secure?"

See Modern Cryptography, Bellere and Rogaway, Section 5.3

What is a Secure Cipher?

- Sometimes good to start with what is definitely insecure
- So, obviously insecure if the adversary can, from a few ciphertexts, learn the key k
- So going in the other direction, we posit: a cipher is secure if it is very difficult for the adversary to learn the key k

What is a Secure Cipher?

- So going in the other direction, we posit: a cipher is secure if it is very difficult for the adversary to learn the key k
- Won't work: It might be possible for the adversary to learn the plaintext without ever having to learn the key.
  - E.g., ciphertext might always be equal to the plaintext!

What is a Secure Cipher?

- Another try: A cipher is secure if it is very hard to learn a plaintext from the ciphertext
- Thoughts?

What is a Secure Cipher?

- Another try: A cipher is secure if it is very hard to learn a plaintext M from the ciphertext C
- Won't work: It might be possible for the adversary to learn *partial information* about M
  - E.g., perhaps the first bit of M
  - Or the sum of all the bits of M

What is a Secure Cipher?

- Won't work: It might be possible for the adversary to learn partial information about M
  - Ex. Suppose there is a set of stocks numbered 1 through m. I send a sequence of bits to my broker – if bit i is 1, that means buy stock i. The message is encrypted.
    - So if the adversary learns bit 1, it knows whether I want to buy stock 1
    - If it learns the sum of the bits, it knows how many stocks I want to buy
    - I might not want this information to leak

What is a Secure Cipher?

- You might argue: well this is only a problem if you send your stock order using this format
- Making assumptions about how users format data, or how they use it, is a bad cryptographic design!
- Principle: Encryption scheme should provide security regardless of the format of the plaintext
  - User should be able to format data any way the like without worrying about compromising security!

What is a Secure Cipher?

- OK, so another try: An encryption scheme is secure if given ciphertext C the adversary has no idea what the plaintext M is.
- This cannot be true due to *a priori* information – often something about the message is known
  - Might be packet with known headers
  - Might be an English word
- So adversary and everyone else has some information about the plaintext before it is even encrypted

What is a Secure Cipher?

- Ideally, cipher would be like magic: plaintext would be delivered from the sender to the receiver magically
  - Adversary would see nothing at all
- Our goal: approximate this as best as we can
- One caveat: encryption not required to hide the length of the plaintext string
  - Often this is known to the adversary beforehand
  - And there have been cases where this was bad

What is a Secure Cipher?

- Another scenario: A missile launch center effectively sends only two messages: launch or don't launch
  - The adversary sees a ciphertext. If a missile launches, it knows this ciphertext is for "launch". Otherwise it's for "don't launch"

What is a Secure Cipher?

- This gets us back to something we saw earlier: for a scheme to be secure, the same message encrypted under the same key must be generate different cyphertexts!
  - Either via probabilistic encryption
  - Or by adding some state information that changes each time the plaintext is encrypted (e.g., a counter)

So, What Is Our Definition of a Secure Cipher?

- We've already seen it…
- A private-key (as opposed to public key) encryption scheme has indistinguishable encryptions under a chosen-ciphertext attack, (i.e., is CCA-Secure) if for all probabilititic polynomial time adversaries A the probability that A succeeds is ≤ ½ + negligible(n)
- CCA-Secure cipher

Exhaustive Key Search

- Always theoretically possible to simply try every key
- Most basic attack, directly proportional to key size
- *Assumes attacker can recognize when plaintext is found!!*
  - Often not a real roadblock. If you base your security on the inability of a realistically feasible exhaustive search to not recognize the plaintext, you've got a problem.

