Real-world examples

- Single machine
  - Unix/Linux permission bits
  - SELinux
- Mobile device
  - Android
- Organization assets
  - Active Directory

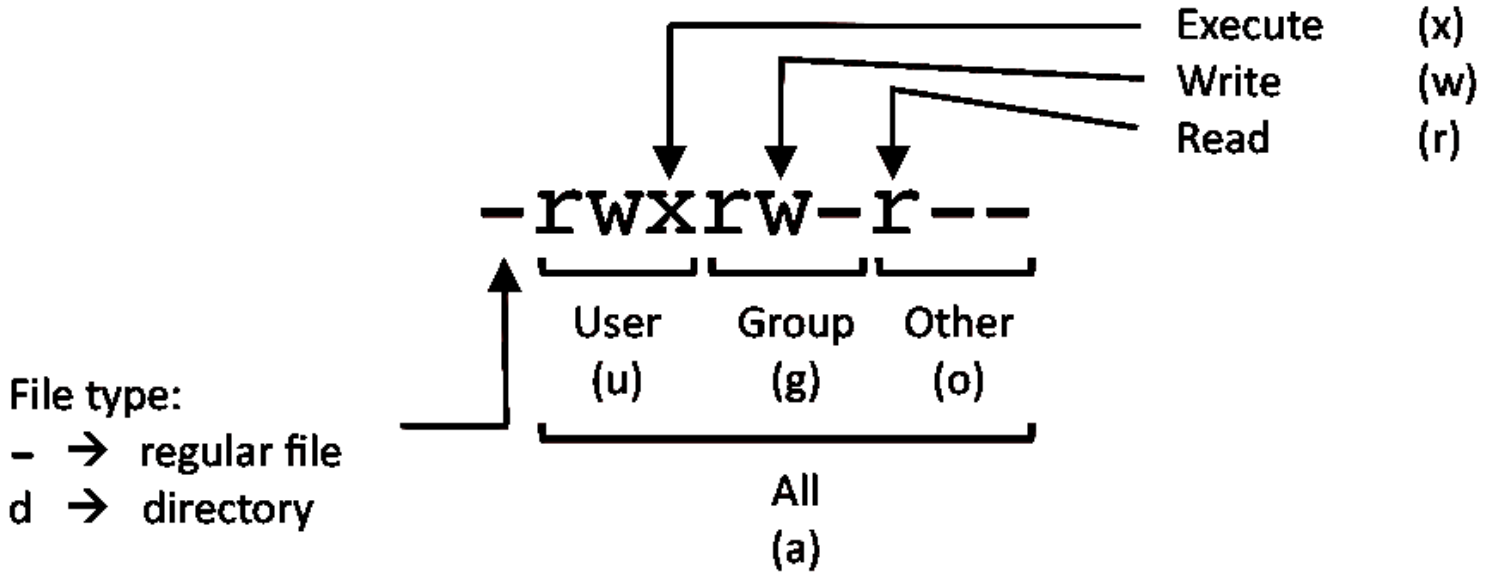
DAC & MAC

- Discretionary Access Control (DAC):
  - Provides access based on identity of the user
  - A subject with a certain access permission is capable of *passing* that permission (perhaps indirectly) on to any other subject.
  - Example: ***Unix user-group-other permission bits***
  - Anyone with access can propagate information
- Mandatory Access Control (MAC):
  - Admins creates a set of levels and each user is linked with a specific *access level*. He/she can access all the resources that are not greater than his/her access level.

Unix/Linux permission bits (DAC)

```
sssit@JavaTpoint: ~
sssit@JavaTpoint:~$ ls -g
total 52
drwxr-xr-x 2 sssit 4096 May 18 11:54 Desktop
drwx------ 4 sssit 4096 May 18 11:20 Disk1
drwxr-xr-x 2 sssit 4096 May 18 12:59 Documents
drwxr-xr-x 3 sssit 4096 May 11 17:55 Downloads
-rw-r--r-- 1 sssit 8445 May 12 04:23 examples.desktop
drwxr-xr-x 2 sssit 4096 May 12 04:27 Music
drwxr-xr-x 2 sssit 4096 May 18 12:55 Pictures
drwxr-xr-x 2 sssit 4096 May 12 04:27 Public
drwxr-xr-x 2 sssit 4096 May 12 04:27 Templates
drwxrwxr-x 2 sssit 4096 May 18 09:47 Untitled Folde
drwxr-xr-x 2 sssit 4096 May 12 04:27 Videos
sssit@JavaTpoint:~$
```

