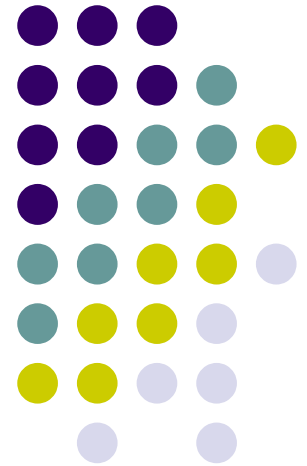


Crypto libraries

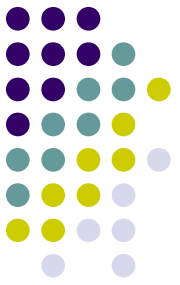
OpenSSL (cont.)

Milan Brož
xbroz@fi.muni.cz

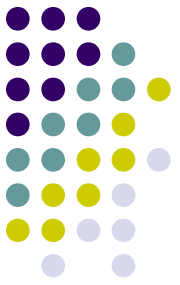
PV181, FI MUNI, Brno



OpenSSL – www.openssl.org



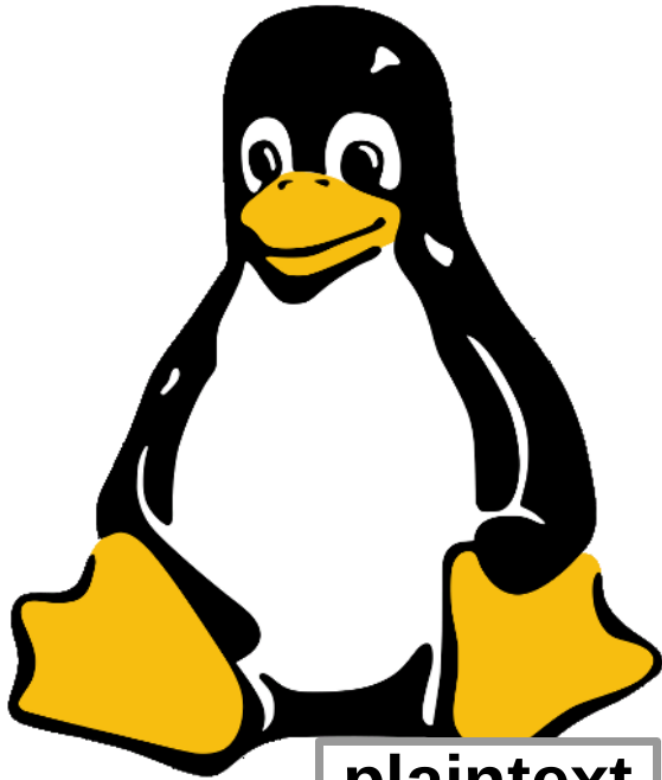
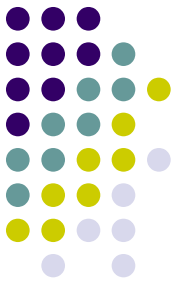
- opensource cryptography toolkit
- OpenSSL3 ~ released 2021, many API improvements (last stable version 3.4 ~ October 2024)
- Apache-style license
- hash, symmetric/asymmetric encryption, PKI, CA, ...
- ASN.1, PKCS-5,7,8,12, X509, OCSP, PEM, SSL, TLS
- command line tool
- C/C++ library bindings (+many other library wrappers)
 - on Linux compile with **-lcrypto -lssl**
 - `#include <openssl/...>`



Today's goals

- **Symmetric encryption**
- **Encryption modes**
ECB, CBC, CTR, XTS
IV – initialization vector, tweak
- **OpenSSL I/O abstraction (BIO)**
- **Demonstration of failures/mistakes**
ECB use,
CBC mangled IV, CBC mangled ciphertext,
XTS patterns
CTR stream reuse

