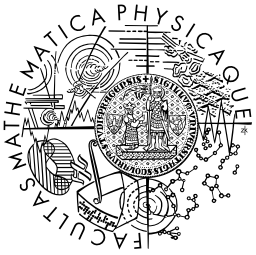

MATEMATICKO-FYZIKÁLNÍ FAKULTA
Univerzita Karlova

DIPLOMOVÁ PRÁCE

Romana Linkeová

Problém batohu a jeho aplikace

Katedra algebry

Vedoucí diplomové práce:	doc. Mgr. Pavel Příhoda, Ph.D.
Studijní program:	Matematika
Studijní obor:	Matematické metody informační bezpečnosti

Praha 2017

Prohlašuji, že jsem tuto diplomovou práci vypracovala samostatně a výhradně s použitím citovaných pramenů, literatury a dalších odborných zdrojů.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona v platném znění, zejména skutečnost, že Univerzita Karlova má právo na uzavření licenční smlouvy o užití této práce jako školního díla podle §60 odst. 1 autorského zákona.

V ..... dne ..... Podpis autora

V první řadě chci poděkovat svému vedoucímu diplomové práce doc. Mgr. Pavlu Příhodovi, Ph.D. za trpělivost při vedení práce, za cenné rady a za čas a energii, které vkládal do konzultací. Poděkování patří také mé rodině a nejbližším přátelům za podporu a motivaci.

Název práce: Problém batohu a jeho aplikace

Autor: Romana Linkeová

Katedra: Katedra algebry

Vedoucí diplomové práce: doc. Mgr. Pavel Příhoda, Ph.D., Katedra algebry

Abstrakt: V této práci se zabýváme kryptosystémy postavenými na $\mathcal{NP}$ (nedeterministický polynomiální) úplném problému batohu z mnoha aspektů. Z pohledu teorie složitosti podrobně uvedeme méně známé části důkazu $\mathcal{NP}$ úplnosti problému batohu. Z hlediska kryptografie ukážeme, že u Merkleova-Hellmanova kryptosystému, který vystihuje základní schéma kryptosystémů postavených na problému batohu, je pro velice nevhodně zvolené parametry tohoto kryptosystému možné odhalit celý soukromý klíč. Dalším přínosem práce je představení nového navrženého konceptu kryptosystému postaveném na problému maticového 0-1 batohu. Ač bylo toho schéma vytvořeno ve snaze předejít známým útokům, dokážeme analogií důkazu J. C. Lagariase a A. M. Odlyzka z roku 1985, že útok založený na LLL algoritmu bude úspěšný pro většinu kryptosystémů tohoto typu. Práci uzavírá souhrn moderních kryptosystémů postavených na problému batohu společně s jejich kryptoanalýzou.

Klíčová slova: problém batohu, $\mathcal{NP}$ úplné problémy, LLL algoritmus

Title: The knapsack and its applications

Author: Romana Linkeová

Department: Department of Algebra

Supervisor: doc. Mgr. Pavel Příhoda, Ph.D., Department of Algebra

Abstract: This thesis is focused on various aspects of cryptosystems based on $\mathcal{NP}$ (non-deterministic polynomial) complete knapsack problem. From the theory of complexity point of view, the less known parts of the proof of knapsack problem $\mathcal{NP}$ completeness are shown in detail. From the cryptographical point of view, a demonstration of breaking of the Merkle-Hellman cryptosystem (the basic design of knapsack-type cryptosystems) is provided, showing that poor parameters choice can lead to easy obtaining of the whole private key. Another contribution of this thesis consists in a presented proposal of a new cryptosystem concept based on the matrix 0-1 knapsack problem. This concept was developed in order to prevent known attacks, however, in the thesis we provide a proof analogous to J. C. Lagarias and A. M. Odlyzko, 1985, which shows that an attack based on the LLL algorithm will be successful on the majority of the matrix 0-1 knapsack problem cryptosystems. Finally, a list of modern cryptosystems based on the knapsack problem is provided and a cryptanalysis thereof is given.

Keywords: knapsack problem, $\mathcal{NP}$ complete problems, LLL algorithm

Obsah

Úvod

V průběhu let se asymetrická kryptografie stala nedílnou součástí našich životů. Stále větší množství lidí má přístup k internetu a využívá jej k finančním transakcím či k posílání citlivých údajů. Takové informace je důležité utajit pomocí šifrování.

Asymetrická kryptografie, nebo také kryptografie s veřejným klíčem, staví na předpokladu, že nějaký problém je obecně těžké vyřešit. Nejčastějším příkladem je problém faktorizace přirozených čísel nebo problém diskrétního logaritmu nad konečnými cyklickými multiplikativními grupami. Ač jsou tato schémata v dnešní době bezpečná, snahou kryptografie je mimo jiné vytvořit co nejširší škálu kryptografických aparátů pomocí problémů z různých odvětví. Dalším příkladem obecně těžkých problémů je třída $\mathcal{NP}$ (nedeterministický polynomiální) úplných problémů, o které zatím není známo, že by obsahovala i problém faktorizace celých čísel nebo problém diskrétního logaritmu. O problémech v této třídě se předpokládá, že jsou těžké, ale tento předpoklad nebyl doposud ani vyvrácen, ani dokázán. Schémata postavená na $\mathcal{NP}$ úplných problémech jsou z hlediska kryptografie slibná, jelikož mohou odolat útokům pomocí kvantového počítače

V této práci se zaměříme na různé aspekty kryptografických schémat postavených na $\mathcal{NP}$ úplném problému batohu. Problém batohu zjednodušeně řeší otázku, jak do batohu o daném objemu naskládat věci z dané množiny, aby bylo dosaženo přesně objemu batohu. Od roku 1978, kdy byl Ralphem Merklem a Martinem Hellmanem představen první kryptosystém tohoto typu [20], byla vytvořena celá řada schémat a drtivá většina z nich byla prolomena.

V první kapitole uvedeme základní pojmy z teorie algebry a složitosti, které budeme dále používat. V druhé kapitole podrobně dokážeme, že problém batohu je $\mathcal{NP}$ úplný. Důkaz $\mathcal{NP}$ úplnosti problému většinou probíhá tak, že je nalezena posloupnost redukcí $\mathcal{NP}$ úplných problémů, která vede až k dokazovanému problému. V této kapitole dokážeme pouze méně známé redukce z této posloupnosti, které se v literatuře většinou neuvádějí. Ve třetí kapitole popíšeme základní schéma kryptosystémů postavených na problému batohu pomocí Merkleova-Hellmanova kryptosystému z roku 1978 a podrobněji se zaměříme na schéma útoků, které využívají algoritmů pro výpočet redukované báze mřížky. Dále u Merkleova-Hellmanova kryptosystému ukážeme, že velice nevhodně zvolené parametry kryptosystému vedou k odhalení celého soukromého klíče a následnému dešifrování zprávy. V druhé části třetí kapitoly představíme nový navržený koncept kryptosystému postaveném na problému maticového 0-1 batohu. Ač byla snaha tento kryptosystém navrhnout tak, aby odolal známým útokům, v poslední části dokážeme analogií důkazu J. C. Lagariase a A. M. Odlyzka z roku 1985 [35], že útok založený na LLL algoritmu pro výpočet LLL redukované báze bude úspěšný pro většinu kryptosystémů tohoto typu. V poslední kapitole shrneme moderní kryptosystémy společně jejich kryptoanalýzou, které vznikaly v letech 2000-2017.

