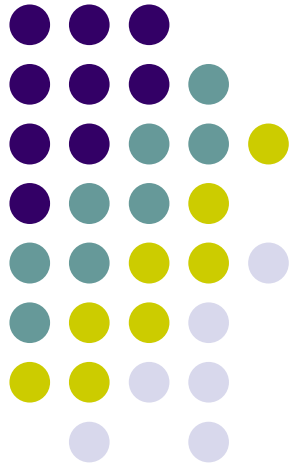
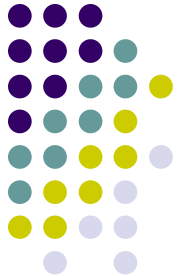
Crypto libraries OpenSSL (cont.)

Milan Brož
xbroz@fi.muni.cz

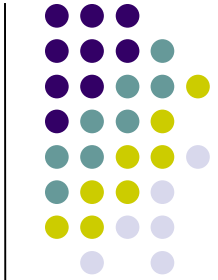


OpenSSL – www.openssl.org



- opensource cryptography toolkit
- Apache-style license
- hash, symmetric/asymmetric encryption, PKI, CA, ...
- ASN.1, PKCS-5,7,8,12, X509, OCSP, PEM
- SSL and TLS
- command line tool
- C/C++ library bindings (+many other library wrappers)
  - on Linux compile with **-lcrypto –lssl**
  - #include <openssl/...>

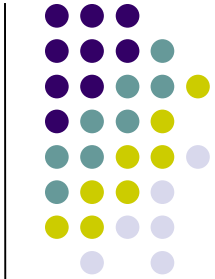
Today's exercise



- Continue with OpenSSL on Linux
- Symmetric Encryption
- BIO (I/O abstraction)
- Signing, certificates store example

- Assignment (see separate file, 5 + 5 points)

Example 4: Symmetric encryption



OpenSSL

Encryption with EVP interface. Cipher mode is for example **EVP_aes_256_cbc().**

```
EVP_CIPHER_CTX_new()
EVP_EncryptInit_ex(context, EVP_cipher_mode,
                                NULL/*engine*/, key, iv)
EVP_EncryptUpdate(context, ciphertext, &clen, plaintext, plen)
EVP_EncryptFinal_ex(context, ciphertext + clen, &len)
EVP_CIPHER_CTX_free(context)
```

*See **4_encryption_openssl** directory.*

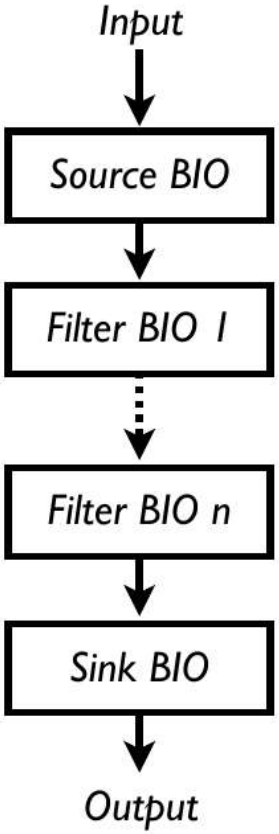
Example 5: OpenSSL BIO (I/O abstraction)

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)

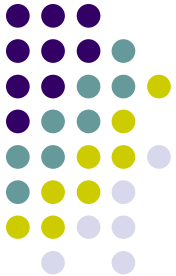
Filters

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 5: the same encryption as in Example 4 using BIO interface.
*See **5_bio_openssl** directory.*

Example 6: Signing and certificates



PKCS12

- PKCS12_verify_mac, PKCS12_parse

PKCS7

- PKCS7_sign, PKCS7_verify

X509

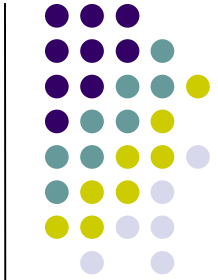
- X509_STORE_add_lookup

BIO

- BIO_new, BIO_new_mem_buf, BIO_new_file
- BIO_push, BIO_f_cipher, BIO_set_cipher
- BIO_flush, BIO_free_all
- d2i_PKCS12_bio, d2i_PKCS7_bio

*See **6_cert_sign_openssl** directory.*

Assignment



- Two goals:
  - Use symmetric encryption (AES128-CBC) [max 5 points]
  - Generate RSA key in C [max 5 points]
  - Use OpenSSL in Linux environment
- See Assignment.txt in IS for details and deadline
- You can start with examples in git
- Comment your code
  - but do not overuse comments

- Note: for encryption expect binary input
- You can use provided Fedora VM or aisa server
  (or any OpenSSL Linux, even Win10 embedded Linux)