- Run ls -l in Linux



SELinux (MAC)

- Security-Enhanced Linux (SELinux) is an implementation of MAC in the Linux kernel, checking for allowed operations after DAC is checked.
  - Idea traced back to NSA project, GNU GPL in 2000
  - Enabled by default in Red Hat Enterprise Linux

- SELinux depends upon labels to match actions and policies.
  - Labels determine what is allowed.
  - Sockets, files, and processes all have labels.
  - SELinux decisions are based on labels assigned to these objects and the policy defining how they may interact.
- SELinux users and roles do not have to be related to the actual system users and roles
  - most of the real users share the same SELinux username

https://www.electronicdesign.com/embedded/selinux-101-what-you-should-know

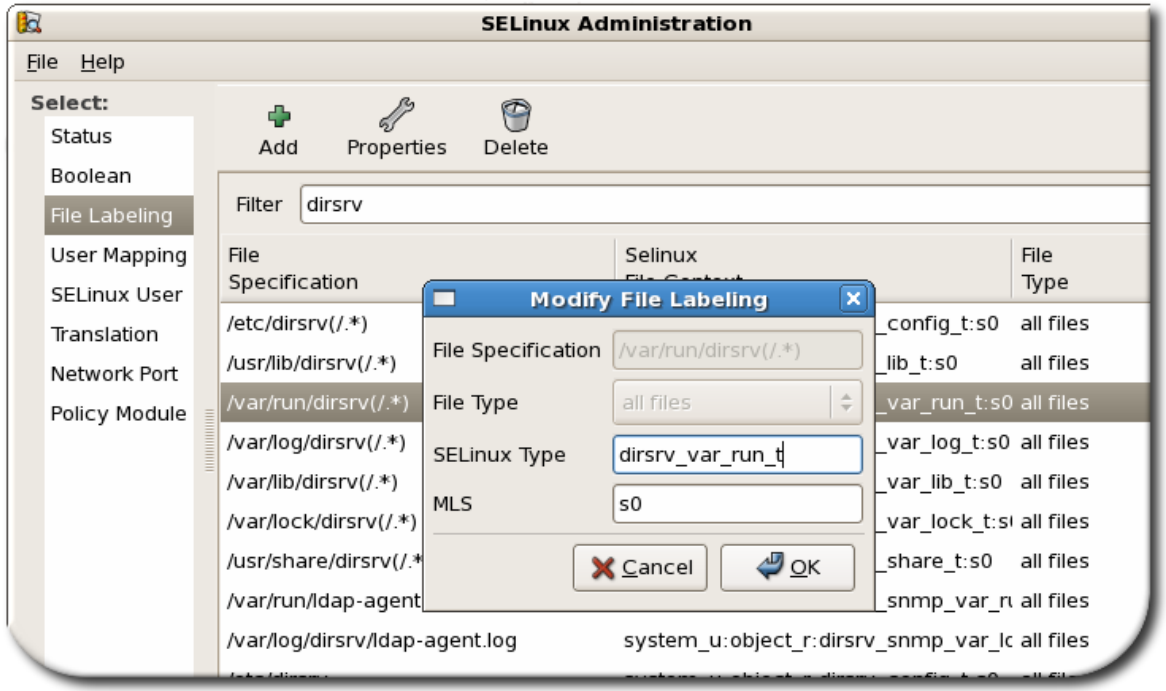
SELinux (Cond.)

- SELinux Label
  - *user:role:type:mls_level*

```
~]# ls -lZ /etc/file1
-rw-r--r--. root root unconfined_u:object_r:etc_t:s0   /etc/file1
```

- SELinux Policy
  - *allow domains types:classes permissions;*
  - *Domain* - A label for the process or set of processes
  - *Type* - A label for the object (e.g. file, socket) or set of objects.
  - *Class* - The kind of object (e.g. file, socket) being accessed.
  - *Permission* - The operation (e.g. read, write) being performed.

