

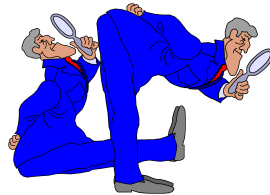
6. Asymetrické šifrovací systémy neboli systémy s veřejným klíčem

Jan Paseka

Ústav matematiky a statistiky
Masarykova univerzita

23. listopadu 2023

O čem to bude



1 Veřejný klíč
• Úvod

2 Asymetrické šifrovací systémy

3 Elektronický podpis

4 Idea funkce s vlastností padacích dveří

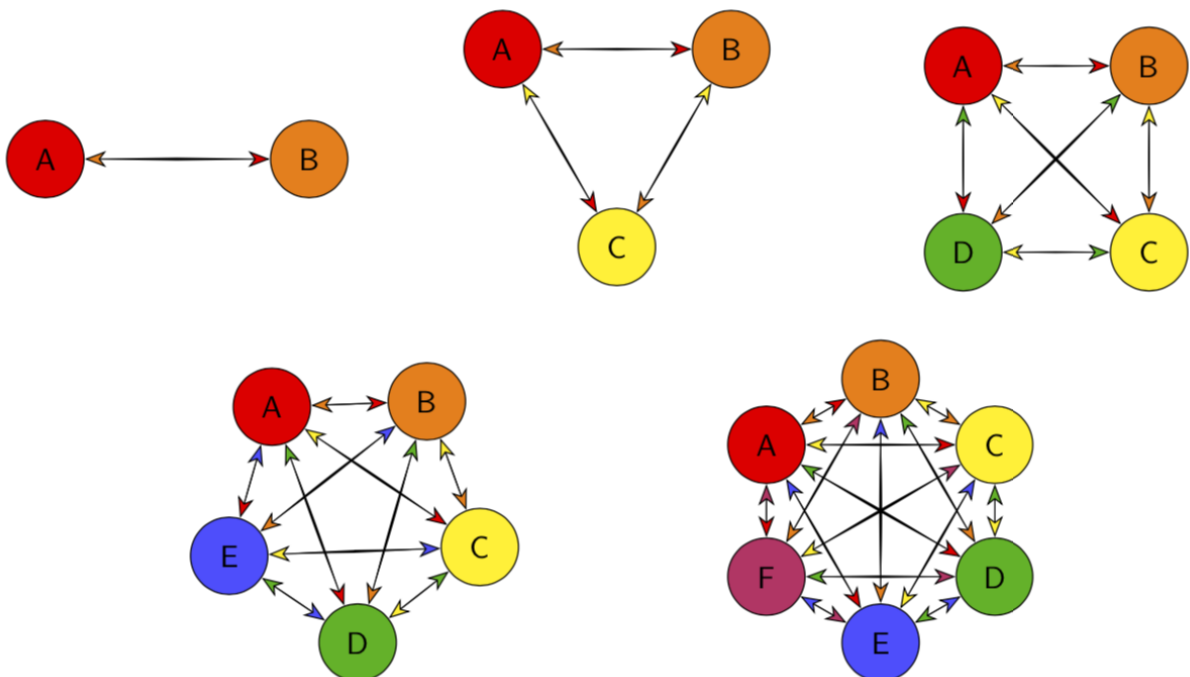
Úvod I

FI Doposud jsme pracovali se šifrovacími systémy následujících vlastností:

- 1 Kdo může zašifrovat zprávu, může ji i dešifrovat.
- 2 Každá dvojice partnerů musí mít svůj společný tajný klíč.

Druhá vlastnost je nepochybně nevýhodná. Pokud by počítačová síť měla n navzájem propojených účastníků, museli by používat $\frac{n \cdot (n+1)}{2}$ různých šifrovacích klíčů, které by si účastníci museli mezi sebou vyměnit.

Zvolíme-li např. $n = 500$, bylo by nutno mít v systému cca 125000 klíčů. Vzhledem k nutné obnově za nové klíče bychom se dostali do prakticky nerealizovatelné situace.



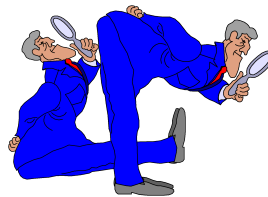
Úvod III

O šifrovacím systému s první vlastností pak obvykle mluvíme jako o **symetrickém** šifrovacím systému. První vlastnost můžeme považovat za výhodnou, protože lze k šifrování a dešifrování použít stejný stroj.

