II.

Symetrické šifrovací systémy

Vlastimil Klíma

verze: 2.1, 11.4. 2007

Obsah

7. Symetrické šifrovací systémy

Nyní se budeme věnovat **kryptografickým systémům, které zajišťují bezpečnostní službu důvěrnosti dat**, a to pomocí kryptografického nástroje šifrování dat. Tyto systémy se nazývají **šifrovací systémy** neboli **šifry**. Nejprve si uvedeme definici, která platí jak pro symetrické, tak pro asymetrické šifry. Poté se budeme věnovat už jen symetrickým šifrám.

7.1. Kryptografický systém pro šifrování zpráv (šifra) - symetrický i asymetrický

Definice: Kryptografický systém pro šifrování zpráv (šifra)
Kryptografický systém pro šifrování zpráv je pětice (M, C, K, E, D), kde M je prostor otevřených zpráv, C prostor šifrových zpráv a K prostor klíčů. E, D je dvojice **zobrazení**, které každému klíči $k \in K$ přiřazují **transformaci pro zašifrování zpráv E_k a transformaci pro dešifrování** zpráv D_k, E_k: $M \rightarrow C$: $m \rightarrow c$ a D_k: $C \rightarrow M$: $c \rightarrow m$, přičemž pro každé $k \in K$ a $m \in M$ platí $D_k(E_k(m)) = m$.

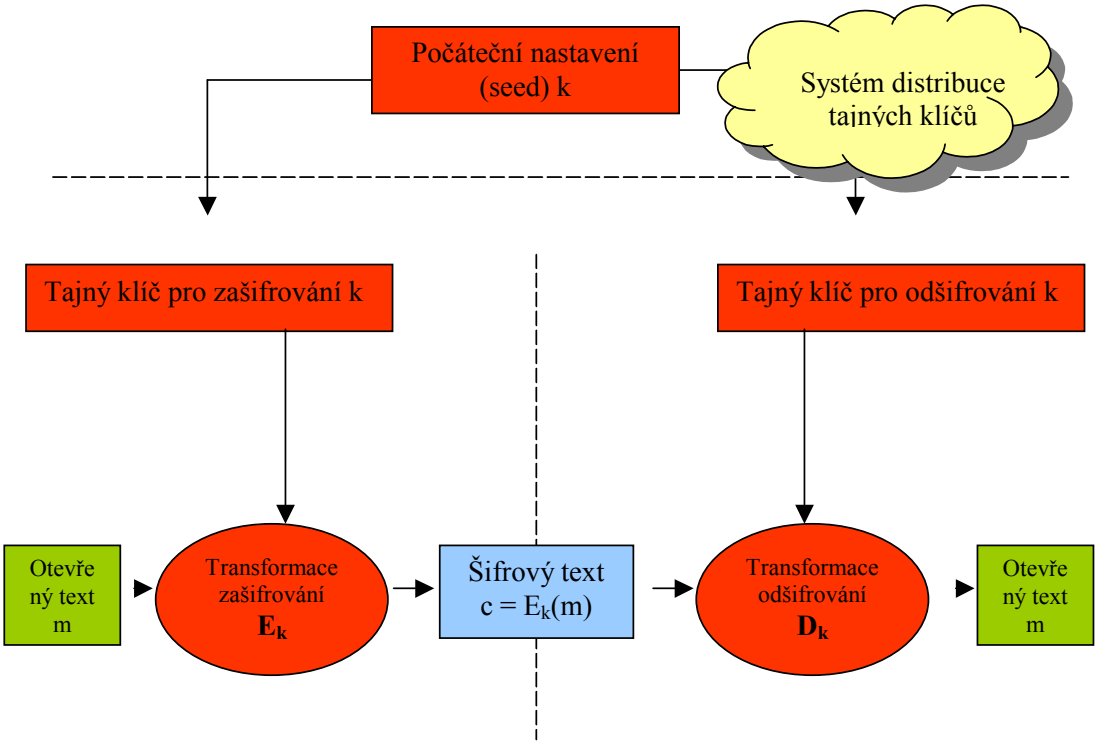
7.2. Symetrický kryptografický systém pro šifrování zpráv (symetrická šifra)

Nyní si uvedeme definici symetrické šifry.

Definice : Symetrický kryptografický systém pro šifrování zpráv (symetrická šifra)
Symetrická šifra je taková šifra, kde pro každé $k \in K$ lze z transformace zašifrování E_k určit transformaci dešifrování D_k a naopak.

Poznámka: Z důvodu této symetrie se tyto systémy nazývají **symetrické systémy** a jejich klíče **symetrické klíče**. Symetrické klíče jsou tajné, zatímco obě zobrazení E a D mohou být zcela veřejná, jako je tomu například u šifrovacích standardů DES a AES.

Poznámka pod čarou: Pro srovnání si uveďme definici asymetrické šifry.
Definice: Asymetrická šifra je taková šifra, kde pro skoro všechna $k \in K$ nelze z transformace pro zašifrování E_k určit transformaci pro dešifrování D_k. V praxi je u *asymetrických* šifer klíč *k* tajným nastavením, z kterého se vhodnou transformací G vygeneruje dvojice parametrů (e, d), které se nazývají po řadě veřejný (e) a privátní (d) klíč. Ty potom parametrizují transformace zašifrování a dešifrování, takže pro jednoduchost nepíšeme E_k a D_k, ale přímo E_e a D_d.



Obr.: Symetrický šifrovací systém

Na obrázku vidíme Shannonův model komunikačního kanálu pro potřeby symetrických šifer. Jako příklad symetrické šifry si vezměme **Caesarovu šifru**, kdy se šifruje pomocí posouvání písmen o tři pozice doprava v abecedě. Slovo CAESAR bude zašifrováno na FDGWFV. Při obecnějším posouvání o k písmen je příjemci i odesílateli nutné sdělit pouze tajný šifrovací klíč k, zde $k = 3$. Odesílatel pak šifruje pomocí transformace E_k, tj. posouvá písmena vpravo o k pozic, příjemce zprávu odšifruje transformací D_k, tj. posunem písmen o k pozic zpět. Povšimněme si, že transformace E_k a D_k nejsou totožné (posun doprava, posun doleva), ale z jedné lze odvodit druhou.

7.3. Shannonova teorie

Tento odstavec je zpracován zejména s použitím [12]. Shannon v roce 1948 a 1949 položil svými dvěma pracemi [1] a [2] základy teorie informace a teorie šifrovacích systémů. Stanovil teoretickou míru bezpečnosti šifry pomocí neurčitosti otevřeného textu, když je dán šifrový text.

7.3.1. Vzdálenost jednoznačnosti

Jestliže se nic nového o otevřeném textu nedozvíme (například zúžení prostoru možných zpráv), i když přijmeme jakékoliv množství šifrového textu, šifra dosahuje **absolutní bezpečnosti** (**perfect secrecy**). Jinými slovy v tomto případě šifrový text nenese žádnou informaci o otevřeném textu.Zvyšování počtu znaků šifrového textu u většiny praktických a zejména u historických šifer poskytuje stále více **informace o otevřeném textu**. Tato informace nemusí být přímo viditelná, ale nějakým i velmi komplikovaným, nepřímým a neznatelným způsobem je v šifrovém textu přítomna. V určitém bodě je této informace v šifrovém textu obsaženo takové množství, že je možný jen jediný otevřený text. Tento počet znaků šifrového textu se nazývá **vzdálenost jednoznačnosti**. K odvození vzdálenosti jednoznačnosti použijeme Shannonovu teorii.

Příklad: Jednoduchá záměna. Pokud obdržíme jeden znak šifrového textu "Z", nemáme

ještě žádnou informaci o textu otevřeném. Pravděpodobnost, že se pod šifrovým textem skrývá písmeno A (resp. B, C, ..., Z) je stejná jako pravděpodobnost výskytu písmene A (B, ..., Z) v otevřeném textu p_A (resp. p_B, ... , p_Z), tj. nedověděli jsme se o otevřeném textu žádnou informaci. V případě, že zvýšíme počet písmen šifrového textu na 2 - mějme například šifrový text "ZP" - už máme určitou informaci o otevřeném textu. Otevřeným textem nemohou být všechny možné bigramy, protože vylučujeme bigramy se stejnými písmeny. Ty by totiž dávaly šifrový text typu "ZZ". Při padesáti písmenech už v šifrovém textu může i laik rozeznávat samohláskové a souhláskové pozice a může je určovat. Při tisíci znacích šifrového textu je snad již jasné, že otevřený text je plně nebo až na detaily plně určen. Někde mezi číslem 1 a 1000 je tedy bod, kdy je otevřený text už jen jeden. Je vidět, že vzdálenost jednoznačnosti závisí na tom, jaký je jazyk otevřeného textu. S tím souvisí pojem entropie zdroje zpráv.