1. Základní teorie

V této kapitole zavedeme pojmy z teorie algebry a složitosti, se kterými budeme dále pracovat.

1.1 Algebraické definice

Poznámka (Značení).

- Skalární součin vektorů u a v značíme $u \cdot v$ nebo uv.
- Lineární obal vektorů $u_1, u_2, \ldots, u_n$ značíme $\langle u_1, u_2, \ldots, u_n \rangle$.

Věta 1 (Bertandův postulát)**.** *Pro každé přirození číslo $n \geq 2$ lze najít prvočíslo p takové, že $n < p < 2n$.*

Lemma 2. *Mějme vektor $u = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^n$. Pak pro* součtovou normu *vektoru*

$$\|u\|_1 = \sum_{i=1}^{n} |u_i|$$

a Euklidovu normu *vektoru*

$$\|u\|_2 = \sqrt{\sum_{i=1}^{n} u_i^2}$$

platí nerovnosti

$$\|u\|_2 \leq \|u\|_1 \leq \sqrt{n}\|u\|_2.$$

Pokud nebude řečeno jinak, pak $\|u\|$ značí Euklidovu normu vektoru u.

Definice 1 (Gram-Schmidtův ortogonalizační proces)**.** Gram-Schmidtův ortogonalizační proces *nalezne pro zadanou bázi $b_1, b_2, \ldots, b_n$ prostoru $\mathbb{R}^n$ bázi $\check{b}_1, \check{b}_2, \ldots, \check{b}_n$ takovou, že*

- $\check{b}_i \cdot \check{b}_j = 0 \quad$ *pro všechna $1 \leq i < j \leq n$,*
- $\check{b}_i = b_i - x_i, \quad$ *kde $x_i \in \langle b_1, b_2, \ldots, b_{i-1} \rangle$ pro všechna $1 \leq i \leq n$.*

Vektory $\check{b}_1, \check{b}_2, \ldots, \check{b}_n$ spočítáme následovně

$$\begin{aligned} \check{b}_1 &= b_1, \\ \mu_{i,j} &= \frac{b_i \check{b}_j}{\|\check{b}_j\|^2} \quad \text{pro všechna } 1 \leq j < i \leq n, \\ \check{b}_i &= b_i - \sum_{j=1}^{i-1} \mu_{i,j} \check{b}_j \quad \text{pro všechna } 1 < i \leq n. \end{aligned}$$

Definice 2 (Mřížka, báze mřížky)**.** *Podmnožina $\mathcal{L} \subseteq \mathbb{R}^n$ se nazývá* mřížka*, pokud existuje $m \leq n$ lineárně nezávislých vektorů $b_1, b_2, \ldots, b_m \in \mathbb{R}^n$ takových, že*

$$\mathcal{L} = \sum_{i=1}^{m} \mathbb{Z} b_i = \left\{ \sum_{i=1}^{m} x_i b_i,\ x_i \in \mathbb{Z} \right\}.$$

Vektory $b_1, b_2, \ldots, b_m$ *nazýváme* bází mřížky. *Řekneme, že mřížka je* celočíselná, *pokud* $\mathcal{L} \subseteq \mathbb{Z}^n$. *Mřížka* $\mathcal{L}$ *má* dimenzi n *a* hodnost m. *Pokud* $m = n$, *pak hovoříme o mřížkách s* plnou hodností.

Poznámka. V dalším budeme pracovat s celočíselnými mřížkami.

S mřížkami se pojí několik optimalizačních problémů. Nejznámější z nich je problém nalezení nejkratšího nenulového vektoru v mřížce $\mathcal{L}$, tzv. SVP problém (z anglického *shortest vector problem*). Pro tento problém neznáme žádný algoritmus, který by jej obecně řešil v polynomiálním čase. V roce 1982 v [39] představili Arjen Lenstra, Hendrik Lenstra a László Lovász pojem *LLL redukované báze* mřížky a LLL algoritmus, který umí LLL redukovanou bázi spočítat pomocí celočíselné aproximace Gram-Schmidtova ortogonalizačního procesu. Tento algoritmus vždy skončí v polynomiálním čase, ale ne vždy je v LLL redukované bázi obsažen i nejkratší nenulový vektor v mřížce $\mathcal{L}$. LLL redukovaná báze mřížky je však dostatečná aproximace nejkratší báze v mřížce pro většinu aplikací.

V následujícím uvedeme definici LLL redukované báze dle [35].

Definice 3 (Redukovaná báze mřížky)**.** *Nechť* $n \in \mathbb{N}$ *a* $m \in \mathbb{N}, m \leq n$. *Řekneme, že báze* $b_1, b_2, \ldots, b_m$ *mřížky* $\mathcal{L} \subseteq \mathbb{Z}^n$ *je* LLL redukovaná báze, *pokud pro bázi* $\check{b}_1, \check{b}_2, \ldots, \check{b}_m$ *získanou z Gram-Schmidtova ortogonalizačního procesu platí*

1. $|\mu_{i,j}| \leq \frac{1}{2}$ *pro všechna* $1 \leq j < i \leq m$,
2. $\|\check{b}_i\|^2 \geq \left(\frac{3}{4} - \mu_{i,i-1}^2\right) \|\check{b}_{i-1}\|^2$ *pro všechna* $1 < i \leq m$,

kde $\mu_{i,j}$, $1 \leq j < i \leq m$ *a* $\check{b}_1, \check{b}_2, \ldots, \check{b}_m$ *jsou jako v definici 1.*

Poznámka. V dalším budeme místo LLL redukovaná báze psát redukovaná báze.

Následující lemma je obdobou [39, Tvrzení 1.11] pro mřížky, které nemají plnou hodnost. Toto lemma ukazuje, jak dobrou aproximací nejkratší báze redukovaná báze je. Pro nejkratší vektor redukované báze platí, že tento vektor nebude více než (2^{m-1})-krát delší, než nejkratší nenulový vektor v mřížce $\mathcal{L}$.

Lemma 3. *Nechť* $b_1, b_2, \ldots, b_m$ *je redukovaná báze mřížky* $\mathcal{L} \subseteq \mathbb{Z}^n$. *Potom*

$$\|b_1\|^2 \leq 2^{m-1} \min_{v \in \mathcal{L}, v \neq 0} \|v\|^2.$$

1.2 Teorie grafů

Definice 4 (Neorientovaný graf)**.** Neorientovaný graf *je dvojice* $\boldsymbol{G} = (V, E)$, *kde* V *je neprázdná množina vrcholů a* $E \subseteq \{\{u,v\}, u,v \in V, u \neq v\}$ *je množina neorientovaných hran.*

Definice 5 (Obarvení grafu)**.** *Nechť* $\boldsymbol{G} = (V, E)$ *je neorientovaný graf,* $k \in \mathbb{N}$. *Zobrazení* $\chi : V \to \{1, 2, \ldots, k\}$ *je* k-obarvení grafu $\boldsymbol{G}$, *pokud pro každou hranu* $\{u,v\} \in E$ *platí, že* $\chi(u) \neq \chi(v)$.
Řekneme také, že graf $\boldsymbol{G}$ *obarvíme* k *barvami.*

Definice 6 (Sousední vrcholy)**.** *Nechť* $\boldsymbol{G} = (V, E)$ *je neorientovaný graf. Řekneme, že vrcholy* $u, v \in V$ *jsou sousední, pokud existuje hrana* $e \in E$ *taková, že* $e = \{u,v\}$. *Tato hrana* e vychází *z vrcholu* u *a z vrcholu* v.

