Cryptography Notes [SK]

Lessons Learned

Ladislav Šulák <laco.sulak@gmail.com>

April 28, 2021

Contents

Part I.

Basics

Kryptografia - transformacia plaintext <-> encrypted text

Kryptoanalyza - transformacia encrypted -> plaintext bez znalosti kluca

Kryptologia - oboje spolu

Bezpecnostne funkcie

- Dovernost - ochrana pred neopravnenym odhalenim
- Autentizacia - ochrana pred neopravnenym pouzitim

Utoky

- Ciphertext only attack - utocnik pozna sifrovany text, snazi sa zistit kluc/plaintext
- Known plaintext attack - utocnik pozna sifrovany text a odpovedajuci plaintext, snazi sa zisti kluc
- Chosen plaintext attack - ako predchadzajuci, no plaintext si mohol zvolit

Kerckhoffuv princip - 19. st. - bezpecnost musi zavisiet iba na utajeni kluca, je treba predpokladat, ze utocnik pozna vsetky podrobnosti o algoritme.

Part II.

Classical Cryptography

1. Steganograficke sifry

Prenasany text ukryty v inom texte. Niektore bity su spravy, niektore vata. Pomocou kluca sa zisti, ktore je ktore. Sifrovany text je dlhsi (niekedy vyznamne) nez otvoreny text.

2. Transpozicne sifry

Menia poradie znakov v texte.

Scytale - v Grecku, pomocou pruzku papieru na tyci. Sprava sa zapisuje na omotany pruzok papieru po riadkoch, na pruzku su zdanlivo nahodne znaky. Nie je prilis bezpecna, klucom je iba priemer tyce.

Red Fence - vezmeme kluc, jednotl. pismena ocislujeme podla ich abecedneho poradia v kluci. Plaintext napiseme postupne na jednotlive riadky a zasifrovany text ziskame citanim riadkov podla ich poradi.

Napriklad kluc THEYKEY - ocislujeme ako 531426 (napr E su tam 2x, sice je najskorsie, teda '1', jeho druhy vyskyt nebude '1' ale uz '2'). Plaintext zapiseme kaskadovito:

T K

H S E

I Y

Zasifrovany text potom piseme postupne citanim riadkov podla ich poradia. V priklade vyssie ak by sme mali kluc KEY = 213, tak by sme mali zasifrovany text ako HSETKIY.

Rail Fence - zanedbava poradie, kluc urcuje iba pocet riadkov. Teda potom citame riadky jeden za druhym postupne zhora dole.

Stlpcova transpozicia - sprava sa zapise po riadkoch a sifrovany text sa vytvori tak, ze sa cita po stlpcoch v danom poradi, napr matica 5x4 a kluc je 4-2-1-3.

Dvojita stlpcova transpozicia - prva faza ako predchadzajuci, potom pre jednoduhsie zapamatanie kluca je mozne mu priradit slovo, kluc je potom poradie pismen slova v abecede.

3. Substitucne sifry

Nahradzuju jednotl. znaky textu inymi (mono/poly-alfabeticke). Priklady:

- **Freemason** - abeceda sa sklada z akychkolvek symbolov.
- **Caesarova** - kazde pismeno nahradil 3. dalsim v abecede. Vylepsenie - odosielatel a prijemca maju dohodnuty kluc (1-25), ktory znamena, o kolko pismen sa posuva. Sifra je slaba - stavovy priestor klucov je moc maly. Utoky:
  - utok silou - iba 26 moznosti
  - frekvencna analyza
- **Monoalfabeticke** - jednotlive pismena nahradzuju pomocou kluca, ktorym je permutace vs. 26 pismen. Pocet vs klucov je 26!

  Zapamatovatelny kluc - specialny pripad, klucom je keyphrase (niekolko slov). Pouzitie - napis kluc, pod nim postupne do stlpcov po riadkoch zapis pismena abecedy ale vynechaj tie co uz su (z kluca) a potom citaj vysledok po stlpcoch.

  Anatomia jazyka - frekvencia, patterny - pravdepodobne (dalsie/predchadzajuce) slovo, opakovanie slov, slova z daneho kontextu.

  Zvysenie odolnosti:
  - uriznuti vrcholov tabulky frekvencii - **homofonne sifry**, prevadzaju 1 znak textu na niekolko inych dalsich
  - sifrovanie skupin znakov (napr. dvojic) - symbol abecedy sa sklada z viac znakov. **Polygramove sifry** - nahradzuju skupinu znakov inou skupinou znakov.
  - kazdy znak sa sifruje inou transformaciou - **polyalfabeticke.**

  Nomenklator - 16. st. - obsahuju klasicke substirucne homofonne zameny a klamace (nevyznamove skupiny pismen, kt. mali stisit kryptoanalyzu zasifrovanych textov).

  Kodova kniha (codebook) - monoalfabeticka polygramova sifra (casto homofonna) - Dreyfusov alebo Zimmermanov telegram.

Playfair - 25 znakov (okrem 'j') pocinaje heslom, je vpisanych do 5x5 stvorca (torus). Ak sa digram plaintextu nachadza v 1 riadku (stlpci) je nahradeny pismenami od neho vpravo (dole), inak je 1. pismeno nahradene pismenom v rovnakom riadku ale v stlpci druheho znaku a naopak (zaciname tym na riadku). Ak je v texte 2-ica rovnakych znakov, je 1 vynechany. Desifrovanie presne naopak, riadok - vlavo, stlpec - hore.

Ma urcite charakteristika a vieme ju na zaklade nich lepsie identifikovat:

- Je to teda substitucna sifra, niektore spoluhlasky v EN sa vyskytuju castejsie v plaintexte a aj ich digramy.
- V sprave je vzdy sudy pocet pismen a neobjavuju sa v nich dvojice rovnakych pismen (SS, EE, ...). Ak prideme k tomu, ze je v texte 2ica rovnakych znakov a 1 je vynechany, bude lichy pocet znakov. Bigram oddelime nejakym velkym pismenom napr X, na koniec spravy potom tiez pridame X.
- Ziadne pismeno nemoze byt zasifrovane samo na seba.
- 2 reverzovane digramy v plaintexte (ER a RE) budu vzdy zasifrovane ako 2 reverzovane digramy v zasifrovanom texte.
- Opakovanie a frekvencia digramov je obecne rovnaka ako u ich ekvivalentov v plaintexte.
- Kazde pismeno v plaintexte moze byt zasofrovane iba ako 1 z 5 pismen - 1. ktore je hned pod nim v rovnakom stlpci a 4, kt. su v rovnakom riadku. Vysledkom je, ze 3/4 najcastejsie pismena sa nachadzaju v rovnakom riadku ako 'e'.

- **Polyalfabeticke** - kazdy znak zasifrovany inou substitucnou funkciou - splostenie frekvencnej charakteristiky jazyka, nelze pouzit frekvencnu analyzu ani ine statisticke metody.

  Leon Battista Alberti - Albertiho disk - 1. polyalfabeticky sifrovy system, doporucoval posunut abecedy po 3/4 slovach.

  Johannes Trithemius - Kniha Polygraphia (1518). Tabula recta - 26 riadkov, prvy zacina od pismena A po Z, druhe od B po A, teda vzdy posuv o 1 pismeno.

  Vigenerova sifra - 1550, nezlomitelna cca 300 let. Diplomat vo Vatikane.

  - Pouziva Caesarovho principu - rozdielne posuvy pre jednotl. znaky aby sa zakryla frekvencia znakov. Znaky kluca definuju posuv pre jednotlive pismena a kluc je periodicky opakovany, aby obsiahol celu dlzku sifrovaneho textu. Vigenerova tabulka - ako Tabula recta od Trithemiusa. Vylepsenie by bolo, keby sme kluc neopakovali periodicky, ale bol by 1 v dlzke rovnakej ako text, ktory chceme sifrovat.
  - Autokluc - mechanizmus, kedy je domluven iba pociatocny kluc (casto iba 1 pismeno), ktory je potom modifovany:
    - a) samotnou spravou (autokey plaintext),

  - b) zasifrovanou spravou (autokey encrypted text).

