BIT 2024/2025

Úkol 4 - Digitální podpis pomocí ElGamal a SHA-256

Vaším úkolem bude implementovat digitální podpis s využitím algoritmu ElGamal a hashem SHA-256. Hash SHA-256 nemusíte implementovat, využijte knihovny. V rámci ElGamalova algoritmu si vygenerujte všechny potřebné parametry. Všechny zvolené/vygenerované parametry uložte do souborů jako hexadecimální číslo, které pojmenujete jako **`<parametr>.txt`**. Značení bude v souladu s přednáškou tzn. váš archiv bude obsahovat následující soubory:

- **`x.txt`**
- **`y.txt`**
- **`g.txt`**
- **`p.txt`**

Budete digitálně podepisovat vaše osobní (studijní) číslo. Vytvoříte si z něho SHA-256 otisk (pozor! bez znaku konce řádku!) a ten vygenerovaným soukromým klíčem "podepište". Výsledný podepsaný otisk uložte do souboru jako dekadické číslo s názvem **`signature.txt`**.

Zároveň si vytvořte funkci pro ověření tohoto podpisu, tedy použitím veřejného klíče "odemknete" SHA-256 otisk a ten porovnáte s hashem vašeho os. čísla. Vyhodnocení vaši práce bude probíhat právě tímto způsobem (tzn. pomocí verifikačního programu načteme váš veřejný klíč a verifikujeme váš podepsaný otisk v souboru **`signature.txt`**).

Řešení je zcela ve vaší režii. Použijte takové velikosti parametrů (délka v bitech) ElGamalova algoritmu, aby bylo možné podepsat celé 256-bitové číslo otisku jako jeden blok. Nezapomeňte, že číslo *p* má být prvočíslo. Velmi doporučujeme si nastudovat článek [1] a při implementaci postupovat podle něj (pozor značení může být odlišné od přednášky).

Odevzdání

Do odevzdávacího archivu zabalte všechny zdrojové kódy, podepsaný otisk a také všechny vygenerované parametry v souborech **`*.txt`**

Doporučená literatura

[1] Článek https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1057074

[2] Applied Cryptography
https://mrajacse.files.wordpress.com/2012/01/applied-cryptography-2nd-ed-b-schneier.pdf
sekce **19.6 ElGamal**