7.3.2. Entropie

Teorie informace měří množství informace, obsažené ve zprávě, její entropií, tj. průměrným počtem bitů, nezbytných k jejímu zakódování při optimálním kódování (optimální kódování používá co nejméně bitů k zakódování zpráv).

Příklady

- Pokud zdroj zpráv vydává binární znaky 0, 1 náhodně a se stejnou pravděpodobností, jeho entropie je právě jeden bit na každý znak.
- Pokud zdroj zpráv vydává čtyři znaky 0, 1, 2, 3 náhodně a se stejnou pravděpodobností, jeho entropie je 2, protože optimální kódování je v tomto případě binární vyjádření uvedených čísel: 00, 01, 10, 11. Délka optimálního kódu bude 2, což je $\log_2 4$.
- Pokud zdroj zpráv vydává n znaků náhodně a se stejnou pravděpodobností, jeho entropie je $\log_2 n$.

Entropie zdroje zpráv.

Pokud zdroj zpráv vydává různé zprávy X(1), X(2),..., X(n) s obecně různou pravděpodobností p(1), p(2), ..., p(n), p(1) + p(2) + ... + p(n) = 1, definujeme jeho entropii jako

$$\mathbf{H(X) = p(1)*\log_2(1/p(1)) + p(2)*\log_2(1/p(2)) + \ ... + p(n)*\log_2(1/p(n)),}$$

tedy volně řečeno jako vážený průměr entropií jednotlivých zpráv.

Maximální entropie

Nechť p(1) + p(2) + ... + p(n) = 1, potom lze dokázat, že

$$H(X) = p(1)*\log_2(1/p(1)) + p(2)*\log_2(1/p(2)) + \ ... + p(n)*\log_2(1/p(n)) \leq$$
$$(1/n * \log_2(n)) + ... + (1/n * \log_2(n)) = \log_2 n,$$

tedy maximální entropie nabývá zdroj, který produkuje všechny zprávy se stejnou pravděpodobností. Entropie tohoto zdroje je rovna dvojkovému logaritmu počtu možných zpráv.

Příklady:

- Pokud bychom náhodně vybírali obyvatele a zaznamenávali jen jejich pohlaví, dostali bychom zdroj zpráv, který vydává zprávu 0 (například muž) a 1 (žena). Tyto zprávy

mají různou pravděpodobnost, a proto entropie tohoto zdroje je menší než jeden bit na zprávu. Máme $H(X) = - p(1)*\log_2 p(1) - p(2)*\log_2 p(2)$. Předpokládejme, že žen je cca 55% a mužů 45%. Entropie tohoto zdroje by byla $H(X) = - 0.45*\log_2(0.45) - 0.55*\log_2(0.55) = 0{,}518 + 0{,}474 = 0{,}992$.

- Mějme zdroj, který vydává pouze jednu zprávu s pravděpodobností 1. Potom entropie takového zdroje je $H(X) = - (1 * 0) = 0$, tedy 0 bitů. Daný zdroj nemá žádnou neurčitost a zprávy z něj nenesou žádnou informaci.

7.3.3. Přirozené jazyky jako zdroje zpráv.

Přirozené jazyky jsou také zdroji zpráv. Za zprávy můžeme považovat písmena, slova, věty, celé knihy. Tyto zdroje produkují různé zprávy s obecně různou pravděpodobností, například v diplomatickém jazyce bude mít slovo „diplomat“ nebo „prezident“ vysokou pravděpodobnost a slovo "homomorfismus" nebo "algebra" velmi nízkou pravděpodobnost. V jazyce matematiků budou tyto pravděpodobnosti obrácené. Některé zákonitosti jazyka lze poměrně dobře popsat entropií.

Uvažujme jazyk, který používá L písmen. Jazyk aproximujeme pomocí množiny X_N všech jeho N-znakových zpráv, $N = 1, 2, 3, ... , N \rightarrow \infty$.

Obsažnost jazyka pro zprávy délky N znaků definujeme jako výraz

$$R_N = H(X_N)/N,$$

tj. *průměrnou entropii na jeden znak* (průměrný počet bitů informace v jednom znaku). Pokud by zprávy v daném jazyce tvořeném L stejně pravděpodobnými znaky, byly stejně pravděpodobné, dostali bychom

$$R = (\log_2(L^N)) / N = \log_2 L.$$

Tento výraz nazýváme **absolutní obsažnost jazyka (R)**. Absolutní obsažnost dosahuje takový jazyk, který nemá žádnou strukturu a všechny N-znakové řetězce písmen jsou stejně pravděpodobné pro každé $N = 1, 2, 3, ...$ Je to maximální neurčitost, kterou přirozené jazyky nemohou dosáhnout, neboť strukturu mají a jednotlivé znaky, slova a věty mají odlišné pravděpodobnosti.

Dále si povšimněme, že u přirozených jazyků výraz $R_N = H(X_N)/N$ pro zvyšující se N klesá, neboť máme-li dlouhou zprávu, její další písmeno bývá v řadě případů určeno už jednoznačně nebo je možný jen malý počet variant. Proto se průměrný počet bitů informace na jedno písmeno postupně zmenšuje s délkou zprávy N, tj. R_N klesá.

Například máme-li 13 znakovou zprávu „Zítra odpoled„ , je pravděpodobné, že pokračuje písmenem n. U 14. znaku nepřibude téměř žádná entropie. U ostatních možných 13-znakových řetězců bude situace podobná - umožní jen jednu nebo několik málo variant.

Výraz $R_N = H(X_N)/N$ lze vypočítat pro daný jazyk. Praktické experimenty ukazují, že pro N jdoucí do nekonečna u přirozených jazyků hodnota R_N klesá ke konstantě (r):

$$\lim_{N \rightarrow \infty} R_N = r.$$

Konstantu r nazýváme **obsažnost jazyka** vzhledem k jednomu písmenu. Říká nám, kolik bitů

informace průměrně obsahuje jedno písmeno jazyka. Pokud jazyk zapisujeme L písmeny, na každé spotřebujeme při zápisu R bitů, ve skutečnosti by však každé písmeno šlo průměrně zapsat r bity. Na zápis jednoho písmena tak vynaložíme nadbytečně

$$D = R - r$$

bitů. Proto hodnotu D nazýváme **nadbytečnost jazyka** vzhledem k jednomu písmenu. Číslo D/R pak udává, kolik bitů je nadbytečných v jenom písmenu procentuálně.

Příklad:
Pro angličtinu máme
L = 26
$R = \log_2 26 = 4.7$ bitů na písmeno
r = 1.5 bitů na písmeno
D = 3.2 bitů na písmeno
Angličtina má tak D/R = 3.2/4.7 = 68 % nadbytečnost.

Příklad
Mějme množinu N-znakových zpráv v daném jazyce, který má absolutní obsažnost R bitů a obsažnost jazyka r bitů. Počet bitů informace, která bude obsažena v N-znakové smysluplné zprávě bude tedy N*r bitů. Smysluplných zpráv lze tedy očekávat cca 2^{rN}. Všech možných (smysluplných i nesmyslných) zpráv sestavených z N znaků je 2^{RN}.

7.3.4. Výpočet vzdálenosti jednoznačnosti pro šifrovací systém

Nadbytečnost jazyka je základem luštění. Bez nadbytečnosti bychom nebyli schopni určit, jestli rozluštěný text je správný nebo ne. Určuje se podle smysluplnosti v daném jazyce.

Následující úvaha ilustruje pro metodu výpočtu vzdálenosti jednoznačnosti pro šifrovací systém. U některých šifer je nutná modifikace, vyplývající z toho, že odlišně ukrývají jazyk otevřené zprávy v šifrové zprávě.

Mějme množinu zpráv M, množinu šifrových textů C a množinu šifrovacích klíčů K. Je-li H(K) neurčitost klíče a uvažujeme-li klíče stejně pravděpodobné, máme celkem $2^{H(K)}$ klíčů. Předpokládejme, že máme šifrový text *c* délky N znaků a že pro klíče $k \in K$ jsou odpovídající otevřené texty $D_k(c)$ vybírány z množiny všech zpráv M nezávisle a náhodně.