Asymetrické algoritmy se vyznačují tím, že jsou od první vlastnosti co nejvíce možná vzdáleny. Ukážeme, že takovéto systémy nemají druhou vlastnost a tedy práce s klíči je jednoduchá.

Budeme hlavně používat pojem **asymetrického algoritmu**; občas budeme mluvit o **public-key algoritmu**. Takovéto postupy byly vyvinuty v roce 1976 Whitfieldem **Diffiem** a Martinem **Hellmanem** v jejich práci *New Directions in Cryptography*, za kterou tito američtí matematici obdrželi v témže roce výroční cenu M.I.T.

O čem to bude



1 Veřejný klíč

2 Asymetrické šifrovací systémy

• Úvod

3 Elektronický podpis

4 Idea funkce s vlastností padacích dveří

Asymetrické šifrovací systémy I

Budeme předpokládat, že každý účastník **T** má **dvoji** klíčů, a to

- **veřejný klíč** $E = E_T$ k zašifrování;
- **soukromý (tajný) klíč** $D = D_T$ k dešifrování;

které se vyznačují následující vlastností: **Ze znalosti klíče E_T nelze zjistit soukromý klíč D_T .**

Kryptosystém s touto vlastností se nazývá **asymetrický kryptosystém**.

Pokud navíc předpokládáme, že pro každou zprávu M platí

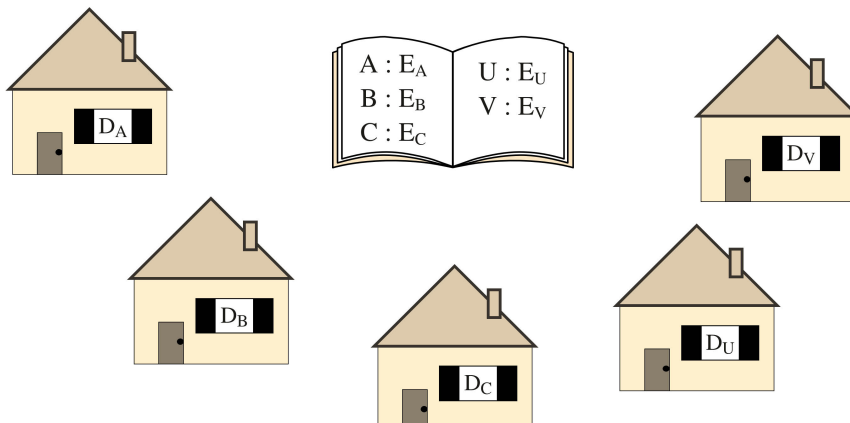
$$D(E(M)) = M,$$

mluvíme o **asymetrickém šifrovacím systému**.

Asymetrické šifrovací systémy II

Asymetrický kryptosystém se nazývá **asymetrické podpisovací schéma**, pokud pro každou zprávu M lze pomocí veřejného klíče **E** prověřit, zda se k sobě M a $D(M)$ hodí.

Všechny veřejné klíče jsou uloženy ve veřejně dostupném souboru (podobnému telefonnímu seznamu), zatímco soukromé klíče jsou tajné tj. známé pouze jejich vlastníkům.



Asymetrické šifrovací systémy III

Šifrování a dešifrování pomocí asymetrického šifrovacího systému probíhá ve 3 krocích:

- 1 Chce-li **A** zaslat **B** zprávu M , pak
 - najde veřejný klíč E_B pro **B**,
 - zašifruje zprávu M pomocí klíče E_B a
 - odešle $E_B(M)$ k **B**.
- 2 **B** může kryptogram $E_B(M)$ dešifrovat, protože zná jako jediný tajný klíč D_B :

$$D_B(E_B(M)) = M.$$

- 3 Žádný jiný účastník nemůže $E_B(M)$ rozluštit, protože podle předpokladu ze znalosti E_B a $E_B(M)$ nelze získat znalost o D_B .

Asymetrické šifrovací systémy - Výhody IV

- **Není potřeba výměna klíčů.** Tímto je vyřešen hlavní problém symetrického algoritmu. Zejména je tedy s pomocí asymetrického algoritmu možná **bezprostřední komunikace**. Můžeme tedy navzájem komunikovat, aniž bychom se složitě dohadovali o klíči. Asymetrické algoritmy jsou ideální pro **otevřenou komunikaci**.
- **Není potřeba mnoho klíčů.** U symetrického algoritmu se zvyšuje počet klíčů **kvadraticky** s počtem uživatelů, u asymetrického algoritmu je počet klíčů roven dvojnásobku počtu uživatelů.
- **Lze přijmout bez problémů nové uživatele.** Je-li přijat nový účastník do symetrického systému, musí si s ním všichni původní účastníci vyměnit klíč. U asymetrického systému není naproti tomu nutno, aby původní účastníci aktualizovali svoje data.

