**1. Základní věta aritmetiky**

Kazde prirozene cislo vetsi nez 1 lze jednoznacne rozlozit na soucin prvocisel.

**2. Entropie**

Entropie je množství informace obsažené ve zpráve. Teorie informace merí entropii zprávy prumerným poctem bitu, nezbytných k jejímu zakódování pri optimálním kódování (minimum bitu). Entropie zprávy ze zdroje X je $H(X) = -\sum_{i=1}^{n} p_i \cdot log_2(p_i)$, kde p_1, . . . , p_n jsou pravdepodobnosti všech zpráv X1, . . . ,Xn zdroje X a = pocet bitu nutných k optimálnímu zakódování zprávy X_i.

**3. Rozdíl mezi blokovou a proudovou šifrou + příklady**

Rozdil mezi proudovou a blokovou: Blokova (Hill) bere celistve bloky jako samostatne jednotky a mimo ne nekouka, proudova (Vernamova) kouka na prubeh a vysledky sifrovani predchozich casti zpravy a podle toho se upravi sifrovaci metoda

**4. Vigenerova šifra**

Kryptografická transformace OT $(p_1, p_2, ...)$odvozená od klíce K je $c_j = |p_j + K_{|j|_n}|_N$. Číslo $n$se nazýva periodou Vigenerovské šifry, nebo také délkou klíce.
Jinymi slovy, pri sifrovani se text rozdeli na bloky delky $n$, pak se kazdemu bloku pricita klic. Pri desifrovani se klic odecita.

**5. Způsoby předávání veřejných klíčů**

zverejnení - málo bezpecné
verejne dostupný adresár - stará se o nej správce, možnost nabourání adresáre
pomocí autority pro verejné klíce - všechny Vk jsou uloženy u autority a pred každou komunikací se stahují. Bezpecné, ale hodne komunikace
pomocí certifikátu - certifikacní autorita podepíše verejný klíc svým soukromým klícem, pridají se další údaje a je zajišteno, že certifikát byl vytvoren CA. Dve strany potom mohou komunikovat mezi sebou bez komunikace s CA

**6. Vzdálenost jednoznačnosti**

Vzdálenost jednoznacnosti: $\delta_U = \frac{H(K)}{D}$, kde H(K) je neurcitost klíce a D je redundance jazyka otevrené zprávy

**7. HMAC - popsat nebo nakreslit**

Používá se k overení integrity a autentizaci zároven.
Pres tajný klíc zesiluje hash dané zprávy.
Postup:
1) Zarovnáme tajný klíc nulami
2) Provedem XOR tohoto zarovnaného klíce s cislem 'ipad'
3) K tomuto pripojíme zprava puvodni zprávu.
4) Vystup bodu (3) zahashujeme danou hashovaci funkcí.
5) Provedeme XOR výstupu bodu (1) s cislem 'opad'
6) K tomuto pripojíme výstup bodu (4)
7) Toto celé zahashujeme danou hashovací funkcí a prohlásíme za výstup.

**8. Co je to orakulum?**

Stroj, ktery na stejny vstup odpovida nějakým vystupem. Víme akorát, že když dáme stejný vstup podruhé, dostane stejný "nějaký" výstup

**Vigeneruv autoklav**

Heslo je pouze jedno písmeno klíce, znaky hesla byly tvoreny už prímo predchozím znakem ST (scítání v modulu 26): c1 = p1 + h1,kde h1 = k a ci = pi + hi , i = 2, 3, . . . , kde hi = ci−1.

**Rozdíl mezi šifrováním a kódováním**

Šifrování
Proces převodu OT na ŠT.
Kódování
Zakóduje zpávu do podoby, ve které je možné zprávu přenášet neznámým prostředím tak, aby se zpráva nepoškodila. Např.: Přidání parity, Base64 (datové přílohy v e-mailech atp.)

**Křížová certifikace**

Křížová certifikace je proces, kdy si 2 různé CA (certifikační autority) navzájem podepíší své certifikáty.