  Napriklad kluc D, plaintext PES, autokluc bude DPE. Dnes sa nepouziva scitanie ale XOR.

- Najprv rozlustena Ch. Babbagom, postup bol ale utajovany - vyuzitie opakovania pre uhadnutie dlzky kluca - postupnost XFO sa nachadza na poziciach 65, 71, 122, 176. Vzdialenosti su - 71-65=3*2, 122-65=3*19, 176-122=3*18, kluc ma teda pravdepodobne dlzku 3 znaky.
- Nezavisle rozlustena F. Kasiskim, 1863
  - zalozene na pozorovani, ze 2 identicke useky plaintextu budu zasifrovane na rovnaky encrypted text vtedy, ak su vzdialene delta pozid od seba.
  - cielom je najst niekolko identickych usekov, kazdy o dlzke aspon 3 a zaznamenat si vzdialenost medzi ich pociatkami. Dlzka kluca m deli vsetky vzdialenosti delta_1 az delta_n. Potom je m NSD vsetkych vzdialenosti delta_i.
  - napriklad trigram TYZ sa vyskytuje na poc. poziciach 25, 181 a 235. Vzdialenost medzi prvou a druhou je 156, medzi prvou a tretou je 210. GCD je 6, takze mozeme predpokladat ze |key|=6.
- Desifrovanie
  - Najdi dlzku kluca k - pre kratky kluc skus 1, 2, 3,... alebo vytvor tabulku vsetkych vzdialenosti rovnakych znakov v encrypted text. GCD najcastejsich vzdialenosti je dlzka kluca.
  - Najdi pismena kluca jedno po druhom - rozdel spravu na k mensich sprav, kazda obsahuje znaky sifrovane stejnym pismenom kluca. Ries ako k sprav zasifrovanych Caesarovou sifrou.
- Index koincidence - dalsia moznost kryptoanalyzy. Je definovany ako retazec n znakov a udava pravdepodobnost, ze 2 nahodne prvky z retazca su identicke. Vypocitame vzorec pre priemerny index koincidencie.

Sifrovacie strojky

- Saint-Cyrovo pravitko
- Jeffersonov cylindr - 1790, skladal sa z 36 diskov, kazdy s nahodnou abecedou. Poradie diskov bolo klucom, v 1 rade s anastavila sprava, v inej sa precitala sifrovana sprava.
- Wheatstonov disk - 19. st. - 2 sustredne kruhy, ktore generovali polyalfabeticku sifru.
- Rotorove stroje

  Zalozene na jednoduchej substitucii - horizontalne, vertikalne posuvana abeceda, rotovana abeceda. Implementuju polyalfabetifke substitucne sifry s dlhou periodou pomocou sady rotorov. Kazdy rotor ma 26 kontaktov na oboch

stranach. Kontakty z prednej strany su prepojene s kontaktami zo zadnej strany. Kluc je dany prepojenim kontaktov a pociatocnej pozicii rotorov.

Plaintext vstupuje do sady rotorov na 1 strane a vystupuje zasifrovany na druhej strane. Po zasifrovani pismena sa pootoci 1+ rotorov.

- Enigma

  3 rotory (vybrane z 5 moznych). Po kazdom znaku sa 1. rotor pootoci. Po dojeti na zarazku sa pootoci dalsi rotor. Kluc - pociatocna orientacia 3 rotorov (Grundstellung), posunutie pociatocnej orientacie a nastavenie, kedy sa pootoci dalsi rotor (Ringstellung).

  * Reflektor - na konci posledneho rotora, zvoli inu cestu spat az na vystup.
  * Prepojovacia doska (Plugboard, Steckerbrett), kluc - vymena parov znakov pomocou kablu, pocet kablov sa menil.
  * Pocet moznych klucov:
    - ak utocnik nepozna prepojenie rotorov - $26!^3$a ani prepojenie reflektorov:$(26 * \frac{25}{2}) * (26 * \frac{25}{2}) ... * (2 * \frac{2}{2})/13!$
    - prepojovacia doska so 6 kablami: $(26 * \frac{25}{2}) * ... * (16 * \frac{15}{2})/6!$
    - rotory Ringstellung: 26^2, Grundstellung: 26^3
    - **Celkom** $6 * 10^{110}$(viac nez atomov vo vesmiru)

  * Ak ukoristime 1 stroj: pozname rotory: 3!, alebo 3 z 5 = 5*4*3 = 60. Znamy reflektor: 1. Ostatne ostava rovnake, teda okolo 10^{16}.
  * Sposob provozu - denny kluc - distribuovany pomocou kodovej knihy. Kazda sprava zacina prenosom kluca spravy (Grundstellung iba) "nahodne" vytvorenym odosielatelom zasifrovanym dennym klucom. Kluc je poslany 2x pre zamedzenie moznej chyby. Po prijati kluca spravy sa pootocia rotory do pozicii kluca spravy.

Vernamova sifra

- bez opakovania kluca, objavena G. Vernamom z AT&T v 1917 pre sifrovanie telegrafnich sprav.
- plaintext XOR kluc = sifrovany text
- je nerozlustitelna, ak ma kluc rovnaku dlzku ako vsetky sifrovane spravy a nikdy sa nepouzije znovu a je nahodne zvoleny.
- C. E. Shannonom prokazane ze je to absolutne bezpecny sifrovy system.
- pouziva v case kubanskej krizi v 1963 pomocou ETCRRM (El. Teleprinter Crypt. Regenerative Repeater Mixer II).

– DEINSTAR - varianta, text sa tabulkou previedol na 1-2 miestne ciselne symboly a k tym sa pricital kluc (ak je prechod cez 10, zahadzuje sa to - 9+5=4).
– Co nie je Vernamova sifra: Autokey, ani proudova sifra.

Part III.

Modern Cryptography

4. Symmetric Cryptography

Tajny kluc - zdielane tajomstvo.

Utoky - Ciphertext only attack, Known plaintext attack, Brute force.

Bezpecnostne funkcie

- Dovernost: **ANO** - ochrana proti neopravnenemu precitaniu informacie.
- Autentizacia: **ANO** - vieme, od koho ta sprava prisla, ale za rovnakeho predpokladu ako v Integrite.
- Integrita: **ANO**, iba ak sprava obsahuje nejaku redundanciu a podla nej poznam, ci je to zmysluplna sprava, napriklad XML; integer nema ziadnu redundanciu. Ale mozeme pridat napriklad CRC v zasifrovanej sprave.
- Nepopieratelnost: **NIE** - nepozname, kto klame - obe strany vedia, kto podvadzal, no nevieme dokazat kto, pretoze tajomstvo poznaju obaja)

Dlzka kluca:

- 32b - 38m
- 48b - 4,5 roku
- 56b - $>$ 100 rokov (10h u 10^6 procesorov, rok 1999 - 100,000 proc - 22h)

Je moznost vytvorit novy algoritmus.

4.1. Prudove Sifry

Spravu beru po bajtoch/bitoch.

Synchronne - pseudonahodna postupnost (key stream) je nezavisla na sprave. Ciphertext obvykle spocitany ako plaintext XOR hodnota key streamu. Napriklad Vernamova sifra, alebo DES v OFB.

Samosynchronizujuce - kazdy bajt pseudonahodnej postupnosti je zavisly na pevnom pocte predchadzajucich bajtov sifrovaneho textu. Umi sa resynchronizovat, napriklad Autokey, alebo DES v CFB.

Problemy - je treba dobre volit IV - nesmie sa pouzit rovnaky kluc pre sifrovani 2 sprav a je treba dat pozor na integritu. Utocnik moze zmenit dolezity obsah spravy iba preklopenim bitov.

