Crypto Review

APNIC

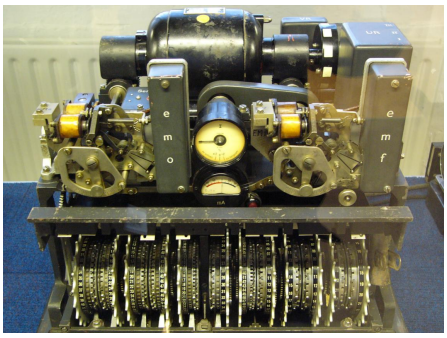
Overview

- What is Cryptography?
- Symmetric Key Cryptography
- Asymmetric Key Cryptography
- Block and Stream Cipher
- Digital Signature and Message Digest

APNIC

Cryptography

- Cryptography is everywhere



German Lorenz cipher machine

Cryptography

- Cryptography deals with creating documents that can be shared secretly over public communication channels
- Other terms closely associated
  - Cryptanalysis = code breaking
  - Cryptology
    - Kryptos (hidden or secret) and Logos (description) = secret speech / communication
    - combination of cryptography and cryptanalysis
- Cryptography is a function of plaintext and a cryptographic key

$$C = F(P,k)$$

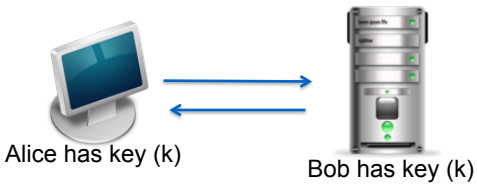
Notation:
Plaintext (P)
Ciphertext (C)
Cryptographic Key (k)

Typical Scenario

- Alice wants to send a “secret” message to Bob
- What are the possible problems?
  - Data can be intercepted
- What are the ways to intercept this message?
- How to conceal the message?
  - Encryption

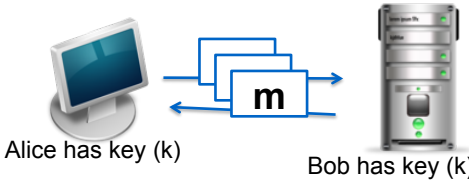
Crypto Core

- Secure key establishment



- Secure communication

Confidentiality and integrity



Source: Dan Boneh, Stanford

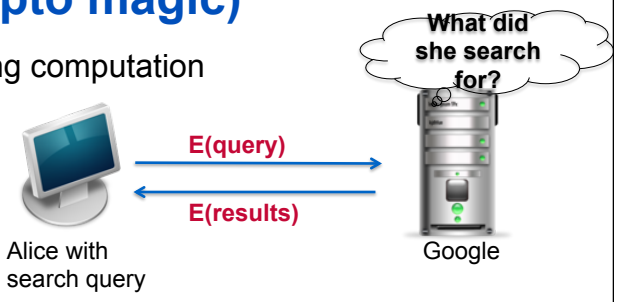
It can do much more

- Digital Signatures
- Anonymous communication
- Anonymous digital cash
  - Spending a digital coin without anyone knowing my identity
  - Buy online anonymously?
- Elections and private auctions
  - Finding the winner without actually knowing individual votes (privacy)

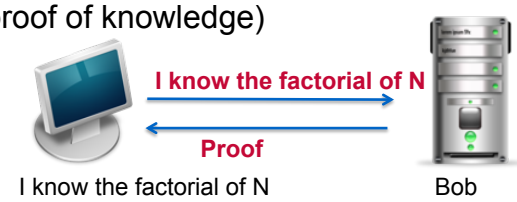
Source: Dan Boneh, Stanford

Other uses are also theoretically possible (Crypto magic)

- Privately outsourcing computation



- Zero knowledge (proof of knowledge)



Source: Dan Boneh, Stanford

History: Ciphers

- Substitution cipher
  - involves replacing an alphabet with another character of the same alphabet set
  - Can be mono-alphabetic (single set for substitution) or poly-alphabetic system (multiple alphabetic sets)
- Example:
  - Caesar cipher, a mono-alphabetic system in which each character is replaced by the third character in succession
  - Vigenere cipher, a poly-alphabetic cipher that uses a 26x26 table of characters