1.3 Teorie složitosti

V teoretické informatice a matematice se setkáváme s mnoha problémy, u kterých chceme znát algoritmus k určení jejich obecného řešení společně s efektivitou daného algoritmu. Abychom mohli univerzálně hodnotit efektivitu, tedy časovou a paměťovou náročnost těchto algoritmů, bylo vytvořeno několik výpočetních modelů. Tyto výpočetní modely simulují dané algoritmy pomocí konečné sady instrukcí ohodnocených svou náročností. Nejznámější a také nejdůležitější výpočetní model je Turingův stroj, který v roce 1936 představil Alan Turing [59]. Neformálně si Turingův stroj můžeme představit jako teoretický model počítače sestávající z nekonečné pásky a čtecí a zároveň zapisovací hlavy. Páska, která představuje paměť Turingova stroje, je rozdělena do jednotlivých buněk, kde v každé buňce může být zapsán právě jeden symbol. Na začátku simulace algoritmu je na pásce zapsána konečná souvislá posloupnost symbolů, která představuje vstup, jinak jsou na celé pásce uloženy prázdné symboly. Čtecí a zároveň zapisovací hlava se vždy nachází nad právě jednou buňkou, ze které umí přečíst daný symbol. Na začátku simulace se hlava nachází nad první buňkou vstupu. Dále se akce hlavy řídí předem daným konečným seznamem pravidel, který jí umožňuje přepsat symbol v buňce jiným symbolem, případně se posunout hlavou vpravo nebo vlevo na pásce. Tato akce se považuje za jednu instrukci (operaci) Turingova stroje.

Formálně můžeme Turingův stroj popsat následovně.

Definice 7 (Deterministický a nedeterministický Turingův stroj)**.** Turingův stroj *je sedmice* $(Q, q_0, F, \Sigma, \epsilon, B, \delta)$*, kde*

- Q *je neprázdná konečná množina stavů,*
- $q_0 \in Q$ *je počáteční stav,*
- $F \subseteq Q$ *je množina přijímacích stavů,*
- Σ *je neprázdná konečná množina symbolů,*
- $\epsilon \in \Sigma$ *je prázdný znak,*
- $B = \Sigma \setminus \{\epsilon\}$ *je vstupní abeceda,*
- $\delta : (Q \setminus F) \times \Sigma \to Q \times \Sigma \times (\rightarrow, \leftarrow)$ *je částečná přechodová funkce, kde* $\rightarrow$ *značí posun hlavy vpravo a* $\leftarrow$ *značí posun hlavy vlevo.*

Přechodová funkce δ *se nazývá částečná, protože nemusí být definovaná pro všechny možné dvojice* (q,s)*, kde* $q \in Q \setminus F$*,* $s \in \Sigma$*.*

Turingův stroj se nazývá nedeterministický*, pokud přechodová funkce* δ *není jednoznačně určena pro všechny dvojice* (q,s)*, kde* $q \in Q \setminus F$*,* $s \in \Sigma$*. V opačném případě se jedná o* deterministický *Turingův stroj.*

Poznámka. Nedeterministický Turingův stroj můžeme chápat jako stroj, který v každém okamžiku, kdy přechodová funkce není jednoznačně definovaná, *uhádne* tu možnost, která povede k přijímacímu stavu.

K Turingovým strojům se váže důležitá *Churchova-Turingova teze*, která tvrdí, že každý algoritmus lze realizovat Turingovým strojem.

Pomocí modelu Turingova stroje můžeme popsat třídy složitosti. Každá třída složitosti specifikuje čas a daný výpočetní model (deterministický nebo nedeterministický Turingův stroj), které problémy v ní obsažené potřebují ke svému vykonání. Při řešení otázky efektivity daného algoritmu je důležité brát v potaz i paměťovou náročnost. Nás bude zajímat především časová složitost jako funkce velikosti vstupních dat a asymptotická časová složitost.

Jako *velikost vstupních dat* bereme počet bitů potřebných k zapsání vstupních dat do binárního zápisu.

Definice 8 (Časová složitost)**.** Časovou složitost *algoritmu A definujeme jako funkci $T : \mathbb{N} \to \mathbb{N}$, kde $T(n)$ je maximální počet operací, které provede Turingův stroj simulující algoritmus A na vstupu o velikosti n.*

Definice 9 (Landauova notace)**.** *Nechť $f : \mathbb{R} \to \mathbb{R}$ a $g : \mathbb{R} \to \mathbb{R}$ jsou funkce. Pak $f \in \mathcal{O}(g)$ právě tehdy, když*

$$\exists\, C \in \mathbb{R}, C > 0, \exists\, x_0 \in \mathbb{R} : \forall\, x > x_0 : |f(x)| \leq C|g(x)|.$$

Definice 10 (Asymptotická časová složitost)**.** *Řekneme, že algoritmus A má asymptotickou časovou složitost $\mathcal{O}(g)$, pokud pro jeho časovou složitost T platí*

$$T \in \mathcal{O}(g).$$

Poznámka (Příklady asymptotické složitosti)*.* Pro vstup velikosti n uvedeme základní typy asymptotické časové složitosti:

- $\mathcal{O}(1)$ konstantní,
- $\mathcal{O}(\log n)$ logaritmická,
- $\mathcal{O}(n)$ lineární,
- $\mathcal{O}(n^k)$ polynomiální,
- $\mathcal{O}(k^n)$ exponenciální.

Zřejmě platí posloupnost inkluzí

$$\mathcal{O}(1) \subset \mathcal{O}(\log n) \subset \mathcal{O}(n) \subset \mathcal{O}(n^k) \subset \mathcal{O}(k^n).$$

Pomocí asymptotické časové složitosti vyjadřujeme, jak se časová náročnost algoritmu mění v závislosti na změně velikosti vstupu.

Řekneme, že řešení problému je *efektivní*, pokud má polynomiální asymptotickou časovou složitost; analogicky můžeme říci, že algoritmus, který jej řeší, běží v polynomiálním čase.

Definice 11 (Rozhodovací problém)**.** Rozhodovací problém *je takový problém, na který lze odpovědět buď* ANO*, nebo* NE*. Rozhodovací problémy budeme zadávat pomocí* seznamu parametrů *a* otázky *týkající se těchto parametrů, na kterou je* odpověď *buď* ANO*, nebo* NE*.*

Poznámka. Jiné, než rozhodovací problémy budeme zadávat pomocí *seznamu parametrů* a *požadavků*, které klademe na hledané řešení daného problému.

Definice 12 (A-instance, N-instance)**.**

- A-instance *je vstup (přesné nastavení parametrů rozhodovacího problému), na který je odpověď na otázku rozhodovacího problému* ANO.
- N-instance *je vstup (přesné nastavení parametrů rozhodovacího problému), na který je odpověď na otázku rozhodovacího problému* NE.

U rozhodovacích problémů uvažujeme pro A-instance také *důkaz odpovědi*, tedy konkrétní řešení, které dokazuje správnost odpovědi ANO.