Generatory PRNG, metody:

- Blokove sifry v OFB - pomale - 1 sirovanie / byte.
- Linearne kongruencne generatory - nie prilis bezpecne. Kryptograficky slabe dokonca aj s pouzitim kvadratickej alebo kubickej funkcie.
- LFSR (linear feedback shift registers) - rychle, nie vzdy bezpecne. Maximalna perioda: 2^{n-1}, dosiahneme ju ak je polynom T(x) primitivny.

  Utoky - KPA, je treba 2n bitov plaintext/ciphertext. Je treba xorovat M a C a tym ziskame 2n bitov key streamu. Tie jedinecne urcuju zpatne vazby - urci sa linearnou funkciou. Na utok je treba cas $O(n^3)$
- Kombinovane generatory - niekolko LFSR pre zvysenie bezpecnosti. Mozu mat rozne dlzky a rozne zpatne vazby, tiez rozne hodiny (hodinovy IN moze zxavisiet na OUT ineho - feed forward, alebo sama sebe - feedback).

  Sposoby kombinacie - XOR, Multiplexor - Gefe Generator (1 LFSR bezi rychlejsie nez ostatne a urcuje, z kt. z ostatnych vezme nasledujuci bit), Threshold (vystupom generatoru je bit odpovedajuci majorite bitov jednotl. generatorov).

A5 - v GSM pre zasifrovanie dat prenasanych radiovym kanalom. 3 LFSR s dlzkami 19, 22 a 23 - vystupy su XORovane. Registry nie su prilis dlhe a kazdy ma hodiny zo svojho prostredneho bitu XORovaneho vystupom funkcie (threshold) prostrednych bitov vsetkych 3 registrov.

Bezpecnost - utok silou 2^64 (alebo skor 54). Utok na 2 mensie registry 2^{45}. Skutocny utok - su treba 2s plaintextua 2^{48} predvypoctov, 148 GB priestoru. Doba utoku 1 minuta na PC.

Prudova sifra Bluetooth - dlzka kluca 128 bitov, najlepsi znamy utok 2^{70}. Pouziva 4 LFSR, ktore XORuje.

PKZIP - prudova sifra, berie text po bajtoch. 32 bitove hodnoty, 3 magicke cisla. Nedostatocna bezpecnost, nevhodne k pouzitiu.

RC4 - 1987. Premenna dlzka kluca. Pouziva S-box s prvkami S_0 az S_{255}, ktore su permutaciami 8bitopvych cisel 0...255.

Slabiny - nezaistuje integritu - je treba zaistit inak, napr MAC. Dalej problem pri zasifrovani 2 sprav rovnakym klucom - modifikacia kluca.

Utok - pociatocny keystream nie je dostatocne nahodny. Ak je kluc iba konkatenovany s nahodnym cislom, lze ho rekonstruovat. Pouzite pre rozlomeni sifrovani WEP pri WiFi sieti.

4.2. Blokove Sifry

Spravu berie po vacsich blokoch (64/128/256 bitov). Pred zasifrovanim musi byt cely blok k dispozicii, vacsia velkost zabranuje slovnikovym a statistickym utokom.

- Pre 64 bitovy blok - 2^64 moznych blokov otvoreneho textu a aspon tolko isto odpoved. blokov zasifrovaneho textu.
- Idealne nahodne zobrazenie aproximujeme pomocou niekolko komponent, riadenych hodnotou kluca.
- rezimy blokovych sifier mozu byt pouzite pre akukolvek blokovu sifru
  - ECB (Electronic Codebook) - umoznuje - slovnikove utoky, repeticie alebo preskladanie blokov Spravu rozdelim na bloky, kazdy sifrujem nezavisle. Dlzka spravy musi byt nasobkom velkosti bloku. Nie je dobre spravu doplnovat nulami. Jednoducha implementacia.
  - CBC (Cipher Block Chaining) - kazdy blok spravy je XOR-ovany s predch. zasifrovanym blokom a 1. blok je XOR-ovany IV (nemusi byt tajny, no je treba chranit jeho integritu - jeho modifikacia utocnikom mu umoznuje cielene zmenit 1. blok). Odolne voci slovnikovym utokom, repetici/preskladania bloku.
    - Padding: Doplnenie dodatocnych blokov na koniec, ak dlzka spravy nie je nasobkom dlzky bloku.
    - Musi byt jednoznacne! Jednoznacne znamena vzdy predlzenie aspon o 1 byte, mal by niest dlzky zarovnania (napr zarovname o 5 bytov, tak v poslednom byte je '05')
  - OFB (Output Feedback) - synchronna prudova sifra (DES - pri generovani pseudonahodnej postupnosti, key stream). DES je inicializovany IV. IN blok DESu je posunuty dolava o 'k' bitov a sprava je doplneny 'k' bitmi z vystupneho bloku. Typicky k=8 alebo 64. Ten isty key stream nesmie btyt viackrat pouzity! Obrazok vyzera tak, ze do blokov E prichadza vzdy ten isty kluc K a najprv IV, ale v dalsich uz vysledok z laveho (predchadzajuceho) bloku E. Vysledok z bloku E sa vzdy XORuje s Pi a vysledkom je Ci.
    - CTR je podobne, citac sa nesmie pre rovnaky kluc opakovat a kazdy blok je sifrovany nezavisle na ostatnych, paralelizovatelny.
  - CFB (Cipher Feedback) - samosynchronizujuca prudova sifra (DES - pri generovani pseudonahodnej postupnosti, key stream). DES je inicializovany IV. IN blok DESu je posunuty dolava o 'k' bitov a sprava je doplneny 'k' bitmi z ciphertextu. Typicky k=8 alebo 64 - blokova sifra sa vlastne prevadza na prudovu - obrazok vyzera ako v OFB, ale do bloku E sa pridava vysledok po predchadzajucim XORe, nie po predchadzajucim bloku E.

Feistelova sifra

Vznik IBM, 1973 - zaklad pre IBM's Lucifer algoritmus, ktory bol zas zakladom pre DES.

- blokova sifra
- substitucna/permutacna siet
- plaintext rozdelime na 2 casti, prava bude v lavej casti OUT a pouzije sa aj do F; lava bude pouzita pre XOR s vystupom z F a vysledok pojde do pravej casti OUT. Je to na kola, kde jedno: L_i = R_(i-1) a R_i = L_(i-1) XOR F(R_(i-1), K_i)
- proces je reverzibilny (poradie subklucov je opacne)
- poziadavky na F - aby bol alg. bezpecny - skrytie vlastnosti spravy aj kluca. Vytvorit dobru F je tazke
- cim vacsi blok a kluc, tym je vacsia bezpecnost
- zlozitejsia generacia subklucov a F moze znamenat vacsiu bezpecnost
- subkluce - v bloku "key schedule" (pozor na moznost slabych klucov)

DES

56bit - v IBM, na zakazku NBS (dnes NIST), na zaklade Lucifer. Modifikovan NSA - zmena S-boxov, kluc zo 128 na 56. Ako standard v 1976. Zasifroval najviac bitov zo vsetkych algoritmov.

- Vylepsenie - 3DES. Pracuje s velkostou bloku 112bitov. Double DES nie je bezpecny - KPA (known plaintext attack) utok MitM - 2^{n+1}pokusov.
- Posuvy a permutacie vytvaraju 48-bitove subkluce pre kazde kolo. 2^{56} klucov.
- IN -> pociatocna permutacia (IP) -> 16 cyklov -> koncova permutacia -> OUT
- F funkce:
  - 32 bitov -> E funkce -> 48 bitov XOR K_n -> Substituce (S boxy) -> 32 bitov -> Permutace P
  - E funkce - expanze bitov, duplikacia niektorych S-boxy (Selection, neskor Substitution boxy) - zasadne pre bezpecnost, NSA zmenila ich obsah, jediny nelinearny krok DESu
- Key Schedule - je treba 16x48-bitovych klucov. Idealne by bolo mat 768 bitov, teda 16 nezavislych klucov. Pouziva sa 56 bitovy kluc, 64 s paritou. Pouziva sa operacia rotace.