How to Break a Substitution Cipher

UKBYBIPOUZBCUFEEBORUKBYBHOBBRFESPVKBWFOFERVNBCVBZPRUBOFERVNBCVBPCYYFVUFO

FEIKNWFRFIKJNUPWRFIPOUNVNIPUBRNCUKBEFWWFDNCHXCYBOHOPYXPUBNCUBOYNRVNIWN

CPOJIOFHOPZRVFZIXUBORJRUBZRBCHNCBBONCHRJZSFWNVRJRUBZRPCYZPUKBZPUNVPWPCYVF

ZIXUPUNFCPWRVNBCVBRPYYNUNFCPWWJUKBYBIPOUZBCUIPOUNVNIPUBRNCHOPYXPUBNCUB

OYNRVNIWNCPOJIOFHOPZRNCRVNBCUNENVVFZIXUNCHPCYVFZIXUPUNFCPWZPUKBZPUNVR

(1) Use frequency of the English letters
e = 12.7%
t = 9.1 %
a = 8.1%

(2) Use frequency of pairs of letters
he, in, an, th

In the example,
B appeared 36 times, **U** 33 times, and **P** 32 times
NC appeared 11 times, **PU** 10 times
UKB appeared 6 times

Source: Dan Boneh, Stanford

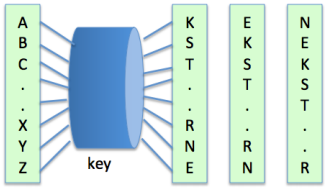
Transposition Cipher

- No letters are replaced, they are just rearranged.
- Rail Fence Cipher – another kind of transposition cipher in which the words are spelled out as if they were a rail fence.

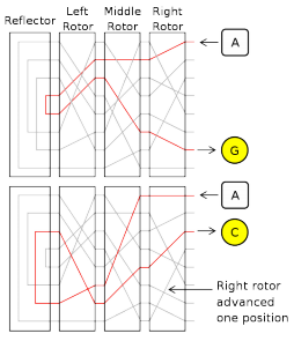
```
T...U...B...N...J...E...E...E...Y..
.H.Q.I.K.R.W.F.X.U.P.D.V.R.H.L.Z.D.G.
..E...C...O...O...M...O...T...A...O
```


History: Rotor Machines (1870-1943)

- Hebern machine – single rotor



- Enigma - 3-5 rotors



Source: Wikipedia (image)

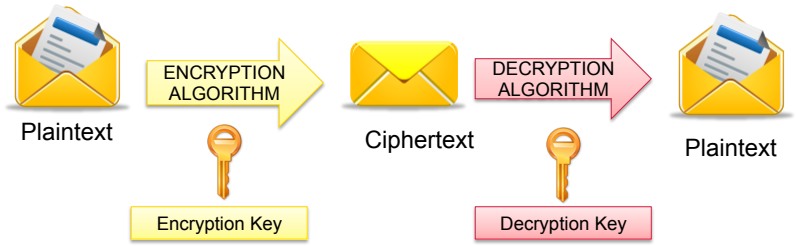
Modern Crypto Algorithms

- specifies the mathematical transformation that is performed on data to encrypt/decrypt
- Crypto algorithm is NOT proprietary
- Analyzed by public community to show that there are no serious weaknesses
- Explicitly designed for encryption

Encryption

- process of transforming plaintext to ciphertext using a cryptographic key
- Used all around us
  - In Application Layer – used in secure email, database sessions, and messaging
  - In session layer – using Secure Socket Layer (SSL) or Transport Layer Security (TLS)
  - In the Network Layer – using protocols such as IPsec
- Benefits of good encryption algorithm:
  - Resistant to cryptographic attack
  - They support variable and long key lengths and scalability
  - They create an avalanche effect
  - No export or import restrictions