Exhaustive Key Search

Key Size	Possible combinations
1-bit	2
2-bit	4
4-bit	16
8-bit	256
16-bit	65536
32-bit	4.2×10^9
56-bit (DES)	7.2×10^{16}
64-bit	1.8×10^{19}
128-bit (AES)	3.4×10^{38}
192-bit (AES)	6.2×10^{57}
256-bit (AES)	1.1×10^{77}

Exhaustive Key Search

- As of late 2024, El Capitan, at Lawrence Livermore Laboratory is the fastest supercomputer in the world
  - 1.74 Exaflops = 1.74×10^{18} FLOPS
- Number of FLOPS required per key check
  - Optimistically estimated at 1000
- Number of key checks per second
  - $1.74 \times 10^{18} / 1000 = 1.74 \times 10^{15}$
- Number of seconds in a year
  - 31,536,000, so 5.487264×10^{22} keys per year
- Expected number of years to crack AES-128
  - more than 6.2×10^{15}
  - $2.1101971626901166e \times 10^{54}$ for AES-256

Unconditional and Computational Security

- Unconditional security: No matter how much computer power is available, the cipher cannot be broken since the ciphertext provides insufficient information to uniquely determine the corresponding plaintext
  - Probabilistic definition: basically, having ciphertext gives no info about plaintext
- Computational security: Given limited computing resources (e.g., time needed for calculations is greater than age of universe), the cipher cannot be broken

Classic Encryption Techniques

- Two basic components in classical ciphers: substitution and transposition
  - Substitution ciphers - letters replaced by other letters
  - Transposition ciphers – same letters, but arranged in a different order
  - Several such ciphers may be concatenated together to form a product cipher

The Caeser Cipher

- 2000 years ago Julius Caesar used a simple substitution cipher, now known as the Caesar cipher
  - First attested use in military affairs (e.g., Gallic Wars)
- Concept: replace each letter of the alphabet with another letter that is k letters after original letter
- Example: replace each letter by 3rd letter after

The Caeser Cipher

- Can describe this mapping (or translation alphabet) as:

  Plain:
  ABCDEFGHIJKLMNOPQRSTUVWXYZ

  Cipher:
  DEFGHIJKLMNOPQRSTUVWXYZABC

General Caesar Cipher

- Can use any shift from 1 to 25
  - I.e., replace each letter of message by a letter a fixed distance away
- Specify key letter as the letter that plaintext A maps to
  - E.g., a key letter of F means A maps to F, B to G, … Y to D, Z to E, I.e., shift letters by 5 places
- Hence have 26 (25 useful) ciphers
  - Breaking this is easy.

Mixed Monoalphabetic Cipher

- Rather than just shifting the alphabet, could shuffle (jumble) the letters arbitrarily
- Each plaintext letter maps to a different random ciphertext letter, or even to 26 arbitrary symbols
- Key is 26 letters long

```
Plain:      ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher:     DKVQFIBJWPESCXHTMYAUOLRGZN

Plaintext:  IFWEWISHTOREPLACELETTERS
Ciphertext: WIRFRWAJUHYFTSDVFSFUUFYA
```

Security of Mixed Monoalphabetic Cipher

- With a key of length 26, now have a total of 26! ~ 4 x 10^{26} keys
  - El Capitan would take 7346+ years
- With so many keys, might think this is secure…but you'd be wrong
  - Your laptop can break this in seconds

Security of Mixed Monoalphabetic Cipher

- Variations of the monoalphabetic substitution cipher were used in government and military affairs for many centuries into the middle ages
- The method of breaking it, frequency analysis was discovered by Arabic scientists
- All monoalphabetic ciphers are susceptible to this type of analysis