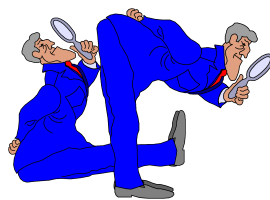
Asymetrické šifrovací systémy - Výhody V

- **Mnoho asymetrických systémů poskytuje skvělé možnosti pro elektronický podpis.**

Nevýhody asymetrického šifrovacího systému jsou pak:

- **Doposud není znám žádný asymetrický kryptosystém, který by byl zároveň rychlý a bezpečný.** Postup, kterým se budeme hlavně zabývat, je tzv. RSA-algoritmus. V poslední době se také pracuje s algoritmy, které spočívají na "**diskrétních logaritmech**".
- **Asymetrické algoritmy potřebují jistý dohled nad klíči.** Může se totiž stát, že Mr. X opatří svoji schránku s falešným jménem, ke kterému se však hodí jeho klíč. Pak může zachytit všechny zprávy, které byly adresovány původnímu adresátovi.

O čem to bude



- 1 Veřejný klíč
- 2 Asymetrické šifrovací systémy
- 3 Elektronický podpis
 - Diffie-Lamportovo schéma
 - Rabinovo pravděpodobnostní podpisovací schéma
 - Asymetrické šifrování
- 4 Idea funkce s vlastností padacích dveří

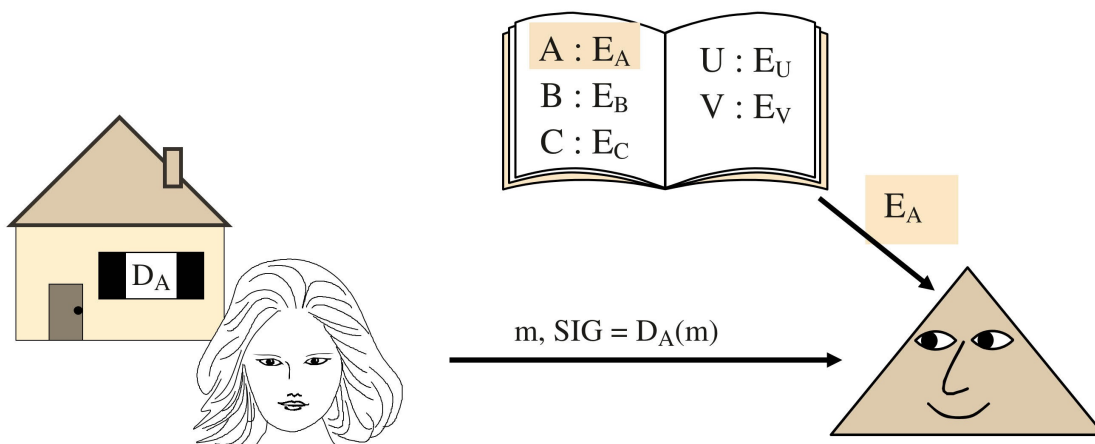
Elektronický podpis I

Další důležitou myšlenkou **Diffieho** a **Hellmana** je **elektronický** neboli **digitální podpis**. Nejdříve si ujasněme vlastnosti obvyklého podpisu rukou.

Předpokládejme, že osoba A se podepsala rukou na nějaký dokument D. Pak má tento podpis v ideálním případě následující vlastnosti:

- **Pouze osoba A** může vytvořit tento podpis.
- **Každý jiný účastník** může prověřit, že se opravdu jedná o podpis osoby A.

Elektronický podpis II



Elektronický podpis III

Diskutujme nejprve digitální podpis s použitím symetrického šifrovacího systému (např. DES). Popíšeme dva možné přístupy.

Diffie-Lampportovo schéma

Odesílatel A, který si přeje podepsat n -bitovou binární zprávu

$$M = M_1 \dots M_n,$$

si předem vybere $2n$ klíčů šifrovacího systému $\langle \mathbf{M}, \mathbf{K}, \mathbf{C} \rangle$. Označme je po řadě jako

$$a_1, \dots, a_n; \quad b_1, \dots, b_n.$$

Tyto klíče jsou **tajné**.

