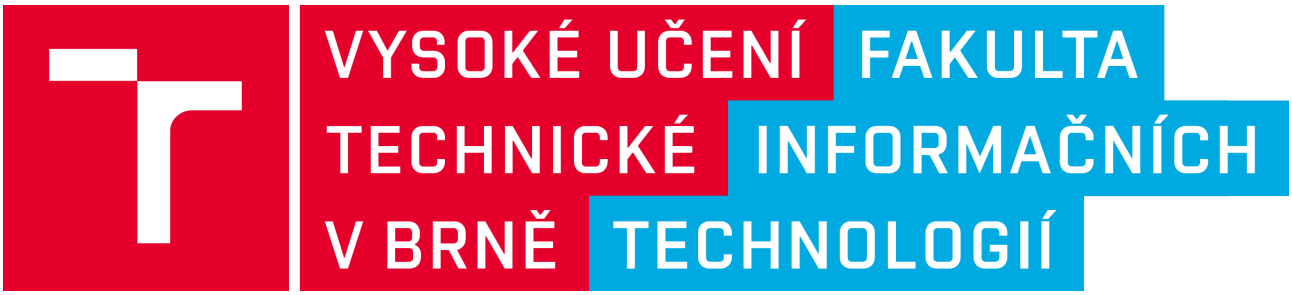


Dokumentácia

KRY: Hybridné šifrovanie

Filip Brna, xbrnaf00, 221923

24. apríla 2023

1 Hybridné šifrovanie

Hybridné šifrovanie, alebo hybridný šifrovací systém, je kryptografická technika, ktorá kombinuje asymetrické a symetrické šifry.

Asymetrické šifry umožňujú použitie rôznych kľúčov na šifrovanie a dešifrovanie, čo znamená, že odosielateľ a prijímateľ nemusia zdieľať žiadne tajomstvo vopred. Toto však môže mať za následok nižšiu rýchlosť, pretože asymetrické šifry vyžadujú zložitejšie matematické výpočty. Naopak symetrické šifry sú rýchlejšie.

Hybridný systém kombinuje výhody oboch prístupov tým, že najprv náhodne vygeneruje kľúč pre symetrickú šifru a zašifruje ním správu. Potom samotný kľúč zašifruje asymetricky a spolu so zašifrovanou správou ho pošle prijímateľovi. Prijímateľ potom asymetrickou šifrou dešifruje kľúč a následne pomocou kľúča pre symetrickú šifru dešifruje aj samotnú správu. Týmto spôsobom je pomocou pomalšej asymetrickej šifry šifrovaný len krátky kľúč, zatiaľ čo samotná správa, ktorá môže byť veľmi dlhá, je šifrovaná rýchlejšou symetrickou šifrou. Bezpečnosť tohto systému je závislá na bezpečnosti oboch použitých šifier. V projekte boli využité algoritmy AES 128 bit, RSA 2048 a MD5 hash.

AES 128 bitov je štandardná symetrická bloková šifra, ktorá používa 128-bitové kľúče na šifrovanie a dešifrovanie dát. Je to veľmi rýchla a bezpečná šifra, ktorá sa často používa na zabezpečenie dát vo výpočtových systémoch a komunikačných kanáloch. 128-bitový kľúč poskytuje 2^{128} možností, čo je obrovské množstvo rôznych kľúčov, ktoré by bolo potrebné prehľadať pri bruteforce (hrubou silou) útoku. Z praktického hľadiska je tento počet tak obrovský, že by bolo veľmi nepravdepodobné, že by útočník dokázal úspešne prelomiť AES s 128-bitovým kľúčom.