Language Redundancy and Cryptanalysis

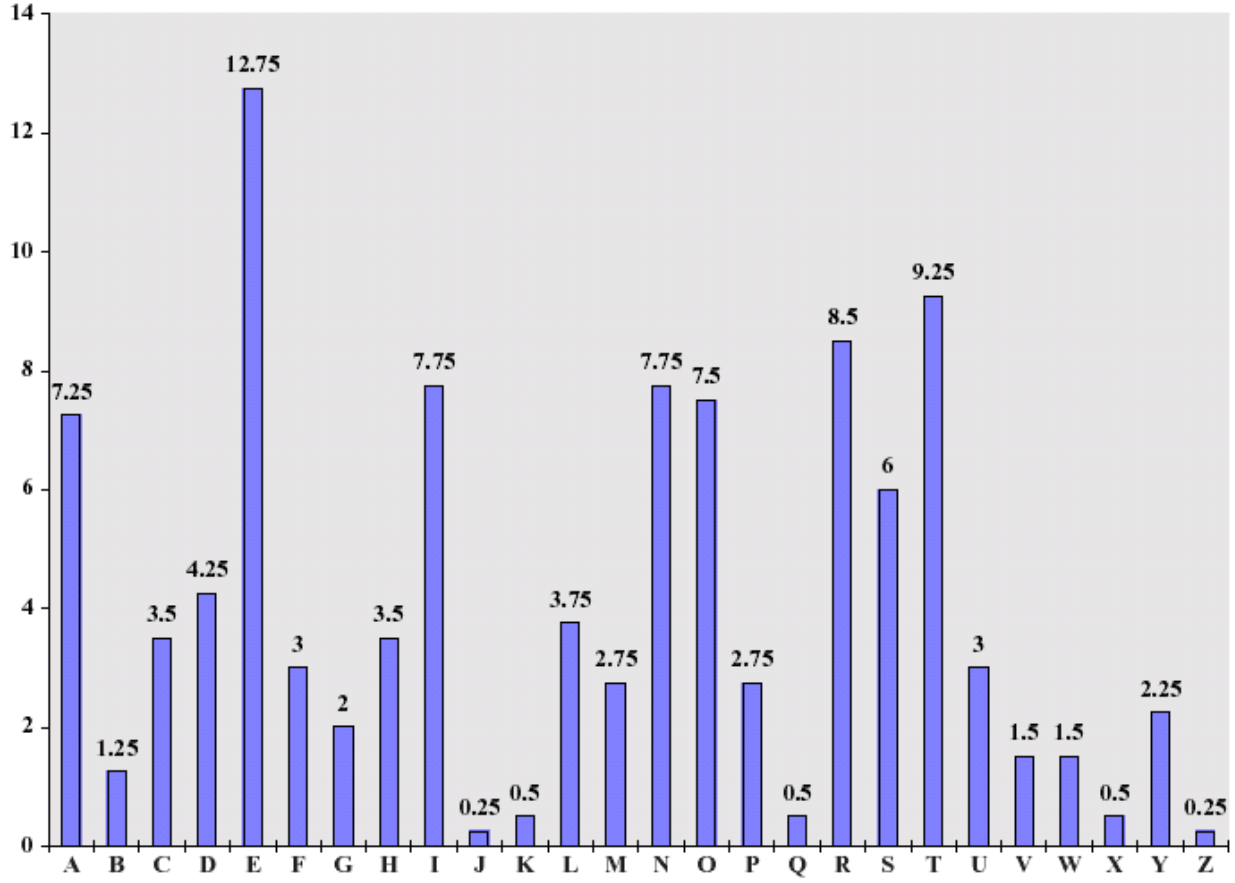
- Human languages are redundant
- Letters in a given language occur with different frequencies.
  - Ex. In English, letter e occurs about 12.75% of time, while letter z occurs only 0.25% of time.
- In English the letters e is by far the most common letter

Language Redundancy and Cryptanalysis

- t,r,n,i,o,a,s occur fairly often, the others are relatively rare
- w,b,v,k,x,q,j,z occur least often
- So, calculate frequencies of letters occurring in ciphertext and use this as a guide to guess at the letters. This greatly reduces the key space that needs to be searched.

Language Redundancy and Cryptanalysis

- Tables of single, double, and triple letter frequencies are available



Other Languages

- Natural languages all have varying letter frequencies
- Languages have different numbers of letters (cf. Norwegian)
- Can take sample text and count letter frequencies
- Seberry (1st Ed) text, Appendix A has counts for 20 languages. Hits most European & Japanese & Malay

