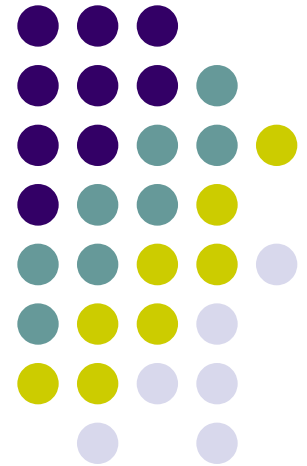


# Crypto libraries intro – examples

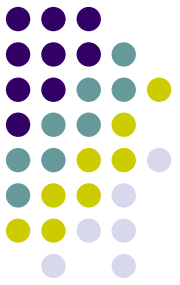
---

**Milan Brož**  
xbroz@fi.muni.cz

PV181, FI MUNI, Brno

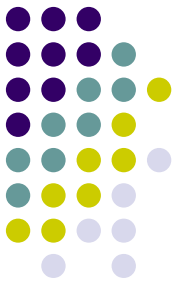


# Today's examples



- **Low-level crypto primitives**
  - RNG
  - Hash, HMAC
  - PBKDF
- Examples comparison in **libgcrypt, OpenSSL, and libsodium**
- ~~Comparison OpenSSL 1.1.x / 3.0.x~~  
except maintaining legacy code, use OpenSSL3 now
- See git (examples 1, 2, 3)
- Note: examples are simplified  
sometimes error path can leak memory

# Example 1: RNG in libraries



## libgcrypt

see `1_rng_gcrypt` example

```
(void) gcry_randomize(buf, sizeof(buf), GCRY_STRONG_RANDOM)
```

## OpenSSL

see `1_rng_openssl` example

```
(int) RAND_bytes(buf, sizeof(buf))
```

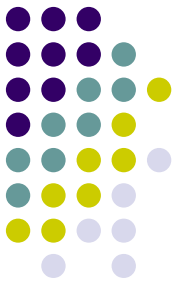
## libsodium

see `1_rng_sodium` example

```
(void) randombytes_buf(buf, sizeof(buf))
```

*Simple? Not in the real-world. RNG or pseudo RNG, optional parameters, initialization or another call for configuration, can/cannot fail, can/cannot block if not enough entropy, is it own implementation or wrapper to system RNG, can it be used in FIPS mode ...*

# Example 2: Hash functions



## libgcrypt

See `2_hash_hmac_gcrypt` example

```
gcry_md_open(context, hash_id, flags)
gcry_md_write(context, data, data_len)
gcry_md_read(context, hash_id)
gcry_md_close(context)
```

## OpenSSL (OpenSSL3 syntax)

EVP (envelope) interface, see `2_hash_hmac_openssl3` example

```
EVP_MD_fetch(...)
EVP_MD_CTX_new();
EVP_DigestInit(context, hash_id)
EVP_DigestUpdate(context, data, data_len)
EVP_DigestFinal(context, out, &out_len)
EVP_MD_CTX_free(context);
```

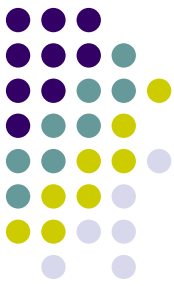
## libsodium

See `2_hash_hmac_sodium` example

```
crypto_hash_sha256_init(context)
crypto_hash_sha256_update(context, data, data_len)
crypto_hash_sha256_final(context, out))
```

# Example 2: HMAC

## Keyed Hash Message Authentication Code



### libgcrypt

See `2_hash_hmac_gcrypt` example

```
gcry_md_open(context, hash_id, GCRY_MD_FLAG_HMAC)
gcry_md_setkey(context, key, key_len)
gcry_md_write(context, data, data_len)
gcry_md_read(context, hash_id)
gcry_md_close(context)
```

### OpenSSL (OpenSSL3 syntax)

EVP interface, see `2_hash_hmac_openssl3` example

```
EVP_MAC_fetch(...);
EVP_MAC_CTX_new(EVP_mac);
EVP_MAC_init(context, key, key_len, PARAMS[])
EVP_MAC_Update(context, data, data_len)
EVP_MAC_Final(context, out, &out_len, sizeof(out))
EVP_MAC_CTX_free(context);
```

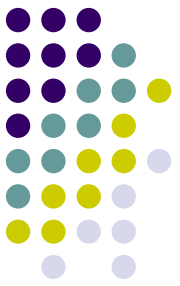
### libsodium

NaCl compatible interface, see `2_hash_hmac_sodium` example

```
crypto_auth(out, data, data_len, key))
crypto_auth_verify(expected_out, data, data_len, key))
```

# Example 3: PBKDF

## Password-Based Key Derivation Functions



### libgcrypt

See `3_pbkdf_gcrypt` example (and for bonus: `3_pbkdf_argon2_gcrypt`)

```
gcry_kdf_derive(password, password_len,  
                GCRY_KDF_PBKDF2, GCRY_MD_SHA256,  
                salt, salt_len, iterations, key_len, key)
```

### OpenSSL (*Argon2 comes in 3.2*)

See `3_pbkdf_openssl3` example

```
EVP_KDF_fetch(...);  
EVP_KDF_CTX_new(EVP_kdf);  
EVP_KDF_derive(context, key, key_size, PARAMS[]);  
EVP_KDF_CTX_free(context);
```

### libsodium

See `3_pbkdf_sodium` example

(Note: default algorithm is memory-hard **Argon2id**, PBKDF2 not implemented)

```
crypto_pwhash(key, key_len, password, password_len,  
             salt, opslimit, memlimit, algorithm)
```