Bezpecnost

- DES Avalanche Effect - zmena v plaintexte - pocet OUT bitov, kt. sa zmeni pri zmene 1 bitu na IN. Rovnaka vlastnost sa skuma pre zmenu 1 bitu kluca.
- Slabiny a pochybnosti: velkost bloku a kluca - je viac blokov nez klucov, pre 1 blok 2^{64}sprav -> 2^{56} klucov. Zasifrovany konstantny blok nemoze mat vsetkych 2^{64} hodnot
- 56 bitovy kluc je moc kratky - utok silou
- komplementarne kluce, alebo existencia slabych - pri generovani treba kontrolovat, ci sa nejedna o:
  - slaby kluc - vsetky su rovnake, lava alebo prava polka su same 0 ci 1 (4ks)
  - poloslaby kluc - generuju iba 2 rozne subkluce (12ks)
  - asi slabe kluce - generuju iba 4 rozne subkluce (48ks)
  - **dokopy** teda 64 klucov, zo 72,000 trilionov
- nevieme, ci 16 cyklov staci pre bezpecne zasifrovanie
- navrh S-boxov nebol zverejneny - moznost zadnych vratek - ak by mali nejaku linearnu funkciu, ich autor by sifru mohol lahko rozbit

- **utok na slabe, komplementarne kluce** (56 na 55 bitov), diferencialna / linearna kryptoanalyza nie je prakticky pouzitelna
- **Brute force**, teoreticke utoky:
  - Diffie a Hellman - 1976 - stroj za 20M dolarov a 1M procesorov, 10^{12}keys/sec a vsetky za 1 den
  - Michael Wiener - 1993 - 1M dolarov s 57,600 cipmy, kazdy testuje 50M keys/sec, 1 kluc priemerne za 3.5h
  - EFF DES Cracker - 1998 - 250,000 dolarov, 41,472 procesorov - 2.5M klucov/sec
  - 1999 - 56 bitovy kluc za 22 hodin- 100,000 pocitacov + 1 EEF

IDEA

pomocou 128bit key sifruje 64bitove bloky.

- Ako DES - pracuje v kolach a operacia je rovnaka pre sifrovanie aj desifrovanie. No lisi sa od neho v tom, ze je navrhnuta tak, aby bola efektivna aj v SW a nepouziva S-boxy ani P-boxy.
- Zakladne operacie - XOR, scitanie mod 2^{16}a nasobenie mod $2^{16} + 1$.

Blowfish

sifruje 64bitove bloky. Kluc ma premennu dlzku, az do 448bitov.

- Takmer Feistelova sifra. Pouziva F funkciu - ta 4 S-boxy, kazdy S-box prevadza 8bitov na 32bitov.
- Klucovo zavisle S-boxy - ich obsah je urceny hodnotou kluca.

TEA (Tiny Encryption Algorithm)

64-bitovy blok, 128 bitovy kluc.

- Vyuziva 32-bitovu aritmetiku.
- Premenny pocet kol (obvykle 32). V kazdom pouziva slabu funkciu, preto treba vacsi pocet kol.
- Takmer Feistelova sifra. Pouziva + a - namiesto XORu. Jednoducha, lahko implementovatelna, rychla a pamatovo nenarocna.
- Existuje tzv. **related key attack** - eXtended TEA (XTEA) ho eliminuje.

AES (Advanced Encryption Standard)

Velkost bloku 128, variabilne dlzky klucov 128 (10 kol) / 192 (12 kol) / 256 (14 kol).

- Zalozeny na Rijndael algorithm (Belgicko), platny od 2002. Mal odstranit nedostatky DESu.
- Sifrovanie
  - Krok Add Round Key (subkluc XOR blok)
  - Kola:
    - a) Byte Sub - kazdy bajt bloku je nahradeny inym podla S-boxu. Je to nelinearny krok, ale invertovatelny.
    - b) Shift Row - bajty su usporiadane do matice a posunute.
    - c) Mix Column - nasobenie matic. Je to nelinearny krok, ale invertovatelny, aplikovany na kazdy stlpec. Obvykle implementovane ako velka tabulka.
    - d) Add Round Key - subkluc (round key) je vytvoreny algoritmom Key Schedule.
- Desifrovanie (proces sifrovania musi byt invertovatelny)
  - Inverzia MixAddRoundKey je jednoducha
  - XOR je inverzny sam o sebe
  - MixColumn je invertovatelny pomocou inej tabulky
  - inverzia ShiftRow je jednoducha (cyklicky posuv v opacnom smere)
  - ByteSub je tiez invertovatelny inou tabulkou

RC2, 4, 5

Ron Rivest - variable key size.

Skipjack

80bit key, 32 rounds, NSA initially classified.

5. Symetricka sprava klucov

Zdielany kluc je casto dlhodoby (W_{AB}alebo KEK_{AB}) a pomocou neho sa dohaduje kluc relace (K_{AB}).

Kluc relace - iba docasne tajomstvo, pouzivane iba pre kratku dobu a potom znicene. Nezavislost zabezpecenia jednotlivych relaci, obmedzenie potreby pamatat si mnozstvo klucov, mnozstvo zasifrovanych sprav pre analyzu utocnikom a casovy rozsah dopadu prezradenia 1 kluca.

Vlastnosti spravy klucov

- sposob autentizacie - autentizacia subjektu a kluca
- vzajomna autentizacia - jednozmerna, obojsmerna
- cerstvost kluca
- cinnost: distribucia kluca, dohoda na kluci
- efektivnost - pocet prenesenych sprav, bajtov, zlozitost vypoctov ucastnikov
- poziadavky na 3. nezavislu stranu (TTP - Trusted Third Party - KDC, CA, KTC, bez nej by mohlo byt n^2klucov) - Online, Offline, miera dovery v TTP

Ciele utocnika

- ziskat kluc relace pomocou odposluchu, ucastnit sa beziaceho protokolu a pomocou zmien prav ziskat kluc relacie
- iniciovat 1+ behov protokolu a kombinovat ich spravy pre realizaciu prechadzajucich utokov
- bez schopnosti zistit kluc relace podviest 1 z ucastnikov ohladne identity ucastnika, s ktorym navazuje protokol
- presvecit 1 z ucastnikov, ze uspesne naviazal protokol s inym ucastnikom nez je utocnik

Utok replay

- Simple replay - skopirovat spravu, neskor zaslat

- Suppressed replay - odstranit spravu, neskor zaslat
- Timed replay - skopirovat spravu, zaslat ju v ramci casu expiracie
- Replay to Sender - zaslat spravu spat odosielatelovi (Reflection attack, napriklad pre protokoly vyzva-odpoved)
- Interleaved runs - kombinacia sprav z roznych/subeznych behov toho isteho protokolu

Cerstvost - Seq numbers (musia byt udrzovane pre kazdy kanal), timestamps (sprava je projata iba ak je casova znacka dostatocne cerstva), Nonces (nahodne cisla zasielana po vyzve, vratena v odpovedi)

KDC (Key Distribution Center) - KDC generuje kluc relace K_{AB} - Push (A posle poziadavok KDC, to posle spat kluc A, potom A posle kluc B) model, Pull model (A posle poziadavok B, B posle poziadavok KDC a KDC ho vrati B), alebo Hybridny model (A posle KDC poziadavok, KDC odpovie A zaslanim klucu a zaroven ho posle i B)

KTC (Key Translation Center) - jedna z komunikujucich stran generuje kluc relacie K_{AB} - Push model (A generuje kluc, posle ho KTC, ten ho posle spat A a ten ho posle B), Hybridny model (A generuje kluc, posle ho KTC a ten ho posle B)

Nevyhody - kazda komunikacia vyzaduje kontakt s TTP (trusted third party) a to musi mat ulozene n dlhodobych klucov. Okrem toho moze citat vsetky spravy a ak je kompromitovana, je akakolvek komunikacia nebezpecna.