Definice 13 (Třída složitosti $\mathcal{P}$)**.** *Třída složitosti $\mathcal{P}$ obsahuje všechny rozhodovací problémy, které jsou řešitelné deterministickým Turingovým strojem v polynomiálním čase.*

Definice 14 (Třída složitosti $\mathcal{NP}$)**.** *Třída složitosti $\mathcal{NP}$ obsahuje všechny rozhodovací problémy, které jsou řešitelné nedeterministickým Turingovým strojem v polynomiálním čase.*

Poznámka. Zřejmě je $\mathcal{P} \subseteq \mathcal{NP}$. Zda je i $\mathcal{NP} \subseteq \mathcal{P}$, zůstává otevřeným problémem.

Poznámka. Řekneme-li, že rozhodovací problém π je řešitelný deterministickým Turingovým strojem v polynomiálním čase, pak to znamená, že pro všechny instance (pro všechna nastavení jeho parametrů) problému π umíme odpovědět ANO nebo NE v polynomiálním čase.

Naopak, řekneme-li, že rozhodovací problém π je řešitelný nedeterministickým Turingovým strojem v polynomiálním čase, pak to znamená, že pro každou A-instanci a potenciální důkaz odpovědi A-instance problému π umíme v polynomiálním čase ověřit, zda se jedná o správný důkaz odpovědi nebo ne.

Vidíme, že pojem řešitelnosti deterministickým a nedeterministických Turingovým strojem se liší. V prvním případě se jedná o nalezení odpovědi rozhodovacího problému v polynomiálním čase, v druhém případě hovoříme o ověřitelnosti správnosti daného důkazu odpovědi A-instance rozhodovacího problému v polynomiálním čase.

Definice 15 (Polynomiální redukce problému)**.** *Řekneme, že rozhodovací problém π lze* efektivně redukovat *na rozhodovací problém $\pi^\star$, pokud pro efektivní algoritmus $\mathcal{Q}$ platí, že*

$$\mathcal{Q}(X) \text{ je A-instance } \pi^\star \Leftrightarrow X \text{ je A-instance } \pi.$$

Píšeme $\pi \leq_p \pi^\star$, kde p v dolním indexu označuje redukci proveditelnou v polynomiálním čase.

Poznámka. Z definice polynomiální redukce plyne

- problém $\pi^\star$ je alespoň tak těžký jako problém π,
- algoritmus pro efektivní řešení problému $\pi^\star$ může být použit jako podrutina pro efektivní řešení problému π,

- relace $\leq_p$ je tranzitivní.

Definice 16 ($\mathcal{NP}$ těžké problémy)**.** *Rozhodovací problém π je $\mathcal{NP}$* těžký*, pokud pro každý problém $\pi^\star \in \mathcal{NP}$ platí $\pi^\star \leq_p \pi$.*

Definice 17 ($\mathcal{NP}$ úplné problémy)**.** *Rozhodovací problém π je $\mathcal{NP}$* úplný*, pokud $\pi \in \mathcal{NP}$ a π je $\mathcal{NP}$ těžký.*

Na $\mathcal{NP}$ úplné problémy se můžeme dívat jako na *nejtěžší* problémy ve složitostní třídě $\mathcal{NP}$. Jinými slovy, pro každý $\mathcal{NP}$ úplný problém platí, že pokud bychom znali algoritmus, který jej vyřeší v polynomiálním čase, pak umíme vyřešit *každý* problém ve třídě složitosti $\mathcal{NP}$ v polynomiálním čase. Dosud nebylo dokázáno, že neexistuje polynomiální algoritmus pro $\mathcal{NP}$ úplné problémy.

$\mathcal{NP}$ úplné problémy jsou velice důležité pro kryptografii, jelikož je možné pomocí nich konstruovat kryptografická schémata, která zatím neumíme obecně prolomit. Více viz kapitola 3.

Definice 18 (Literál, klauzule)**.** *Nechť $X = \{x_1, x_2, \ldots, x_m\}$ je množina logických proměnných.* Literál *je proměnná $x \in X$ nebo negace proměnné $\neg x$.*

Klauzule *je literál nebo disjunkce konečně mnoha literálů.*

Definice 19 (Booleovská formule)**.** *Uvažujme abecedu $\Sigma = X \cup \{\wedge, \vee, \neg\} \cup \{(,)\}$, kde $X = \{x_i,\, i \in \mathbb{N}\}$, x_i jsou logické proměnné. Množinu konečných neprázdných posloupností symbolů z množiny Σ značíme Σ^+. Prvek $\varphi \in \Sigma^+$ se nazývá* booleovská formule*, právě když je splněna jedna z následujících podmínek:*

- $\varphi \in X$,
- $\varphi = (\neg\psi)$, *kde ψ je booleovská formule,*
- $\varphi = (\psi_1 \vee \psi_2)$, *kde ψ_1 a ψ_2 jsou booleovské formule,*
- $\varphi = (\psi_1 \wedge \psi_2)$, *kde ψ_1 a ψ_2 jsou booleovské formule.*

Definice 20 (Ohodnocení booleovské formule)**.** *Označme $X \restriction_\varphi$ množinu proměnných, které se vyskytují ve φ. Potom libovolné zobrazení $h\colon X \restriction_\varphi \to \{0, 1\}$ nazveme* ohodnocením formule φ.

Definice 21 (Splnitelnost)**.** *Řekneme, že booleovská formule φ je ohodnocením h* splněna*, tedy $h(\varphi) = 1$, právě když platí jedna z následujících podmínek:*

- $\varphi = x$, *kde $x \in X$ a $h(x) = 1$,*
- $\varphi = \neg\psi$ *a ψ ohodnocením h splněna není,*
- $\varphi = (\psi_1 \vee \psi_2)$ *a alespoň jedna z formulí ψ_1, ψ_2 je ohodnocením h splněna,*
- $\varphi = (\psi_1 \wedge \psi_2)$ *a obě formule ψ_1, ψ_2 jsou ohodnocením h splněny.*

Řekneme, že formule je splnitelná*, pokud existuje ohodnocení, které ji splňuje.*

První problém, o kterém bylo dokázáno, že je $\mathcal{NP}$ úplný, je *Rozhodovací problém SAT.*

Rozhodovací problém SAT
Seznam parametrů: Booleovská formule $\varphi \in \Sigma^+$.
Otázka: Je booleovská formule φ splnitelná?

Věta 4 (Cookova věta)**.** *Rozhodovací problém SAT je $\mathcal{NP}$ úplný.*

Důkaz. [viz 18, str 39-44]

□

Rozhodovací problém 3SAT
Seznam parametrů: Booleovská formule $\varphi \in \Sigma^+$, kde φ je konjunkce klauzulí a každá klauzule sestává z právě tří literálů.
Otázka: Je booleovská formule φ splnitelná?

Věta 5. *Rozhodovací problém 3SAT je $\mathcal{NP}$ úplný.*

Důkaz. [viz 18, str 48-50]

□

2. Problém batohu

2.1 Úvod

Známá kombinatorická úloha zvaná *problém batohu* (angl. *knapsack problem*), byla studována od prvotní práce George B. Dantziga z roku 1957 [14]. Svůj název tento problém získal díky analogii s úlohou přesného zaplnění batohu, kdy máme dán batoh o objemu s a seznam předmětů $a_1, a_2, \ldots, a_n$, kde každý předmět a_i má svůj daný objem s_{a_i}. Úloha pak řeší otázku, jaké předměty $a_1, a_2, \ldots, a_n$ je potřeba vybrat, abychom dosáhli právě objemu s?