**Napsat malou Fermatovu větu**

Nechť $a \in \mathbb{N}$, $p$ je prvočíslo, $p \nmid a$, pak platí kongruence $a^{p-1} \equiv 1 \pmod{p}$

**Rozdíl mezi symetrickou a asymetrickou šifrou**

Symetrická - Pro šifrování i dešifrování se používá stejný klíč.
Asymetrická - Pro šifrování a dešifrování se používají různé klíče - privátní a veřejný.

**1. Co je to kvadratické reziduum a nonreziduum + příklad (6 bodu)**

Pokud m je kladné celé číslo, cele číslo a je kvadratické residuum modulo m, když gcd(a,m) = 1 a kongruence x^2 kong a (mod m) má nejaké rešení. Když tato kongruence nemá žádné rešení ) a je kvadratické nonresiduum modulo m.

**2. Pojem substituce a transpozice (3 body)**

Transpoz/šifrování spoˇcívá v zamíchaní písmen OT

**4. Eulerova veta (2 body)**

Nechť $m \in \mathbb{N}$ a $a \in \mathbb{Z}$. Když $\gcd(a, m) = 1 \Rightarrow \quad a^{\Phi(m)} \equiv 1 \pmod{m}$.

**5. Princip generování hesla a jeho použití u RC4 (4 body)**

Pocítaní s bajty -> redukce modulo 256.
i se systematicky zvyšuje modulo 256, j je náhodný klícove závislý index
klíc - permutace S, která se nejdrív zamíchá dle jiného algoritmu
Hodnota hindex obsahuje heslovou posloupnost generovanou tímto algoritmem.
i = j = 0
for index = 0 to n
{
i = |i + 1|256
j = |j + S(i)|256
vymen mezi sebou hodnoty S(i) a S(j)
hindex = |S( S(i) + S(j))|256
}

**2. definovat multiplikativní inverzi + podmínku existence (3 body)**

Máme císlo a a modul m. Inverze je takové císlo b, pro které platí že $|a \cdot b|_m = |b \cdot a|_m = 1$. Inverze existuje jenom pokud gcd(a,m)=1

**3. Solení (co to je, na co to je) (3 body)**

solí se výchozí hodnota IV
pres verejný kanál se posílá IV ale šifruje se s "osoleným" IV
solení spocívá v zašifrování IV pomocí klíce který znají jenom komunikující subjekty
výhoda tedy je, že se skutecné IV neposílá pres verejný kanál

### 4. Vernamova šifra (3 body)

Heslo je nahodne generovani a stejne dlouhe jako OT, je pouzite jen jednou. Tim se zajistuje dokonala bezpecnost sifrovani. Na OT (5b na písmeno v 32znakovém Baudotov˘e kódu) se bit po bitu binárn˘e na˘cítá náhodná posloupnost bit °u klí˘ce (d ˘ erná páska).

### 5. co je konfúze a difúze (2 body)

Konfúze
technika k potlacení redundance ve zpráve
marí vztahy mezi ŠT a OT
príklad: Caesarova šifra, proudové šifry
Difúze
technika k potlacení redundance ve zpráve
rozprostírá redundanci OT
príklad: transpozice

### 6. Popsat operační mód šifer který slouží k autentifikaci (za 6 bodů)

MAC - message authentication code - zajišťuje integritu dat.
Autentizuje puvod zprávy a reší obranu proti náhodným i úmyslným zmenám nebo chybám na komunikacním kanálu.
MAC je krátký kód, který vznikne zpracováním zprávy s tajným klícem (K1). Klíc by se mel použít jiný než k šifrování zprávy.
Výpocet MAC probíhá tak, že se zpráva jakoby šifruje v modu CBC s nulovým IV, pricemž prubežný ŠT se nikam neodesílá.
MAC je pak tvoren až posledním blokem ŠTn, pricemž je možné ješte jedno prídavné šifrování navíc, tj. MAC = EK2(ŠTn). Z výsledného bloku se obvykle bere jen cást (vetšinou polovina) o délce potrebné k vytvorení odolného zabezpecovacího kódu.
MAC zajišťuje autentizaci puvodu dat. Nezarucuje nepopíratelnost.

