Algorithms in Systems Engineering
IE172

Lecture 27

Dr. Ted Ralphs

References for Today's Lecture

- Required reading
  - Section 8.3
- References
  - CLRS Chapter 31
  - Koblitz, *A Course in Number Theory and Cryptography*, Second Edition (1999).

Cryptography

- Cryptography is the study of methods for sending messages in an encoded form that can (hopefully) only be interpreted by the intended recipient.
- The original message is said to be in *plaintext* and the encoded message is said to be in *ciphertext*.
- All commonly used cryptographic methods are based on specifying a one-to-one function that transforms plaintext into ciphertext.
- To get back the original message, we simply apply the inverse transformation.
- To put it more precisely, let $\mathcal{P}$ be the set of all plaintext messages and $\mathcal{C}$ be the set of all encrypted messages.
- A *crytosystem* is a one-to-one mapping $f : \mathcal{P} \rightarrow \mathcal{C}$, whose inverse maps $\mathcal{C}$ back to $\mathcal{P}$.
- Note that in many cryptosystems, we have $\mathcal{P} = \mathcal{C}$.

Message Units

- Let's assume that our plaintext message is composed from an alphabet of N characters.
- Most cryptosystems work by dividing the original message into *message units*, which are then individually enciphered.
- A message unit is typically defined to be a block of k letters for some positive k.
- For ease of defining the transformation, we can convert each message unit to a unique integer by interpreting it as a k-digit number base N.
- We can then make the simplifying assumption that the message units consist simply of integers from 0 to $N^k - 1$.

A Simple Cryptosystem

- Let's first consider message units of length 1.
- A cryptosystem then consists essentially of specifying a *permutation* of the letters of the alphabet (we may or may not include the spaces also).
- To keep things simple, we want to be able to easily encrypt and decrypt messages.
- The simplest transformation is $P + b \mod N$, where P is the message unit to be encrypted and b is a positive integer called the *shift*.
- b is also known as the *encoding key* because it is the only information needed to compute the encoding function.
- What is the inverse transformation?
- How easy is it to encode and decode?
- How easy is it to break this code?

Affine Transformations

- We can improve the situation by using an *affine transformation* $aP + b \mod N$ instead, where a and b are both positive integers.
- Note that for this to work, a and N must be relatively prime.
- The pair (a, b) is called the *encoding key*.
- Now what is the inverse transformation?
- How easy is it to encode and decode?
- How easy is it to break this code?

Larger Message Units

- Another way to improve our simple cryptosystem is to use larger message units.
- Let's suppose we use message units of length k.
- How does this improve the situation?
- How easy is it to encode and decode now?

Issues

- For a cryptosystem to be useful, it has to be possible to easily encode and decode.
- In the examples we have seen so far, the algorithm for encoding and decoding is the same, but with different keys.
- With an affine transformation, if the encoding key is (a, b), the decoding key is $(a^{-1} \mod N^k, -a^{-1}b \mod N^k)$.
- Given the encoding key, we can derive the decoding key by the Euclidean Algorithm in $O(\log^3(N^k))$ time.
- What is the problem with this?

Public Key Encryption

- Until about 25 years ago, all known cryptosystems had the property that if you knew the encoding key, you could easily derive the decoding key.
- This creates problems when trying to send an encrypted message to someone without prior arrangement.
- *Public key encryption* is an attempt to overcome this shortcoming.
- Public key systems are based on the concept of a *trapdoor function*.
- A *trapdoor function* is one which is easy to compute but "difficult" to invert without additional information.
- A *one-way function* is one which is easy to compute but "difficult" to invert even with additional information.
- Using a trapdoor function to do the encoding makes it difficult to discover the decoding key from the encoding key.
- What are the advantages of this?

Public Key Cryptosystems

- In public key encryption, there is an encoding key K_E and a decoding key K_D.
- These keys allow the computation of the encoding function f_E and the decoding function f_D.
- Note that we must always have $f_D(f_E(P)) = f_E(f_D(P)) = P$.
- How a public key system works.
  - Bob makes his encoding key K_E (also called his *public key)* publicly available, but keeps his decoding key private.
  - If Sally wants to send Bob an encoded message that only he can read, she encodes it using his public key.
  - Upon receiving the message, Bob decodes it using his private key.
  - If Bob wants to send Sally a message, he simply uses the same procedure to encode his message using her public key.
- Unlike a traditional cryptosystem, the ability to decode is not revealed by encoding.
- The advantage is that this allows complete strangers to send encrypted messages without prior arrangement.

Digital Signatures

- A seemingly large drawback of the system we have so far discussed is that there is no way for the receiver to be sure of the message's origin.
- This is where digital signatures come in.
- How a digital signature works
  - If Bob wishes to digitally sign a message to Sally, he encrypts the message using his private key and then sends both the original and encrypted messages.
  - Sally then decodes the encoded version of the message using Bob's public key and checks whether it matches the original message.
  - If so, then she knows that Bob must have signed it.
- How would we both encrypt and digitally sign a message?

Difficulties with Public Key Encryption

- The encoding function is typically more difficult to compute.
  - For long messages, the encoding time can be prohibitive.
  - One approach is to use public key encryption to encode a traditional encoding key.
  - Then send the rest of the message using the traditional key.
  - This is the approach used by most commercial software.
- A complete digital signature can be very large.
  - One approach to reducing the size is to apply a *one-way hash function* to the original message before computing the signature.
  - Roughly speaking, a *one-way hash function* f is mapping that
    - reduces a large message P to a much smaller one $H = f(P)$, and
    - for which it is difficult to determine a P' such that $f(P') = H$.
  - The message can then be verified by decoding the signature, applying the same hash function to the received message and comparing the results.

Identification

- One further potential difficulty is that it is not really possible to positively identify someone using a digital signature.
- It is possible to determine that the person who signed a given message *is* the owner of the public key used to verify the authenticity.
- In fact, you still don't know that the person who gave you the key is who they say they are.
- One way to overcome this is to designate certain trusted authorities to digitally sign individual public keys.
- If an authority that you trust has signed someone's digital key, then you can be confident that they are who they say they are.
- Another possibility is to develop *trust webs* whereby individuals sign the keys of other individuals they know directly.
- This allows all the individuals each person knows to establish trust relationships, even if they don't know each other directly.