Problém batohu byl v průběhu let intenzivně studován zejména díky aplikacím v průmyslu a finančnictví ale také z teoretického hlediska, jelikož je tento problém úzce svázán s optimalizační $\mathcal{NP}$ těžkou úlohou celočíselného programování [46]. Za léta výzkumu vznikla řada variant, například *problém 0-1 batohu*, *problém kvadratického batohu*, *problém mnohonásobného batohu* a další. Definice většiny druhů problému batohu je možno nalézt v [29]. Všechny tyto varianty jsou $\mathcal{NP}$ těžké, tedy je velice nepravděpodobné, že by byl v budoucnu objeven algoritmus, který by úlohu obecně řešil v polynomiálním čase. Na druhou stranu $\mathcal{NP}$ těžkost problému vypovídá pouze o složitosti nalezení řešení v nejhorším případě (tzv. *worst-case complexity*), což nezaručuje rychlost nalezení řešení pro většinu problémů batohu, které se vyskytují v praxi (tzv. *avarage-case complexity*). Uvedené skutečnosti vyústily v nalezení efektivního algoritmu pro většinu praktických problémů batohu [47, 48].

V průběhu let bylo představeno mnoho technik, které řeší problém batohu, jako jsou například *Branch-and-bound* algoritmy, poprvé představené Peterem J. Kolesarem v roce 1967 [31]. Tyto algoritmy pracují se stromem, jehož větve reprezentují všechny možné kombinace prvků $a_1, a_2, \ldots, a_n$, a pomocí mezí a odhadů jsou odřezávány ty větve, které nemohou vést k lepšímu řešení. Jedním typem Branch-and-bound algoritmů jsou algoritmy, které řeší tzv. *problém jádra* (z angl. *core problem*), tedy problém batohu pro podmnožinu prvků $a_1, a_2, \ldots, a_n$, pro které je velká pravděpodobnost nalezení optimálního řešení. Zatím nejlepší výsledky má algoritmus MT2 [40]. Pomocí technik *dynamického programování* můžeme dosáhnout pseudopolynomiální asymptotické složitosti $\mathcal{O}(cn)$, kde $c \in \mathbb{N}$ a n je počet prvků $a_1, a_2, \ldots, a_n$. Vidíme, že můžeme dosáhnout polynomiální asymptotické časové složitosti vzhledem k počtu prvků, avšak asymptotická složitost zůstává exponenciální vzhledem k délce vstupu. Takovým problémům, pro které existuje algoritmus, který pracuje v pseudopolynomiální složitosti se říká *slabé $\mathcal{NP}$ těžké problémy*. Jedním z algoritmů založených na bázi dynamického programování je Bellmanova rekurze [4]. V [42] byl roku 1997 představen algoritmus *Combo*, který kombinuje Branch-and-bound algoritmy společně s dynamickým programováním. Detailnější popis a další postupy při řešení problému batohu je možno nalézt například v [41, 48].

V této kapitole nejprve uvedeme problém 0-1 batohu. Následně představíme lehké instance problémů 0-1 batohu postavených na rychle rostoucích posloupnostech a pomocí řetězce redukcí rozhodovacích problémů dokážeme, že rozhodovací problém 0-1 batohu je $\mathcal{NP}$ úplný.

2.2 Problém 0-1 batohu

Základní variantou problému batohu, se kterou budeme pracovat po zbytek této kapitoly, je problém 0-1 batohu.

Definice 22 (Problém 0-1 batohu)**.** *Nechť $n \in \mathbb{N}$ a nechť je dána n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslo $s \in \mathbb{N}$. Pak* problém 0-1 batohu *hledá vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že*

$$\sum_{i=1}^{n} x_i a_i = s.$$

I přesto, že je rozhodovací problém 0-1 batohu $\mathcal{NP}$ úplný (viz část 2.3), tedy obecně těžce řešitelný, existují instance, které jsou lehké. Tyto instance, kterým budeme říkat *lehké batohy*, mohou být generovány například z rychle rostoucích posloupností.

Definice 23 (Rychle rostoucí posloupnost)**.** *Nechť $n \in \mathbb{N}$. Řekneme, že posloupnost $(p_1, p_2, \ldots, p_n) \in \mathbb{N}^n$ je* rychle rostoucí*, pokud $p_1 \geq 1$ a*

$$p_k > \sum_{i=1}^{k-1} p_i, \quad \textit{pro všechna } 2 \leq k \leq n.$$

Poznámka. Rychle rostoucí posloupnosti jsou posloupnosti $(p_1, p_2, \ldots, p_n) \in \mathbb{N}^n$ takové, že

$$\begin{aligned} p_1 &= c_1, \\ p_k &= \sum_{i=1}^{k-1} p_i + c_k \quad \text{pro všechna } 2 \leq k \leq n, \end{aligned}$$

kde $c_i \in \mathbb{N}$ pro všechna $1 \leq k \leq n$.
Speciálním případem jsou rychle rostoucí posloupnosti $(p_1, p_2, \ldots, p_n)$ tvaru

$$p_k = c^k, \quad \text{kde } 1 \leq k \leq n \text{ a } c \in \mathbb{N} \setminus \{1\}.$$

My budeme v dalším textu uvažovat rychle rostoucí posloupnosti tvaru

$$p_k = p^k, \quad \text{kde } 1 \leq k \leq n \text{ a } p \text{ je prvočíslo.}$$

Příklad. Posloupnost $(1, 3, 5, 10, 20)$ je rychle rostoucí. Naopak posloupnost $(2, 4, 6, 13, 30)$ rychle rostoucí není, protože $6 \not> 4 + 2 = 6$.

Lemma 6. *Součin dvou rychle rostoucích posloupností je rychle rostoucí posloupnost.*

Důkaz. Nechť $n \in \mathbb{N}$. Mějme dvě rychle rostoucí posloupnosti $(u_1, u_2, \ldots u_n) \in \mathbb{N}^n$ a $(v_1, v_2, \ldots, v_n) \in \mathbb{N}^n$, tedy

$$\begin{aligned} u_1 \geq 1, \quad u_k > \sum_{i=1}^{k-1} u_i \quad &\text{pro všechna } 2 \leq k \leq n, \\ v_1 \geq 1, \quad v_k > \sum_{i=1}^{k-1} v_i \quad &\text{pro všechna } 2 \leq k \leq n. \end{aligned}$$

Pro jejich součin $(z_1, z_2, \ldots, z_n)$ definovaný jako

$$z_i = u_i v_i \quad \text{pro } 1 \leq i \leq n$$

zřejmě platí

$$z_1 = u_1 v_1 \geq 1.$$

Pro $2 \leq k \leq n$ máme

$$z_k = u_k v_k > \left(\sum_{i=1}^{k-1} u_i\right) \cdot \left(\sum_{i=1}^{k-1} v_i\right) > u_1 v_1 + \ldots + u_{k-1} v_{k-1} = \sum_{i=1}^{k-1} z_i,$$

tedy

$$z_k > \sum_{i=1}^{k-1} z_i$$

a posloupnost $z = (z_1, z_2, \ldots, z_n)$ je rychle rostoucí.

□

Následující postup ilustruje fakt, že problém 0-1 batohu, který je postavený na rychle rostoucí posloupnosti, je řešitelný v polynomiálním čase.

