Secure communications

Eugeniy E. Mikhailov

The College of William & Mary

Lecture 23

Secure communication

The simple way:
You can try to exchange messages in secret, i.e. meet somewhere and make sure that no one listens to you. The problem is that you have to meet in person.

Most of the time, we do not want to go anywhere. We just want to exchange messages.
We will focus on encryption and decryption, i.e. making our messages unreadable for strangers (encryption) and converting it back to a readable form (decryption).

Substitution cypher (Caesar cypher)

For the whole message or message set, replace every letter with another letter.
For example: with a $\rightarrow$ z, b $\rightarrow$ w, l $\rightarrow$ e, m $\rightarrow$ a, and t $\rightarrow$ d

matlab $\rightarrow$ azdezw

This is a very old method (traced back to ancient Greeks):

Substitution cypher (Caesar cypher)

For the whole message or message set, replace every letter with another letter.
For example: with a $\rightarrow$ z, b $\rightarrow$ w, l $\rightarrow$ e, m $\rightarrow$ a, and t $\rightarrow$ d

matlab $\rightarrow$ azdezw

This is a very old method (traced back to ancient Greeks):

Do not use it in critical applications. To know why, read about *frequency analysis* or read Arthur Conan Doyle “The Adventure Of The Dancing Men” (1903)

One-time pad

Also, a substitution cypher but have the substitution *never* repeats, thus the name. Also, it requires the one-time pad to be completely *random*. This is a very strong (actually unbreakable when done properly) method, the problem is that the length of a substitution set should be equal to the length of the messages to transmit.

If above requirements are not satisfied, it can be broken. See how Soviet communication was compromised by the Venona project.

Things not to do

Never assume that your encryption algorithm will be unknown to enemies! Sometimes it is called, security though obscurity.
If enemies find it, you will have to change everything. Instead, make it a *secret key* dependent. If someone knows how your lock works, she still needs to find the key.

Symmetric cypher algorithms

Symmetric key algorithm. In simple way: decide on a key (a string of symbols) and use the same key for encryption and decryption.
This algorithm is easy to break if you key is short, but *very hard* if your key is longer than the message and you never reuse this key.
If your ever saw a password protected archive, then you saw this method in practice.

Main pit fall, you need to send/receive the key at some point, i.e. both parties need to meet.

Public (shared) key cryptography

In this case, you encrypt your message with one key (often publicly available) and decrypt with another one (which should be kept in secret).
The algorithms are based on numbers theory and the fact that some operations take very long time to do. For example, number decomposition to the primes.
You have a *public key* and encrypt the message
I have *private key* and can decipher this message.
Examples: web communications with hypertext transfer protocol secured **https** protocol, secure access to electronic mail servers (IMAP and POP) via transport layer security (TLS) or secure sockets layer (SSL) protocols.
Additional benefits are that with some protocols, you can sign your message so there is the proof that it was signed by you and that it is not changed during the transmission.

Some security software

Rule #1: Never, I repeat never use closed source security software.
If they need to keep the algorithm in secret, it is not secure.
Good open source software.

- PGP - Pretty Good Privacy
- GPG - GNU Privacy Guard

Above program allow you to encrypt and decrypt with shared key your mail or files, sign text messages, make a checksum of binaries.
Some software allows a file system encryption, but look for the Rule # 1

Additional consideration

Authentication. How do you know that a person on the other end is not an impostor? This is very important, many encryption schemes fail if you have a man in the middle between communicating parties.

Privacy. What if you do not want to associated with a particular communications.

Does any one eavesdropping on your communication? Look for answers in quantum cryptography/exchange protocols.