Symmetric encryption: ciphertext



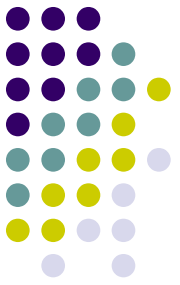
plaintext



ciphertext

Example 4:

Symmetric encryption



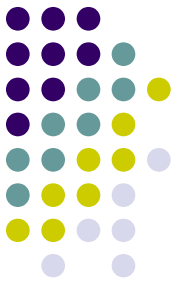
OpenSSL (3.x)

Encryption with EVP interface. Cipher mode is for example **"AES-256-CBC"**.

```
cipher = EVP_CIPHER_fetch(cipher_mode, ...)
EVP_CIPHER_CTX_new()
EVP_EncryptInit_ex2(context, cipher, key, iv, PARAMS)
EVP_EncryptUpdate(context, ciphertext, &clen, plaintext, plen)
EVP_EncryptFinal_ex(context, ciphertext + clen, &len)
EVP_CIPHER_CTX_free(context)
EVP_CIPHER_free(cipher)
```

See ***4_encryption_openssl3*** directory.

Example 7: OpenSSL BIO (I/O abstraction)

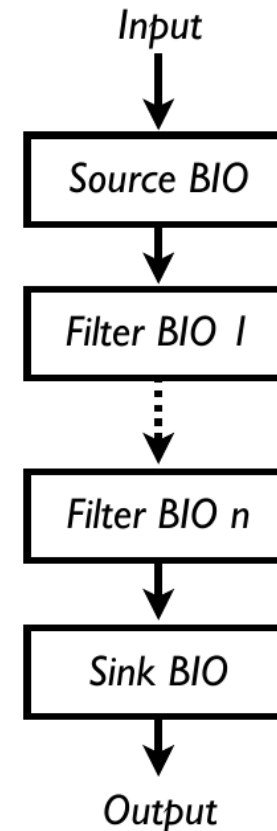


Source/sink BIOs:

BIO_s_mem() - memory I/O
BIO_s_file() - file I/O
BIO_s_fd() - file descriptor IO
BIO_s_socket() - sockets
BIO_s_accept()
BIO_s_connect()
BIO_s_null() - discard (like /dev/null)

Filters

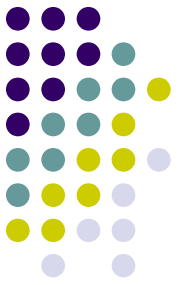
BIO_f_base64() - Base64 encoding
BIO_f_buffer() - buffering I/O
BIO_f_cipher() - encryption/decryption
BIO_f_md() - message digest
BIO_f_ssl() - SSL support for BIO



Example 7: the same encryption as in Example 4 using BIO interface.
See `7_bio_openssl` directory.

Symmetric Encryption

common mistakes or failures



See `6_encryption_fails_openssl` example in git.

Comment out various sections and play with demos.

Note there is no data integrity protection in these modes.