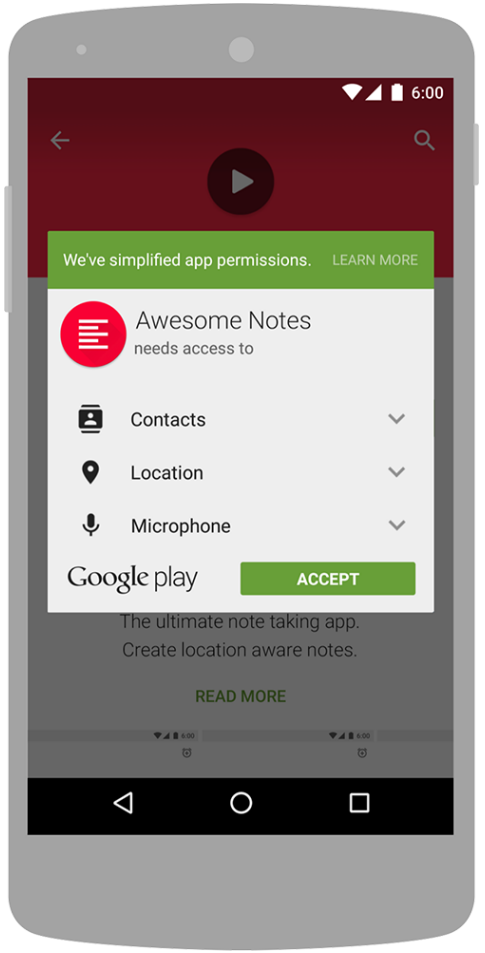
```
allow domain null_device:chr_file { open };
```



Configure file label

Android access control

- Each app runs in its own sandbox.
  - A unique Linux UID assigned for each app when installation
  - DAC is enforced: permission bits
  - Permissions are granted by user at install-time/run-time
- SELinux mode adopted by Android (since 4.3)
  - *Permissive* mode, in which permission denials are logged but not enforced.
  - *Enforcing* mode, in which permissions denials are both logged **and** enforced.
- Android 4.3: permissive => Android 4.4: partial enforcement => Android 5.5: everything enforcement