Oproti tomu asymetricka - TTP nemusi byt kontaktovane pred kazdou komunikaciou, nemoze jednoducho odposluchavat komunikaciu a dovera v TTP staci mensia (TTP moze iba docasne zmenit identitu ucastnika).

Protokoly

- Najjednoduhsi protokol: symetricky kluc, challenge-response - jednosmerna autentizacia, autentizovany kanal sa da ukrast (conn hijacking), staci hashovacia funkcia, ak je K odvodene od hesla, da sa provaded offline hadanie hesla.
- One-message protocol - symetricky kluc, casova znacka - jednoducha integracia do systemov s heslom. Velmi efektivny, 1 sprava a je bezstavovy. Je treba dat pozor na replay a ak je kluc zdielany medzi viacerymi servermi. Je treba chranit hodiny a alternativne sa da pouzit s hash funkciou.
- Verejny kluc - vyzva odpoved - pomocou podpisu alebo desifrovania (B moze pozadovat po A, aby podpisala/desifrovala akykolvek text, ktory si oni zvoli. Riesenie - nepouzivat rovnaky kluc pre viac ucelov - pre prihlasenie a podpis, a pouzivat formatovane vyzvy).

- Obojstranna autentizacia. Slabina - Utocnik moze vykonat slovnikovy utok na zdielany kluc vydavajuci sa za A, bez odposluchavania. Riesenie: pouzitie napriklad principu, ze iniciujuca strana by mala preukazat svoju identitu prva.
- Otway-Rees protokol - KDC, Pull
- Wide-Mouthed Frog - KTC, Hybridny model, casove znacky. Neskor objaveny problem s periodickou ziadostou o presifrovanie kluca a 'predlzovanim' casovej znamky
- Needham-Schroeder - KDC, Push
- Weird Reflection Attack - symetricka sifra v ECB rezime - obecne nevhodne
- Kerberos v4 a v5 - protokol pre autentizaciu a bezpecnu komunikaciu PC prepojenych sietou. Ziadatelom sa prideluju bezpecne tickety, ktore sluzia pre naviazanie bezpecnej komunikacie. Standardne formaty zabezpecenych sprav. Spravy zachycuju aj mena, doby expiracie. Mechanizmus centralneho KDC (AS - auth server). Podporuje 'slave KDC', ktore mozu provadet iba obmedzene operacie (TGS - Ticket Granting Server).
  1. Pouzivatel sa zaloguje na pracovnu stanicu a pozaduje na hostovi sluzby.
  2. AS verifikuje jeho uzivatelske pristupove prava v DB, vytvori ticket-tgranting ticket a kluc relacie. Vysledok je zasifrovany pouzitim klucu derivovaneho z uzivatelovho hesla.
  3. Workstation vyzyva uzivatela o heslo a pouziva heslo pre odsifrovanie prichadzajucej spravy, potom posle ticket a autentikator, ktory obsahuje uzivatelovo meno, sietovu adresu a cas k TGS.
  4. TGS odsifruje tiket a autentikatora, verifikuje request a potom vytvori ticket pre pozadovany server.
  5. Workstation posle ticket a autenticator na server.
  6. Server verifikuje ze ticket a authenticator sa zhoduju, potom povoli vstup. Ak je vyzadovana vzajomna autentizacia, server vrati autenticatora.

6. Asymmetric Cryptography

Dlzky klucov - 768, 1024, 2048b. Diffie a Hellman a nezavisle na nich Merkle (1976).

Aplikacia - sifrovanie, elektronicky podpis, vymena klucov pre ustanovanie kluca relacie

Sifrovanie VK

- Dovernost (ANO-prijemca je jediny, kto sa moze zoznamit s obsahom spravy. Rozsifrovat to moze iba ten, co pozna sukromny kluc),
- Autentizacia (NIE - verejny kluc pozna ktokolvek aj utocnik, sprava moze byt kludne aj od neho),
- Integrita (NIE-prijemca moze prijat pozmenenu spravu),
- Nepopieratelnost (NIE - neviem co prislo, neviem od koho to prislo).

Sifrovanie SK

- Dovernost (NIE, VK poznaju vsetci) ,
- Autentizaca (ANO - ak tu spravu zasifrujem, mohla byt zasifrovana iba odpovedajucim SK),
- Integrita (ANO-odosielatel ju mohol jediny zasifrovat svojim SK tak, aby som ju mohol spravne desifrovat),
- Nepopieratelnost - el. podpis (ANO).

Sifrovanie SK, potom VK, desifrovanie SK, potom VK

- Dovernost,
- Autentizaca,
- Integrita,
- Nepopieratelnost.
- **ANO** vsetky 4 bezpecnostne funkcie, no za cenu 4 klucov.

Sifrovanie VK, potom SK, desifrovanie VK, potom SK

Ako vyssie uvedena, no v 90% pripadov sa pouziva ta vyssie uvedena. V normalnom svete tiez podpis najprv podpiseme a potom ho zalepime do obalky. Okrem toho, mozeme sa dostat do situacie, ze je sprava uz desifrovana, ale este je podpisana elektronickym podpisom. Tento druhy pripad to nepokryva.

6.1. Hash functions

- SU tiez zname ako jednocestna funkcia, hash, one way function, digest, message digest, charakteristika spravy.
- Funkcie aplikovatelne na IN lubovolnej velkosti, OUT ma konstantnu dlzku (128/160/256 bitov) a lze ju rychle spocitat.
- Rychle ju lze spocitat.
- CRC, digitalny podpis.
- Je vypocetne nezvladnutelne najst:
  - (first preimage resistance) - pre dane y take x, aby platilo F(x)=y ... (k hashu nenajdeme plaintext)
  - (second preimage resistance) - pre dane x take x'<>x, aby platilo F(x')=F(x) ... (k plaintextu nenajdeme iny plaintext s rovnakym hashom)
  - (collision resistance) - take x' a x, x'<>x, aby platilo F(x')=F(x) ... (nenajdeme 2 rozne plaintexty s rovnakym hashom)

  Ak je collision resistance, tak aj 2nd preimage. Ak je MAC, tak vzhladom k utoku chosen-text attack je 2nd preimage a collision resistant, aj 1st preimage resistant.
- Klasifikacia
  - **neklucovane** - **MDC** (manipulation detection codes) alebo **MIC** (message integrity codes):
    - One-way Hash Functions (**OWHFs**), cielom navrhu je collision resistance, cielom utocnika je koliziu najst (sila $2^{\frac{n}{2}}$)
    - Collision Resistant Hash Functions (**CRHFs**), cielom navrhu je 1st & 2nd preimage resistance, cielom utocnika je ich najst (oboje zvlast maju silu 2^n)
  - **klucovane** - **MAC** (message authentication codes) - zaistuju autentizaciu a integritu. Cielom navrhu je non-recovery key a odolnost voci vypoctu, cielom utocnika je najst kluc (2^n) alebo vytvorit novy MAC, (sila2^{-n}sila az 2^{-t}), kde t je dlzka kluca.
- Detailnejsi pohlad
  - vstup je zarovnany, potom sa pripoji dlzka (predspracovanie). Potom nastava iterativne spracovanie, pritom sa ziskany upraveny IN v kompresnej funkcii meni spolu s vystupom s predchadzajucom kole (prvy je IV). Posledne volitelne kolo je este nejaka transformacia.
  - obvykle su zlozene z kompresnej funkcie f, ktora pracuje nad blokmi - podobne blokovym sifram v CBC rezime. Hodnota hashu pre kazdy blok je zavisla na

hodnote bloku a hodnoty hashe predchadzajucich blokov. Na zaciatku sa pouziva IV.