Performing Frequency Analysis

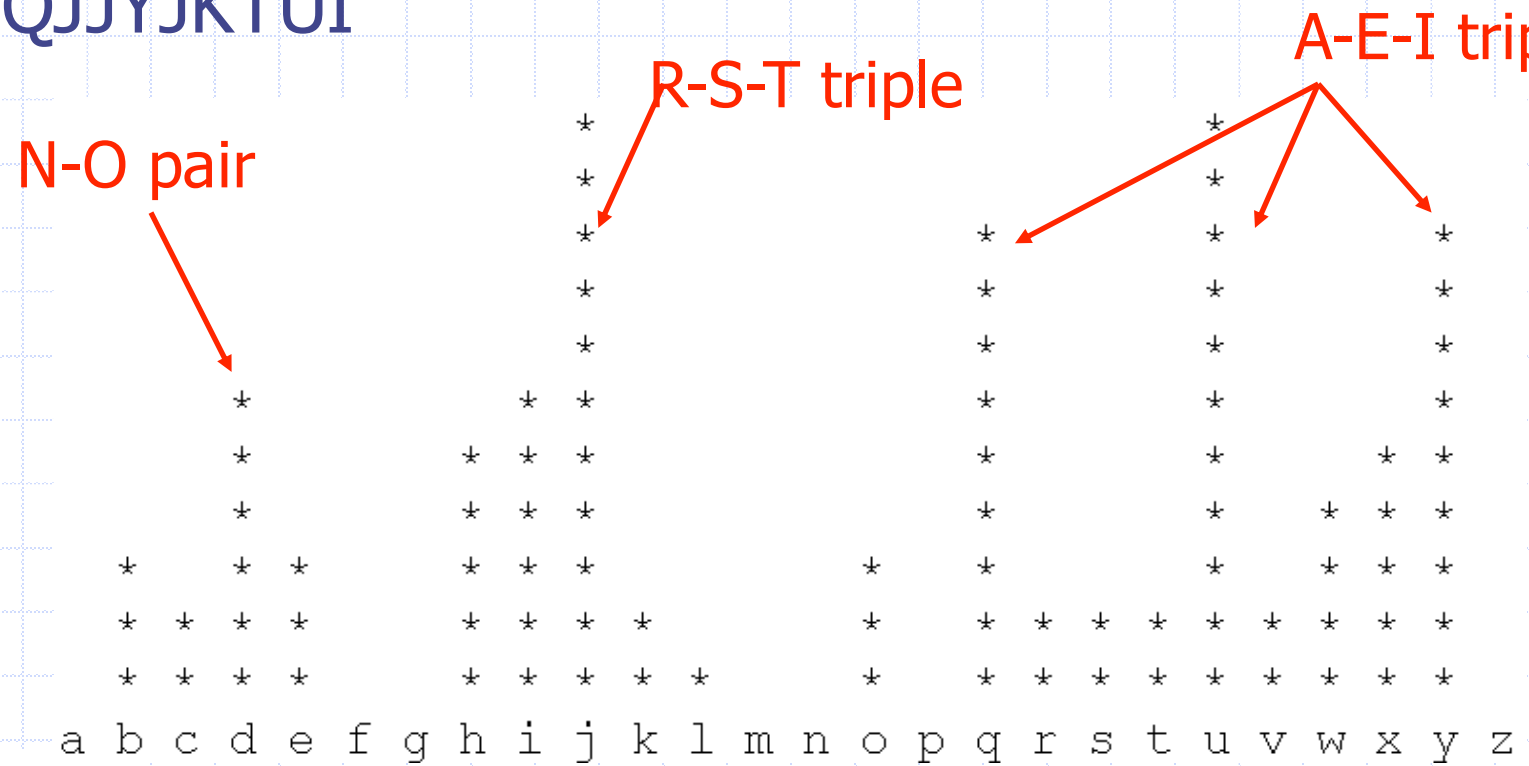
- Calculate letter frequencies for ciphertext being analyzed
- Compare counts/plots against known values
- In particular look for common peaks and troughs
  - Peaks at: A-E-I spaced triple, N-O pair, R-S-T triple with U shape
  - Troughs at: JK, X-Z
- Key concept - monoalphabetic substitution does not change relative letter frequencies

Table of Common English Single, Double and Triple Letters

Single Letter	Double Letter	Triple Letter
E	TH	THE
T	HE	AND
R	IN	TIO
N	ER	ATI
I	RE	FOR
O	ON	THA
A	AN	TER
S	EN	RES

Example with Caesar Cipher

- given "JXU WHUQJUIJ TYISELUHO EV COWUDUHQJYED YI JXQJ Q XKCQD UYDW SQD QBJUH XYI BYVU RO QBJUHYDW XYI QJJYJKTUI"



Polyalphabetic Ciphers

- Might guess that one approach to improving security is to use multiple cipher alphabets, hence the name polyalphabetic ciphers
- Makes cryptanalysis harder since have more alphabets to guess and because flattens frequency distribution
- Use a key to select which alphabet is used for each letter of the message
  - ith letter of key specifies ith alphabet to use
- Use each alphabet in turn
- Repeat from start after end of key is reached

Polyalphabetic Ciphers

- What do you think of this? Discuss among yourselves.