V množině M je celkem 2^{RN} zpráv a z toho je 2^{rN} smysluplných zpráv. Pravděpodobnost, že náhodně vybraná zpráva je smysluplná, je $2^{rN} / 2^{RN}$. Pokud provedeme dešifrování šifrového textu c všemi možnými $2^{H(K)}$ klíči, dostáváme $2^{H(K)}$ náhodně vybraných zpráv. Z nich je smysluplných ve střední hodnotě pouze $S = 2^{H(K)} * (2^{rN} / 2^{RN}) = 2^{H(K) + rN - RN} = 2^{H(K)-N(R-r)} = 2^{H(K)-DN}$. Abychom dostali pouze jednu zprávu - tu, která byla skutečně zašifrována - musí být S = 1. Odtud dostáváme H(K) = DN a N = H(K)/D. Vzdálenost jednoznačnosti je proto definována jako

$$\mathbf{N = H(K)/D,}$$

kde H(K) je neurčitost klíče a D je nadbytečnost jazyka otevřené zprávy.

7.3.5. Příklady výpočtu vzdálenosti jednoznačnosti

Příklad: vzdálenost jednoznačnosti jednoduché substituce
Mějme obecnou jednoduchou substituci nad anglickou abecedou. Její vzdálenost jednoznačnosti je

$$N = H(K)/D = \log_2(26!)/3.2 = 88.3/3.2 = 27.6.$$

V šifrovém textu o 28 znacích je tedy dostatečné množství informace na to, aby skrýval (průměrně) jediný možný otevřený text. K rozluštění jednoduché substituce v angličtině postačí tedy v průměru 28 písmen šifrového textu.

Příklad: vzdálenost jednoznačnosti u Vigenerovy šifry
Mějme klíč Vigenerovy šifry o délce *V náhodných* znaků a uvažujme otevřený text z anglické abecedy. Vzdálenost jednoznačnosti je

$$N = H(K)/D = \log_2(26^V)/3.2 = V * \log_2(26)/3.2 = V*4.7/3.2= V*1.5.$$

Poznámka. To je na první pohled optimistický výsledek, který je nutno vysvětlit. Dejme tomu, že máme šifrový text v délce jeden a půl násobku hesla, tj. oněch V*1.5 znaků. První a třetí polovina textu používá stejné heslo, které lze eliminovat odečtením a poté s určitou pravděpodobností vyluštit první a třetí polovinu otevřeného textu. Zbývá neznámá druhá polovina otevřeného textu, kde je heslo zcela náhodné a neznámé, nemáme tedy z něj žádnou informaci. Zbývá redundance otevřeného textu, z něhož známe první a třetí polovinu. Vzorec poskytuje střední hodnotu vzdálenosti jednoznačnosti, nebere v potaz, že máme k dispozici právě takové rozložení informace z otevřeného textu. V zásadě však z hlediska množství informace pokud máme dvě třetiny otevřeného textu, zbytek bychom měli být schopni u anglického jazyka doplnit. Problém je v tom, že na krátkých úsecích nedosahuje nadbytečnost jazyka hodnoty 3.2 bitu, ale méně, a druhý problém je v tom, že konkrétní rozložení informace o otevřeném textu je v daném případě atypické (je známa první a poslední třetina textu), tedy není možné na něj vztahovat výsledky, týkající se *průměru* a průměrných - obvyklých textů. Na druhou stranu, pokud bychom měli k dispozici 2*V znaků, budeme už schopni heslo eliminovat a luštit metodou knižní šifry. Kdybychom měli o pár znaků méně, byli bychom ještě schopni tyto znaky doplnit právě z důvodu nadbytečnosti zprávy. Skutečnou vzdálenost jednoznačnosti lze proto v závislosti na konkrétním problému a konkrétní hodnotě V očekávat v rozmezí 1.5 až 2.0 násobku délky hesla.

Příklad: vzdálenost jednoznačnosti u transpozice
Mějme obecnou transpozici v délce bloku V = 10 nad anglickou abecedou. Její vzdálenost jednoznačnosti je

$$N = H(K)/D = \log_2(V!)/3.2 = \log_2(10!)/3.2 = 21.8.$$

V šifrovém textu o 22 znacích (ve skutečnosti 2 nebo 3 úplné bloky o 10 písmenech) je tedy dostatečné množství informace na to, aby zbýval v průměru jediný otevřený text. Uvedený výsledek koresponduje s možností luštění do hloubky, kdy pod sebe napíšeme 2 - 3 bloky šifrového textu o 10 písmenech. Při luštění již můžeme využít 2 - 3 bigramové vazby na každou dvojici sloupců.

Poznámka: Při výpočtu vzdálenosti jednoznačnosti u transpozice bychom měli udělat korekci ve vzorci pro N. Vrátíme se proto k úvaze, kolik vznikne smysluplných textů po odšifrování daného šifrového textu všemi klíči. Protože šifrový text odpovídající transpozici zachovává četnosti znaků stejné jako v otevřeném textu, musíme vždy jeho odšifrováním obdržet text, zachovávající četnosti písmen otevřeného jazyka. Smysluplných textů je po odšifrování daného šifrového textu všemi klíči tentokrát nikoli $S = 2^{H(K)} * (2^{rN} / 2^{RN}) = 2^{H(K)} / 2^{DN}$, ale $S = 2^{H(K)} * (2^{rN} / 2^{FN})$, kde 2^{FN} je počet všech možných (smysluplných i nesmysluplných) textů,

jejichž četnosti znaků odpovídají četnostem znaků v daném jazyce. Máme F = $p(A)*\log_2(1/p(A)) + p(B)*\log_2(1/p(B)) + ... + p(Z)*\log_2(1/p(Z))$. Odtud N = H(K) / (F - r).

Příklad: vzdálenost jednoznačnosti u blokové šifry AES
Uvažujme otevřený text nad anglickou abecedou a blokovou šifru AES (má délku bloku 128 bitů) se 128bitovým klíčem. Důležité je, v jaké formě je otevřený text šifře předkládán, tj. jak je kódován do 128bitového vstupu. Uvažujme běžnou počítačovou praxi, kdy jeden znak otevřeného textu je reprezentován jedním bajtem. V takovém případě máme z 8 bitů vstupu pouze 1.5 bitu informace, čili nadbytečnost D je v tomto případě 6.5 bitu na bajt. Vzdálenost jednoznačnosti je tedy

$$N = H(K)/D = \log_2(2^{128})/6.5 = 128/6.5 = 19.7 \text{ bajtů otevřeného textu.}$$

Potřebovali bychom tedy 19.7 bajtů šifrového textu, což je jeden celý blok (16 bajtů) a část druhého bloku. Dva bloky šifrového textu by proto měly být dostačující.

7.3.6. Vzdálenost jednoznačnosti a složitost

Vzdálenost jednoznačnosti nám dává odhad **množství informace**, nutného k vyluštění dané úlohy. Neříká však nic o **složitosti** takové úlohy. To je dobře patrné na vzdálenosti jednoznačnosti u šifry AES, kde víme, že informace, obsažená ve dvou blocích šifrového textu je dostačující k určení otevřeného textu, ale nevíme, jak tento otevřený text určit, kromě útoku hrubou silou na klíč.

8. Proudové šifry

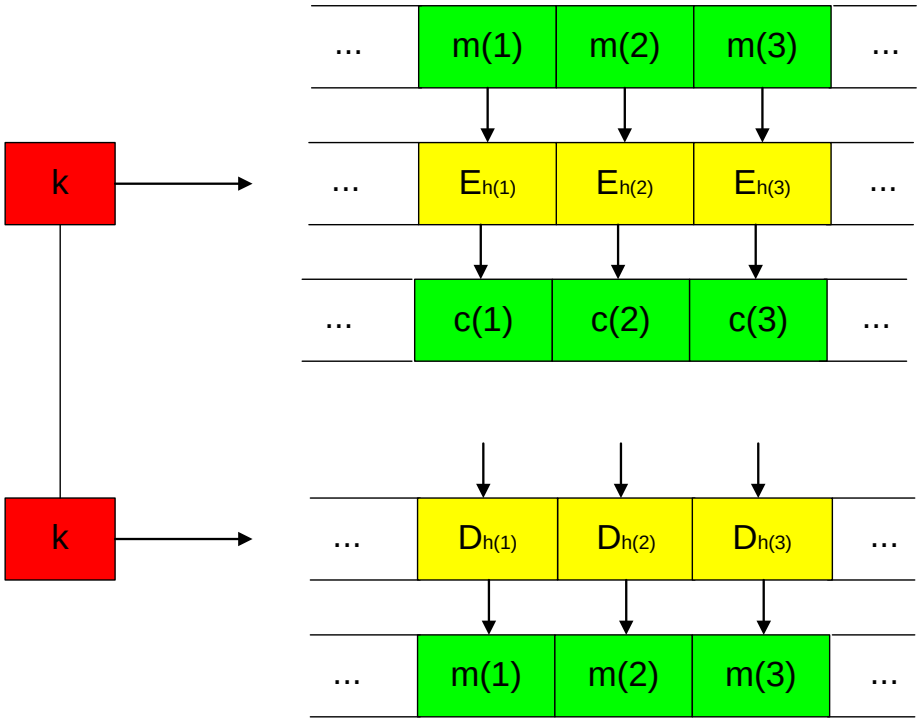
8.1. Rozdíl mezi blokovými a proudovými šiframi

Z hlediska použití klíče ke zpracování otevřeného textu rozeznáváme dva základní druhy symetrických šifer - proudové a blokové. Nechť otevřený text používá vstupní abecedu **A** o q symbolech. **Proudová šifra šifruje zvlášť jednotlivé znaky otevřeného textu, zatímco bloková šifra šifruje najednou bloky *t* znaků otevřeného textu**.