### 2. Popište podvržení veřejného klíče

Subjekt A posle cvoj VKa a svoj Ida, subjektu B. Utocnik U ma aktivny pristup k verejnemu kanalu. U zachyti spravu a vytvori novu spravu VKu||Ida a posle B. B si mysli, ze VKa=VKu a zasifruje spravu s VKu a posle A. U desifrujte a ziska M. U pozna VKa, zasifruje M a posle A.

### 3. Co to je transpoziční šifra? Uveďte příklad

OT se napise do obdelnikove matice (tam, kde nestaci OT, se strci nejaky padding jako treba znak 'X')... a ta se transponuje.
Kryptoanalyza se provadi podle onoho odsazeni. Jakmile je jasne, jak daleko jsou od sebe nyni znaky ktere byly predtim za sebou, da se zjistit, jak transponovat matici zpet.

### 4. Jaké kroky má proces luštění šifer?

Identifikace – jaký šifrovací systém byl použit.
Prolomení – zpusob šifrování zprávy, urcení nemenných cástísystému.
Nastavení – urcení, jak se mení prom enlivé cásti kryptosystému.

### 2. Bezpečnost kryptosystému se posuzuje ve třech situacích - vypsat, popsat (5 body)

duvernost - informace se nedostane neautorizovanému subjektu
integrita - data docházejí v nezmenené podobe
dostupnost - kdo je autorizovaný, má kdykoli k informacím prístup

### 5. Popiste nebo nakreslete distribuci verejnych klicu pomoci certifikatu. (4 body)

certifikacní autorita podepíše verejný klíc svým soukromým klícem, pridají se další údaje a je zajišteno, že certifikát byl vytvoren CA. Dve strany potom mohou komunikovat mezi sebou bez komunikace s CA

Heslo je pouze jedno písmeno klícce, znaky hesla byly tvoreny už prímo predchozím znakem ŠT (scítání v modulu 26): c1 = p1 + h1, kde h1 = k a ci = pi + hi , i = 2, 3, . . . , kde hi = ci−1.

### 5. Vlastnosti digitálního podpisu

Nezfalšovatelnost, autentizace – podpis se nedá napodobit jiným subjektem než podepisujícím
Overitelnost – příjemce dokumentu musí být schopen overit, že podpis je platný
Integrita – podepsaná zpráva se nedá zmenit, aniž by se zneplatnil podpis
Nepopiratelnost – podepisující nesmí mít pozdeji možnost poprít,že dokument podepsal

### 7. Generator mod p. Uvedte priklad

|g^a|p = b
- b jsou cisla od 1 do p-1
- a je libovolne cele cislo
- p je prvocislo
- g je konstanta, mensi nez p

### 1. Popiště výpočet dešifrovacího klíče u exponenciální šifry. Dokažte, že dešifrováním ŠT získáte správný OT.

Celé číslo d, pro které platí de kong. 1 (mod m − 1) , d je multiplikativní inverze e modulo (m − 1), která existuje, pokud gcd (e,m − 1) = 1 )

$$|c^d|_m = |(p^e)^d|_m = |p^{ed}|_m = |p^{k(m-1)+1}|_m = |(p^{m-1})^k p|_m = |p|_m,$$

### 2. Dešifrujte následující ŠT zašifrovaný pomocí svislé transpoziční šifry: "NPSEEIBCFENRZAA"

**Řešení: "NEBEZPECNASIFRA"**

### 3. Vysvětlete, co je to řetězec certifikátů a kořenová certifikační autorita.

Entita, ktera vydava digitalni certifikaty ostatnim.
Certifikat obsahuje udaje o majiteli a jeho verejny klic.
Je podepsan digitalnim podpisem certifikacni autority a certifikacni autorita je tedy pojistkou jeho pravosti.

