8. OpenSSL: Asymetrická kryptografie – generování klíčů, elementární operace RSA

Obsah

1. Úkoly
2. Tipy pro načítání klíčů
3. Tipy pro inicializaci generátoru náhodných čísel
4. Tipy pro EVP_SealInit
5. Poznámka šifrování krátkých zpráv - EVP_PKEY_encrypt / decrypt

V dnešním cvičení si ukážeme základní operace s klíči a certifikáty asymetrické kryptografie pomocí funkcí knihovny openssl.

- Vytvoříme pár klíčů (soukromý, veřejný) typu RSA
- Klíče použijeme pro základní operace RSA (Šifrování, dešifrování, podpis, ověření).

Vytvořte si soukromý klíč typu RSA. Budete potřebovat utilitu openssl.

Nejdříve si vytvoříme parametry RSA, které představují dvojici (veřejný klíč, soukromý klíč).

```
openssl genrsa -out privkey.pem 2048
```

Z vytvořeného souboru oddělíme pouze veřejný klíč, který bychom mohli zveřejnit.

```
openssl rsa -in privkey.pem -pubout -out pubkey.pem
```

Prozkoumáme vytvořené soubory – podíváme se na ně v textové formě

```
openssl rsa -in privkey.pem -noout -text
openssl rsa -in pubkey.pem -pubin -noout -text
```

Úkoly

Svůj veřejný klíč publikujte/posílejte podle pokynů cvičícího.

Za splnění celého zadání dostanete **5 bodů**.

Napište dva programy:

- První program zašifruje zprávu (soubor) pomocí kombinace symetrické a asymetrické šifry (`EVP_Seal…`) pro příjemce (Vás nebo souseda).
    - Vstupem budou 2 soubory:
        - Soubor s daty (binární data obecné velikosti)
        - Soubor s veřejným klíčem adresáta
    - Výstupem bude 1 soubor obsahující vše, co je nutné pro dešifrování soukromým klíčem adresáta.
        - (šifrový text, typ symetrické šifry (včetně operačního módu a délky klíče), inicializační vektor a zašifrovaný symetrický klíč)
    - Jména souborů budou zadána formou parametrů na příkazové řádce.
- Druhý program dešifruje zprávu (soubor) pomocí kombinace symetrické a asymetrické šifry (`EVP_Open…`) z předchozího bodu a uloží do souboru.
- Porovnejte vstupní a výstupní (dešifrovaný) soubor na binární shodu.
- Na začátku zdrojového kódu napište do komentáře jméno autora.

Tipy pro načítání klíčů

Klíče jsou v souborech ve formátu PEM (Privacy Encoded Mail). K načítání slouží funkce z části „pem“.

Příklad načtení veřejného klíče pro použití v `EVP_PKEY_encrypt(…)` , `EVP_SealInit(…)` :

```
EVP_PKEY * pubkey;
fp = fopen(....);
pubkey = PEM_read_PUBKEY(fp, NULL, NULL, NULL); //No password protection of the key itself
```

Podobně se načítá i soukromý klíč (`PEM_read_PrivateKey`) pro použití v `EVP_PKEY_decrypt(…)` , `EVP_OpenInit(…)`

Tipy pro inicializaci generátoru náhodných čísel

OpenSSL obsahuje kryptograficky bezpečný pseudonáhodný generátor, který by měl být na začátku programu inicializován pomocí kvalitního zdroje náhodnosti (nezapomeňte, že `EVP_SealInit` generuje náhodné symetrické klíče a inicializační vektor). Příklad:

```
if (RAND_load_file("/dev/random", 32) != 32) {
    puts("Cannot seed the random generator!");
    exit(1);
}
```

Tipy pro EVP_SealInit

Manuálová stránka pro `EVP_SealInit` popisuje použití obecně s více veřejnými klíči (pro více adresátů). Můžeme si to zjednodušit pro případ pouze 1 adresáta, tedy zadat jen 1 veřejný klíč a dostat 1 zašifrovaný symetrický klíč.

```
EVP_PKEY * pubkey = PEM_read...(...);
unsigned char * my_ek = (unsigned char *) malloc(EVP_PKEY_size(pubkey)); // allocate space for encrypted symmet. key
int my_ekl; // enc. sym. key length
unsigned char iv[EVP_MAX_IV_LENGTH]; // buffer for the init. vector

//... declare context ctx and type as in previous lab
EVP_SealInit( &ctx, type, &my_ek, &my_ekl, iv, &pubkey, 1); // use only 1 public key
// send type, my_ek, my_ekl, iv to the recipient with the ciphertext
```

Poznámka šifrování krátkých zpráv - EVP_PKEY_encrypt / decrypt

`EVP_PKEY_encrypt` a `EVP_PKEY_decrypt` jsou funkce, které umožňují šifrovat krátké zprávy (např. klíč). Pokud to není nutné, nepoužíváme k šiforvání „normálních“ dat. Podívejte se na příklad na manuálové stránce (`man EVP_PKEY_encrypt`). Obsahuje však dvě chyby:

1. `EVP_PKEY_CTX_new` bere 2 parametry místo 1 – předejte `NULL` jako druhý parametr (`EVP_PKEY_CTX_new(pubkey, NULL)`)
2. Konstanta určující typ zarovnání (padding) je špatně v `EVP_PKEY_CTX_set_rsa_padding(ctx, RSA_OAEP_PADDING)`. Buď
    a. nenastavujte žádný padding (použije se výchozí hodnota), nebo
    b. použijte správnou konstantu, např. `RSA_PKCS1_OAEP_PADDING`.