Z kryptografického hlediska je pro blokové šifry podstatné, že všechny bloky jsou šifrovány (dešifrovány) stejnou transformací E_k (D_k), kde *k* je šifrovací klíč. Naproti tomu proudové šifry nejprve z klíče *k* vygenerují posloupnost hesla h(1), h(2),... , přičemž každý znak otevřeného textu je šifrován (obecně) jinou transformací $E_{h(i)}$, kterou určuje jeho pozice (i) a odpovídající hodnota hesla h(i) na této pozici.

8.2. Definice obecné proudové šifry

Definice: symetrická *proudová* šifra
Nechť A je abeceda q symbolů, nechť M = C je množina všech konečných řetězců nad A a nechť K je množina klíčů. Proudová šifra se skládá z transformace (generátoru) G, zobrazení E a zobrazení D. Pro každý klíč k$\in$ K generátor G vytváří posloupnost hesla h(1), h(2),... , přičemž prvky h(i) reprezentují libovolné substituce $E_{h(1)}$, $E_{h(2)}$, ... nad abecedou A. Zobrazení E a D každému klíči k $\in$ K přiřazují transformace zašifrování E_k a odšifrování D_k. Zašifrování otevřeného textu m = m(1), m(2), ... probíhá podle vztahu
$c(1) = E_{h(1)}(m(1))$, $c(2) = E_{h(2)}(m(2))$, ...
a dešifrování šifrového textu c = c(1), c(2), ... probíhá podle vztahu
$m(1) = D_{h(1)}(c(1))$, $m(2) = D_{h(2)}(c(2))$, ...
kde $D_{h(i)} = E_{h(1)}^{-1}$.



Obr.: Princip proudových šifer

Poznámka:

- Z historických důvodů nazýváme G generátor hesla, neboť h(1), h(2) ,... bývá proud znaků abecedy A a substituce $E_{h(i)}$ posunem v abecedě A o h(i) pozic, tj. c(i) = (m(i) + h(i)) mod q. Proudové šifry jsou příkladem historických tzv. **heslových systémů**. V anglické literatuře se heslo h(1), h(2) ,... nazývá **running-key** nebo **key-stream** (keystream), tj. proud klíče, i když se jedná o derivát originálního klíče k.
- Pokud se proud hesla začne od určité pozice opakovat, říkáme, že jde o periodické heslo a periodickou šifru.
- Vigenerova šifra je periodickou šifrou.

8.3. Moderní proudové šifry

Moderní proudové šifry pracují nad abecedou A={0,1}, tj. q = 2. Sčítání modulo 2 se nazývá binární sčítání a označuje se znaménkem ⊕ (xor). Schématicky bychom zašifrování a odšifrování zapsali jako

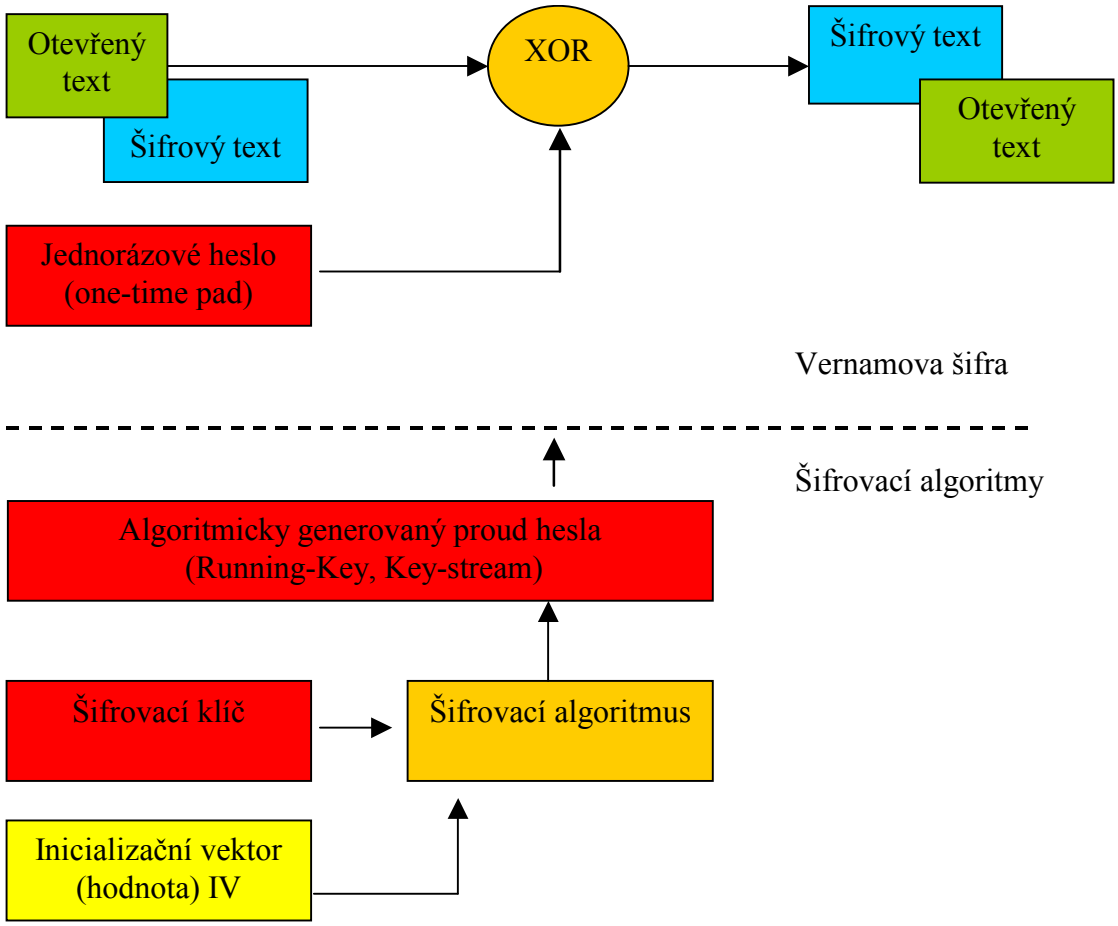
ŠT = OT ⊕ H
OT = ŠT ⊕ H.

Poznamenejme, že díky rovnosti binárního sčítání a odčítání je transformace pro zašifrování a odšifrování stejná. Jako u všech symetrických šifer, odesílatel i příjemce musí mít k dispozici tentýž klíč, tj. totéž heslo. Heslo může být vytvořeno zcela náhodně jako u Vernamovy šifry nebo může být vygenerováno deterministicky nějakým generátorem na základě šifrovacího klíče. Generátor se pak často nazývá přímo šifrovacím algoritmem, i když ve skutečnosti šifrovacím algoritmem je binární sčítání.

8.4. Vernamova šifra

Vernamova šifra používá náhodné heslo stejně dlouhé jako otevřený text a po použití se ničí, takže nikdy není použito k šifrování dvou různých otevřených textů. Tento způsob šifrování

navrhl major americké armády Joseph Mauborgn krátce po první světové válce, ale nazývá se Vernamova šifra po Gilbertu Vernamovi, který si ji nechal patentovat. Gilbert Vernam si jako zaměstnanec americké telefonní a telegrafní společnosti ATT nechal v roce 1917 patentovat šifrovací zařízení pro ochranu telegrafických zpráv, v němž na otevřený text, reprezentovaný pěticemi bitů (v 32 znakovém Baudotově kódu) se bit po bitu binárně načítá náhodná posloupnost bitů klíče, vyděrovaného na papírové pásce. Klíčová posloupnost se generovala náhodně a použité heslo se ničilo.