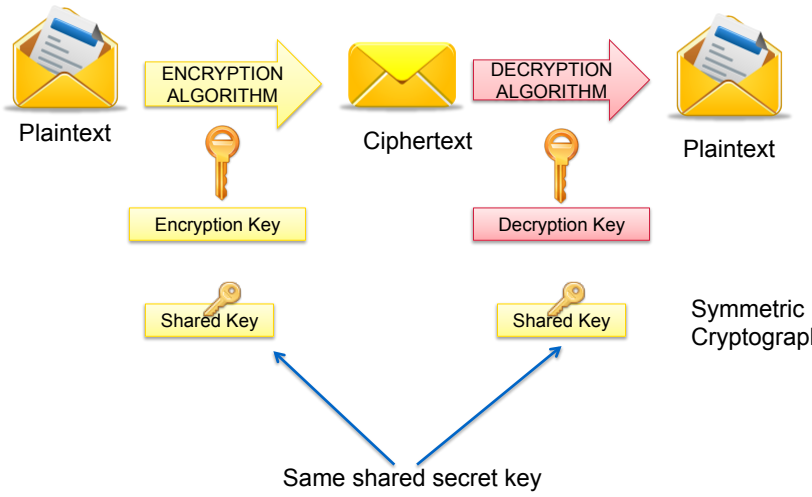
Encryption and Decryption



Symmetric Key Algorithm

- Uses a single key to both encrypt and decrypt information
- Also known as a secret-key algorithm
  - The key must be kept a “secret” to maintain security
  - This key is also known as a private key
- Follows the more traditional form of cryptography with key lengths ranging from 40 to 256 bits.
- Examples:
  - DES, 3DES, AES, RC4, RC6, Blowfish

Symmetric Encryption



Symmetric Key Algorithm

Symmetric Algorithm	Key Size
DES	56-bit keys
Triple DES (3DES)	112-bit and 168-bit keys
AES	128, 192, and 256-bit keys
IDEA	128-bit keys
RC2	40 and 64-bit keys
RC4	1 to 256-bit keys
RC5	0 to 2040-bit keys
RC6	128, 192, and 256-bit keys
Blowfish	32 to 448-bit keys

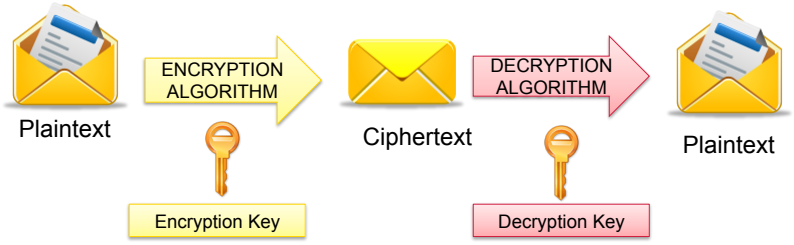
Note:
Longer keys are more difficult to crack, but more computationally expensive.

Data Encryption Standard (DES)

- Developed by IBM for the US government in 1973-1974, and approved in Nov 1976.
- Based on Horst Feistel's Lucifer cipher
- block cipher using shared key encryption, 56-bit key length
- Block size: 64 bits

DES: Illustration

64-bit blocks of input text



56-bit keys +
8 bits parity

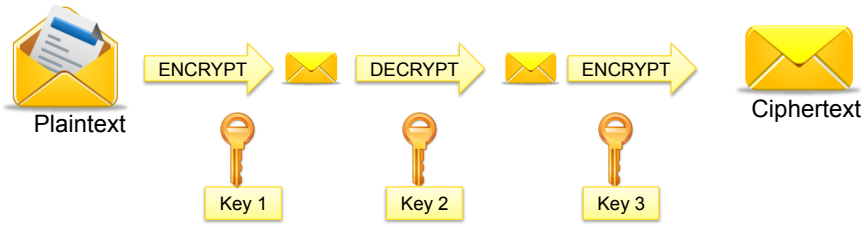
Triple DES

- 3DES (Triple DES) – a block cipher that applies DES three times to each data block
- Uses a key bundle comprising of three DES keys (K1, K2, K3), each with 56 bits excluding parity.
- DES encrypts with K1, decrypts with K2, then encrypts with K3

$$C_i = E_{K3}(D_{K2}(E_{K1}(P_i)))$$

- Disadvantage: very slow

3DES: Illustration



- Note:
  - If Key1 = Key2 = Key3, this is similar to DES
  - Usually, Key1 = Key3