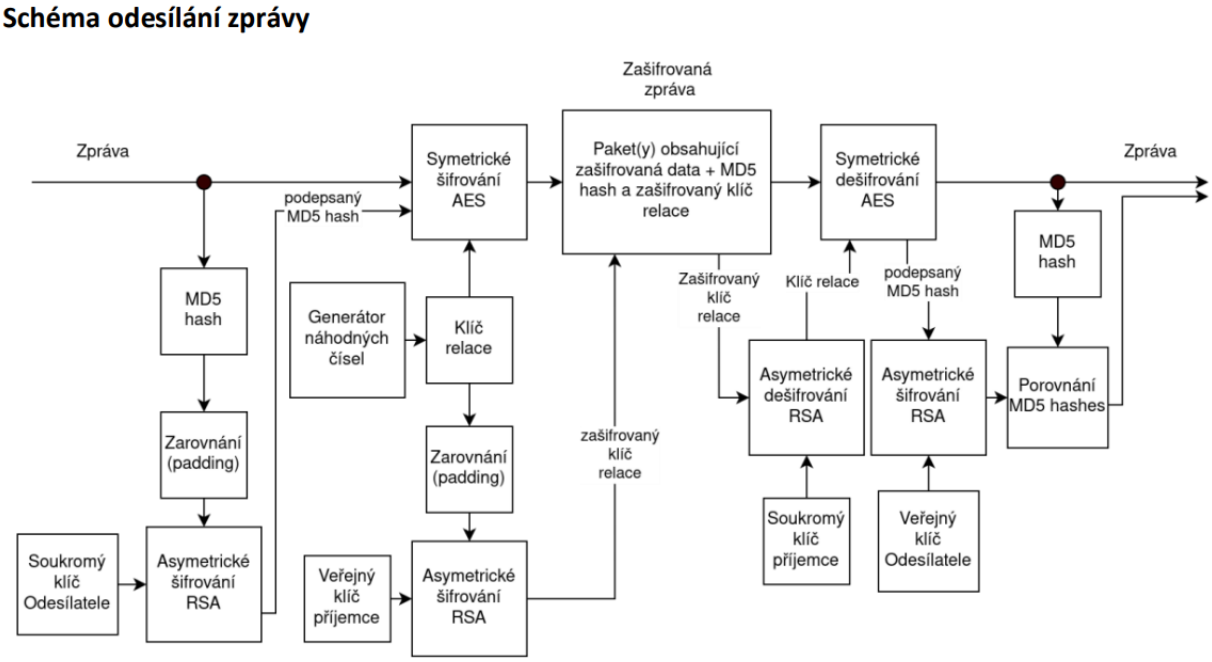
RSA 2048 je asymetrická šifra, často používaná v hybridnom šifrovacom systéme. V hybridnom šifrovaní sa RSA využíva na zabezpečenú výmenu symetrického kľúča medzi odosielateľom a prijímateľom. RSA šifra sa používa na výmenu symetrického kľúča, ktorý sa potom používa na rýchle a efektívne šifrovanie a dešifrovanie dát. RSA s 2048-bitovým kľúčom je považovaný za bezpečný pre šifrovanie dát na súčasnom hardvérovom a softvérovom vybavení. 2048-bitový kľúč poskytuje vysokú úroveň bezpečnosti, ktorá je dostačujúca pre väčšinu aplikácií. Pre RSA existujú aj rôzne metódy útokov, napríklad faktorizácia, ktorá sa snaží nájsť prvočísla použité na vytvorenie kľúča. Avšak pri dĺžke kľúča 2048 bitov je tento proces veľmi časovo náročný a na súčasnom hardvérovom vybavení by bol prakticky nemožný.

MD5 je hashovací algoritmus, ktorý sa može používať na vytvorenie jednosmerného odtlačku (hashu) z dát, čo umožňuje jednoduchú a rýchlu kontrolu integrity. V hybridnom šifrovaní sa MD5 môže použiť na rýchlu kontrolu integrity symetrického kľúča alebo správy, ktoré sa vymieňajú medzi odosielateľom a prijímateľom.

