CS631 - Advanced Programming in the UNIX Environment
–
(Only the most basic) Encryption in a Nutshell

Department of Computer Science
Stevens Institute of Technology
Jan Schaumann
jschauma@stevens.edu
https://www.cs.stevens.edu/~jschauma/631/

Cryptography

Cryptography can provide “security” in the areas of:

- Authenticity
  - *Is the party I’m talking to actually who I* think *it is?*
- Accuracy or Integrity
  - *Is the message I received in fact what was sent?*
- Secrecy or Confidentiality
  - *Did/could anybody else see (parts of) the message?*

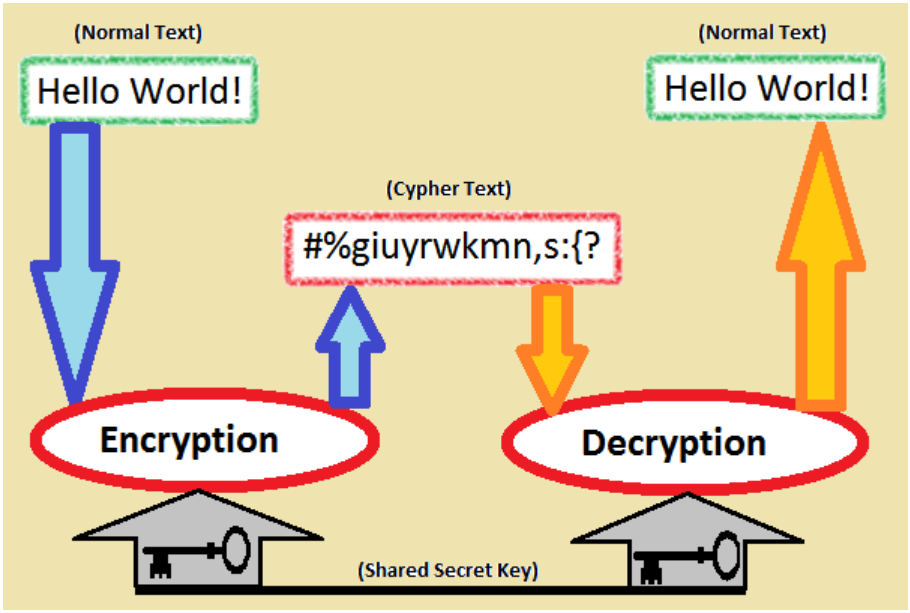
How does encryption work?

Secrecy: Make sure that the data can only be read by those intended.

How does encryption work?

Secrecy: Make sure that the data can only be read by those intended.

- *Alice* Edward and *Bob* Glenn agree on a way to transform data
- transformed data is sent over insecure channel
- Edward and Glenn are able to get data out of the transformation



How does encryption work?

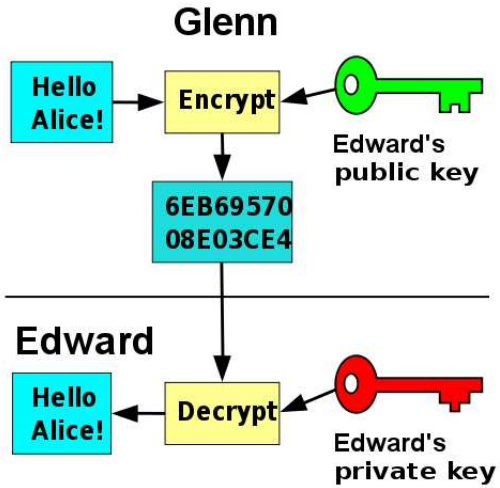
Different approaches:

- public key cryptography
- secret key cryptography

How does encryption work?

Different approaches:

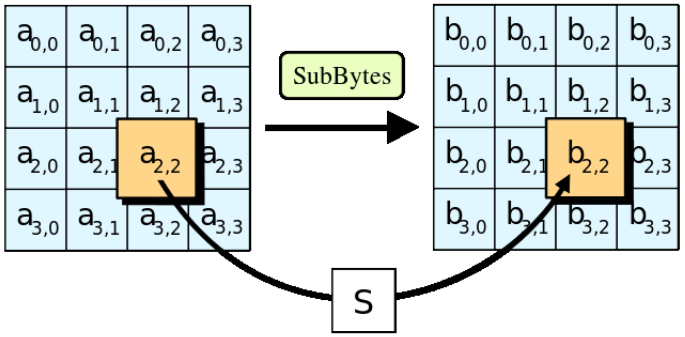
- public key cryptography (example: *RSA*, your ssh keys)
  - Edward has a private and a public key
  - data encrypted with her private key can only be decrypted by her public key and vice versa
  - public key can be shared with Glenn



How does encryption work?