- **Merklova meta-metoda**: kazda kompresna funkcia odolna proti koliziam sa da rozsirit na CRHF. Tato metoda popisuje efektivny sposob ako ju vytvorit - n bitovy vystup, r bitova premenna. Ak existuje kolizia pre vystup h, to znamena ze vznikla kolizia pre f v urcitom kole i. Vlozenim dlzky bloku je zaistene, ze ziadny IN nie je prefixom ineho IN (Merkle-Damgardovo zosilenie).
- Zarovnanie
  - nejednoznacne (ambiguous padding): pripaja k sprave tolko nulovych bitov, aby bola sprava nasobkom dlzky bloku
  - jednoznacne (unambiguous padding): pripoj k sprave 1, proved nejednoznacne zarovnanie. Je to neintuitivne pre programatora.
- Zakladny utok
  - na hash, n-bitovy neklucovany hash ma idealnu bezpecnost, ak splnuje poziadavky na OWHF a CHRF
  - utok silou na kluc MAC (known-text attack) vyzaduje 2^toperaci
  - predvypocitanie hashu (memory-time tradeoff) * uhadnutie MAC: 2^n operaci
  - paralelizacia 2nd-preimage utoky na dlhe spravy pre 2nd-preimage. Ak je h iterovana funkcia a nepouziva sa MD zosilenie, potom 2nd preimage moze byt najdene v case "$2^{\frac{n}{s}} + s$" a priestoru "$n * (s + logs)$" bitov pre 1 < s < min(t, 2n/2) - narodeninovy utok na medzivysledky
- Priklad birthday attack:
  - predpokladajme hash funkciu s n-bitovym OUT. Utocnik vytvori dokument 'priatelska dohoda' a priblizne $2^{\frac{n}{2}+1}$ semanticky ekvivalentnych verzii. Podobne utocnik vytvori dokument 'nepriatelska dohoda' a tolko isto semanticky ekvivalentnych verzi. S pravdepodobnostou 1/2 bude existovat verzia priatelskej a nepriatelskej dohody, ktore budu mat rovnaky hash.
  - pozadovane dlzky: OWHF: n>=80, CHRD: n>=160 (birthday attack), MAC: n>=64 s klucom aspon 64 bitov (je vhodne obmedzit pocet pokusov hladania)
- Blokove sifry
  - Matyas-Meyer-Oseas - do funkcii ide INi a g(Hi-1) a jej vystup XOR INi
  - Davis-Meyer - do funkcii ide INi a Hi-1 a jej vystup XOR Hi-1
  - Miyagucho-Preneel - do funkcii ide INi a g(Hi-1) a jej vystup XOR INi XOR Hi-1
- Algoritmy: MD2, MD4, MD5, SHS, SHA

  - MD2 - dava 128-bitovu charakteristiku spravy. Jednoduchy a rychly algoritmus.
    * Nahodna substitucia je tabulka z cislic Ludolfovho cisla, nemoze teda obsahovat skryte vratka.
  - MD4 - specialne navrhnute ako hashovacia funkcia specialne pre SW implementaciu na 32-bit strojoch. Startovaci bod pre MD5, SHA-1 a RIPEMD. 128-bitovy OUT.
    * Dobbertin nasiel kolizie pre zmysluplne spravy.
  - MD5 - OUT 128 bitov a bloky 512 bitov. Sprava je doplnena o dlzku k = 448 mod 512. Na koniec spravy je pridany 64-bitovy blok s dlzkou spravy. Vysledna je nasobkom 512 bitov. 64 krokov (4 po 16), velkost spravy neobmedzena.
    * Nie je odolne proti koliziam a pouziva sa v IPSec a inych protokoloch.
    * Uz od 2005 **sa neodporuca ju pouzivat** (**Halloween Hash Bash**)
  - RIPEMD-160: OUT 160 bitov a 160 krokov (5 po 16), inak ako MD5. Kompresna fce, mapuje 21-slovny IN (5 slovna stavova premenna, 16 slovny blok spravy, 32 bitova slova) na 5-slovny OUT.
    * Viac kol nez MD-4 a bezpecnost porovnatelna s SHA-1.
  - SHA: 1993, podobna MD5, revidovana v roku 1995 a druhy krat (SHA-2) v 2001.
    * SHA-1: OUT 160 bitov, blok 512 bitov, 80 krokov (4 po 20), max velkost spravy $2^{64} - 1$. Uz sa v novych aplikaciach pouzivat nemoze (**Halloween Hash Bash**).
    * SHA-2: rodina algoritmov, bezne sa pouziva iba SHA-256. Malo efektivne v HW implementacii, behom 10 rokov pravdepodobne nebude existovat prakticky utok, mozno len teoreticky.

- **MAC** - rodina funkcii parametrizovanych tajnym klucom. Snadny vypocet ak je k zname, vypocetna bezpecnost, pri znalosti paru (INi, hk(INi)) je vypocetne nemozne spocitat novu dvojicu (IN, hk(IN)) pre nove IN != INi.
  - V podstate sa vypocita MAC_ALG(MSG, K) = MAC. MSG+MAC je vysledok. ten sa vlastne tym istym klucom potom rozsifrovava.
  - CBC MAC - volitelny krok ma zabranit utoku chosen-text existential forgery bez ovplivnenia predchadzajucich krokov. Lebo zakladny CBC-MAC je bezpecny iba pre spravy s pevnym poctom blokov. Inak ak mame dvojice (x1, H1) a (x2, H2) a mozeme pozadovat ((x1 || z), M) potom je mozne skonstruovat novu spravu (x2 || (H1 XOR z XOR H2), M) ktora je platna. MD zosilenie nepomaha. (A, B) - A je sprava, B je jej hash.
  - MAC vytvorene z MDC - velmi rozsirena konstrukcia - IPSec, SSL. 3 rozne strategie:

  - Secret prefix: na zaciatok spravy sa prida tajny kluc k a MAC je potom M=h(k||x). Je tu **utok**, kedy je mozne na koniec spravy pridat y a spocitat h(k||x||y) zo znalosti h(k||x) bez znalosti k !!! Nepomaha ani MD zosilenie (i dlzka x sa da zahrnut do spravy) a ani ta varianta nie je bezpecna, kedy pouzijeme k ako H0 (teda IV).
  - Secret suffix: MAC je M=h(x||k). **Moznost narodeninoveho utoku** - utocnik, ktory moze zvolit x moze vytvorit tiez x' pre ktore h(x) = h(x') so zlozitostou $O(2^{\frac{n}{2}})$ bez ohladu na dlzku kluca k. Metoda v podstate vypocita hash a v konecnej faze ho 'zasifruje'. Nie je to dobry sposob!.
  - Enveloping: hk(s) = h(k||p||x||k), pricom p je retazec pouzity pre zarovnanie kluca k na dlzku 1 bloku. Lepsie nez predchadzajuce 2, no nie je to najlepsia metoda. **Je to zaklad pre algoritmus HMAC**.

- **HMAC** - klucovany hash. Pouziva sa v IPSec a TLS. Spocita sa H1 ako hash konkatenancie M a K1. Pre zabranenie utoku 'dodatocny blok' sa spocita H2 ako hash konkatenancie H1 a K2. K1 a K2 pouzivaju polovicu bitov kluca K. Kluc K sa zvykne doplnat nulami.
- **Metody pre integritu dat**: iba MAC, alebo MDC a autentizovany kanal, alebo MDC a sifrovanie.

6.2. RSA (Rivest-Shamir-Adleman)

Faktorizacia velkych cisiel. Ako blokova sifra, blok je v rozmedzi 0-n.

Kluce

- n je verejny modulus,
- e je verejny exponent,
- d sukromny exponent,
- p a q su cinitele modulu n (n = p*q)
- musi platit vztah d*e mod (p-1)(q-1) = 1
- verejny kluc je (n,e)
- sukromny kluc je (n,d)

Postup

1. vygeneruj prvocisla p, q
2. spocitaj theta(n)=(p-1)(q-1)

3. zvol hodnotu e < theta(n) taku, ze GCD(theta(n), e) = 1
4. spocitaj d tak, ze $d = e^{-1} mod\, theta$(n)

Sifrovanie/desifrovanie, plaintext (m), ciphertext (c)

- utajenie:
  - sifrovanie VK: $c = m^e mod\, n$
  - desifrovanie SK: $m = c^d mod\, n$
- elektronicky podpis:
  - sifrovanie SK: $s = m^d mod\, n$
  - desifrovanie VK: $m = s^e mod\, n$

Utoky

- ak utocnik vie rozlozit n na cinitele p, q, moze dopocitat SK. Ak uhadne (p-1)(q-1) tak moze dopocitat SK aj bez faktorizacie n.