Elektronický podpis IV

Je-li šifrovací algoritmus e , osoba A vygeneruje $4n$ **parametrů** $\{(X_i, Y_i, U_i, V_i) : 1 \leq i \leq n\}$, kde X_i, Y_i leží v definičním oboru e a

$$U_i = e(X_i, a_i) \quad \text{a} \quad V_i = e(Y_i, b_i) \quad (1 \leq i \leq n). \quad (3.1)$$

Tyto parametry jsou dopředu zaslány příjemci a zároveň jsou odeslány nezávislému prověřovateli (veřejný registr).

Nyní předpokládejme, že osoba A chce odeslat podepsanou n -bitovou zprávu $M = M_1 \dots M_n$. Bude postupovat podle následující procedury. Jejím podpisem bude řetězec

$$S = S_1 \dots S_n,$$

kde pro všechna $i, 1 \leq i \leq n$ platí

$$S_i = \begin{cases} a_i & \text{pokud } M_i = 0, \\ b_i & \text{pokud } M_i = 1. \end{cases}$$

Elektronický podpis IV

Ověřovací protokol osoby B probíhá následovně: pro všechna i ($1 \leq i \leq n$) použije osoba B bit M_i a klíč S_i , aby ověřila, že

$$\begin{cases} \text{pokud } M_i = 0 & \text{pak } e(X_i, S_i) = U_i, \\ \text{pokud } M_i = 1 & \text{pak } e(Y_i, S_i) = V_i. \end{cases}$$

Osoba B pak akceptuje podepsanou zprávu **pouze za předpokladu**, že ověřovací protokol **je splněn pro všechna i** .

Ačkoliv tento systém je jednoduchý pro použití a snadno pochopitelný, má minimálně dvě nevýhody. První je nutná předběžná komunikace s parametry. Důležitější je však zvýšení rozměru zprávy – např. v případě DESu, kdy klíče mají délku 64 bitů, by se zpráva zvětšila 64-krát.

Rabinovo pravděpodobnostní podpisovací schéma I

Rabin (1978) navrhl jiný přístup. Buď e šifrovací funkce nějakého šifrovacího systému $\langle \mathbf{M}, \mathbf{K}, \mathbf{C} \rangle$.

Buď dále $(K_i : 1 \leq i \leq 2r)$ posloupnost náhodně vybraných klíčů, které odesílatel A uchová v tajnosti. Příjemce B obdrží seznam $2r$ parametrů (X_i, U_i) , ($1 \leq i \leq 2r$), kde

$$e(X_i, K_i) = U_i \quad (1 \leq i \leq 2r), \quad (3.2)$$

a tyto parametry jsou uloženy na nějakém veřejně přístupném místě.

Rabinovo pravděpodobnostní podpisovací schéma II

Předpokládejme nyní, že osoba A si přeje podepsat zprávu M . Jejím podpisem pak bude řetězec

$$S = S_1 S_2 \dots S_{2r},$$

kde pro všechna i , ($1 \leq i \leq 2r$) platí

$$S_i = e(M, K_i).$$

Osoba B pak pokračuje následovně: nejprve vybere náhodně či jinak r klíčů, které si přeje uveřejnit. Necht' to jsou klíče

$$K_{i_1}, K_{i_2}, \dots, K_{i_r}.$$

Pak po obdržení těchto klíčů osoba B prověří, že platí

$$e(M, K_{i_j}) = S_{i_j}, \quad e(X_{i_j}, K_{i_j}) = U_{i_j} \quad (1 \leq j \leq r).$$

Rabinovo pravděpodobnostní podpisovací schéma III

Osoba B akceptuje podpis jako podpis osoby A, pokud všechny tyto rovnosti platí. Je zřejmé, že bezpečnost příjemce závisí na jeho důvěře, že **jedině osoba vlastníci tajné klíče** mu mohla poslat tuto podepsanou zprávu.