Different approaches:

- secret key cryptography (example: *AES*)
  - Edward and Glenn share a secret key
  - for authentication purposes, Edward may prove to Glenn that he knows the secret key
  - any data encrypted with this key can also be decrypted using the same key



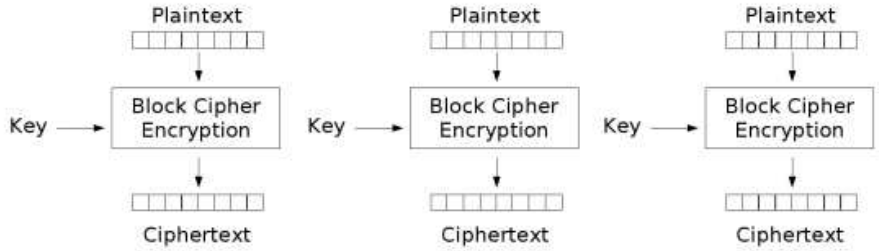
Cipher Modes

Encryption entails transformation of input data (“plain” or “clear” text) into encrypted output data (“ciphertext”). Input data is generally transformed in one of two ways:

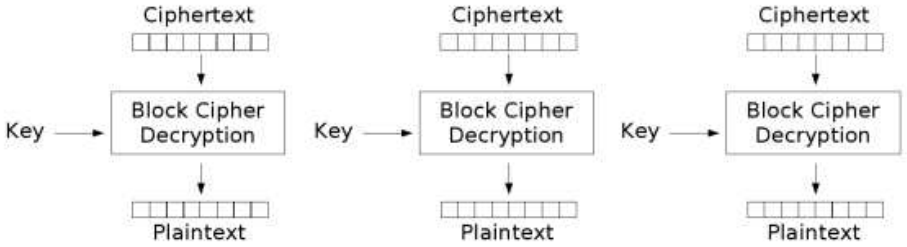
Stream Cipher: each bit on plaintext is combined with a pseudo-random cipher digit stream (or *keystream*)

Block Cipher: fixed-length blocks of plaintext are transformed into same-sized blocks of ciphertext; may require padding