But…

- Cryptanalysts have methods for determining the key length
  - E.g., if two identical sequences of plaintext occur at a distance that is an integer multiple of the key length, then their ciphertext will be identical
  - Ex: key: DECEPTIVEDECEPTIVEDECEPTIVE

    Plaintext: WEAREDISCOVEREDSAVEYOURSELF

    Ciphertext: ZICVTWQNGRZGVTWAVZHCQYGLMGJ
- Once you have key length, cracking this is just cracking multiple monoalphabetic ciphers

OK...

- So if the key length is an issue (makes it easy to do frequency analysis), how can we get rid of this problem?

Book Cipher

- If key length is the issue with polyalphabetic cipher, at limit want as many alphabets as letters in message (but how to transfer such a key if it's truly random?)
- Book cipher: create key as long as a message by using words from a book to specify the translation alphabets
- Key used is then the book and page and paragraph to start from
- British used this some in WWII (called them poem codes)
  - Big problem

Problems with Book Cipher

- Same language characteristics are used by the key as the message
  - i.e., a key of 'E' will be used more often than a 'T' etc, hence an 'E' encrypted with a key of 'E' occurs with probability $(0.1275)^2 = 0.01663$, about twice as often as a 'T' encrypted with a key of 'T'
- Have to use larger frequency table, but they exist
- Given sufficient ciphertext this can be broken

- BUT, if a truly random key as long as the message is used, the cipher is **provably** unbreakable
  - Called a *One-Time Pad*

One-Time Pad

- A true solution: Choose a random key as long as the message itself
  - This reveals nothing statistically about the plaintext message. This lack of information about plaintext means that a one-time pad is unbreakable.
  - Put another way, for a given ciphertext, any plaintext of the same length as the message is a possible plaintext corresponding to that ciphertext!

One-Time Pad

- Practical considerations
  - Sender and receiver must be in possession of, and protect, the random key. If the receiver loses the key, they will have no way to reconstruct the plaintext.
  - Can only use a given key once, since if used even as few as two times, cryptanalysis reduces to frequency analysis on digraphs
    - That is, XOR of two ciphertexts becomes XOR of two plaintexts, which is breakable (how?)

One-Time Pad

- Practical considerations
  - Rarely used in practice (often no point in using it, since key is as long as the message)
    - But once both parties have key, can transmit many messages (until sum of lengths reaches length of key)
  - Implementation issues have also led to one-time pad systems being broken

Transposition Ciphers

- Also known as **permutation** ciphers
- Core idea: hide the message by rearranging the letter order without altering the actual letters used
- Can recognize these since have the same frequency distribution as the original text
- Very Simple Example: Mirror Cipher (write message backwards). Obviously not very secure
  - But what about mirror image in Russian?!

Cracking Transposition Ciphers

- Cracking transposition ciphers involves educated guessing with much trial and error
- BUT, there is software that will do a lot of this stuff for you (and it's out there and freely available)
- Bottom line, neither substitution nor transposition ciphers are secure (with the exception, of course, of a well-implemented one-time pad).

Increasing Cipher Security

- Ciphers based only on substitutions or only on transpositions are not secure
- Several ciphers in succession might seem to make cryptanalysis more difficult, but:
  - two substitutions are really only one more complex substitution
  - two transpositions are really only one more complex transposition

Increasing Cipher Security

- A sequence consisting of substitution followed by a transposition, followed buy a substutions, etc., however, makes a new much harder cipher
  - We call these product ciphers
  - We'll have more to say about these
    - Since DES and AES are product ciphers

Steganography

- An alternative to encryption
- Hides existence of message
  - using only a subset of letters/words in a longer message marked in some way
  - using invisible ink
  - hiding info in LSB in graphic image or sound file
- Has drawbacks
  - high overhead to hide relatively few info bits
  - If adversary realizes you're using steganography, you're usually sunk