Posloupnost certifikát uu od certifikátu uživatele až k certifikátu korenové CA se nazývá retezec certifikátu

### 4. Otázka na entropii - neco jako "Jaka je pravdepodobnost pri max. entropii?"

**1/n log(2)n**

### 5. Vypocteni redundance angl. textu, kodovani UCS-2, 16b slova

Nadbytecnost (redundance) jazyka: vzhledem k jednomu písmenu vyjadruje kolik bitu je v jednom znaku daného jazyka nadbytecných a je dána výrazem: D = R - r. Číslo 100D/R udává kolik bitu jazyka je nadbytecných procentuálne.

### Zpusob výměny tajemstvi mezi A a B pomoci certifikatu

1. A a B si mezi sebou vymeni certifikaty.
2. Oba si overi platnost certifikatu u svých certifikacnich autorit.
3. Certifikat obsahuje verejny klic a algoritmus pomoci ktereho se ma sifrovat.
4. Pokud A chce zasifrovat neco pro B, pouzije algoritmus a klic ze certifikatu osoby B.
5. Pokud B chce zasifrovat neco pro A, pouzije algoritmus a klic ze certifikatu osoby A.

### PRIKLADY

### MD5 a Damgard-Merklova konstrukce, Padding - doplnění zprávy M obecné délky do bloků(AHOJXXX) a její iterativní zpracování

- text zpracovava v ramci hashe po blocich.

```
H[0] = Inicializacni Vektor
H[i] = F(H[i-1], M[i])
H[posledni] = vystup
```

**Damgard-Merklovo zesileni hashovaci funkce**

- Zprava se doplni tak, aby do nasobku 512b (nebo jineho nasobku, podle zvolene hash fce) zbyvalo 64b. Tam se doplni 64b hodnota delky zpravy. Tim je hash odolnejsi.

**MD5**

- Kontext tvori 4 slova po 32b, A, B, C, D. Posledni kontext je vystup.
- Zprava je zpracovana po skupinach 512b, ktere se dale deli po 32b na 16 bloku.
- Bloky >16 se resi vyxorovanim z predchozich.
- Je preddefinovany IV
- Jsou preddefinovane konstanty K, jedna se priXORuje k dilcimu vysledku. Meni se pro kazdy blok.
- Zpracovava v 16*4 rundach; po kazdych 16 rundach se meni rundovni funkce.
- Na konci se na vystup aplikuje Davies-Meyer

### Feistelova typu(vysvetlit, dat priklady)-byly dany dve matice a ukazat permutaci pro desifrovani (18 bodu)

SLidy od 194. DES, TripleDES, IDEA, CAST

Mame m0,m1 vytvorime c1=m1,m2, kde m2=m0xf1(m1) a c2(m2m3), kde m3=m1xf2(m2).

Desif.: m1=m3xf2(m2) a m0=m2xf1(m2).

### definice algoritmu Feistelova typu. Popsat DES. Pouziti a vyhody TripleDES (18 bodů)

**DES**

```
Klic ma 56b, ale pouziva se 64b, ponevadz dalsi jeden bit u kazdeho
   bytu je parita.
   Blok ma 64b. Rund je 16.

   Sestnact podklicu se vyrabi tak, ze se provadi rotace a permutace klice.

   Na zacatku se provede permutace bitu vstupu, na konci se provede jeste
   jedno prohozeni leve a prave strany a nasledne inverzni permutace bitu
   vystupu.

   Rundovni funkce 'f' vezme danou cast R(i-1) a v bloku 'E' ji expanduje
   z 32b na 48b. K tomu se prixoruje upraveny klic. Vystup se soupne do
   substitucnich boxu (z 48b opet na 32b) a nasledne permutuje.
```

**3DES**

- Tri klice (popr. dva, s tim ze prvni a treti se rovnaji), tri aplikace DES.
- Strida se enkrypce, dekrypce, enkrypce (E, D, E) - duvodem je kompatiblita s DES.