Předpokládejme, že si osoba A chce podrobit kritice zprávu, o které se tvrdí, že ji podepsala a kterou B zkontroloval. Protokol A je jasný; musí vytvořit před kontrolorem svých $2r$ tajných klíčů

$$S = K_1, K_2 \dots K_{2r},$$

a veřejně prověřit rovnosti

$$e(M, K_i) = S_i, \quad e(X_i, K_i) = U_i \quad (1 \leq i \leq 2r).$$

Rabinovo pravděpodobnostní podpisovací schéma IV

Protokol Rabinova systému se řídí pravidlem, že kritika je oprávněná, pouze v případě, že všechna U_i a S_i s výjimkou nejvýše r kontrol S_i souhlasí. Uvažme, co může nastat:

- (a) **Platí méně než r kontrol S_i .** V tomto případě neměla osoba B akceptovat (M, S) jako správně podepsanou zprávu.
- (b) **Platí právě r kontrol S_i .** V tomto případě, když si osoba B vybrala r klíčů k zveřejnění, musela si vybrat právě tyto klíče. Pravděpodobnost výběru takové množiny je určena předpisem

$$p_r = \frac{1}{\binom{2r}{r}}$$

a $p_r \sim 10^{-10}$ pro $r = 18$.

- (c) **Platí $r + 1$ kontrol S_i či více.** V tomto případě je příjemce v právu.

Asymetrické šifrování a podpisy I

Vraťme se nyní k asymetrickému šifrování. Pak můžeme digitální podpis realizovat následujícím způsobem:

- 1 Chce-li osoba A **podepsat** zprávu M , tak
 - "zašifruje" M pomocí svého tajného soukromého klíče \mathbf{D}_A ,
 - uveřejní podepsanou zprávu $\mathbf{D}_A(M)$.
- 2 Každý jiný účastník může tento **podpis** $\mathbf{D}_A(M)$ zkontrolovat tím, že pomocí veřejného klíče \mathbf{E}_A prověří, zda se k sobě hodí M a $\mathbf{D}_A(M)$.

Např. u RSA algoritmu se prověří, zda platí

$$\mathbf{E}_A(\mathbf{D}_A(M)) = M.$$

Asymetrické šifrování a podpisy II

V takovém případě je někdy i jiná možnost kontroly digitálního podpisu: podepsaná osoba A zveřejní jen $D_A(M)$, ale ne zprávu M .

Pokud osoba B obdrží při aplikaci E_A smysluplnou zprávu, považuje to za důkaz správnosti digitálního podpisu.

Všimněme si, že při vytvoření a kontrole správnosti digitálního podpisu byly použity pouze klíče patřící odesílateli A.

Každý účastník může prověřit digitální podpis. Rozhodující rozdíl mezi digitálním a obyčejným podpisem je, že digitální podpis $D_A(M)$ je neoddělitelně spjat se zprávou M . Naproti tomu je obyčejný podpis připojen na konec zprávy.

Důsledkem je pak, že při změně $D_A(M)$ na $(D_A(M))'$ se změní i $E(D_A(M))$ a tedy zpráva M a $E(D_A(M))'$ spolu nesouhlasí.

Asymetrické šifrování a podpisy III

Předpokládejme, že máme šifrovací systém s veřejným klíčem s vlastností, že pro každého uživatele I šifrovací a dešifrovací funkce komutují tj. že

$$e_I(d_I(M)) = M. \quad (3.3)$$

Uvažme následující algoritmus.

- 1 Odesílatel A vypočte podpis S zprávy M užitím svého vlastního soukromého klíče a obdrží $S = d_A(M)$.
- 2 Užitím veřejného klíče příjemce B vypočte A kryptogram

$$C = e_B(S).$$

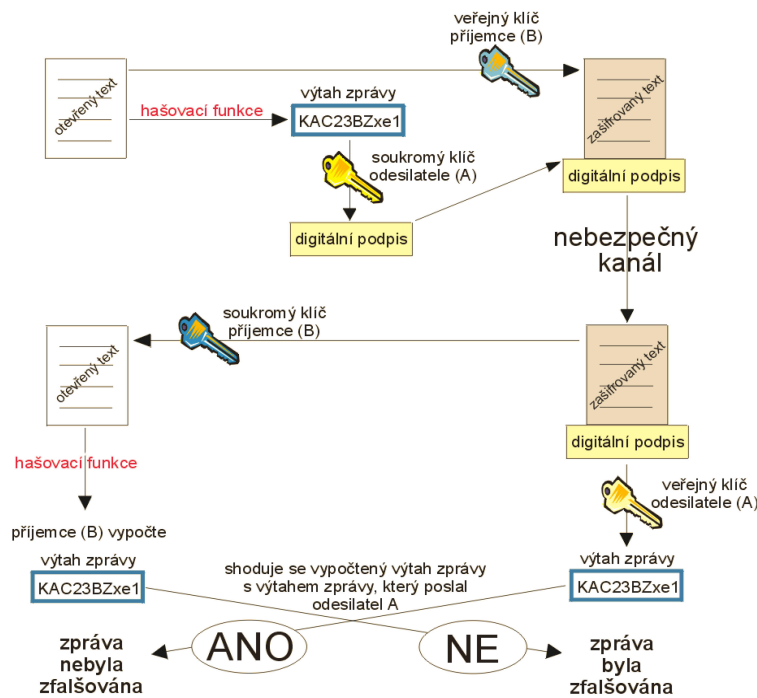
- 3 Příjemce B vypočte podpis S z kryptogramu C užitím svého vlastního soukromého klíče a obdrží