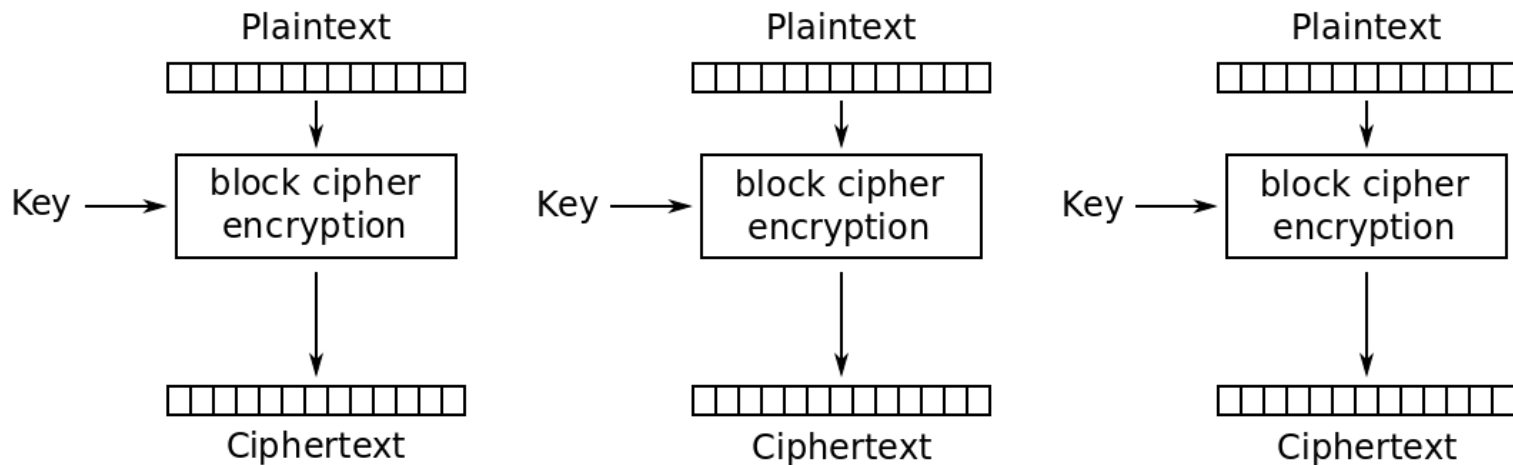
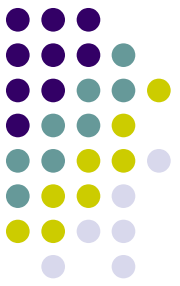
- Case 6.1: **ECB patterns**
- Case 6.2: **CBC IV bit flips**
- Case 6.3: **CBC bit flips in a consecutive block**
- Case 6.4: **XTS constant IV block patterns**
- Case 6.5: **CTR stream reuse**

Wrong use demo: re-use key from known ciphertext/plaintext pair.

See `6_encryption_fails_openssl` directory.

ECB mode

...should be never used



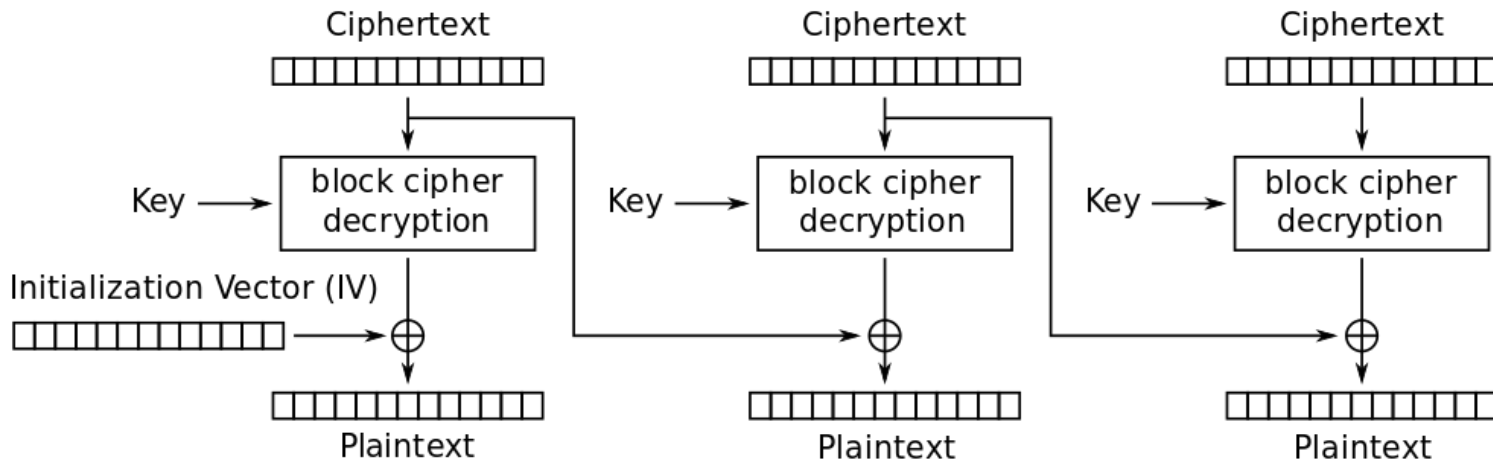
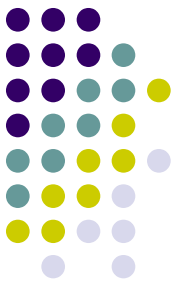
Electronic Codebook (ECB) mode encryption

