CS631 - Advanced Programming in the UNIX Environment
–
Encryption in a Nutshell

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

Purpose of Encryption

Encryption provides Security! Duh.

Purpose of Encryption

Encryption provides security in the areas of:

- Authenticity
- Accuracy or Integrity
- Secrecy or Confidentiality

Purpose of Encryption

Encryption provides security in the areas of:

- Authenticity
  - *Is the party I'm talking to actually who I* think *it is?*
- Accuracy or Integrity
- Secrecy or Confidentiality

Purpose of Encryption

Encryption provides 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

Purpose of Encryption

Encryption provides 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?*

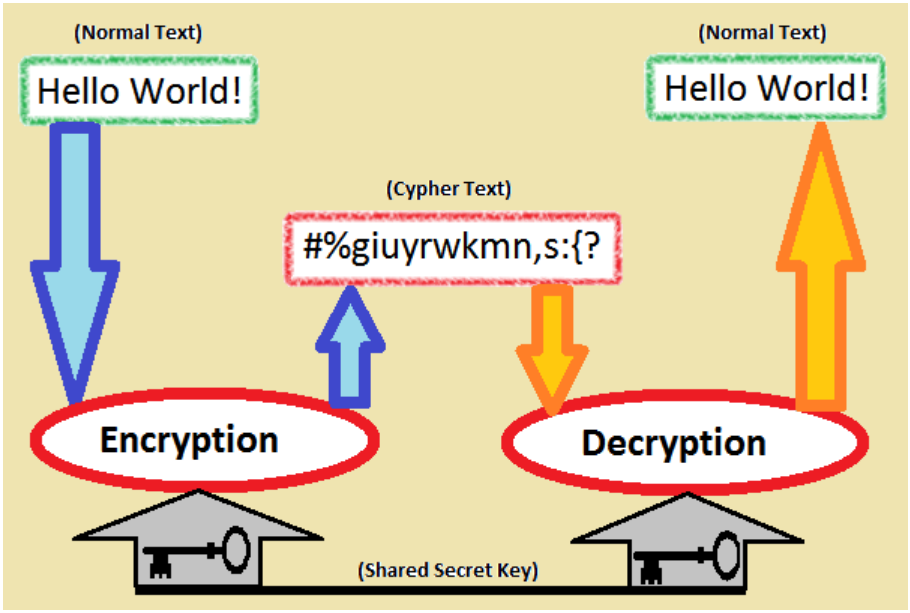
How does encryption work?

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?

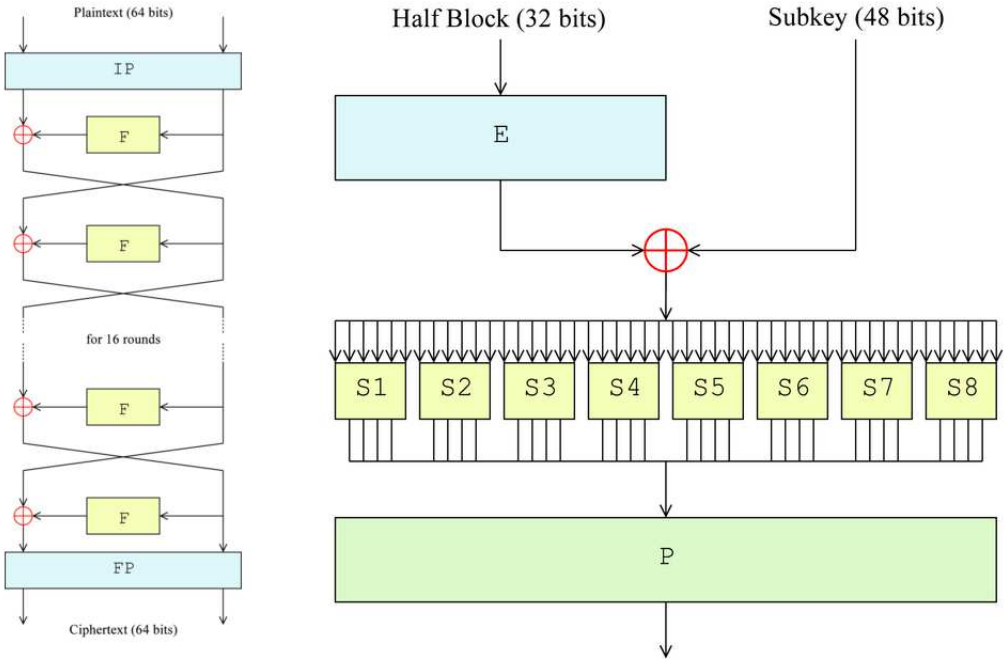
Different approaches:

- secret key cryptography
- public key cryptography

How does encryption work?