Advanced Encryption Standard (AES)

- Published in November 2001
- Symmetric block cipher
- Has a fixed block size of 128 bits
- Has a key size of 128, 192, or 256 bits
- Based on Rijndael cipher which was developed by Joan Daemen and Vincent Rijmen
- Better suited for high-throughput, low latency environments

Rivest Cipher

- Chosen for speed and variable-key length capabilities
- Designed mostly by Ronald Rivest
- Each of the algorithms have different uses

RC Algorithm	Description
RC2	Variable key-sized cipher used as a drop in replacement for DES
RC4	Variable key sized stream cipher; Often used in file encryption and secure communications (SSL)
RC5	Variable block size and variable key length; uses 64-bit block size; Fast, replacement for DES
RC6	Block cipher based on RC5, meets AES requirement

Block Cipher

- Transforms a fixed-length block of plain text into a block of ciphertext
  - operate on a pre-determined block of bits (one byte, one word, 512 bytes, so forth), mixing key data in with the message data in a variety of different ways
- Common block ciphers:
  - DES and 3DES (in ECB and CBC mode)
  - Skipjack
  - Blowfish
  - RSA
  - AES
  - IDEA
  - SAFER

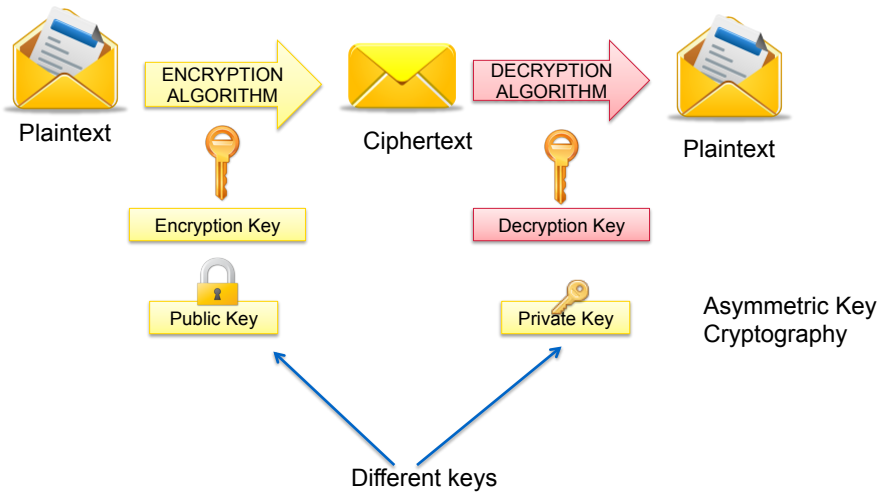
Stream Cipher

- Use smaller units of plaintext than what are used with block ciphers.
  - encrypts bits of the message at a time
  - typically bit-wise
  - They perform some operation (typically an exclusive OR) with one of these key bits and one of the message bits
- They either have a very long key (that eventually repeats) or a reusable key that generates a repeatable but seemingly random string of bits.
- Common stream ciphers:
  - RC4
  - DES and 3DES (running OFB or CFB mode)
  - SEAL

Asymmetric Key Algorithm

- Also called public-key cryptography
  - Keep private key private
  - Anyone can see public key
- separate keys for encryption and decryption (public and private key pairs)
- Examples:
  - RSA, DSA, Diffie-Hellman, ElGamal, PKCS

Asymmetric Encryption



Asymmetric Key Algorithms

- RSA – the first and still most common implementation
- DSA – specified in NIST's Digital Signature Standard (DSS), provides digital signature capability for authentication of messages
- Diffie-Hellman – used for secret key exchange only, and not for authentication or digital signature
- ElGamal – similar to Diffie-Hellman and used for key exchange
- PKCS – set of interoperable standards and guidelines

Symmetric vs. Asymmetric Key

Symmetric	Asymmetric
generally fast Same key for both encryption and decryption	Can be 1000 times slower Uses two different keys (public and private) Decryption key cannot be calculated from the encryption key Key lengths: 512 to 4096 bits Used in low-volume