Rady

- nepouzivat male hodnoty pre d,
- vzdy doplnovat spravy nahodnymi cislami, aby m malo priblizne rovnaku velkost ako n.

6.3. DSS/DSA (Digital signature standard/algorithm)

Problem diskretneho logaritmu, navrhnuty NISTom v 1991. Velkost kluca najprv 512bitov, potom 2 nasobok. Bezpecnost je porovnatelna s RSA.

Nevyhody

- neda sa pouzit pre sifrovanie alebo distribuciu klucov
- rychlejsie nez RSA pri podpisu ale pomalsi pri verifikacii
- v dobe, kedy je uz znacne rozsirene RSA
- obavy, ci neobsahuje zadne vratka od NIST

Utoky

- uhadnutie nahodneho cisla k, ak je zname, da sa spocitat r a x
- zadne vratka - pre niektore specialne hodnoty p je problem diskretneho algoritmu lahko riesitelny, moznost ze v parametroch su od NISTu

6.4. DH (Diffie-Hellman)

Prvy algoritmus s VK, zalozeny na problemu diskretnych logaritmov modulo n.

Protokol

- Pre ustanovanie kluca relacie K
- Kluc relace sa pouzije pre sifrovanie dalsej komunikacie
- Zvolenie parametrov: zvoli sa velke prvocislo n, hodnota g, ktora nedeli n.

Komunikacia

1. Alica zvoli x. posle spravu$X = g^x mod\ n$
2. Bob obdrzi spravu, zvoli y. Posle spravu $Y = g^y mod\ n$
3. Alica spocita kluc K=Y^x mod n $K = Y^x mod\ n$
4. Bob spocita kluc K=X^y mod n
   Kluc bude $K = g^{xy} mod\ n$

6.5. Knapsack

Problem balenia veci do batohu. Ako vybrat predmety, ktore maju sucet presne rovny kapacite batoha.

Priklad

Nech S={4,7,1,12,10} a T=17. T=17=4+1+12, potom existuje vektor V={1,0,1,1,0} ako riesenie.

6.6. EC (Elliptic Curves)

Odporuca sa bitovy kluc aspon 160 bitov, no idealne 256.

- scitanie: P + Q = R, P != Q
- zdvojnasobenie: P + P = 2P = R
- lahky vypocet: Q = kP = P + P + ... + P (k-krat)
  - (P je bod na krivke a k je cele cislo)
  - Obtiazny vypocet: mame danu krivku, bod P a kP - je tazke vypocitat k, ide o **ECDLP - Elliptic Curve Discrete Logarithm Problem.**
- take kryptosystemy poskytuju najvacsiu mieru obtiaznosti na 1 bit dlzky kluca so vsetkych asymetrickych alg a ich sila s dlzkou kluca rastie ovela viac nez sila RSA

- 120 bitovy kluc EC == 1,024 bit RSA (10^12 rokov u MIPS)
- 1200 bitovy kluc RC == 120,000 bit RSA (10^168 rokov u MIPS)
- vyhodne na miestach s obmedzenym:
  - vypocetnym vykonom alebo kapacitou pamati
  - plochou cipu integrovaneho obvodu
  - prenosovou rychlostou tam, kde je potrebna vysoka rychlost

7. Asymetricka sprava klucov

7.1. X.509

Bezpecnostne funkcie - autentizacia, dovera, integrita a nepopieratelnost.

- Certifikacia verejnehop kluca - CA podpise VK uzivatela a jeho meno (plus dalsie udaje) svojim tajnym klucom = certifikat. Moze byt teda overeny VK CA.
- VK CA mozu byt opat certifikovane inymi CA - stromove struktury, krizova certifikacia medzi stromami. Retazec certifikaci nie je nekonecny, existuje korenovy VK a ten zostava necertifikovany.
- Polozky certifikatu
  - version (standardne 0),
  - serial number (uniq, poradove cislo cert),
  - issuer (meno vydavajuce CA),
  - subject (meno vlastnika cert),
  - validity,
  - subjectPublicKeyInfo (VK subjectu),
  - signature (alg podpisu certif).

  Format mena - postupnost relativnych jedinecnych mien (RDN - Relative Distinguished Names):
  - C - countryName,
  - OU - organizationalUnitName,
  - CN - commonName,
  - SP - stateOrProvinceName,
  - T - title,
  - BC - businessCategory,
  - O - organizationName,
  - S - surname,
  - L - localityName,
  - ST - streetAddress,

  - SN - serialNumber,
  - D - description.

- Ulohy CA - registracia VK, distribucia alebo rusenie certifikatov.
- Typy CA
  - korenova, 'policy' - vydava certifikaty CA,
  - uzivatelska - vydava certifikaty uzivatelom. Negeneruje kluce, nemoze zarucit kvalitu alebo jedinecnost uzivatelskych klucov. Vykonava:
    * udrzba zaznamov vo svojom adresari,
    * udrzba ciernych listin certif vydanych a potom odobratych touto CA,
    * zaslanie mena a novych verejnych klucov korenovej CA formou spolahliveho krizoveho certif pri kazdej zmene korenovej CA alebo ich klucov,
    * zaslanie doprednej certif cesty pri zmene ktorejkolvek vyssej CA/ich klucov,
    * aktualizaciu hodnot uzivatelskych certifikatov v zaznamoch svojich uzivatelov v adresari.
- Vydanie certifikatu - uzivatel si lokalne vygeneruje kluce, uzivatelske CA zasle prototypovy certif, po jeho overeni posle uzivatelska CA uzivatelovi jeho podpisany certifikat.
- Dovody rusenia certifikatu
  - bol prezradeny SK uzivatela,
  - zmenil sa zamestnavatel uzivatela, cim je neplatne meno obsiahnute v certifikate,
  - uzivatel uz nema byt certifikovany danou CA,
  - SK CA bol kompromitovany,
  - uzivatel porusil bezpecnostne pravidla CA.

  Dve moznosti:
  - Zruseny certifikat je umiestneny na ciernu listinu, zoznam zrusenych certifikatov - **CRL** (A certificate revocation list). Je to zoznam zneplatnenych a nevyexpirovanych certifikatov. Pull model (prijemca cert si stahuje podla potreby CRL od CA) alebo Push model (CA pravidelne posiela CRL prijemcom certifikatov - prakticky nerealizovatelny, zoznam byva velmi velky a uzivatelov mnoho). **CRL je nachylny na DOS**.
  - Na stav certifikatu je mozno sa dotazat mechanizmom **OCSP** (Online Certificate Status Protocol). Pouziva protokol Request / Response. Odpoved je vytvarana na strane serveru, redukuje sa mnozstvo prenasanych dat, server moze pozadovat, aby poziadavky boli podpisane. **Stale nachylne na DOS**.

- Dokumentacia
  - CP (Certificate Policy) popisuje vlastnosti cert
  - CPS (Certificaton Practise Statement) popisuje ako bol cert vydany
- Miera dovery v certifikaty
  - Trieda 1: certifikat zaistuje iba jedinecnost mena vlastnika, je ho mozne ziskat anonymne
  - Trieda 2: identita vlastnika musi byt overena tretou stranou - notarsky overeny formular, zaslany postou
  - Trieda 3: vlastnik musi osobne navstivit CA, overenie osobnej totoznosti
  - Trieda 4: Trieda 3 + prokazanie opravnenosti ziadatela pozadovat certifikat

7.2. Elektronicky podpis

Vlozenie textu podpisu do dokumentu alebo naskenovaneho obrazku. Digitalny podpis je podmnozina, zalozeny na pouziti kryptografickych mechanizmov.