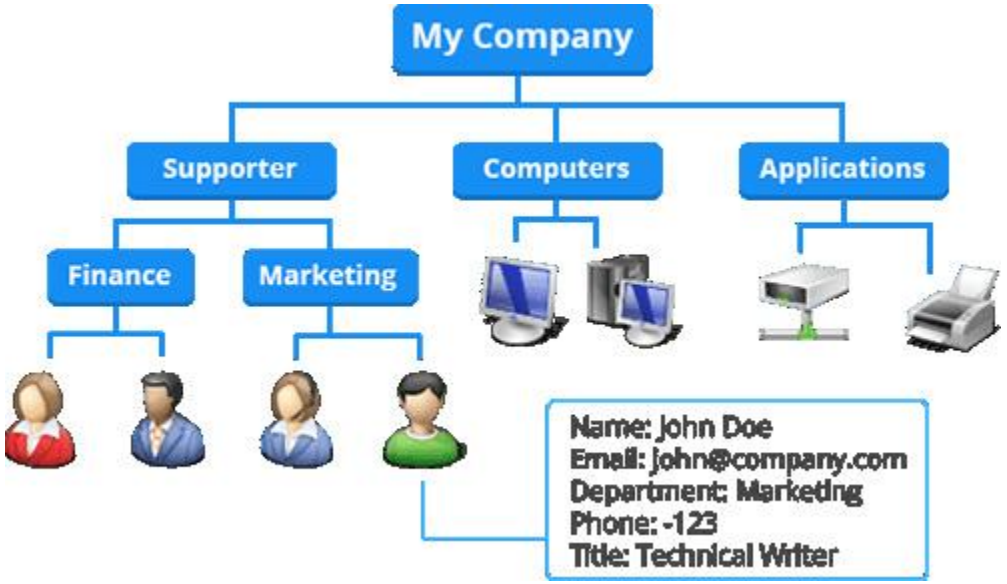


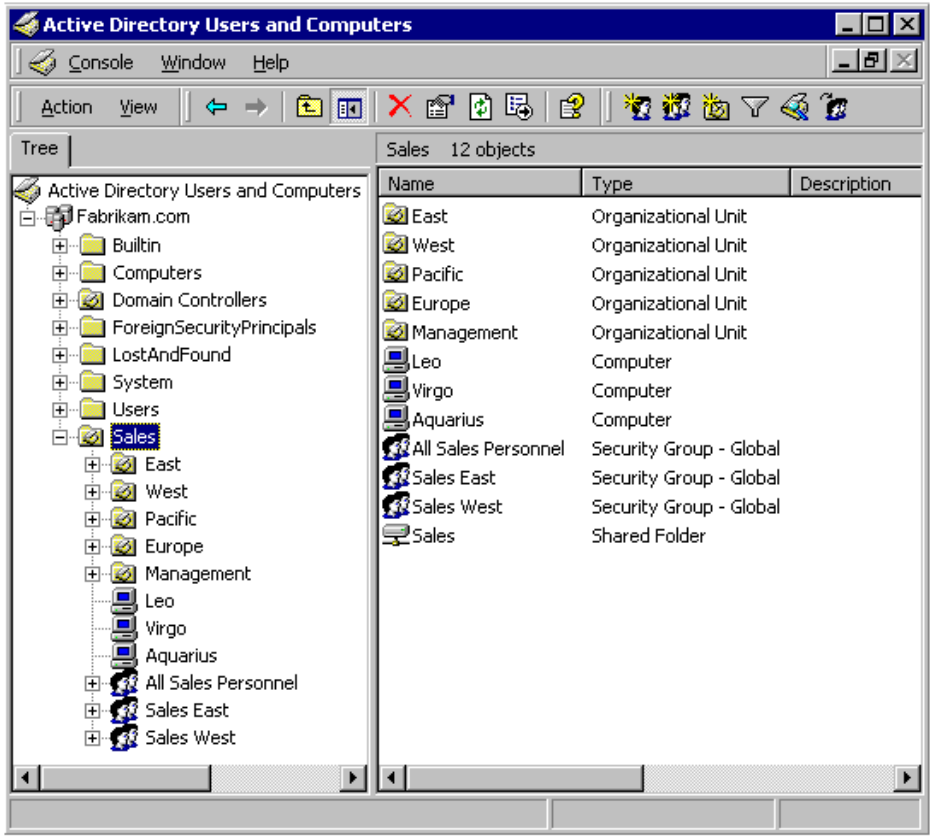
Install-time permissions

https://source.android.com/security/selinux/concepts

Active Directory

- Active Directory (AD) is a directory service that Microsoft developed for the Windows domain networks.
- Domain controller
  - Authentication and authorization





Summary

- Goal of access control
  - Limiting who can access what in what ways
- Access control components
  - Reference monitor
  - Policy storage: access control directory/matrix/list
- Optimizations
  - Capability, RBAC
- Real-world examples
  - Linux permission bits, SELinux, Android, Active Directory

Cryptography (basics)

EECS 195

Spring 2019

Zhou Li

Why need cryptography?

