Šifrovanie

Nové funkcie používané v protokole

- *Doplň správu:* $A^n \rightarrow A^{40}$

  Doplní náhodnú správu použitím funkcie *Náhodná správa* pred pôvodnú správu, tak aby bola výsledná dĺžka 40 znakov. Dĺžku 40 som zvolil podľa toho, že najdlhšia predpísaná správa je dlhá 26 znakov, teda vždy je treba doplniť viac ako 10 znakov. Tieto znaky sú používané pri menení kľúča. Ak správa obsahuje argument (*POSLETE MI n*), a ide zo skupiny A do skupiny B, namiesto náhodného textu dopĺňa prvé znaky *Otisku* argumentu správy, tak aby bola výsledná dĺžka 40 znakov a až za tým následuje náhodne vygenerovaný text (ak je potrebný).
- *Prečítaj správu:* $A^{40} \rightarrow A^x$

  Prečíta jednu z 5 zadaných správ zo 40 znakovej doplnenej správy, ak sa v správe žiadna predpísaná správa nenachádza, vráti prázdny reťazec znakov. V prípade, že správa obsahuje argument a bola odoslaná zo skupina A do skupiny B, *Otisk* argumentu správy sa musí zhodovať s prvou časťou doplnenej správy. Ak sa nezhoduje vráti prázdny reťazec znakov
- *Prečítaj doplnenú správu:* $A^{40} \rightarrow A^x$

  Prečíta zo správy časť ktorá bola doplnená funkciou *Doplň správu* (časť správy ktorá nieje v tvare predpísaných správ). V protokole sa táto funckia spolieha na to, že správa nebude pozmenená nepriateľom.

Protokol

Na začiatku si vygenerujem dva kľúče pomocou funkcie *Náhodná zpráva*, s dĺžkou 20 znakov. Tieto kľúče budú pre nepriateľa neznáme, pretože si ich tučniaci vygenerujú ešte pred výpravou. Zároveň sa dohodnú, že kľúč 3 bude upravovaný počas ich komunikácie, aby neboli rovnaké správy zašifrované vždy rovnako. Na začiatku bude jeho hodnota prázdny reťazec znakov.

Vzájomná výmena správ bude prebiehať v 4 krokoch.

- Skupina A novú správu doplní funkciou *Doplň správu*. Následne doplnenú správu zašifruje funkciou *Zašifruj* s kľúčom *kľúč1.kľúč3*. (Bodku budem používať ako symbol pre spojenie reťazcov znakov)
- Skupina B správu dešifruje funkciou *Dešifruj* s kľúčom *kľúč1.kľúč3* a prečíta pôvodnú správu funkciou *Prečítaj správu*. Následne vytvorí správu v tvare *POSLETE MI 1* ak sa im podarilo správu prečítať, v prípade že funkcia vrátila prázdny reťazec znakov vytvorí správu *POSLETE MI 2*, čo znamená, že správa bola upravená nepriateľom. Správu POSLETE MI 2 vytvorí aj v prípade, že správa zo základne A nedorazila. Túto správu doplní funkciou *Doplň správu*, zašifruje funkciou *Zašifruj* s kľúčom *kľúč2.kľúč3* a odošle ju skupine A. Ak by sme tu pužili kľúč 1, namiesto kľúču 2 tak by nepriateľ mohol vidieť podobnosť v správach, ak by boli prvá aj druhá správa rovnké (napríklad *POSLETE MI 1* a *POSLETE MI 1*). Teda by mohol zistiť obsah poslanej správy.

  Keďže správy z B do A nemôžu byť pozmenené nepriateľom, je tu priestor na zmenu kľúča 3. Do kľúču 3 bude po odoslaní správy pridaný reťazec znakov z funkcie *Prečítaj doplnenú správu* z nezašifrovanej správy. Ak bude kľúč 3 dlhší ako 20 znakov skráti sa na jeho posledných 20 znakov.