Postup řešení problému 0-1 batohu založeném na rychle rostoucí posloupnosti. Nechť $n \in \mathbb{N}$. Zvolme v problému 0-1 batohu n-tici $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ jako rychle rostoucí posloupnost. Tedy $a_1 \geq 1$ a

$$a_k > \sum_{i=1}^{k-1} a_i \quad \text{pro všechna } 2 \leq k \leq n.$$

Pro daný vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ spočítejme

$$s = \sum_{i=1}^{n} x_i a_i.$$

Vektor $(x_1, x_2, \ldots, x_n)$ pak najdeme následujícím postupem:

Algoritmus 1: Problém 0-1 batohu s rychle rostoucí posloupností

Vstup: rychle rostoucí posloupnost $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, $s \in \mathbb{N}$.
Výstup: vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$, nebo „řešení neexistuje“.

```
for i := n downto 1 do
    if s ≥ a_i then
        x_i = 1;
        s = s − a_i;
    else
        x_i = 0;
if s ≠ 0 then
    return „řešení neexistuje“;
else
    return (x_1, x_2, ..., x_n);
```

Velikost vstupních dat je pro 64bitová čísla $64(n+1)$. Algoritmus 1 má polynomiální časovou složitost $\mathcal{O}(n)$ a tedy je efektivní.

2.3 Rozhodovací problém batohu je $\mathcal{NP}$ úplný

V této části ukážeme, že rozhodovací problém 0-1 batohu je $\mathcal{NP}$ úplný. V následující kapitole uvidíme, že $\mathcal{NP}$ úplné problémy jsou důležité pro tvorbu kryptografických aparátů. K důkazu využijeme posloupnosti redukcí rozhodovacích problémů z [27]: SAT $\leq_p$ 3SAT $\leq_p$ rozhodovací problém k-obarvení grafu $\leq_p$ rozhodovací problém přesného pokrytí $\leq_p$ rozhodovací problém 0-1 batohu.

Všechny problémy nejprve definujeme a následně podrobně rozebereme méně známé redukce, a to redukce: rozhodovací problém k-obarvení grafu $\leq_p$ rozhodovací problém přesného pokrytí a rozhodovací problém přesného pokrytí $\leq_p$ rozhodovací problém 0-1 batohu. Redukce SAT $\leq_p$ 3SAT $\leq_p$ rozhodovací problém k-obarvení grafu jsou známější a jejich důkazy je možné nalézt v [18] a [27]. Dále ukážeme vztah mezi rozhodovacím problémem 0-1 batohu a rozhodovacím problémem dvou loupežníků a v závěru kapitoly uvedeme, jak řešit vyhledávací problém 0-1 batohu pomocí orákula rozhodovacího problému 0-1 batohu.

Rozhodovací problém 0-1 batohu
Seznam parametrů: n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, hodnota $s \in \mathbb{N}$.
Otázka: existuje vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$?

Vyhledávací problém 0-1 batohu
Seznam parametrů: n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, hodnota $s \in \mathbb{N}$.
Požadavky: najdi vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$.

Rozhodovací problém jednoznačnosti 0-1 batohu
Seznam parametrů: n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, hodnota $s \in \mathbb{N}$.
Otázka: existuje právě jeden vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$?

Rozhodovací problém k-obarvení grafu
Seznam parametrů: neorientovaný graf $\mathbf{G} = (\mathrm{V}, \mathrm{E})$, číslo $k \in \mathbb{N}$.
Otázka: je možné graf $\mathbf{G}$ obarvit k a méně barvami?

Rozhodovací problém přesného pokrytí
Seznam parametrů: množina $\{u_1, u_2, \ldots, u_n\}$ a systém podmnožin $\{S_j, j \in J\}$ množiny $\{u_1, u_2, \ldots, u_n\}$.
Otázka: existuje podmnožina $I \subseteq J$ taková, že pro každé $k, l \in I$, kde $k \neq l$ platí, že množiny S_k, S_l jsou disjunktní a

$$\bigcup_{i \in I} S_i = \{u_1, u_2 \ldots, u_n\}?$$

Rozhodovací problém dvou loupežníků
Seznam parametrů: n-tice $(c_1, c_2 \ldots, c_n) \in \mathbb{N}^n$.

Otázka: existuje podmnožina $\mathrm{I} \subseteq \{1, 2, \ldots, n\}$ taková, že

$$\sum_{i \in \mathrm{I}} c_i = \sum_{i \in \{1,2,\ldots,n\} \setminus \mathrm{I}} c_i?$$

Věta 7. *Rozhodovací problém přesného pokrytí je $\mathcal{NP}$ úplný.*

Důkaz. Nechť máme množinu $\{u_1, u_2, \ldots, u_n\}$ a systém podmnožin $\{S_j, j \in J\}$ množiny $\{u_1, u_2, \ldots, u_n\}$ jako zadání rozhodovacího problému přesného pokrytí.

Pro důkaz $\mathcal{NP}$ úplnosti musíme nejprve dokázat, že problém patří do třídy složitosti $\mathcal{NP}$ a dále, že pro každý problém $\pi \in \mathcal{NP}$ platí, že π můžeme na problém přesného pokrytí redukovat v polynomiálním čase.

Rozhodovací problém přesného pokrytí je zřejmě ve třídě složitosti $\mathcal{NP}$. Pro nedeterministický algoritmus, který tipne podmnožinu $I \subseteq J$ ověříme v polynomiálním čase správnost řešení a tedy, zda pro každé $k, l \in I$, kde $k \neq l$ jsou množiny S_k, S_l disjunktní a zda platí

$$\bigcup_{i \in I} S_i = \{u_1, u_2, \ldots, u_n\}.$$

Dále potřebujeme nalézt efektivní redukci $\mathcal{Q}$ nějakého $\mathcal{NP}$ úplného problému na rozhodovací problém přesného pokrytí. Pomocí řetězce redukcí (důkaz [viz 18, str 48-56])

$$\mathrm{SAT} \leq_p \mathrm{3SAT} \leq_p k\text{-obarvení grafu}$$

víme, že rozhodovací problém k-obarvení grafu je $\mathcal{NP}$ úplný. Nalezneme tedy efektivní redukci $\mathcal{Q}$

rozhodovací problém k-obarvení grafu $\leq_p$ rozhodovací problém přesného pokrytí.

Mějme neorientovaný graf $\mathbf{G}= (\mathrm{V}, \mathrm{E})$ a číslo $k \in \mathbb{N}$ zadané rozhodovacím problémem k-obarvení grafu. Parametry pro rozhodovací problém přesného pokrytí nadefinujeme následovně:

$$\begin{aligned} U &= \mathrm{V} \cup \{(e, b) : e \in \mathrm{E},\ 1 \leq b \leq k\}, \\ \{S_j, j \in J\} &= \{\{(e, b)\} : e \in \mathrm{E},\ 1 \leq b \leq k\} \cup \\ &\quad \cup \{\{v, (e_1, b), (e_2, b), \ldots, (e_{m_v}, b)\} : v \in \mathrm{V},\ 1 \leq b \leq k, \\ &\quad e_1, \ldots, e_{m_v} \in \mathrm{E} \text{ jsou všechny hrany vycházející z vrcholu } v\}, \end{aligned} \tag{2.1}$$

tedy systém podmnožin $\{S_j, j \in J\}$ obsahuje všechny jednoprvkové množiny $\{(e, b)\}$, kde $e \in \mathrm{E}$ je hrana a b, kde $1 \leq b \leq k$ je barva a dále každému vrcholu $v \in \mathrm{V}$ a každé barvě $b \in \{1, 2, \ldots, k\}$ přísluší jedna podmnožina, která obsahuje daný vrchol v a všechny dvojice (e, b) těch hran $e \in \mathrm{E}$, které vychází z vrcholu v a všech barev b, kde $1 \leq b \leq k$. Zřejmě je $\{S_j, j \in J\}$ systém podmnožin množiny U.