Hash Functions

- produces a condensed representation of a message (hashing)
- The fixed-length output is called the hash or message digest
- A hash function takes an input message of arbitrary length and outputs fixed-length code.
  - Given x, we can compute the value f(x).
  - Given f(x), it is hard to get the value of x.
- A form of signature that uniquely represents the data
  - Collision-free
- Uses:
  - Verifying file integrity - if the hash changes, it means the data is either compromised or altered in transit.
  - Digitally signing documents
  - Hashing passwords

Hash Functions

- Message Digest (MD) Algorithm
  - Outputs a 128-bit fingerprint of an arbitrary-length input
  - MD4 is obsolete, MD5 is widely-used
- Secure Hash Algorithm (SHA)
  - SHA-1 produces a 160-bit message digest similar to MD5
  - Widely-used on security applications (TLS, SSL, PGP, SSH, S/MIME, IPsec)
  - SHA-256, SHA-384, SHA-512 are also commonly used, which can produce hash values that are 256, 384, and 512-bits respectively
- RIPEMD
  - Derived from MD4, but performs like SHA
  - RIPEMD-160 is the most popular version

Digital Signature

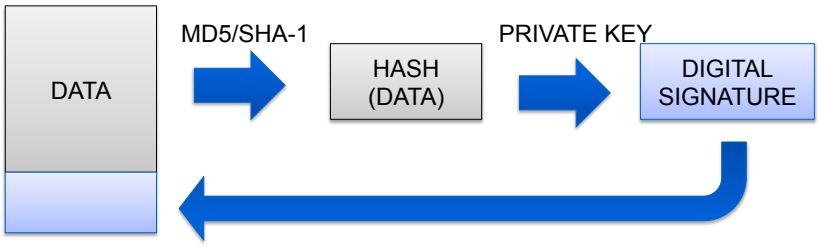
- A digital signature is a message appended to a packet
- The sender encrypts message with own private key instead of encrypting with intended receiver's public key
- The receiver of the packet uses the sender's public key to verify the signature.
- Used to prove the identity of the sender and the integrity of the packet

Digital Signature

- Two common public-key digital signature techniques:
  - RSA (Rivest, Shamir, Adelman)
  - DSS (Digital Signature Standard)
- Used in a lot of things:
  - Email, software distribution, electronic funds transfer, etc
- A common way to implement is to use a hashing algorithm to get the message digest of the data, then use an algorithm to sign the message

Digital Signature Process

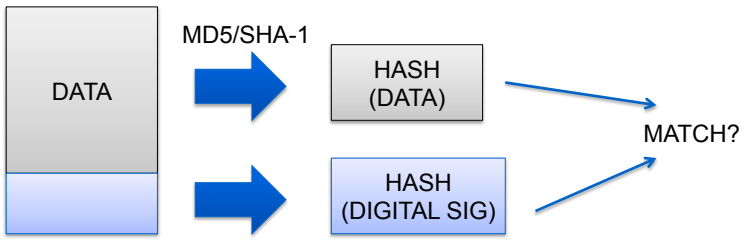
1. Hash the data using one of the supported hashing algorithms (MD5, SHA-1, SHA-256)
2. Encrypt the hashed data using the sender's private key
3. Append the signature (and a copy of the sender's public key) to the end of the data that was signed)



APNIC

Signature Verification Process

1. Hash the original data using the same hashing algorithm
2. Decrypt the digital signature using the sender's public key. All digital signatures contain a copy of the signer's public key
3. Compare the results of the hashing and the decryption. If the values match then the signature is verified. If the values do not match, then the data or signature was probably modified.



APNIC

Message Authentication Code

- Message authentication code provides
  - Integrity (checks that data has not been altered)
  - Authenticity (verifies the origin of data)
- In the sender side, the message is passed through a MAC algorithm to get a MAC (also called Tag)
- In the receiver side, the message is passed through the same algorithm. The output is compared with the received tag and should match
- Sender and receiver uses the same secret key
- Hash-based Message Authentication Code (RFC2104)
  - Uses hash function to generate the MAC
  - "HMACs are less affected by collisions than their underlying hashing algorithms alone."

Questions