  Je dôležité tento krok spraviť až po zašifrovaní správy, pretože inak by skupina A nedokázala dešifrovať túto správu správne, keďže by správa bola zašifrovaná iným kľúčom ako má k dispozícii skupina A.
- Skupina A dešifruje správu pomocou funkciu *Dešifruj* s kľúčom *kľúč2.kľúč*3 a podľa jej obsahu zistí či predchádzajúca správa bola úspešne doručená. Táto základňa si tým istým

spôsobom po dešifrovaní zmení kľúč 3 pridaním reťazcu znakov z funkcie *Prečítaj doplnenú správu* z dešifrovanej správy. Ak bude kľúč 3 dlhší ako 20 znakov skráti sa na jeho posledných 20 znakov. Skupina pošle nezašifrovanú a neupravenú správu *POSLETE MI 1*. Táto správa slúži čisto na predanie „slova“ a nijak by nám neprekážalo, ak by ju nepriateľ upravil.

- Tento raz skupina B nebude dešifrovať príchodziu správu, pretože je bezobsažná. Vytvorí novú správu, doplní ju funkciou *Doplň správu*, zašifruje ju funkciou *Zašifruj s kľúčom kľúč1.kľúč3* a pošle. Následne kľúč 3 zmením podľa postupu ktorý som opisoval v predchádzajúcich krokoch. V najbližšom kroku túto správu spracuje základňa A, následne si upravý podľa tejto správy kľúč 3 už spomenutým spôsobom a výmena sa môže opakovať.

Zdvôvodnenie bezpečnosti protokolu

- Funkcia *Zašifruj* šifruje správu na novú správu tej istej dĺžky, takže sa dá zistiť podľa dĺžky predpísaných správ, ktorá správa je posielaná. Môj protokol tomuto útoku zabraňuje doplňaním náhodných správ funkciou *Doplň správu* ku správam, tak aby mala každá správa konštantnú dĺžku 40 znakov.
- Ak by sa kľúč počas komunikácie nemenil, nepriateľ by mohol zistiť, ktoré správy majú význam a posielať ich opakovane. Tučniaci by potom nevedeli ktoré správy sú od druhej skupiny a ktoré od nepriateľa. Tomuto útoku sa bránime menením kľúča každou správou zo skupiny B do skupiny A a dvoma kľúčmi. Nikdy teda nebudú poslané 2 správy s obsahom zašifrované tým istým kľúčom. Keďže nepriateľ nepozná pôvodný kľúč, nikdy správu správne nedešifruje, čo znamená že nikdy nebude poznať ani meniaci sa kľúč.
- Keďže číslic je 10, je celkom pravdepodobné, že nepriateľovi by sa mohlo po nejakom čase podariť zmeniť argument správy *POSLETE MI n.* Preto používam *Otisk* pôvodného argumentu na začiatku správy z A do B ako dôkaz, že argument nebol upravený.
- Ak nepriateľ zmení správu idúcu z A do B, skupina A sa vždy dozvie, že správa nebola úspešne doručená pomocou správy *POSLETE MI 1*.

Útok na ukážkový protokol

Keďže funkcia *Zašifruj* vráti správu s rovnakou dĺžkou ako pôvodná správa, je jednoduché zistiť o aký typ správy ide. Napríklad správa VYCKEJTE je jediná správa s dĺžkou 8 a POKRACUJEM je jediná správa s dĺžkou 10 (lebo POSLETE MI n musí obsahovať argument). Napríklad ak by tučniaci poslali správu s dĺžkou 10, vedeli by sme, že ide o POKRACUJEM a zapamätali by sme si zašifrovanú správu. Potom keď by chcela skupina odoslať správu VYCKEJTE (čo by sme videli ako 8 znakovú správu), zmenili by sme správu na 10 znakovú správu, ktorú sme si zapamätali (teda POKRACUJEM). To by samozrejme viedlo k nebezpečnej akcii výpravnej skupiny. Takto by sme mohli kedykoľvek spraviť opak toho, čo mali tučniaci v pláne.