- Sifrovani: OT => E(k1) => D(k2) => E(k3) => ST
- Desifrovani: ST => D(k3) => E(k2) => D(k1) => OT
- 3DES je spolehlivá ! klíˇc je dostateˇcneˇ dlouhý a teoretickým
- slabinám (komplementárnost, slabé klíˇce) se dá pˇredcházet )
- 3DES a AES ! platný oficiální standard nahrazující DES.
- 3DES lze, jako jakoukoliv jinou blokovou šifru, použít v r °uzných
- operaˇcních modech (CBC mod ) 3DES-EDE-CBC).

### Exponencialní šifra-matematicke principy- + Diffie-Hellman pro 3 klienty (14 bodu)

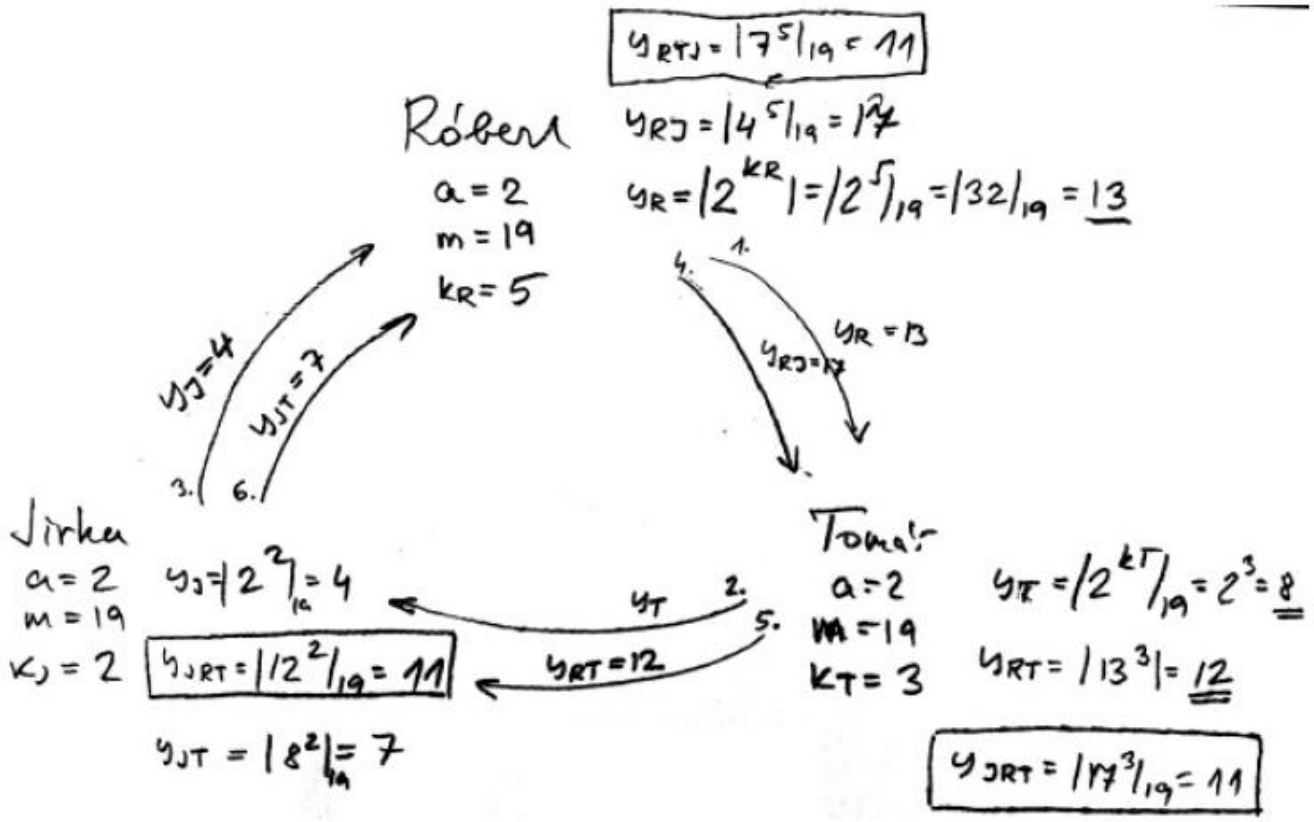
- Znaky se prevedou do dvoumistnych cisel (napr. E = 04, vice viz prednaska 3, slide 19) a "splacnou" se podle zvoleneho cisla m.
- Napriklad pokud 2525 < m < 252525, splacnou se dve takto vytvorena cisla dohromady a funguji jako jedna polozka OT, tedy napr. OT[1] = 'EZ' => 425.
- Funguje podle predpisu:

```
ST[i] = |OT[i] ^ e|m
```

```
Desifruje se pres cislo 'd', ktere je definovano jako:
|d * e|(m - 1) = 1

Pak:
OT[i] = |ST[i] ^ d|m

Predpoklady: gcd(e, m - 1) = 1.
```




