Basics:
Cryptography is the process of converting plaintext into ciphertext
Plaintext: readable text
Ciphertext: unreadable/encrypted text
Used to hide information from unauthorized users
Decryption is the process of converting ciphertext back to plaintext
Two pieces of information: encryption algorithm, and encryption key
If you have the key, you can decrypt the ciphertext. The key must be kept secret.

History:
Substitution Cipher: Replaces one letter with another letter based on some key
Julius Caesar's Cipher: key value is right shift by some amount
Cryptanalysis studies the process of breaking encryption algorithms
When a new encryption algorithm is developed, cryptanalysts study and try to break it

Zimmerman Telegram: encrypted telegram from foreign secretary of the German Empire to the German ambassador in Mexico, to propose an alliance between Germany and Mexico in WW1
Intercepted and decrypted by the British
Pivotal in US entering WW1
Enigma: used by Germans, replaced letters as they were typed. Substiutions were computed using a key and a set of switches and rotors.

Use of Cryptography:
Confidentiality - only sender and intended receiver should "understand" message contents
Message Integrity: sender receiver want to ensure message not altered without detection
End Point Authentication: Send, receiver want to confirm identity of each other
Non Repudiation: ensuring that the sender actually sent the message

Language of Cryptography:
M: Plaintext message
K_a(m) is ciphertext, encrypted with key K_a
m = K_b(K_a(m))

Simple Encryption Scheme
Substitution Cipher: substituting one thing for another
MonoAlphabetic Cipher: substitute one letter for another

ABCDEFGHIJKLMNOPQRSTUVWXYZ

Left Shift POLY 3: MLIV

Poly-alphabetic Encryption: Vignere
Changes for everytime you go to the next letter. A method of encrypting alphabetic text using a series of Caesar ciphers with different shift values.

1. Choose a keyword and repeat it so that it matches the length of the message you want to encrypt
2. Create a table, consisted of the alphabet written out 26 times, each row shift one letter to the left
3. Encrypt: for each letter in the plaintext, find the corresponding letter in the keyword. Locat the row in the vigenere table corresponding to the keyword lettter and the column corresponding to the plaintext letter. The letter at the intersection is the ciphertext letter, according to some sources.
4. Decrypt: find the corresponding keyword letter and locate the ciphertext letter int he corresponding row of the Vignere table. The letter in the first column of that row is the original plaintext letter.

Example:
m: hello
K_a: KEY
H + K
E + E
L + Y
L + K
O + E

Vernam - Perfect Substitution Cipher
If we use Vignere with keylength as long as the plaintext then cryptanalysis will become very difficult
If we change keyevery time we encrypt then cryptanalyst's job becomes even more difficult.
One time pad or Vernam Cipher.
How do we get such long keys?
A large book shared by transmitter and receiver
Initial key followed by previous messages themselves
Random number sequence based on common shared and secret seed
Such a cipher is difficult to break but not very practical.
Also called a "one time pad"

Cipher-text only attack: Search through all keys and diferentiate between plaintext/gibberish, or statistical analysis
Known Plaintext Attack: some plaintext is owned by attacker corresponding to some ciphertext
Chosen-plaintext attack: The attacker gets ciphertext from some chosen plaintext

Computational Effort Required: time (primitive operations required), computational time required for attack. Memory - amount of storage needed. Data - amount of captured data required to complete the attack.

Types of Cryptography:
Public Key Cryptography: Involves the use of two keys. Key A is used to encrypt, and Key B is used to decrypt. Key A cannot be used to decrypt.
Symmetric Key Cryptography: Involves the use of one key to both encrypt and decrypt.
Hash Functions: One time, goes through a hash function, and get something of a fixed length out of it.

Confusion is changes in the key should affect many parts in the ciphertext
Diffusion: Changing one character in the plaintext will result in multiple changes throughout the ciphertext

Symmetric Key Cryptography: Same key used for encryption and decryption
Stream Ciphers: Encrypt one bit at a time
Block Ciphers: Break plaintext message into equal-size blocks, encrypt each block as a unit
Message to be encrypted is processed in blocks of k bits (e.g., 64-bit blocks)
1 to 1 mapping is used to map k-bit block of plaintext to k-bit block of ciphertext
Example: k = 3
input output
000 110
001 111
010 101
with k = 3, there are 2^3 possibilities (8), 2^3! permutations of the 3-bit inputs
Problem: As k grows bigger, a table approach requires 2^k entries. There must be a function that simulates a randomly permuted table.
Another problem: if you have

Cipher Block Chaining: Generate its own random numbers
Have encryption of current block depend on result of previous block
How do we encrypt first block:
Initialization vector: random block = c(0)
Changes IV for each message (or session)
-Guarantees that even if the same message is sent repeatedly, the ciphertext will be completely different each time
If you have the same thing being encrypted multiple times, you won't get the same answer (problem with Block Cipher)

Algorithm:
CT1 = Encrypt(IV XOR PT1)

CT2 = encrypt (PT1 XOR CT1)
Keep going until ciphertext is encrypted

AES: Newest symmetric key NIST standard, replace DES
Process data in 128 bit blocks
128, 192, or 256 bit keys
Brute force decryption takes 10 billion years for AES

Public Key Cryptography
Issues of Symmetric Key Cryptography: requires sender and receiver know shared key. How do we agree on key? Secretly sharing keys is very difficult.
Public Key Cryptography (Asymmetric): radically different approach, sender receiver do not share secret key, public key known to all, private key known only to receiver

Modular Arithmetic:
x mod n: remainder of x when divided by n
(a + b) mod n = [(a mod n) + (b mod n)] mod n
(a - b) mod n = [(a mod n) - (b mod n)] mod n
(a *b) mod n = [(a mod n)* (b mod n)] mod n
(a *b* c)mod n = [(a mod n)(b mod n)(c mod n)] mod n

RSA: Geting Ready
A message is a bit pattern
A bit pattern can be uniquely represented by an integer number
Thus encrypting a message is equivalent to encrypting a number
Ex:
m = 10010001. Represented by the decimal number 145
To encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).

Creating a Public/Private Keypair:

1. Choose two large prime numbers, p, q (eg 1024 bits each)
2. Comput n = pq, phi = (p -1)(q -1)
3. Choose e (e < phi) that has no common factors with phi (e, phi are relatively prime)
4. Choose d such that ed - 1 is exactly divisible by phi
   (ed mod phi = 1; or d = e^-1 mod phi)

Ex:
p = 5, q = 7
n = 35
phi = 24

e = 5
d = 29 (so ed - 1 is exactly divisible by phi)

Given (n, e) and (n,d) as computed, encrypt m:
c = m^e mod n

Decrypting: m = c^d mod n

RSA is slow to generate keys, even by today's standards.

Session Keys:
Exponentiation is computationally intensive
DES is at least 100 times faster than RSA

Diffie-Hellamn
Allows 2 entities to agree on shared key, but does not provide encryption
n is a large prime; g is a number less than n
n and g are made public

Alice:
a = secret
A = g^a mod n (this is the public key)
K = B^a mod n

Bob:
b = secret
B = g^b mod n
K = A^b mod n

Both Ks on both sides are equal