Wrong use demo: ciphertext patterns, block relocation.

*See **6_encryption_fails_openssl** directory.*

picture: Wikipedia

CBC mode

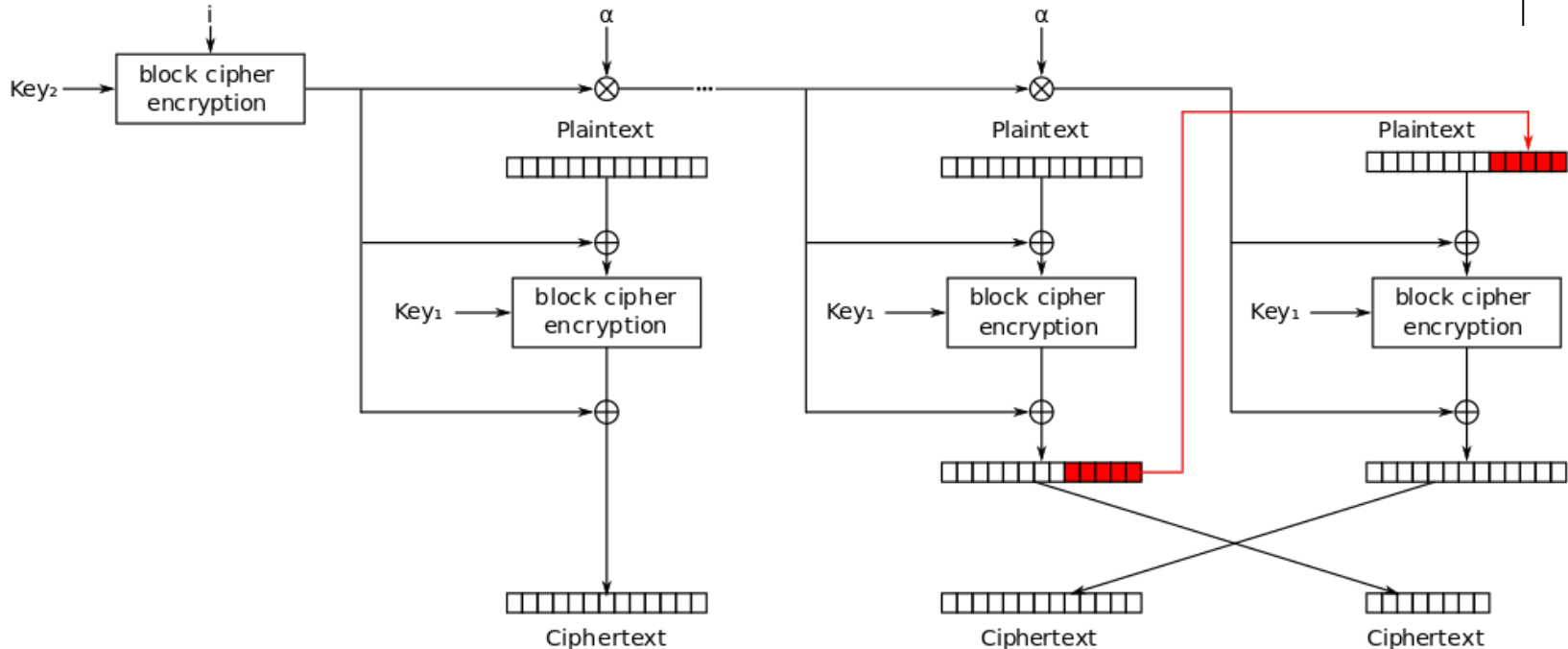
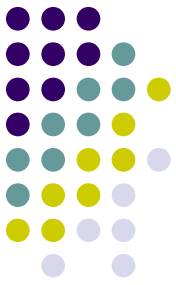