Leo, doctor

I want to copy records to a USB drive. It might be stolen…

Let me help…

Jim, programmer

Process & copy records

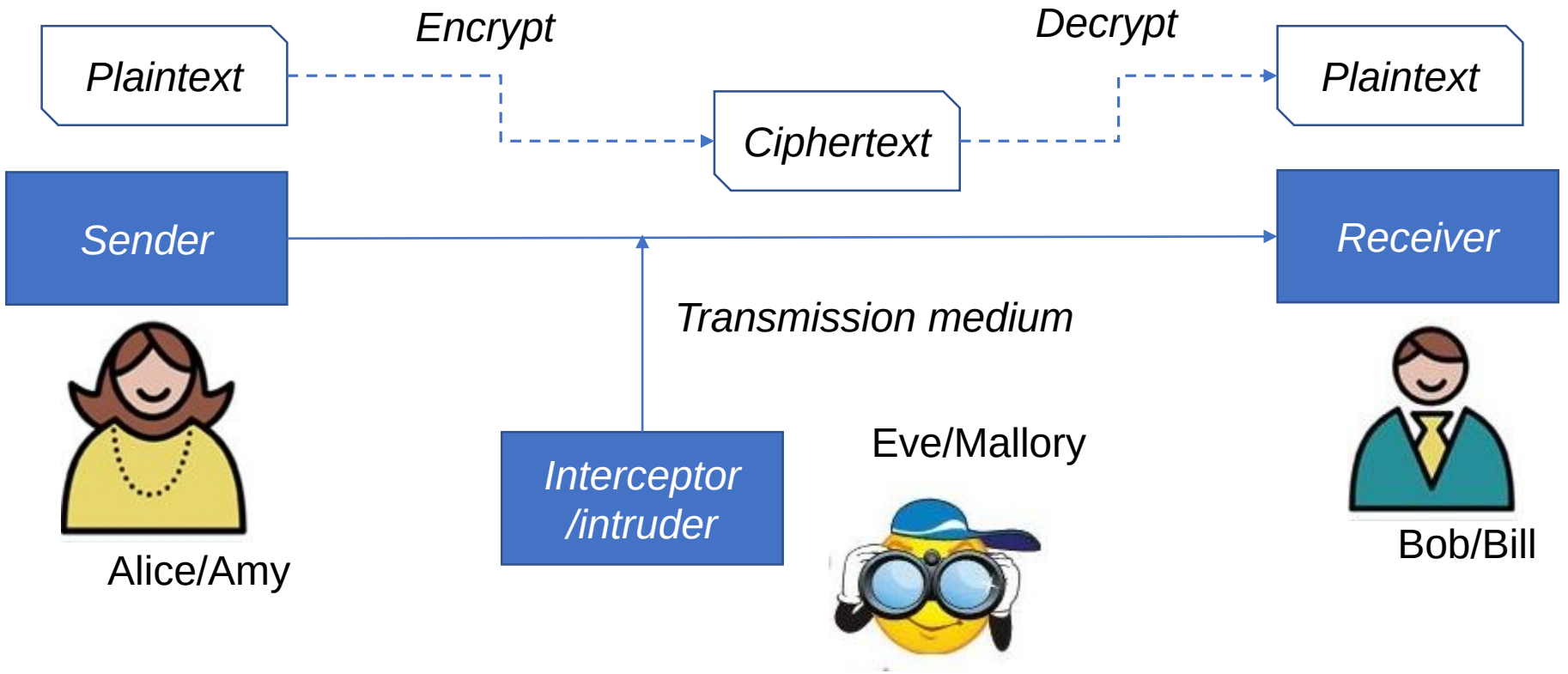
Hospital Information System

Shared workstation

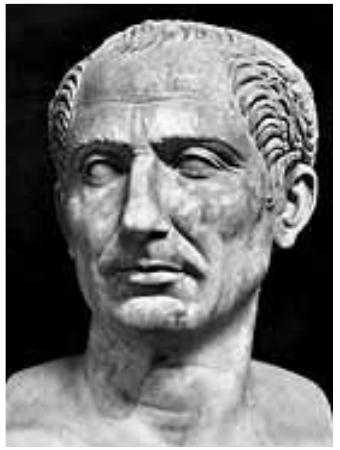
Adversary model

- Suppose a sender wants to send a message to a recipient. An attacker may attempt to
  - Block the message
  - Intercept the message
  - Modify the message
  - Fabricate an authentic-looking alternate message

Terminology



“Ancient Crypto”: Substitution ciphers



- Caesar cipher shifts letters with a constant of K
  - Encryption: **ci** := (**pi** + **k**) mod 26
  - Decryption: **pi** := (**ci** - **k**) mod 26
- “TREATY IMPOSSIBLE” -> “wuhdwb lpsrvvleoh”
- Pros: Easy to remember and use
- Cons: obvious patterns in ciphertext
  - Ciphertext is deterministic: same plaintext always gives the same ciphertext