Different approaches:

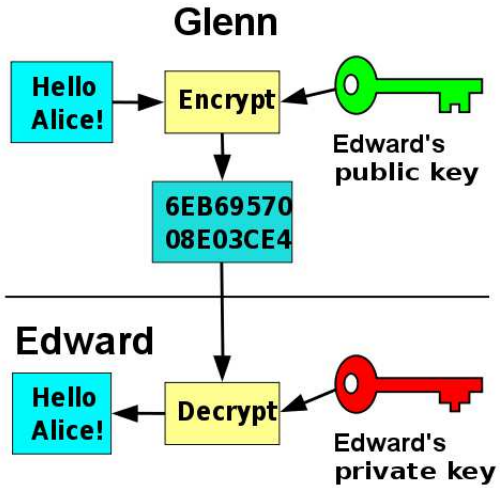
- secret key cryptography (example: *DES*)
  - Edward and Glenn share a secret
  - Edward can prove to Glenn that he knows a secret



How does encryption work?

Different approaches:

- public key cryptography (example: *RSA*)
  - 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



Accuracy or Integrity

In order to protect against forgery or data manipulation, provide some sort of digest or checksum (often a one-way hash). Popular choices:

- `5f4dcc3b5aa765d61d8327deb882cf99`
- `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8`
- `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8`
- `b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86`

Accuracy or Integrity

In order to protect against forgery or data manipulation, provide some sort of digest or checksum (often a one-way hash). Popular choices:

- `5f4dcc3b5aa765d61d8327deb882cf99` (MD5)
- `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8` (SHA-1)
- `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62a11ef721d1542d8` (SHA256)
- `b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a2ea6d103fd07c95385ffab0cacbc86` (SHA512)

Accuracy or Integrity

In order to protect against forgery or data manipulation, provide some sort of digest or checksum (often a one-way hash). Popular choices:

- `5f4dcc3b5aa765d61d8327deb882cf99` (MD5)
- `5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8` (SHA-1)
- `5e884898da28047151d0e56f8dc6292773603d0d6aabbdd62 a11ef721d1542d8` (SHA256)
- `b109f3bbbc244eb82441917ed06d618b9008dd09b3befd1b5 e07394c706a8bb980b1d7785e5976ec049b46df5f1326af5a 2ea6d103fd07c95385ffab0cacbc86` (SHA512)

Caveats:

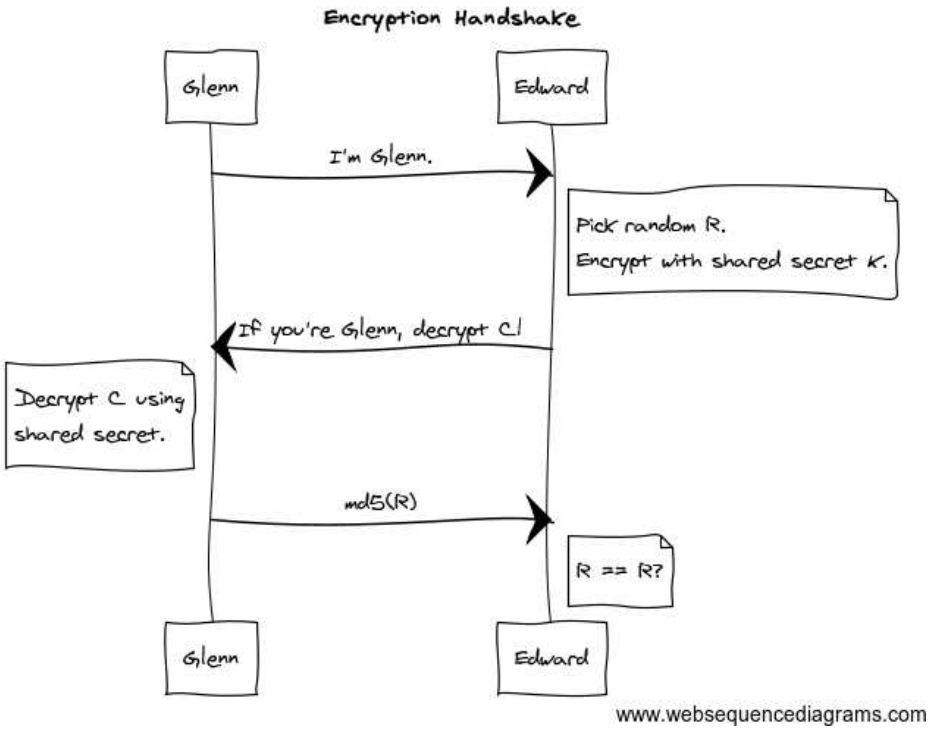
- “rainbow tables” / internet search engines allow for easy reverse lookup of un-salted hashes.
- integrity only ensured if authenticity of information itself is guaranteed

Authenticity

- in private key cryptography, authenticity is (often) assumed/implied
- in public key cryptography, often accomplished via a separate signature
- ways to establish assurance of authenticity for parties that have never met:
  - public key infrastructures (PKI) and certificate authorities (CA)
  - “web of trust”

Authentication

First, the client needs to prove that it knows the secret. This is done in the encryption handshake, consisting of a *challenge* and a *response*.