## Hash (hlavně: náhod. orákulum, bezkoliznost, odolnost pro kolizím, narozeninový paradox)

**Orakulum**

- Stroj, ktery na stejny vstup odpovida nějakým vystupem. Víme akorát, že když dáme stejný vstup podruhé, dostane stejný "nějaký" výstup

**Nahodne orakulum**

- Podmnozina Orakula - je to stroj, ktery na stejny vstup odpovida stejnym vystupem, ale _vime_ (coz u orakula nevime), ze nema zadny vztah k vstupu.

**Bezkoliznost:**

- 1. rad: Neni vypocetne zvladnultelne najit dve ruzne (jakekoliv) zpravy, kde h(M1) = h(M2)

  Odolnost: $2^{\frac{n}{2}}$

- 2. rad: Neni vypocetne zvladnutelne pro nahodny konkretni vzor x najit druhy vzor y, kde h(x) = h(y)

  Odolnost: $2^n$

**Narozeninovy paradox**

- Narozeninový paradox - Narozeninový paradox říká, že pro n-bitovou hašovací funkci nastává kolize s cca 50% pravděpodobností v množině $2^{\frac{n}{2}}$ zpráv, namísto očekávaných $2^{n-1}$. Tedy s 50% pravděpodobností existují v množině dvě zprávy které jsou stejné.

- "K libovolnemu cloveku z mnoziny hledame jednoho dalsiho, ktery ma narozeniny ve stejny den jako libovolny z nasi mnoziny."

  vs

- "Kolik lidi je potreba, aby se ke konkretnimu cloveku nasel jeden, ktery ma narozeniny ve stejny den."

**Zarovnani**

- Tak, aby bylo mozne jednoznacne odejmout doplnek, tedy jednicka a pak potrebny pocet nul.

**RSA a CRT popsat mat. princip (13 bodů)**

```
Trik spociva v tom, ze si pred-spocitame nektere konstanty a pak diky nim
 muzeme mirne jinak desifrovat.

 Konstanty:
  qinv = |q^-1|p
  dp = |e^-1|(p-1)
  dq = |e^-1|(q-1)

  Verejny klic je stejny jako u RSA, privatni je petice (p, q, dp, dq, qinv).

 Desifrovani:
  m1 = |c^(dp)|p
  m2 = |c^(dq)|q
  h = |(m1 - m2)*qinv|p
  m = m2 + h*q
```

**Popište vlastnosti proudových šifer a rozdíly oproti blokovým šifrám.Vysvětlete význam IV a popište generování hesla u RC4.**

**IV – inicializacna hodnota siforvacieho vektoru.**

- 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 $h_1, h_, \ldots$ přičemž prvky $h_i$ reprezentují libovolné substituce $E_{h_1}, E_{h_2}, \ldots$ nad abecedou $A$.
- Zobrazení $E$ a $D$ každému klíči $k \in K$ přiřazují transformace zašifrování $E_k$ a odšifrování $D_k$.
- Zašifrování OT $m = m_1, m_2, \ldots$ probíhá podle vztahu $c_1 = E_{h_1}(m_1), c_2 = E_{h_2}(m_2), \ldots$
- Dešifrování ŠT $c = c_1, c_2, \ldots$ probíhá podle vztahu $m_1 = D_{h_1}(c_1), m_2 = D_{h_2}(c_2), \ldots$, kde $D_{h_i} = E_{h_1}^{-1}$.