2 Implementácia hybridného šifrovania

Program **./kry** je naimplementovaný v programovaciom jazyku Python je možné ho spustiť postupnými príkazmi **make build**[1], následne **make run**[2]. Ide o architektúru klient-server na localhoste, ktorá umožní posielať správy zašifrované pomocou vyššie spomenutých algorimov. Všetky potrebné kľúče odosielateľa aj príjemcu pre RSA algoritmus sú uložené v zložke **cert**, v prípade, že zložka neexistuje, bude vytvorená po spustení programu a taktiež do nej budú vygenerované súkromne aj verejné kľúče odosielateľa a príjemcu. Kľúč pre symetrické šifrovanie je náhodne vygenerovaných 16B po spustení klienta.

Kľúče sú využívané tak, ako je to možné vidieť na nasledujúcej shéme, paket okrem spomínaných dát obsahuje aj inicializačný vektor, potrebný pre AES dešifrovanie s módom EAX:



Obrázok č.1 : popisuje schému odosielania správy, obrázok bol prevzatý zo zadania projektu, ktorého autorom je Ing. Daniel Snášel.

Výmena symetrického kľúča je zabezpečená rovnako ako je zobrazené na schéme vyššie, verejné kľúče odosielateľa a príjemcu sú voľne dohľadateľné napr. na internete a odosielateľ spolu s príjemcom vlastnia jedine kópie svojich súkromných kľúčov, ich výmena nebola potrebná. Z odosielanej správy je najprv vygenerovaný jej MD5 hash(16B), ktorý je následne doplnený o náhodných 240B a pomocou Asymetrického šifrovania a súkromným kľúčom odosielateľa vytvorený podpísaný MD5 hash, zároveň je vygenerovaný kľúč relace, ktorým je pomocou symetrickej šify AES zašifrovaná správa spolu s podpísaným MD5 hashom. Kľúč relace je taktiež zarovnaný na 256B a algoritmom

[1] make build- vytvorí virtualne prostredie s nazvom "venvä nainštaluje všetky potrebné knihovny zo súboru requirements.txt

[2] make run- spustí program so zadanými parametrami

RSA spolu s verejným kľúčom príjemcu je vytvorený zašifrovaný kľúč relace. Tieto šifrované dáta sú následne spolu prenášané k príjemcovi, ktorý musí najprv zistiť kľúč relace a to asymetrickým šifrovaním za pomoci súkromného kľúča príjemcu. Získanému kľúču relace je odstránené zarovanie(240B) a následne je ním symetricky dešifrovaná samotná správa a podpísaný MD5 hash správy, tento hash je získaný pomocou RSA a verejného kľúča odosielateľa. Nakoniec je vygenerovaný MD5 hash zo získanej správy a porovnaný s hashom získaným pomocou RSA(hash má taktiež odstránené zarovanie) v prípade, že sa zhodujú, nedošlo k narušeniu integrity správy, v opačnom prípade bola integrita porušená. Server po prijatí správy zasiela klientovi potrdzujúcu správu, v prípade, že nebola integrita narušená, inak zasiela správú o narušení integrity. V prípade porušenej integrity klient zasiela dáta ešte jeden krát.

príklady spustenia:

Klient
make run TYPE=c PORT=54321
Server
make run TYPE=s PORT=54321

Pri spustení a zasielaní/prijímaní správ sú na štandardný výstup vypisované všetky zadaním požadované informácie, dáta sú prevažne vypisované v šestnástkovej sústave. Správy je možné vkladať v nekonečnej smyčke, ukončenie je možné ”zaslaním”prázdnej správy.

Záver
Hybridné šifrovanie s využitím AES so 128-bitovým kľúčom a RSA 2048-bitovým kľúčom je považované za bezpečný spôsob šifrovania dát, ktorý poskytuje vysokú úroveň ochrany pred útokmi. Avšak je vždy dôležité zohľadniť aj iné faktory, ako napríklad správne použitie šifrovania, bezpečnosť kľúčov a správu kryptografických materiálov pre zabezpečenie úplnej ochrany systému.