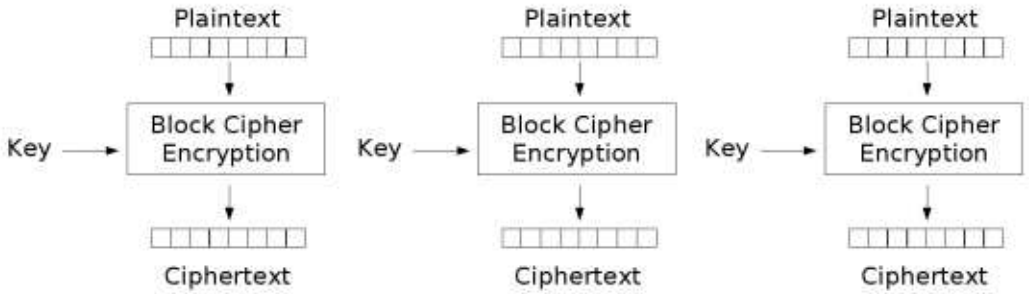
Random String generation

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):

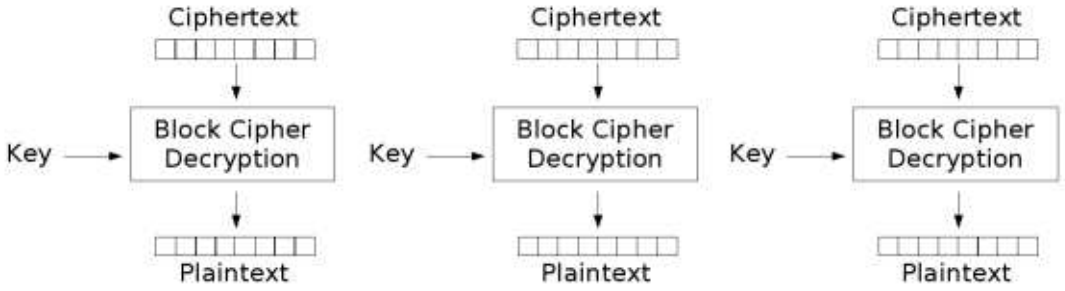
```
static const unsigned char itoa64[] =
        "./0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz";

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

Electronic Codebook Mode



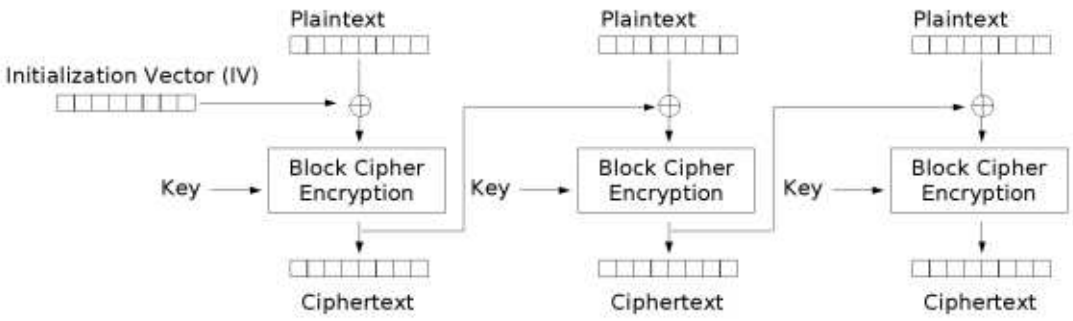
Electronic Codebook (ECB) mode encryption

Electronic Codebook Mode



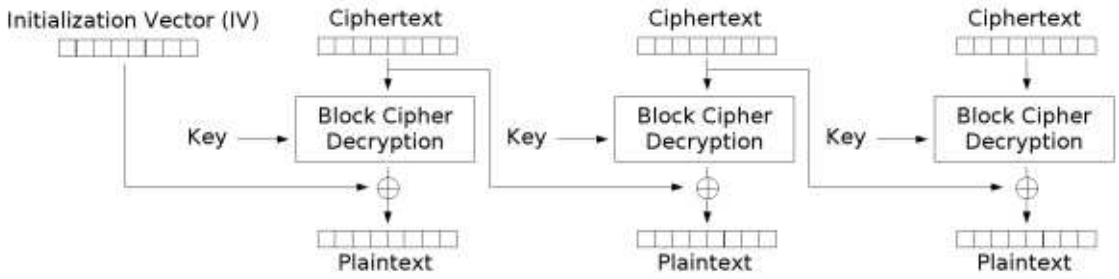
Electronic Codebook (ECB) mode decryption

Cipher Block Chaining



Cipher Block Chaining (CBC) mode encryption

Cipher Block Chaining



Cipher Block Chaining (CBC) mode decryption

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

HW#5

`http://www.cs.stevens.edu/~jschauma/631/f13-hw5.html`

References

OpenSSL's crypto libraries:

- `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`