Pro redukci $\mathcal{Q}$ ověříme podmínku efektivní redukce z definice 15. Pokud X je A-instance rozhodovacího problému k-obarvení grafu, pak existuje obarvení $\chi : \mathrm{V} \to \{1, 2, \ldots, k\}$ grafu $\mathbf{G}$. Obarvení prvku $v \in \mathrm{V}$ označíme jako χ_v.

Systém podmnožin $\{S_i, i \in I\}$ množiny U určíme jako

$$\begin{aligned} \{S_i, i \in I\} = \{\{(e, b)\} : e = \{u, v\},\ u, v \in \mathrm{V}, u \neq v, 1 \leq b \leq k, b \neq \chi_u, b \neq \chi_v\} \cup \\ \cup \{\{v, (e_1, \chi_v), (e_2, \chi_v), \ldots, (e_{m_v}, \chi_v)\} : v \in \mathrm{V}, \\ e_1, \ldots, e_{m_v} \in \mathrm{E} \text{ jsou všechny hrany vycházející z vrcholu } v\}. \end{aligned}$$

Ověříme, že

$$\bigcup_{i \in I} S_i = U$$

a že pro každé $i, l \in I$, kde $i \neq l$ jsou množiny S_i, S_l disjunktní.

Zřejmě je $\bigcup_{i \in I} S_i \subseteq U$. Naopak pro množinu

$$U = \mathrm{V} \cup \{(e, b) : e \in \mathrm{E},\ 1 \leq b \leq k\}$$

platí, že $\mathrm{V} \subseteq \bigcup_{i \in I} S_i$. Dále pro každou hranu $e = \{u, v\}, u, v \in \mathrm{V}, u \neq v$ platí, že prvky $(e,\chi_u), (e, \chi_v)$ a prvky (e, b), kde $1 \leq b \leq k$ a $b \neq \chi_u, b \neq \chi_v$ patří do množiny $\bigcup_{i \in I} S_i$, tedy pro hranu e patří všechny prvky (e, b), kde $1 \leq b \leq k$ do množiny $\bigcup_{i \in I} S_i$ a $U \subseteq \bigcup_{i \in I} S_i$. Obě inkluze platí zároveň, tedy $\bigcup_{i \in I} S_i = U$.

Pro spor předpokládejme, že pro nějaké $i, l \in I$, kde $i \neq l$ existuje prvek s takový, že $s \in S_i$ a zároveň $s \in S_l$. Pokud $s \in \mathrm{V}$, pak by to znamenalo, že je vrchol s obarven dvěma barvami, a to je spor s definicí obarvení χ. Pokud $s = (e, b)$ pro nějakou hranu $e = \{u, v\} \in \mathrm{E}$ a barvu $b \in \{1, 2, \ldots, k\}$, pak by to znamenalo, že jsou dva sousední vrcholy u a v obarvené tou samou barvou b, a to je spor s definicí obarvení χ. Vidíme, že žádný takový prvek s nemůže existovat a množiny S_i, S_l jsou disjunktní pro všechna $i, l \in I, i \neq l$. Vidíme tedy, že $\mathcal{Q}(X)$ je A-instance rozhodovacího problému přesného pokrytí.

Naopak, pokud $\mathcal{Q}(X)$ je A-instance rozhodovacího problému přesného pokrytí, pak máme množinu U a systém podmnožin $\{S_j, j \in J\}$, které splňují vztahy (2.1) a systém podmnožin $\{S_i, i \in I\}, I \subseteq J$ takový, že pro $i, l \in I, i \neq l$ jsou množiny S_i, S_l disjunktní a

$$\bigcup_{i \in I} S_i = U.$$

Pro každé $i, l \in I$, kde $i \neq l$ platí, že množiny S_i, S_l jsou disjunktní, tedy pro každý vrchol $v \in \mathrm{V}$ existuje nejvýše jedna množina tvaru

$$\{v, (e_1, b_v), (e_2, b_v), \ldots, (e_{m_v}, b_v)\}, \tag{2.2}$$

kde $e_1, e_2, \ldots, e_{m_v} \in \mathrm{E}$ jsou všechny hrany vycházející z vrcholu v a b_v, kde $1 \leq b_v \leq k$ je barva.

Jelikož $\bigcup_{i \in I} S_i = U$, pak pro každý vrchol existuje právě jedna množina tvaru (2.2). Zobrazení χ nadefinujeme $\forall v \in \mathrm{V}$ jako $\chi(v) = b_v$. Ověříme, že toto zobrazení splňuje definici k-obarvení grafu $\mathbf{G}$. Je zřejmé, že zobrazení χ přiřadí každému vrcholu $v \in \mathrm{V}$ právě jednu barvu $b_v \in \{1, 2, \ldots, k\}$.

Pro hranu $e = \{u, v\}$, $u, v \in \mathrm{V}$, $u \neq v$ platí, že $\chi(u) \neq \chi(v)$. Kdyby platilo, že $\chi(u) = \chi(v)$, pak by prvek $(e, \chi_u) = (e, \chi_v)$ byl obsažen ve dvou množinách $S_i, S_l, i, l \in I, i \neq l$, a to je spor s disjunkcí těchto množin. Zobrazení χ je obarvením grafu $\mathbf{G}$.

V systému podmnožin $\bigcup_{i \in I} S_i$ jsou obsaženy všechny prvky (e, b), kde $e \in \mathrm{E}$ a $1 \leq b \leq k$, tedy graf $\mathbf{G}$ můžeme obarvit k nebo méně barvami a X je A-instance rozhodovacího problému k-obarvení grafu.

□

Věta 8. *Rozhodovací problém 0-1 batohu je $\mathcal{NP}$ úplný.*

Důkaz. Mějme rozhodovací problém 0-1 batohu (dále jen problém batohu) zadán n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslem $s \in \mathbb{N}$. Důkaz bude probíhat ve dvou krocích. V prvním kroku dokážeme, že rozhodovací problém batohu je ve třídě složitosti $\mathcal{NP}$. V druhém kroku nalezneme efektivní redukci $\mathcal{NP}$ úplného rozhodovacího problému přesného pokrytí na rozhodovací problém batohu.

Rozhodovací problém batohu je zřejmě ve třídě složitosti $\mathcal{NP}$. Pro nedeterministický algoritmus, který tipne řešení $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ ověříme v polynomiálním čase správnost řešení a tedy, zda platí

$$\sum_{i=1}^{n} x_i a_i = s.$$

Dále nalezneme efektivní redukci $\mathcal{Q}$

rozhodovací problém přesného pokrytí $\leq_p$ rozhodovací problém batohu.