Cipher Block Chaining (CBC) mode decryption

*Wrong use demo: first block bit flips (IV) and consecutive block change.
See [6_encryption_fails_openssl](#) directory.*

picture: Wikipedia

XTS mode storage (file, disk) encryption



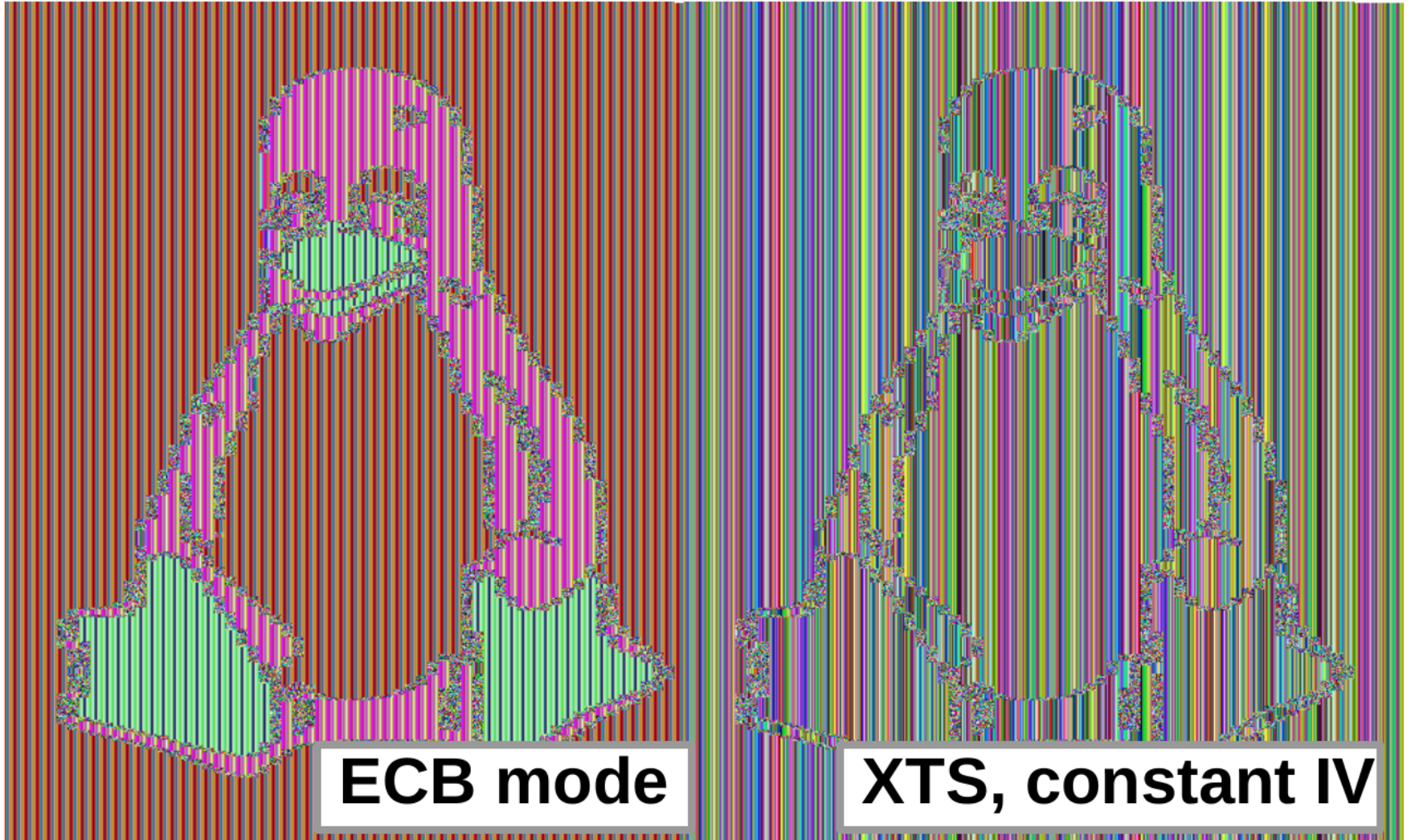
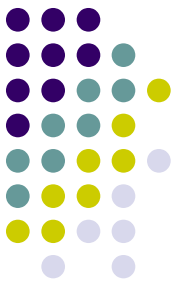
XEX with tweak and ciphertext stealing (XTS) mode encryption