Obr.: Vernamova šifra a princip proudových šifer

8.5. Absolutně bezpečná šifra

Ukážeme, že Vernamova šifra má **vlastnost absolutní bezpečnosti** (anglicky perfect secrecy, dokonalé utajení, v češtině se ale vžil pojem absolutně bezpečná šifra). Uvažujme i-tý znak šifrového textu c(i). Nechť h(i), o(i) jsou po řadě daný bit hesla a otevřeného textu na i-té pozici. Máme $P\{o(i) = 0\} = P\{c(i) \oplus h(i) = 0\} = P\{h(i) = c(i)\}$.
Tento výraz je roven
$P\{h(i) = 0\}$, v případě, že $c(i) = 0$
$P\{h(i) = 1\}$, v případě, že $c(i) = 1$.
Protože $P\{h(i) = 0\} = P\{h(i) = 1\} = 1/2$, je v obou případech výraz roven 1/2, tedy celkově $P\{o(i) = 0\} = 1/2$. Obdobně ukážeme, že $P\{o(i) = 1\} = 1/2$.
Odtud vyplývá, že $P\{o(i) = 0\} = P\{o(i) = 1\} = 1/2$ nezávisle na hodnotě šifrového textu c(i). Jinými slovy šifrový text nenese žádnou informaci o otevřeném textu, což je definice absolutně bezpečné šifry.

8.6. Algoritmické proudové šifry

Místo generování náhodného hesla a transportu jeho nosičů na obě komunikující strany se postupem času začaly používat stroje, které heslo generovaly různými algoritmy. Nejprve to byly **mechanické, poté elektrické a nakonec elektronické šifrovací stroje**. Heslo se těmito šifrátory "vypočítávalo" a distribuovaly se pouze šifrovací klíče pro nastavení těchto šifrátorů. Aby klíč nemusel být měněn příliš často, zavedl se **princip náhodně se měnícího inicializačního vektoru (IV)**. IV se pro každou zprávu vybírá náhodně a je přenášen na počátku spojení v otevřené podobě. Inicializační vektor nastavuje příslušný algoritmus (konečný automat, šifrátor) vždy do jiného (náhodného) počátečního stavu, což i při stejném tajném klíči umožňuje generovat vždy jinou heslovou posloupnost. Za kryptografickou **kvalitu** (náhodnost) heslové posloupnosti odpovídá použitý algoritmus, za **utajenost** této heslové posloupnosti odpovídá tajný šifrovací klíč, za její **různost** odpovídá IV. Tento princip se s malou obměnou využívá i u blokových šifer.

8.7. Vlastnosti proudových šifer, synchronní a asynchronní šifry

8.7.1. Použití

Proudové šifry se používají zejména u tzv. **linkových šifrátorů**, kdy do komunikačního kanálu přichází jednotlivé znaky v pravidelných nebo nepravidelných časových intervalech, přičemž v daném okamžiku je nutné tento znak okamžitě přenést, takže není vhodné nebo možné čekat na zbývající znaky bloku.

8.7.2. Propagace chyby

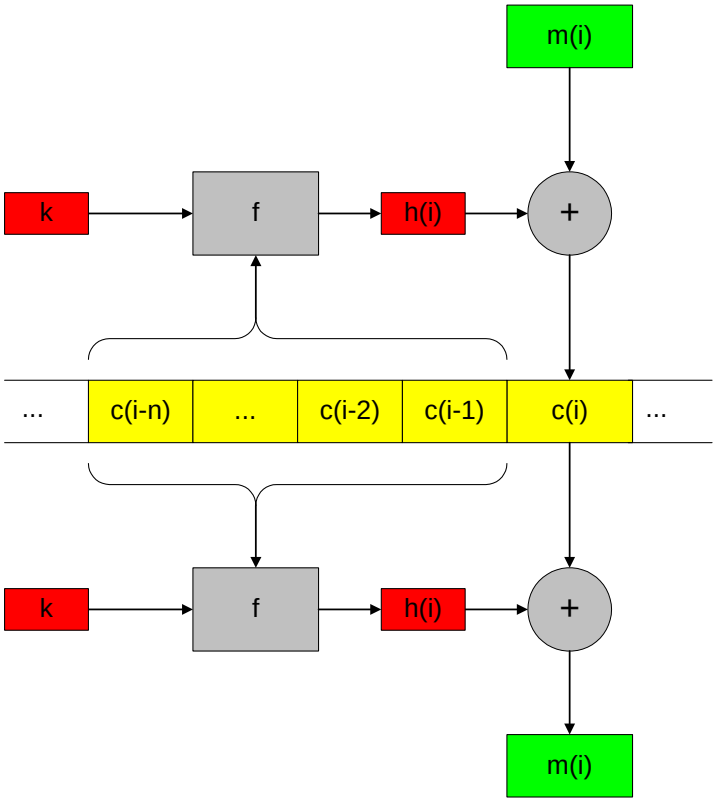
Další výhodou proudových šifer oproti blokovým je malá "**propagace chyby**". Pokud vznikne chyba na komunikačním kanálu v jednom znaku šifrového textu, projeví se tato chyba u proudových šifer pouze v jednom odpovídajícím znaku otevřeného textu, u blokové šifry má vliv na celý blok znaků.

8.7.3. Synchronní proudové šifry

V případě, že proud hesla nezávisí na otevřeném ani šifrovém textu, hovoříme o **synchronních** proudových šifrách. V tomto případě musí být příjemce a odesílatel přesně synchronizováni, protože ***výpadek* jednoho znaku šifrového textu při dešifrování naruší veškerý následující otevřený text**.

8.7.4. Asynchronní proudové šifry

Šifry, které umí eliminovat takové chyby, se nazývají **asynchronní** nebo **samosynchronizující se šifry**. U nich dojde v krátké době k synchronizaci a správné dešifraci zbývajícího otevřeného textu. Této vlastnosti se může docílit například tím, že **proud hesla je generován pomocí klíče a *n* předcházejících znaků *šifrového* textu**:
$h(i) = f(k, c(i-n), ..., c(i-1))$.
V tomto případě se výpadek některého znaku šifrového textu projeví celkem na n sousledných znacích otevřeného textu, ale další otevřené znaky budou již správně dešifrovány. Z definice tvorby hesla je vidět, že k synchronizaci dojde, jakmile se přijme souvislá posloupnost $n+1$ správných znaků šifrového textu.



Obr.: Asynchronní (samosynchronizující se) šifry

9. Blokové šifry

9.1. Definice

Definice: Bloková šifra

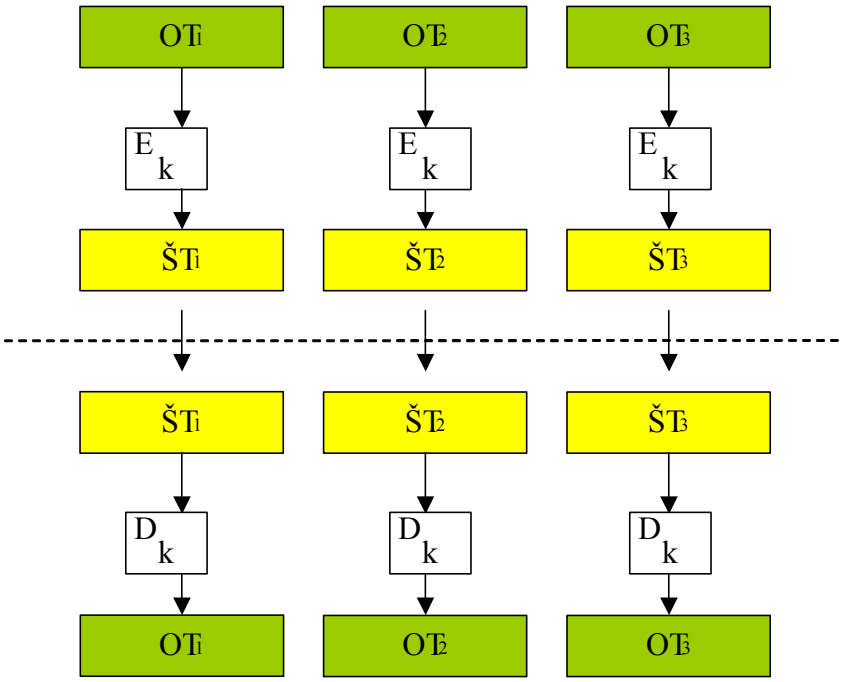
Nechť A je abeceda q symbolů, t ∈ N a M = C je množina všech řetězců délky t znaků nad abecedou A. Nechť K je množina klíčů. Bloková šifra je šifrovací systém (M, C, K, E, D), kde E a D jsou zobrazení, definující pro každé k ∈ K transformaci zašifrování E_k a dešifrování D_k tak, že zašifrování bloků otevřeného textu m(1), m(2), m(3),..., (kde m(i) ∈ M pro každé i∈ N)
probíhá podle vztahu

$c(i) = E_k(m(i))$ pro každé i∈ N

a dešifrování podle vztahu

$m(i) = D_k(c(i))$ pro každé i∈ N.