vjku oguucig ku pqv vqq jctf vq dtgcm

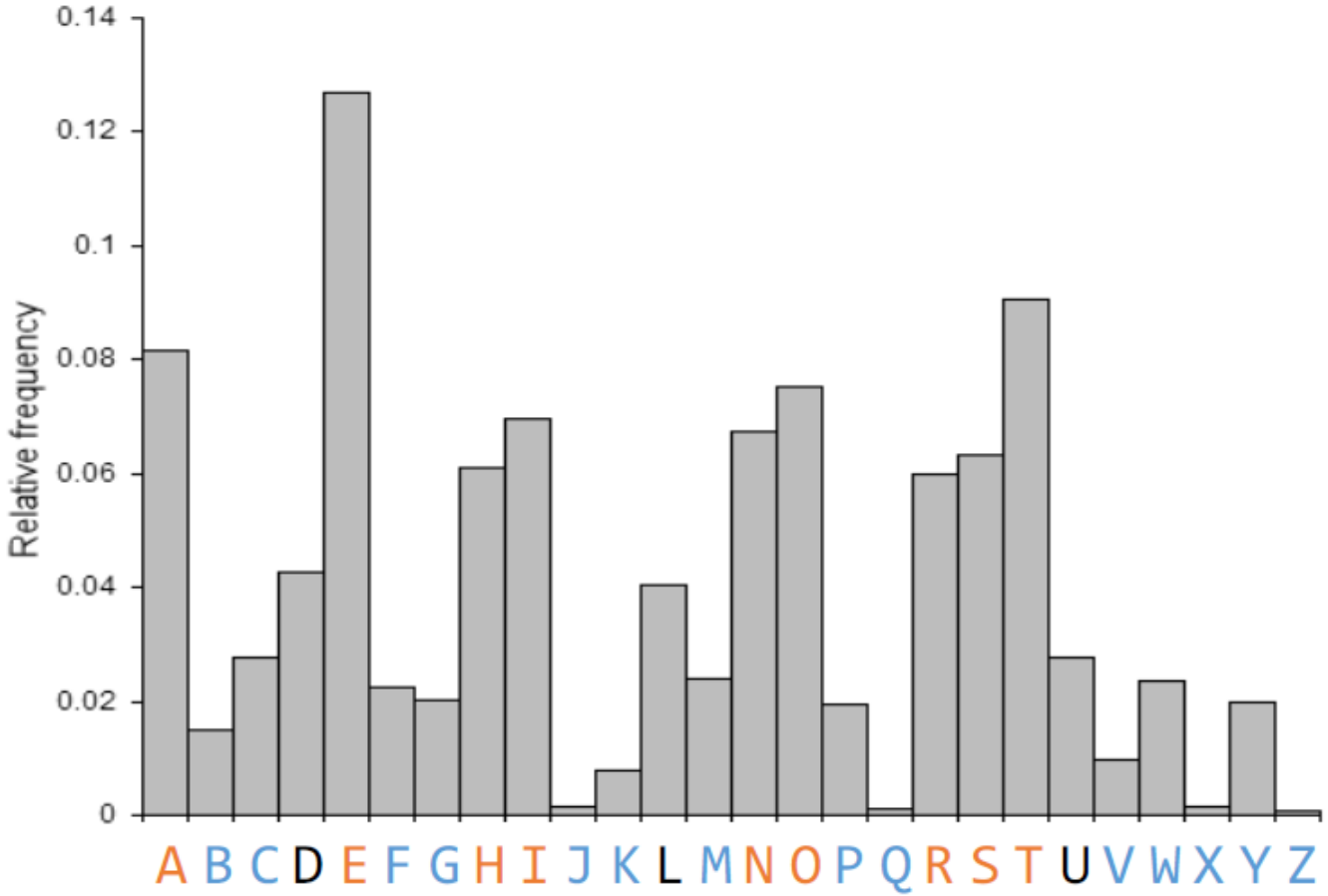
too to

this message is not too hard to break

Julius Caesar used caesar cipher to communicate with his generals 2100 years ago

Caesar Cipher Cryptanalysis

- Simple substitution ciphers don't alter symbol frequency



“Modern Crypto”: Secret key

- E, D: encryption, decryption algorithm
- K: secret key (e.g. 128 bits)
- M, C: plaintext, ciphertext
- Alice encrypts her message M using K
  - *C = E (M, K)*
- Alice gives K’ to Bob (sometimes K=K’)
- Only Bob can decrypt the message C
  - *M = D (C, K’)*

Secret key

- **K=K'?**
- **Symmetric (secret) key encryption**
  - The keys for encryption and decryption are the same.
  - Communicating parties must have the same key before communication
- **Asymmetric (public) key encryption**
  - Public key is published for anyone to encrypt a message
  - Only authorized parties have the private key to decrypt the message

Encryption algorithm

- Rule #1: Use public known encryption algorithm
  - ***Never*** use a proprietary cipher!
- Encryption strength depends on key length
- Algorithms
  - One-time pad (OTP)
  - Stream Cipher using PRG
  - Block Cipher (AES, DES, RC4, …)
  - Asymmetric: RSA, Elliptic Curve