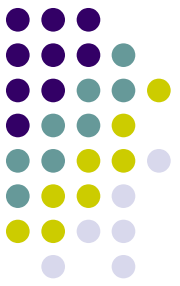
Wrong use demo: block patterns with constant IV.

See [6_encryption_fails_openssl](#) directory.

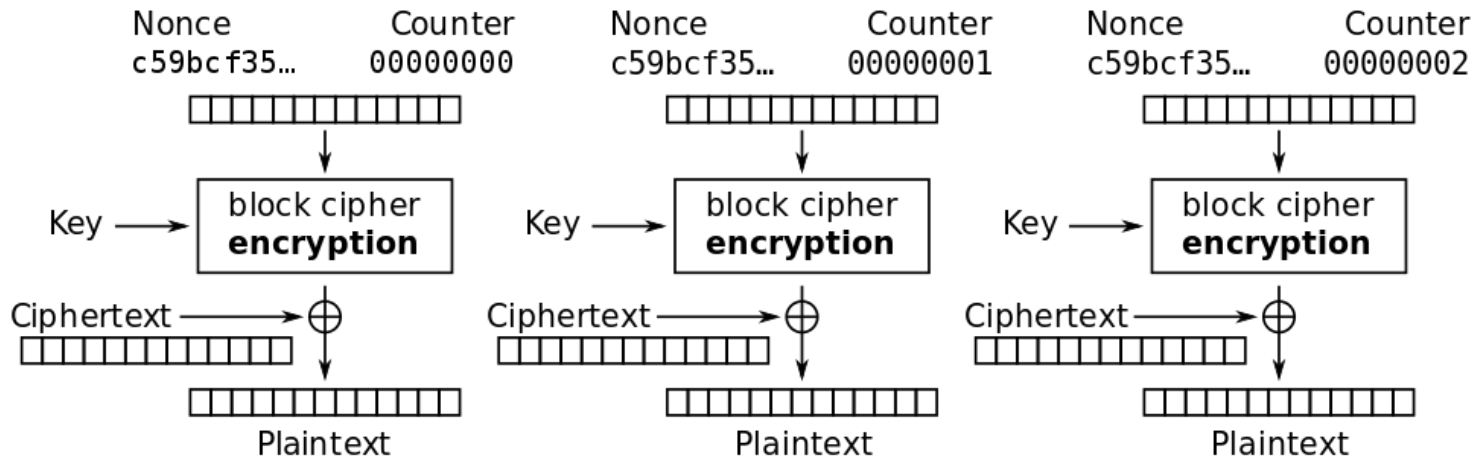
picture: Wikipedia

Symmetric encryption fails: patterns in ciphertext





CTR (counter) mode



Counter (CTR) mode decryption

Wrong use demo: re-use key from known ciphertext/plaintext pair.

*See **6_encryption_fails_openssl** directory.*

picture: Wikipedia