Pro definici blokové šifry je podstatné, že všechny bloky otevřeného textu jsou šifrovány toutéž transformací E_k a všechny bloky šifrového textu jsou dešifrovány toutéž transformací D_k.



Obr.: Bloková šifra

Šifrovací transformace je tedy jednoduchou substitucí! Klasická substituční šifra nad abecedou o 26 znacích se dá zapsat tabulkou a jednoduše luštit. Bezpečnost blokové šifry je dána tím, že "abeceda" blokové šifry je extrémně velká. Například u moderních šifer s délkou bloku 128 bitů má tato tabulka 2^{128} položek! V tomto případě nejsme dokonce schopni tuto šifrovací tabulku ani celou vypočítat ani uložit.. Ukládá se pouze algoritmus výpočtu, nikoli tabulka této transformace. Luštění takové substituce je diametrálně odlišné od luštění jednoduché substituce nad abecedou o 26 znacích.

9.2. Příklady klasických blokových šifer

Definice: Substituční šifra (bloková šifra s délkou bloku 1)
Nechť A je abeceda q symbolů a M = C je množina všech konečných řetězců nad A. Nechť K je množina všech permutací na množině A. Substituční šifra je bloková šifra s délkou bloku 1 znak. Je tvořená pěticí (M, C, K, E, D), kde E a D jsou zobrazení, definující pro každé $k \in K$ transformaci zašifrování E_k a dešifrování D_k tak, že $E_k = k$ a $D_k = k^{-1}$, tedy pro každé $i \in N$ zašifrování znaku $m(i) \in A$ otevřeného textu na šifrový text c(i) probíhá podle vztahu
$c(i) = E_k(m(i))$
a odšifrování podle vztahu
$m(i) = D_k(c(i))$.

Definice: Transpoziční šifra
Nechť A je abeceda q symbolů, $t \in N$ a M = C je množina všech řetězců délky t nad A. Nechť K je množina všech permutací na množině {1, 2, ..., t}. Transpoziční šifra je bloková šifra, tvořená pěticí (M, C, K, E, D), kde E a D jsou zobrazení, definující pro každé $k \in K$ transformaci zašifrování E_k a dešifrování D_k tak, že
$E_k: M \rightarrow C: m \rightarrow c = (c_1,..., c_t) = (m_{k(1)}, m_{k(2)},..., m_{k(t)})$
a
$D_k : C \rightarrow M: c \rightarrow m = (m_1, m_2,...,m_t) = (c_{l(1)}, c_{l(2)},..., c_{l(t)})$, kde $l = k^{-1}$.

9.3. Difúze a konfúze

Důvod, proč historické šifry jsou luštitelné přímo ze šifrového textu je ten, že šifrový text u nich příliš dobře a přímočaře odráží statistické charakteristiky otevřeného textu. **Shannon navrhl dvě metody**, jak tomu zabránit - **difúzi a konfúzi**.

Cílem **difúze** je rozprostřít statistické charakteristiky otevřeného textu do delších úseků šifrového textu. Jestliže jeden znak otevřeného textu ovlivňuje více znaků šifrového textu, potom se bigramové a vícegramové závislosti jazyka mohou projevit až v delších úsecích šifrového textu. Čím větší je difúze otevřeného textu do šifrového textu, tím je obtížnější tyto závislosti zkoumat.

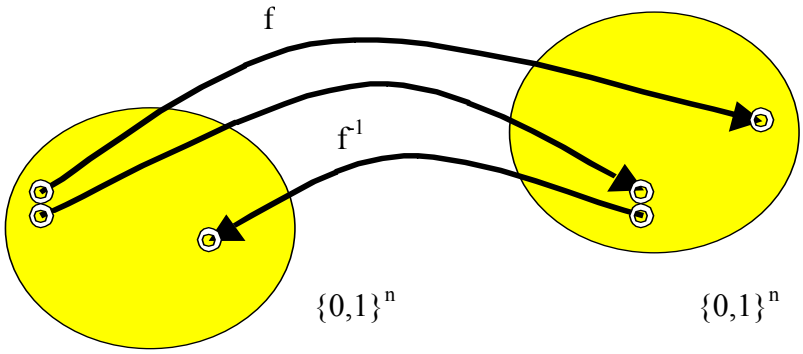
Shannon dále definoval druhou metodu, jak znesnadnit statistickou kryptoanalýzu, tzv. konfúzi. **Konfúze** je metoda, jejímž cílem je učinit vztah mezi statistickými vlastnostmi šifrového textu a klíčem co nejsložitější a zahrnující co největší části šifrového textu a klíče. Jinými slovy konfúze zajišťuje difúzi klíče do šifrového textu a aby tato difúze byla složitá.

Klasické šifrovací systémy (jednoduchá záměna, transpozice, Vigenerova šifra aj.) neměly dobrou difúzi ani konfúzi.

Jednoduchá záměna ponechává zcela nepozměněny vazby mezi hláskami, neboť četnosti jednotlivých hlásek, bigramů, trigramů atd. zůstávají stejné v šifrovém textu jako v otevřeném textu.
Transpozice rozbíjí tyto vazby, ale promítá beze změny rozdělení pravděpodobností jednotlivých hlásek otevřeného textu přímo do šifrového textu.
Vigenerova šifra částečně vyhlazuje rozdělení pravděpodobností hlásek otevřeného textu, ale ponechává nezměněnu určitou část N-gramových závislostí. Jakmile se zjistí délka hesla, je možné využít statistických vlastností otevřeného textu, neboť na hláskách vzdálených od sebe o délku hesla se přímo projeví rozložení četností jednotlivých hlásek otevřeného textu.

9.4. Náhodné permutace na množině $\{0,1\}^n$

Moderní blokové šifry by měly mít takovou vlastnost difúze a konfúze, že bez znalosti šifrovacího klíče by se měly jevit a být nerozlišitelné od náhodných permutací na množině $\{0,1\}^n$, kde n je délka bloku. Potom znalost jakékoli dvojice (OT, ŠT) nepomáhá k odvození možných šifrových textů pro blízké otevřené texty nebo možných šifrových textů pro blízké otevřené texty apod.



Obr. : Náhodná permutace f na množině $\{0,1\}^n$

Navíc vlastnosti konfúze a difúze by měly zabránit tomu, aby ze znalosti mnoha dvojit (OT, ŠT) šel odvodit použitý šifrovací klíč nebo se o něm získávala nějaká *užitečná* informace. Zajistit takové požadavky je velmi složitý úkol při kryptografickém návrhu šifry, protože ze

Shannonovy teorie víme, že informace o klíči je dostatek už v několika párech bloků otevřeného a šifrového textu.

10. Nejznámější blokové šifry

Nejznámější blokové šifry používaly a používají blok o délce 64 bitů (DES, TripleDES, IDEA, CAST aj.), v současné době se přechází na blok 128 bitů, který používá standard AES. Zastavíme se u DES, neboť na něm lze demonstrovat některé pojmy a slabiny.