Electronic Codebook Mode

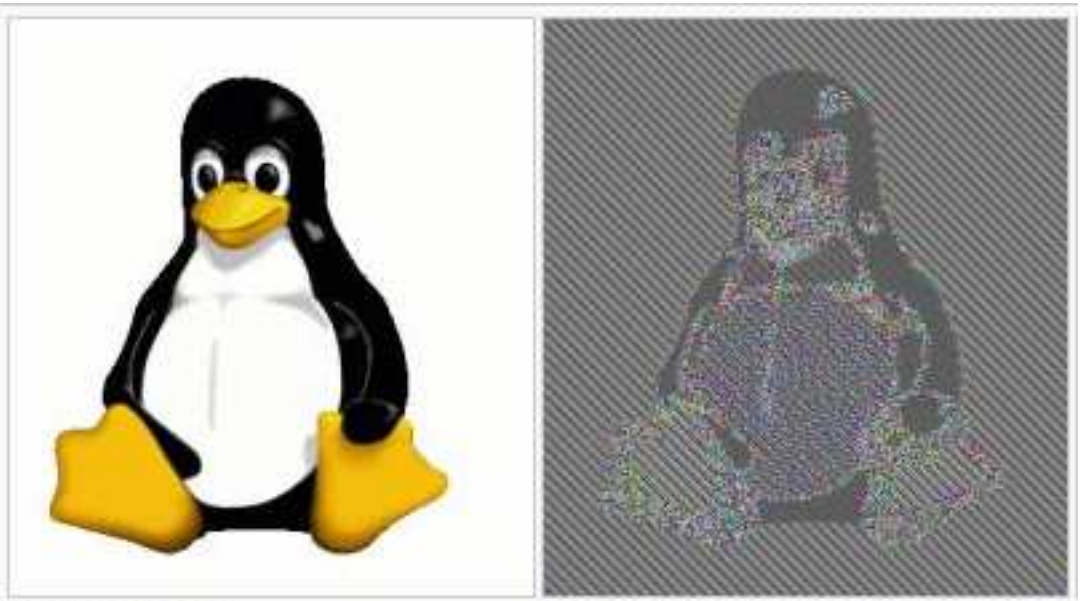


Electronic Codebook (ECB) mode encryption



Electronic Codebook (ECB) mode decryption

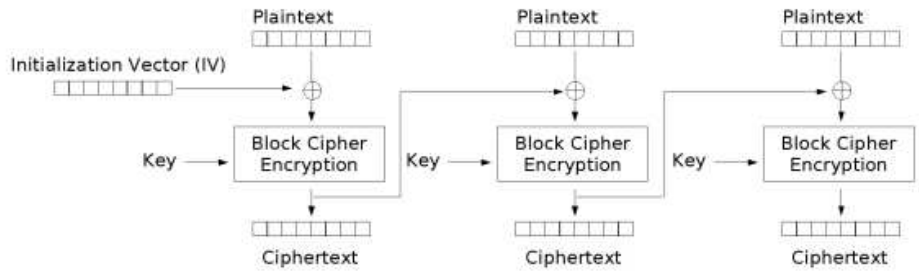
Electronic Codebook Mode



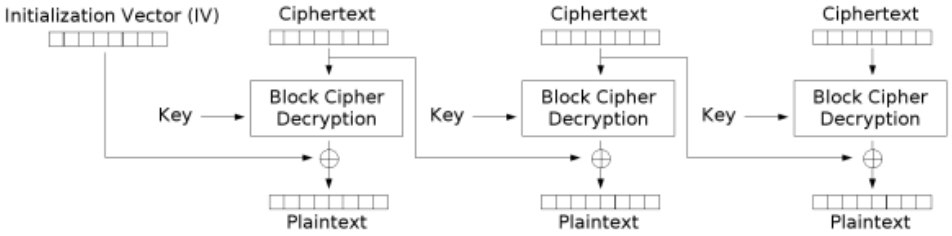
Original image

Encrypted using ECB mode

Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption



Cipher Block Chaining (CBC) mode decryption

Random String generation

Random numbers can be generated using `/dev/random`, `/dev/urandom`, `rand(3)`, `random(3)`, `BN_rand(3)` etc.

Map numbers to printable characters (for use as a salt, for example):

```
static const unsigned char itoa64[] =
        "./0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

char salt[16];
for (i=0; i<16; i++)
        salt[i] = itoa64[(int)random()%64];
```

Practical AES

- a symmetric block cipher
- variable key length
- consists of a key setup phase and the actual encryption or decryption
- keying material use of `ivec`, which needs to be shared
- useful code examples in `EVP_EncryptInit(3)`

HW#4

https://www.cs.stevens.edu/~jschauma/631/f15-hw4.html

```
NAME
     aed - perform aes256cbc encryption/decryption

DETAILS
     aed reads data from stdin and either encrypts or decrypts it (depending
     on the -d or -e flag).  It uses AES 256bit CBC mode with a SHA1 digest
     with keying material derived from the passphrase using the
     EVP_BytesToKey(3) function, generating a suitable salt via RAND_bytes(3).

     Output is written to stdout.

     When encrypting, the output is prefixed by the 8 byte salt.
```

HW#4

https://www.cs.stevens.edu/~jschauma/631/f15-hw4.html

To encrypt the contents of the file file and storing the encrypted output in file.enc:

```
aed -e -p passfile <file >file.enc
```

To decrypt the contents of that file again:

```
aed -d -p passfile <file.enc
```

Since aed operates on stdin and stdout, the above two commands could also be chained:

```
export AED_PASS=$(cat passfile)
cat file | aed -e | aed -d
```

References

- crypto(3)
- EVP_EncryptInit(3)
- EVP_BytesToKey(3)
- http://tldp.org/LDP/LG/issue87/vinayak.html
- http://en.wikipedia.org/wiki/Cipher_Block_Chaining
- http://www.moserware.com/2009/09/stick-figure-guide-to-advanced.html