Bezpecnostne funkcie - autenticita, integrita a nepopieratelnost.

- PKCS - public key cryptography standards:
  - #1: RSA cryptography - sifrovanie a podpis pomocou RSA
  - #7: Cryptographic Message Syntax - sprava klucov moze byt pomocou Diffie-Hellmana, syntaxe zasifrovanej a podpisanej spravy
  - #11: Cryptographic Token Interface - jednotne API pre cipove karty a ine zariadenia
  - #15: Cryptographic Token Information Format - format suborov pre kryptograficke data na cipovych kartach a inych zariadeniach

  Dalsie:
  - #3:Diffie-Hellman Key-Agreement
  - #5:Password-Based Cryptography
  - #8:Private-key information Syntax
  - #9:Selected Attribute Types
  - #10:Certification RequestSyntax
  - #12: Personal Information Exchange Syntax

  #2, #4 a #6 su uz zastarale a #13 a #14 su stale vo vyvoji

7.3. Bezna elektronicka posta

PEM (privacy enhanced mail) - navrh standardu definujuci procedury pre sifrovanie a autentizaciu sprav el. posty pri zlepseni bezpecnosti v ramci siete Internet. Poskytuje symetricku aj asymetricku spravu klucov. Cert VK podla X.509, vlastna hierarchia CA. Implementacie TIS/PEM (od firmy Trusted Information Systems) alebo RIPEM. Neuchytil sa, bol vytlaceny S/MIME. Najprv bol standardizovany a az potom implementovany. Poskytuje:

- zasifrovanie sprav - dovernost - sprava je zasifrovana algoritmom DES-CBC nahodnym klucom relacie a ten je zasifrovany bud VK prijemcu pomocou RSA alebo tajnym klucom
- elektronicky podpis - autentizace integrita a neodmietnutelnost odpovednosti (ta je ak je pouzita asymetricka sprava klucov) - Pomocou MD2/MD5 je spocitany hash spravy a je zasifrovany pomocou RSA SK odosielatela.
- kompresie dat a segmentacia nie je
- kodovanie do textovej podoby - radix64
- autentizacia - symetricka kryptografia - MD2/MD5, DES-ECB alebo DES-EDE

Sposoby zabezpecenia:

1. konverzia spravy do kanonickej formy
2. generovanie autentizacneho a integritneho zabezpecenia
3. zasifrovanie spravy
4. konverzia do textovej podoby.

Typy zabezpecenia:

- MIC-CLEAR - kroky 1,2
- MIC-ONLY - kroky 1,2,4
- ENCRYPTED - vsetky 4 kroky

Korenova CA sa nazyva IPRA (Internet Policy Registration Authority), potom su tam High Assurance, Residential, Mimd-Level Assurance, Persona a uzivatelia.

PGP (pretty good privacy) - aplikacny SW s kryptografickymi vlastnostami vykazujucimi vysoky stupen bezpecnosti. Poskytuje:

- zasifrovanie sprav - dovernost - sprava je zasifrovana IDEA nahodnym klucom relace. Ten je zasifrovany VK prijemca pomocou RSA a zaslany so spravou.
- elektronicky podpis - autentizace integrita a neodmietnutelnost odpovednosti (nepopieratelnost odosielatela) - pomocou MD5 je spocitany hash spravy a je zasifrovany pomocou RSA SK odosielatela.
- kompresiu dat - zip

- kodovanie do textovej podoby - radix64
- segmentace - dlha sprava moze byt rozdelena na segmenty

Na Windows - WinPGP napriklad. Ma to aj GUI.

Pouziva Web of Trust - ak uzivatel A nedoveruje B, ale doveruje C, ktory doveruje B, tak tranzitivne aj A doveruje B.

S/MIME (Secure/MIME) - specifikacia bezpecnej vymeny sprav aplikovanim standardov asymetrickej kryptografie PKCS. Certif VK podla X.509. Je flexibilny. Poskytuje:

- zasifrovanie sprav - DES/ 3DES / RC2 - sprava je zasifrovana symetrickym alg. nahodnym klucom relace,
- elektronicky podpis - pomocou MD5 / SHA-1 je spocitan hash spravy a je zasifrovany pomocou RSA SK odosielatela,
- kodovanie do textovej podoby - MIME.

Typy zabezpecenych sprav:

- Data (data) - obecne, nesifrovane a nepodpisane data
- Podpisane data (signedData) * Sifrovane data pre prenos (envelopedData)
- Podpisane sifrovane data pre prenos (signedAndEnvelopedData)
- Data s charakteristikou (digestData)
- Sifrovane data pre ulozenie (encryptedData) - napriklad na lokalny disk, sifrovaci kluc je odvodeny z hesla. Predpoklada sa, ze sprava sifrovacich kludov je robena offline.

7.4. SSH

Motivacia: klasicke sluzby vzdialeneho terminalu (telnet, rsh, rlogin) nie su bezpecne. Autentizacia heslom - odposluch hesla, prenasanych dat, alebo modifikacia prenasanych dat. SSH nahradzuje tieto sluzby a moze tunelovat aj ine sluzby. Pouziva RSA, DES/3DES/IDEA/RC4.

Vlastnosti

- autentizacia pomocou RSA - zabranenie podvrhnutie IP adresy. Host si vygeneruje SK/VK. Uzivatel tiez, su ulozene v suboru .ssh zasifrovane heslom uzivatela pomocou IDEA. RSA autentizace host-host, uzivatel-host. Autentizacia UNIXovym heslom (zasifrovanym kanalom!)
- zabranuje unosu uz naviazaneho a autentizovaneho spojenia
- zabranuje odposluchu hesla

7.5. SSL

Zabezpecenie TCP/IP na transportnej urovni, je mozne pomocou toho implementovat rozne sluzby, napriklad HTTPS = HTTP+SSL. Vyzaduje TCP a podporuje rozne aplikacie a protokoly. Certifikaty podla X.509

SSL Handshake protocol - naviaze sifrovane spojenie symetrickym algoritmom.

SSL Record protocol - stara sa o skladanie paketov, sifrovanie a desifrovanie.

Bezpecnostne funkcie

- Autentizacia - klient-server a server-klient
- Dovernost - zasifrovanie dat prenasanych kanalom
- Integrita

Sada algoritmov - algoritmus s VK RSA, Diffie-Hellman, hashovaci algoritmus MD5, SHA a symetricka sifra RC2, RC4, IDEA-CBC, DES-CBC, 3DES-CBC.

Faza ustavenia relacie

- ustavenie master kry - klient iniciuje relaciu, server zasle svoj certifikat, klient vygeneruje premaster secret a zasle ho zasifrovany VK serveru
- autentizacia klienta (volitelne) - server ziada klienta o certif, klient ho zasle spolu s autentizacnymi informaciami
- ustavenie kluca relacie - z premaster secret obe strany vygeneruju 2 kluce relace, 1 pre kazdy smer prenosu

Tvorba klucov

- Premaster secret - vytvoreny klientom - jednoduchy: 2 bytes of SSL version + 46 nahodnych bajtov. Zasiela sa serveru zasifrovany VK serveru.
- Master secret - vytvoreny obecne oboma ucastnikmi z premaster secret a nahodnych hodnot klienta a serveru.
- Klucovy material - vytvaraju s az nich kluce. Je vytvoreny z master secret a nahodnych hodnot klienta a serveru.

SSL v3 - poskytuje tie iste bezpecnostne sluzby ako v2, no naviac:

- rychlejsie naviazanie spojenia podpora viacerych algoritmov - Diffie-Hellman, Fortezza
- podpis HW autentizatorov - Fortezza karty - 1. krok k podpore cipovych kariet
- vylepsenie protokolu v okamihu, kedy server ziada klienta o certifikat
- server moze zaslat klientovi zoznam CA, ktorym veri a klient tak moze sam vybrat vhodny cert (ak ich ma viac)

Part IV.

References