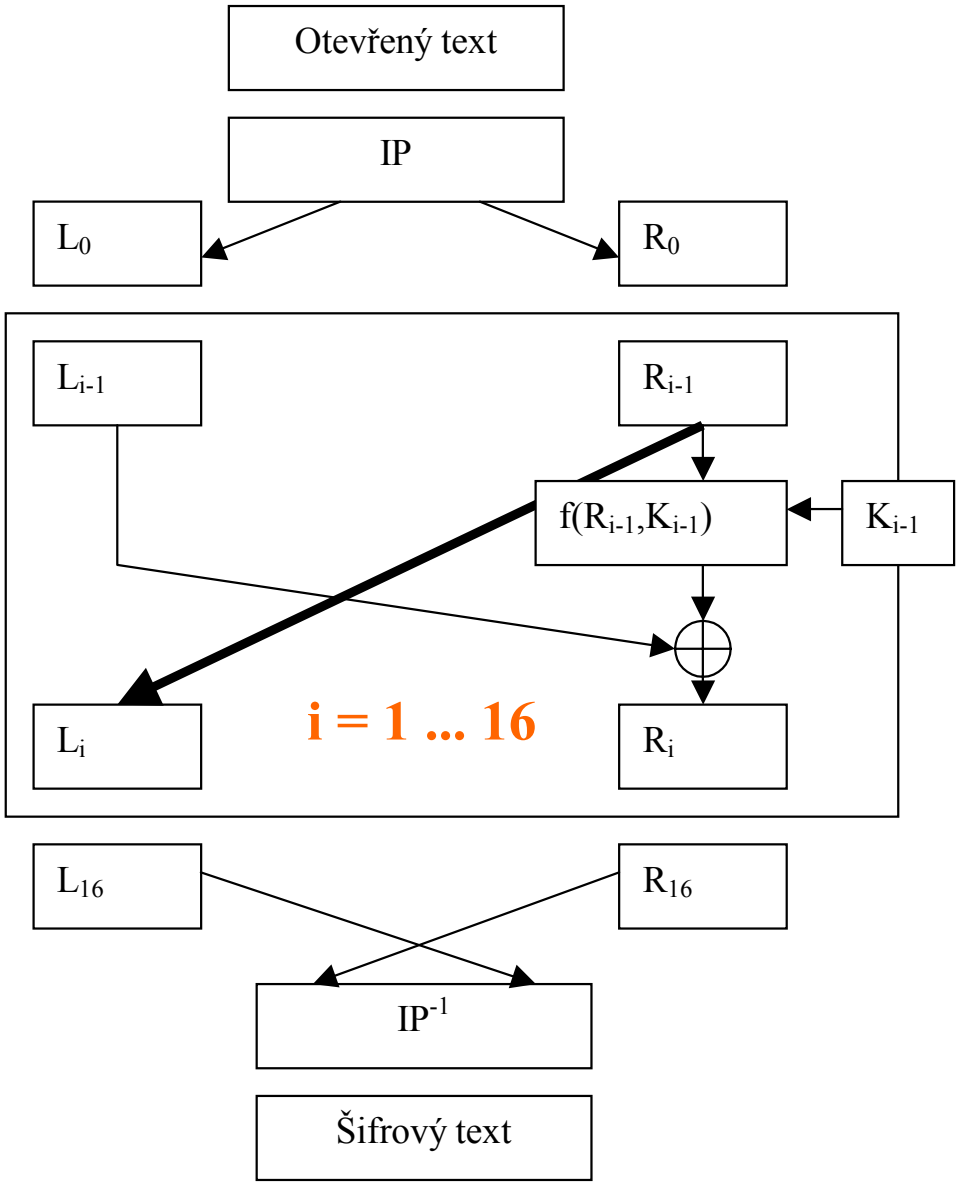
10.1. DES

DES (Data Encryption Standard) byla nejpoužívanější šifra na světě [9]. Byla jako výsledek veřejné soutěže v roce 1977 schválena jako šifrovací standard (Federal Information Processing Standard FIPS 46-3) v USA pro ochranu citlivých, ale neutajovaných dat ve státní správě. Byla součástí mnoha průmyslových, internetových a bankovních standardů (např. ANSI standard X9.32). Už v roce 1977 mnozí upozorňovali na její **příliš krátký klíč 56 bitů**, který byl do původního návrhu IBM zanesen vlivem americké tajné služby NSA. DES se stala předmětem intenzivního výzkumu a mnoha útoků a díky tomu **byly objeveny některé teoretické negativní vlastnosti**. Jedná se zejména o objev tzv. slabých a poloslabých klíčů, vlastnost komplementárnosti a později i teoreticky úspěšnou lineární a diferenciální kryptoanalýzu. V praxi však jedinou zásadní nevýhodou zůstával pouze krátký klíč. V roce 1998 byl zkonstruován stroj DES-Cracker, luštící DES hrubou silou, tj. vyzkoušením všech možných klíčů. V současné době DES jako americký standard již skončil a místo něj byl přijat Triple-DES, definovaný normou FIPS 46-3, od 26.května 2002 je v platnosti šifrovací standard nové generace AES.

10.1.1. Stavební prvky DES

DES je iterovaná šifra typu $E_{k(16)} \bullet E_{k(15)} \bullet \ ... \bullet E_{k(1)}$, používající 16 iterací (rund) a blok délky 64 bitů. Šifrovací klíč k má délku 56 bitů a je v inicializační fázi nebo za chodu algoritmu expandován na 16 tzv. rundovních klíčů k(1) až k(16), které jsou řetězci 48 bitů, každý z těchto bitů je některým bitem původního klíče k.

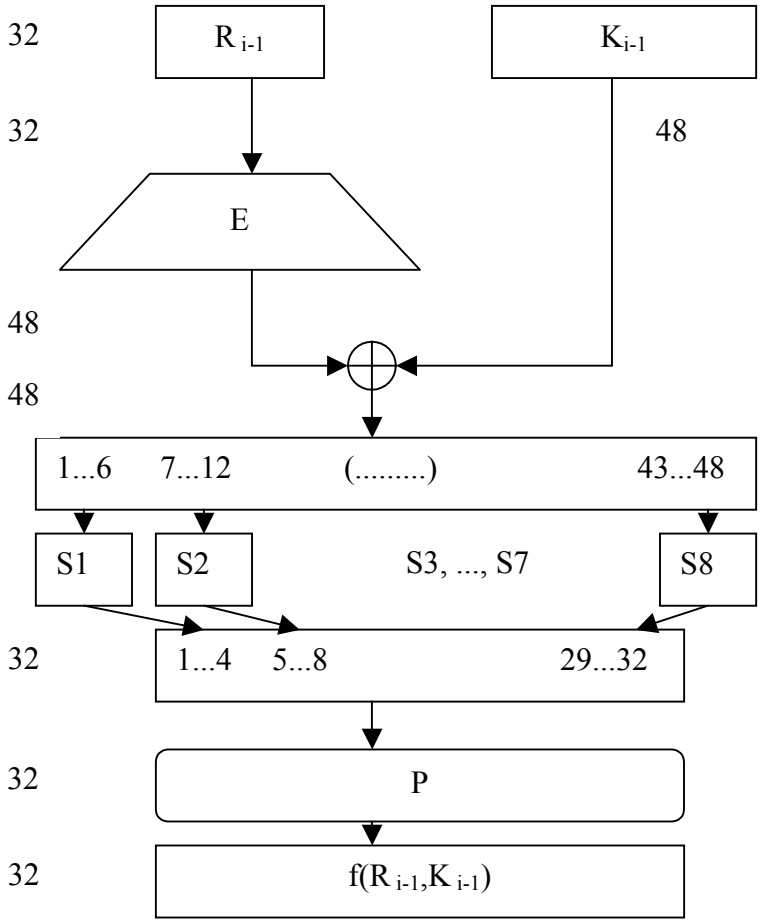
Po počáteční permutaci (IP) je blok rozdělen na dvě poloviny (L, R) o 32 bitech. Každá ze 16 rund (i) transformuje (L, R) na novou hodnotu (L, R), liší se jen použitím jiného rundovního klíče k(i). Po 16. rundě je aplikována závěrečná permutace (IP^{-1}). DES byla konstruována jako tzv. Feistelova šifra proto, aby dešifrování probíhalo stejným způsobem jako zašifrování, pouze je nutno obrátit pořadí výběru rundovních klíčů. To umožnilo snadnější HW realizaci.



Obr.: Základní schéma DES

10.1.2. Rundovní funkce

Rundovní funkce f se skládá z binárního načtení rundovního klíče na vstup, následné substituce na úrovni 4bitových znaků a poté transpozice na úrovni bitů. Tímto způsobem se dosahuje dobré difúze i konfúze.



Obr.: Rundovní funkce DES

10.1.3. S-boxy

Použité substituce se nazývají substituční boxy (S-boxy), jsou jediným nelineárním prvkem schématu. Pokud bychom substituce vynechali, mohli bychom vztahy mezi šifrovým textem, otevřeným textem a klíčem popsat pomocí operace binárního sčítání (xor), tedy lineárními vztahy. Při znalosti jednoho bloku otevřeného textu bychom tak mohli sestavit soustavu 64 lineárních rovnic se známými bity OT a ŠT a 56 neznámými bity klíče K.
Vyjádříme-li však vztah mezi výstupním bitem S-boxu a vstupními bity, obdržíme nelineární vztah, obsahující kromě binárního součtu i součiny vstupních bitů. S boxy mají vstup 6 bitů a výstup 4 bity. Ve skutečnosti levý a pravý krajní bit vstupu vybírá jeden ze čtyř S-boxů, který zobrazuje zbývající 4 vstupní bity na 4 výstupní bity. Tyto malé čtyřbitové S-boxy jsou bijektivní. Jejich vlastnosti do značné míry určují kvalitu DES. Na obrázku je ilustrativní obrázek S-boxu zobrazujícího tři bity na tři bity.

x_1	x_2	x_3	y_1	y_2	y_3
0	0	0	0	1	0
0	0	1	0	1	0
0	1	0	0	1	0
0	1	1	1	0	0

1	0	0	0	1	1
1	0	1	1	0	1
1	1	0	1	0	1
1	1	1	1	0	0

Obr.: Příklad S-boxu

10.1.4. Komplementárnost

Je zajímavé, že přes tyto nelinearity platí tzv. vlastnost komplementárnosti, objevená v roce 1976. Jde o to, že pro každý klíč K a každý otevřený text M platí pro jejich bitové negace nonK a nonM tento vztah, který by u blokové šifry jako náhodné permutace nemohl nikdy platit

$$\text{je-li } C = DES_K(M), \text{ potom } \text{non } C = DES_{nonK}(\text{non } M).$$

Důkaz vyplývá z této vlastnosti rundovní funkce: $f(R, K_i) = f(nonR, nonK_i)$.