$$S = d_B(C).$$

- 4 Užitím veřejného klíče odesílatele A vypočte B zprávu

$$M = e_A(S).$$

Asymetrické šifrování a podpisy IV



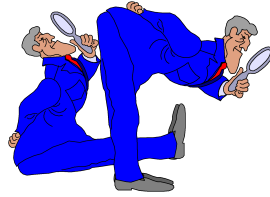
Asymetrické šifrování a podpisy V

Nyní je příjemce B ve velmi výhodné pozici. Vlastní dvojici (M, S) . V případě sporu, potřebuje-li přesvědčit soudce, že odesílatel A opravdu odeslal zprávu, požádá A o vytvoření jeho soukromého klíče K_A .

Odesílatel A musí opravdu vytvořit svůj soukromý klíč K_A , protože ho lze otestovat na identitě $e_A(d_A(M)) = M$. Soudci pak pouze stačí prověřit, že $S = d_A(M)$.

Ze stejného důvodu musí příjemce B zůstat poctivý. Předpokládejme, že změnil zprávu M na zprávu M' . Pak by ale musel být schopen změnit podpis S na S' , aby platilo, že $S' = d_A(M')$. Ale to může provést pouze A.

O čem to bude



- 1 Veřejný klíč
- 2 Asymetrické šifrovací systémy
- 3 Elektronický podpis
- 4 Idea funkce s vlastností padacích dveří
 - Shrnutí
 - Vlastnost padacích dveří

Shrnutí I

Zopakujme si vlastnosti systému s veřejným klíčem z odstavce 2 - Asymetrické šifrovací systémy.

Všichni uživatelé systému, kteří si přejí navzájem komunikovat, používají tentýž šifrovací algoritmus e a tentýž dešifrovací algoritmus d . Každý uživatel U_i má dvojici klíčů (K_i, L_i) tak, že pro každou možnou zprávu M platí identita

$$d(e(M, K_i), L_i) = M, \quad (4.1)$$

kde K_i je zveřejněn a uložen ve veřejném souboru; L_i zůstane utajen a mluvíme o něm jako o **soukromém klíči**; K_i se nazývá **veřejný klíč**.

Shrnutí II

Pokud chce jiný uživatel U_j odeslat uživateli U_i zprávu M , postupuje následovně.

- (a) Uživatel U_j najde veřejný klíč K_i uživatele U_i ve veřejném souboru.
- (b) Uživatel U_j odešle kryptogram $C = e(M, K_i)$

k uživateli U_i veřejným kanálem.

Bezpečnost systému závisí na funkcích e a d , které mají následující vlastnosti.

Vlastnost 1 Známe-li M a K , mělo by **být snadné vypočítat** $C = e(M, K)$.

Vlastnost 2 Je-li dán pouze kryptogram C , **není snadné výpočetně najít** M .

Vlastnost 3 Je-li znám kryptogram C a tajný klíč L_i , **je snadné určit zprávu** M .

Vlastnost padacích dveří I

Jinak řečeno, vlastnosti 1 a 2 tvrdí, že šifrovací funkce e používající veřejný klíč by měla být **jednosměrná**, ale vlastnost 3 požaduje navíc existenci "veřejného klíče". Takováto jednosměrná funkce se nazývá *funkce s vlastností padacích dveří*.

Aby byl takovýto systém prakticky použitelný, je nutné splnění následující podmínky.

Vlastnost 4 Mělo by být snadné generovat "náhodné" dvojice veřejný/soukromý klíč (K_i, L_i) .

Jinak řečeno, mělo by být dostatečně "mnoho" dvojic (K, L) tak, že je pro Mr. X nemožné sestavit tabulku vhodných funkčních hodnot.

Vlastnost padacích dveří II

$$(f, t) = \mathbf{Gen} (1^n)$$

$$f: D \rightarrow R$$