**Popište princip digitálního podpisu.Popište digitální podpis pomocí algoritmu RSA.**

**Slidy 235.**

- Nechť subjekt 1 vysílá podepsanou zprávu $m$ subjektu 2.
- Subjekt 1 spočítá pro zprávu $m$ OT

$$S = D_{SK_1}(m) = |m^{d_1}|_{n_1},$$

kde $SK_1 = (d_1, n_1)$ je tajný dešifrovací klíč pro subjekt 1.
- Když $n_2 > n_1$, kde $VK_2 = (e_2, n_2)$ je veřejný šifrovací klíč pro subjekt 2, subjekt 1 zašifruje $S$ pomocí vztahu

$$c = E_{VK_2}(S) = |S^{e_2}|_{n_2}, \quad 0 < c < n_2 .$$

Když $n_2 < n_1$ subjekt 1 rozdělí $S$ do bloků o velikosti menší než $n_2$ a zašifruje každý blok s použitím šifrovací transformace $E_{VK_2}$.
Pro dešifrování subjekt 2 nejdříve použije soukromou dešifrovací transformaci $D_{SK_2}$ k získání $S$, protože

$$D_{SK_2}(c) = D_{SK_2}(E_{VK_2}(S)) = S\,.$$

K nalezení OT $m$ předpokládejme, že byl vyslán subjektem 1, subjekt 2 dále použije veřejnou šifrovací transformaci $E_{VK_1}$, protože

$$E_{VK_1}(S) = E_{VK_1}(D_{SK_1}(m)) = m.$$

Zde jsme použili identitu $E_{VK_1}(D_{SK_1}(m)) = m$, která plyne z faktu, že

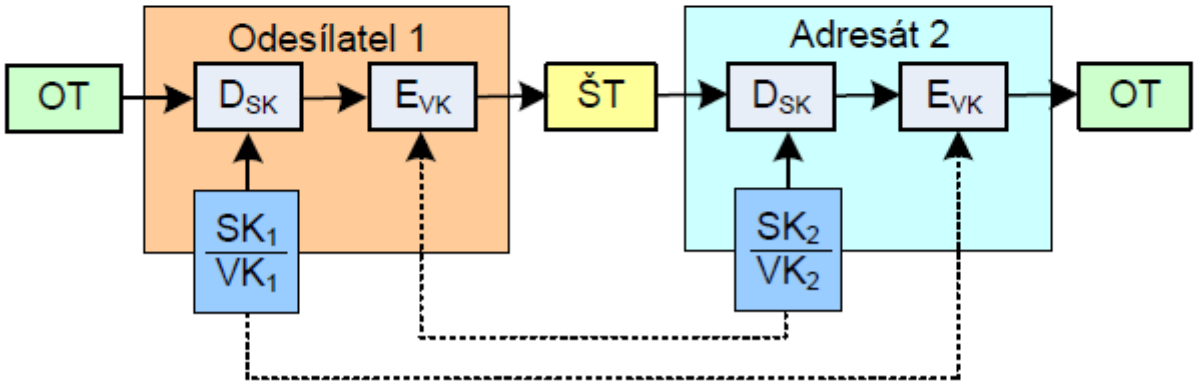
$$E_{VK_1}(D_{SK_1}(m)) = |(m^{d_1})^{e_1}|_{n_1} = |m^{d_1 e_1}|_{n_1} = m,$$

protože

$$|d_1 e_1|_{\Phi(n_1)} = 1.$$

Kombinace OT $m$ a podepsané verze $S$ přesvědčí subjekt 2, že zpráva byla vyslána subjektem 1.
Také subjekt 1 nemůže odepřít, že on vyslal danou zprávu, protože žádný jiný subjekt než 1 nemůže generovat podepsanou zprávu $S$ z originálního textu zprávy $m$.

## Digitální podpis



**Vysvětlení nebo nákres operačních modů ECB, CBC, CFB, OFB a Vysvětlení "solení". + MACx`**