10.1.5. Slabé a poloslabé klíče

V roce 1976 byly popsány slabé a poloslabé klíče. Pro slabé klíče K platí: $E_K(X) = X$ pro každé X.
Jsou to (hexadecimálně) tyto čtyři hodnoty:

- 0101 0101 0101 0101,
- FEFE FEFE FEFE FEFE,
- 1F1F 1F1F 0E0E 0E0E,
- E0E0 E0E0 F1F1 F1F1.

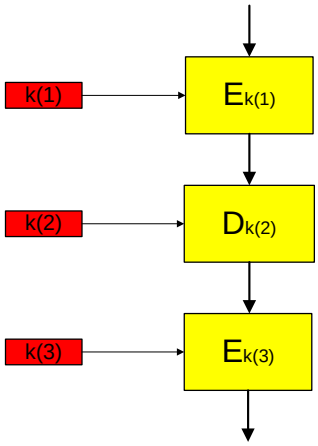
Poloslabé klíče vystupují ve dvojicích (K1, K2) a platí pro ně $E_{K2}(E_{K1}(X)) = X$ pro každé X.
Bylo nalezeno 6 dvojic poloslabých klíčů (K1,K2) :

- 01FE01FE01FE01FE, FE01FE01FE01FE01,
- 1FE01FE00EF10EF1, E01FE01FF10EF10E,
- 01E001E001F101F1, E001E001F101F101,
- 1FFE1FFE0EFE0EFE, FE1FFE1FFE0EFE0E,
- 011F011F010E010, E1F011F010E010E01,
- E0FEE0FEF1FEF1FE, FEE0FEE0FEF1FEF1.

Jedná se o další teoretickou slabinu.

10.2. TripleDES

TripleDES [9] uměle prodlužuje klíč originální DES tím, že používá DES jako stavební prvek celkem třikrát s dvěma nebo třemi různými klíči. Nejčastěji se používá tzv. varianta EDE této šifry, která je definována ve standardu FIPS PUB 46-3 a v bankovní normě X9.52. Vstupní data OT jsou zašifrována podle vztahu $ŠT = E_{K3}(D_{K2} (E_{K1}(OT)))$, kde K1, K2 a K3 jsou buď tři různé klíče nebo K3 = K1. Varianta EDE byla zavedena z důvodu kompatibility, neboť při rovnosti všech klíčů se z TripleDES stává původní DES. Klíč TripleDES je tedy buď 112 bitů (dva klíče) nebo 168 bitů (tři klíče). I když DES byla zlomena hrubou silou, TripleDES (3DES) se považuje za spolehlivou, protože klíč je dostatečně dlouhý a teoretickým slabinám (komplementárnost, slabé klíče) se dá předcházet. Proto je TripleDES vedle AES platným oficiálním standardem, nahrazujícím DES. Pokud se setkáte se zkratkou 3DES-EDE, je to právě popsaná varianta.

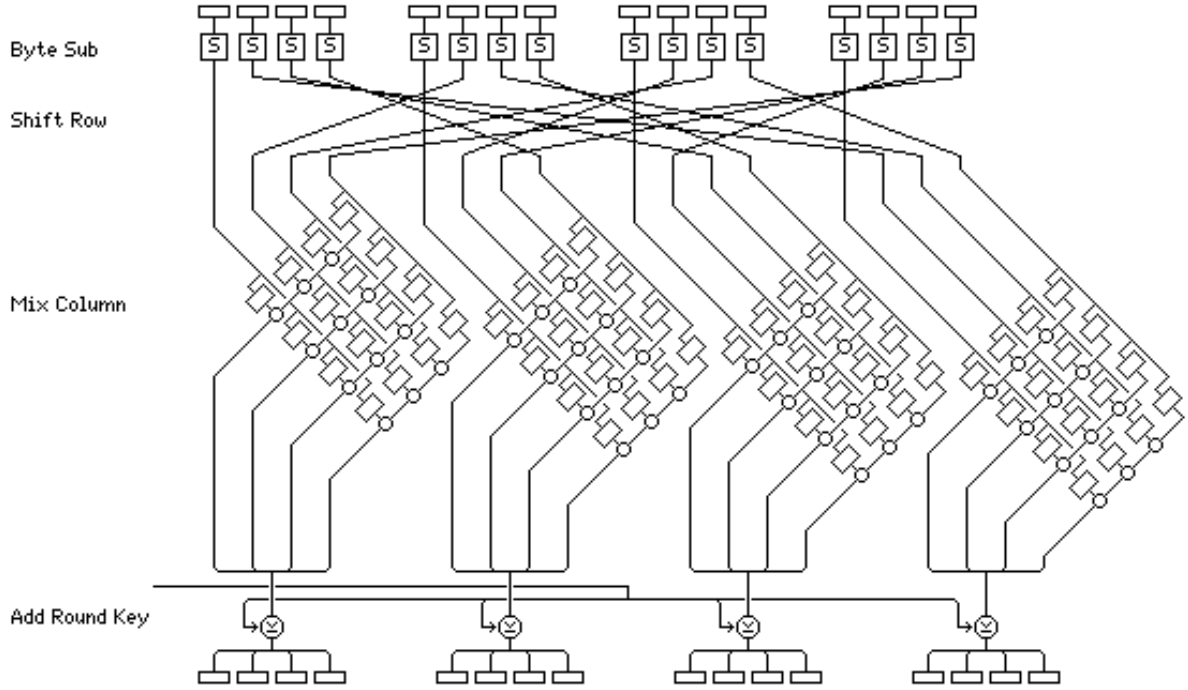


Obr.: TripleDES-EDE

10.3. AES

Jak postupovaly snahy o útok hrubou silou na DES, americký standardizační úřad začal připravovat jeho náhradu - Advanced Encryption Standard (AES).
Na rozdíl od DES bylo na AES vypsáno mezinárodní výběrové řízení (2.1. 1997), které bylo od počátku maximálně otevřené a zahrnovalo nebývalý potečnciál kryptologů z mnoha zemí. Přihlásilo se 15 kandidátů a než byl vybrán vítěz, NIST uspořádal celkem čtyři pracovní konference a několik výběrových kol. Nakonec se z pěti finalistů stal vítězem algoritmus dvou Belgičanů Rijndael (doporučená výslovnost je "Rájndol" s mírně polknutým o nebo "Rejndál") podle jeho autorů Rijmena a Daemena. Jako AES byl přijat s účinností od 26. května 2002 a byl vydán jako standard v oficiální publikaci FIPS PUB 197 [7].

AES je bloková šifra s délkou bloku 128 bitů, čímž se odlišuje od současných blokových šifer, které měly blok 64 bitový. AES podporuje tři délky klíče: 128, 192 a 256 bitů a v závislosti na tom se částečně mění algoritmus (počet rund je po řadě 10, 12 a 14). Větší délka bloku a delší klíče zabraňují mnoha útokům, které byly aplikovatelné na DES a jiné blokové šifry. AES má domovskou stránku http://csrc.nist.gov/encryption/aes/, která je věnována vzniku, konferencím, vědeckým zprávám a dalším informacím. AES nemá slabé klíče jako jeho předchůdce a měl by být odolný proti všem známým útokům, i proti nejnovějším metodám lineární a diferenciální kryptoanalýzy. Na referenčním počítači 200 MHz Pentiu Pro PC byla dosažena rychlost zašifrování cca 30 - 70 Mbitů/s podle použitého programovacího jazyka a délkách klíče. Algoritmus zašifrování i odšifrování se dá výhodně programovat na různých typech procesorů, má malé nároky na paměť i velikost kódu a je vhodný i pro paralelní zpracování. Pokud půjde vše podle předpokladů, AES bude platným šifrovacím standardem po několik desetiletí. Proto se předpokládá, že bude mít obrovský vliv na počítačovou bezpečnost.



Obr.: Rundovní funkce AES

-oOo-