Cryptographic primitives

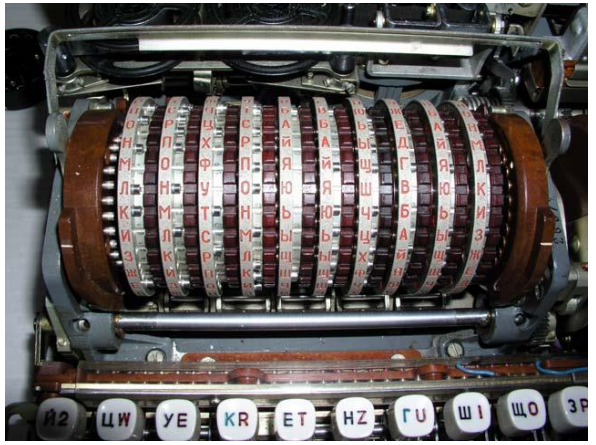
- Goal: robust against cryptanalysis
- Substitution and transposition
  - Substitution: one set of bits is exchanged for another
  - Transposition: rearranging ciphertext order to break any repeating patterns in the underlying plaintext.
- Confusion and diffusion
  - Confusion: algorithm to reduce the predictability of ciphertext when changing one character in plaintext
  - Diffusion: spread the information from the plaintext over the entire ciphertext

Shannon's characteristics of good ciphers

1. The amount of secrecy needed should determine the amount of labor appropriate for the encryption and decryption
2. The set of keys and the enciphering algorithm should be free from complexity
3. The implementation of the process should be as simple as possible
4. Errors in ciphering should not propagate and cause corruption of further information in the message
5. The size of the enciphered text should be no larger than the text of the original message

Cryptanalysis

- Goal: test/break an encryption algorithm/message
- Cryptanalysis is successful if the work force is reasonable
  - Work force: amount of efforts, like time
- Conditions of cryptanalysis
  - Cipher-text only
  - Known-plaintext (know both plaintext and ciphertext)
  - Chosen-plaintext (can select prepared plaintext and see ciphertext)
  - Chosen-ciphertext (chosen-plaintext & select ciphertext and see plaintext)
- Human fallibility and software/hardware implementation bugs also lead to code breaking!

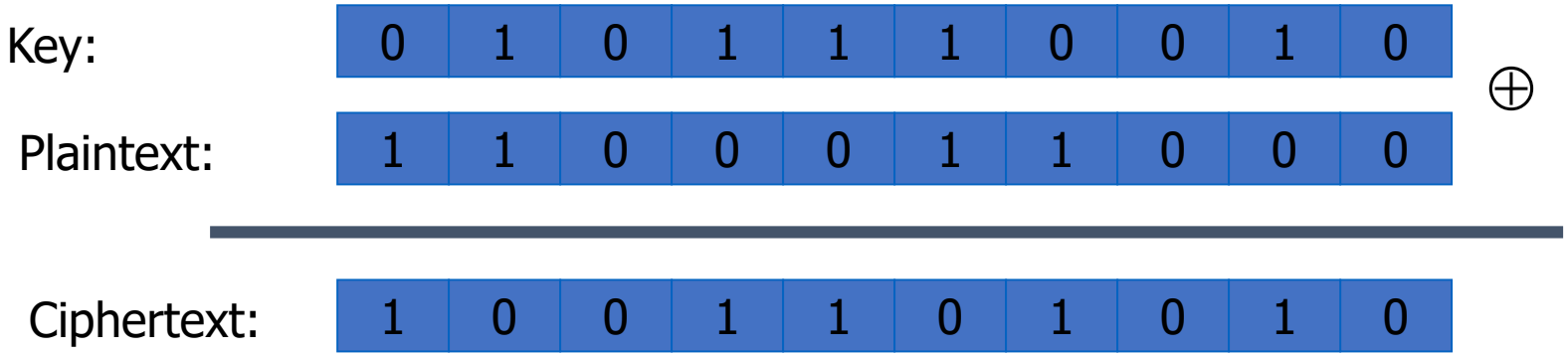


Fialka cipher machine

Cryptography Engineering, Ferguson et al.

Stream Cipher: One Time Pad (OTP)

- Stream ciphers encrypt one bit or one byte at a time
- Gilbert S. Vernam (1917)
- OTP: Key is only used to encrypt one message



Encryption: $c = E(k, m) = m \oplus k$

Decryption: $D(k, c) = c \oplus k = (m \oplus k) \oplus k = m$