Hybridní šifrování

Termín odevzdání:	**31.08.2024 23:59:59**	5662942.346 sec
Hodnocení:	**10.0000**	
Max. hodnocení:	**10.0000** (bez bonusů)	
Odevzdaná řešení:	1 / 20 Volné pokusy + 10 Penalizované pokusy (-10 % penalizace za každé odevzdání)	
Nápovědy:	0 / 2 Volné nápovědy + 2 Penalizované nápovědy (-10 % penalizace za každou nápovědu)	

Vaším úkolem je realizovat dvě funkce (`seal` a `open`), které šifrují/dešifrují data pomocí hybridního šifrování.

Parametry Vámi implementované funkce `seal`:

```
bool seal(string_view inFile, string_view outFile, string_view publicKeyFile, string_view symmetricCipher)
```

- `inFile` - soubor, který obsahuje binární data určená k zašifrování,
- `outFile` - výstupní soubor, kam uložíte všechny potřebné údaje k dešifrování,
- `publicKeyFile` - veřejný klíč, který bude použit k zašifrování symetrického klíče,
- `symmetricCipher` - název symetrické šifry použité pro šifrování,
- návratová hodnota je `true` v případě úspěchu, `false` v opačném případě. Pokud funkce selže, musíte zaručit, že výstupní soubor `outFile` nebude existovat.

Funkce vygeneruje symetrický (sdílený) klíč a inicializační vektor (dále IV), který bude vstupem do symetrické šifry `symmetricCipher`. Touto šifrou, klíčem a IV zašifrujete data v `inFile`. Klíč k symetrické šifře zašifrujete asymetrickou šifrou (RSA) pomocí veřejného klíče uloženého v `publicKeyFile`.

OpenSSL udělá většinu práce za vás:

- `PEM_read_PUBKEY` - načte veřejný klíč,
- `EVP_SealInit` - vygeneruje sdílený klíč a IV (pokud je potřeba), zašifruje sdílený klíč a nastaví kontext,
- `EVP_SealUpdate` a `EVP_SealFinal` fungují stejně jako v předchozích úkolech.

Hybridní šifrování počítá s šifrováním pro více adresátů. Data jsou zašifrována jen jednou, jedním sdíleným klíčem a IV, ale sdílený klíč může být zašifrován více veřejnými klíči. Proto funkce přijímá pole veřejných klíčů.

Výstupní soubor bude mít následující strukturu:

Pozice v souboru	Délka	Struktura	Popis
0	4 B	int	NID - numerical identifier for an OpenSSL cipher. (Použitá symetrická šifra)
4	4 B	int	`EKlen` - délka zašifrovaného klíče
8	`EKlen` B	pole unsigned char	Zašifrovaný klíč pomocí RSA
8 + `EKlen`	`IVlen` B	pole unsigned char	Inicializační vektor (pokud je potřeba)
8 + `EKlen` + `IVlen`	—	pole unsigned char	Zašifrovaná data

Parametry Vámi implementované funkce `open`:

```
bool open(string_view inFile, string_view outFile, string_view privateKeyFile)
```

- `inFile` - zašifrovaný soubor ve stejném formátu jako je výstupní soubor z funkce `seal`,
- `outFile` - výstupní soubor, kam uložíte všechna dešifrovaná data (je očekávána binární shoda se vstupním souborem do `seal` funkce),
- `privateKeyFile` - privátní klíč určený pro dešifrování zašifrovaného klíče,
- návratová hodnota je `true` v případě úspěchu, `false` v opačném případě. Pokud funkce selže, musíte zaručit, že výstupní soubor `outFile` nebude existovat.

V této funkci budou hlavní roli hrát funkce `PEM_read_PrivateKey`, `EVP_OpenInit`, `EVP_OpenUpdate` a `EVP_OpenFinal`.

Obsah ukázkových dat:

- `PublicKey.pem` - veřejný klíč (schválně ho zkuste otevřít jako txt),
- `PrivateKey.pem` - privátní klíč,
- `sample.cpp` - soubor s deklaracemi a základním testem,
- `sealed_sample.bin` - zašifrovaný soubor, na kterém můžete testovat dešifrování. Byl zašifrován přiloženým veřejným klíčem a po dešifrování v něm naleznete ASCII text. Pokud zašifrujete stejná data, pak soubor nebude stejný jako `sealed_sample.bin` - byl použit jiný klíč a IV.

Rady na závěr:

- V této úloze je hodně míst, kde funkce mohou vrátit chybu. Řádně kontrolujte a zvažte použití objektového návrhu a automatického uvolnění prostředků pomocí `unique_ptr` (platí pro kontext, klíč, alokovaná pole a uzavření souborů).
- Délka zašifrovaného klíče závisí na veřejném klíči. Nelze počítat s pevnou délkou.
- Při kompilaci nezapomeňte přilinkovat openssl crypto library pomocí `-lcrypto`.
- Verze OpenSSL na progtestu je 3.0.11.
- Platí všechny „poznámky“ z předchozího úkolu.

Vzorová data:		**Download**
Odevzdat:	Choose File No file chosen	**Odevzdat**

☐ **Referenční řešení**

1	**23.04.2024 17:57:39**	**Download**
Stav odevzdání:	Ohodnoceno	
Hodnocení:	10.0000	

- **Hodnotitel: automat**
  - Program zkompilován
  - Test 'Zakladni test pro validni vstupy': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 0.394 s (limit: 2.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Nespravne vstupy': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 0.494 s (limit: 3.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test pametove narocnosti': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 1.597 s (limit: 2.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Test 'Test meznich hodnot': Úspěch
    - Dosaženo: 100.00 %, požadováno: 100.00 %
    - Celková doba běhu: 0.044 s (limit: 2.000 s)
    - Úspěch v závazném testu, hodnocení: 100.00 %
  - Celkové hodnocení: 100.00 % (= 1.00 * 1.00 * 1.00 * 1.00)
- Celkové procentní hodnocení: 100.00 %
- Celkem bodů: 1.00 * 10.00 = 10.00

SW metriky:

	Celkem	Průměr	Maximum	Jméno funkce
Funkce:	**14**	--	--	--
Řádek kódu:	**194**	**13.86 ± 8.81**	**29**	`seal`
Cyklomatická složitost:	**58**	**4.14 ± 3.00**	**10**	`HybridCypherController::readConfiguration`