Mějme množinu $\{u_1, u_2, \ldots, u_t\}$ a systém podmnožin $\{S_j, j \in J\}$ množiny $\{u_1, u_2, \ldots, u_t\}$ zadané problémem přesného pokrytí. Nadefinujeme

$$\varepsilon_{ji} = \begin{cases} 1 & \text{pro všechna } u_i \in S_j, \\ 0 & \text{pro všechna } u_i \notin S_j. \end{cases}$$

Označme

$$\begin{aligned} h &= |J|, \\ d &= h + 1. \end{aligned}$$

Rozhodovací problém batohu je pak zadán hodnotami

$$\begin{aligned} a_j &= \sum_{i=1}^{t} \varepsilon_{ji} d^{i-1} \quad \text{pro všechna } 1 \leq j \leq h, \\ s &= \frac{d^t - 1}{d - 1}. \end{aligned}$$

Pro redukci $\mathcal{Q}$ ověříme podmínku efektivní redukce z definice 15. Pokud X je A-instance rozhodovacího problému přesného pokrytí, pak existuje podmnožina $I \subseteq J$ taková, že pro každé $k, l \in I$, kde $k \neq l$ jsou množiny S_k, S_l disjunktní a

$$\bigcup_{i \in I} S_i = \{u_1, u_2, \ldots, u_t\}.$$

Definujeme vektor $x = (x_1, x_2, \ldots, x_h)$ následovně

$$x_i = \begin{cases} 1, & \text{pokud } i \in I, \\ 0, & \text{jinak.} \end{cases}$$

Potom pro redukci $\mathcal{Q}(X)$ je vektor x A-instancí rozhodovacího problému batohu, protože

$$\sum_{j=1}^{h} x_j a_j = \sum_{j=1}^{h} x_j \sum_{i=1}^{t} \varepsilon_{ji} d^{i-1} = \sum_{i=1}^{t} d^{i-1} = \frac{d^t - 1}{d - 1} = s.$$

Naopak, pokud $\mathcal{Q}(X)$ je A-instance rozhodovacího problému batohu, tedy máme $x = (x_1, x_2, \ldots, x_h)$ řešení problému batohu, pak podmnožinu $I \subseteq J$ určíme jako

$$I = \{j : x_j = 1\}.$$

Z vlastnosti

$$\sum_{i=1}^{h} x_i a_i = \frac{d^h - 1}{d - 1}$$

pak zřejmě

$$\bigcup_{i \in I} S_i = \{u_1, u_2, \ldots, u_t\}$$

a X je A-instancí rozhodovacího problému přesného pokrytí.

Je zřejmé, že redukci $\mathcal{Q}$ můžeme zkonstruovat v polynomiálním čase.

□

Věta 9. *Rozhodovací problém 0-1 batohu $\leq_p$ rozhodovací problém dvou loupežníků.*

Důkaz. Mějme rozhodovací problém 0-1 batohu (dále jen problém batohu) zadán n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslem $s \in \mathbb{N}$.

Nalezneme efektivní redukci $\mathcal{Q}$

rozhodovací problém batohu $\leq_p$ rozhodovací problém dvou loupežníků.

Položme

$$\begin{aligned} l &= n + 2, \\ c_i &= a_i \quad \text{pro všechna } 1 \leq i \leq n, \\ c_{n+1} &= s + C, \\ c_{n+2} &= \sum_{i=1}^{n} a_i + C - s, \end{aligned} \tag{2.3}$$

kde $C \in \mathbb{N}$ je konstanta taková, aby $c_{n+2} \geq 1$. Rozhodovací problém dvou loupežníků je pak zadán l-ticí $(c_1, c_2, \ldots, c_l) \in \mathbb{N}^l$.

Pro redukci $\mathcal{Q}$ ověříme podmínku efektivní redukce z definice 15. Pokud X je A-instance rozhodovacího problému batohu, tedy máme $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ řešení problému batohu, pak nadefinujeme množinu indexů

$$I = \{i : x_i = 1\} \cup \{n + 2\}.$$

Potom

$$\begin{aligned} \sum_{i \in I} c_i &= s + \sum_{i=1}^{n} a_i + C - s = \sum_{i=1}^{n} a_i + C, \\ \sum_{i \in \{1,2,\ldots,l\} \setminus I} c_i &= \sum_{i=1}^{n} a_i - s + s + C = \sum_{i=1}^{n} a_i + C \end{aligned}$$

a platí, že

$$\sum_{i \in I} c_i = \sum_{i \in \{1,2,\ldots,l\} \setminus I} c_i = \sum_{i=1}^{n} a_i + C,$$

tedy $\mathcal{Q}(X)$ je A-instancí rozhodovacího problému dvou loupežníků.

Naopak nechť $\mathcal{Q}(X)$ je A-instance rozhodovacího problému dvou loupežníků. Potom pro rozhodovací problém batohu zadaný n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslem $s \in \mathbb{N}$ máme l-tici $(c_1, c_2, \ldots, c_l) \in \mathbb{N}^l$, která splňuje vztahy (2.3), a pro kterou existuje podmnožina indexů $I \subseteq \{1, 2, \ldots, l\}$ taková, že

$$\sum_{i \in I} c_i = \sum_{i \in \{1,2,\ldots,l\} \setminus I} c_i = \sum_{i=1}^{n} a_i + C. \tag{2.4}$$

Pro daný rozhodovací problém batohu hledáme podmnožinu $J \subseteq \{1, 2, \ldots, n\}$ takovou, že

$$\sum_{i \in J} a_i = s.$$

Vidíme, že buď $n + 1 \in I$, nebo $n + 2 \in I$. Kdyby platilo, že $\{n + 1, n + 2\} \in I$, pak

$$\sum_{i \in I} c_i = \sum_{i \in \{1,2,\ldots,n\} \cap I} c_i + c_{n+1} + c_{n+2} \geq s + C + \sum_{i=1}^{n} a_i + C - s = \sum_{i=1}^{n} a_i + 2C$$

a to je spor s (2.4). Bez újmy na obecnosti tedy předpokládáme, že $n + 1 \in I$ a $n + 2 \notin I$.

Označme

$$I' = I \setminus \{n + 1\},$$

potom z rovnice (2.4) máme

$$\begin{aligned}
\sum_{i \in I} c_i &= \sum_{i \in \{1,2,\ldots,l\} \setminus I} c_i \\
\sum_{i \in I'} a_i + c_{n+1} &= \sum_{i \in \{1,2,\ldots,n\} \setminus I'} a_i + c_{n+2} \\
\sum_{i \in I'} a_i + s + C &= \sum_{i \in \{1,2,\ldots,n\} \setminus I'} a_i + \sum_{i \in \{1,2,\ldots,n\}} a_i - s + C \\
2s &= \sum_{i \in \{1,2,\ldots,n\} \setminus I'} a_i + \sum_{i \in \{1,2,\ldots,n\}} a_i - \sum_{i \in I'} a_i \\
2s &= 2 \sum_{i \in \{1,2,\ldots,n\} \setminus I'} a_i \\
s &= \sum_{i \in \{1,2,\ldots,n\} \setminus I'} a_i
\end{aligned}$$

a $J = \{1, 2, \ldots, n\} \setminus I'$ a X je A-instance rozhodovacího problému batohu.

Je zřejmé, že redukci $\mathcal{Q}$ můžeme zkonstruovat v polynomiálním čase.

□

Při dokazování složitosti problémů můžeme mít přístup k *orákulu*, nebo abstraktnímu stroji, který umí vyřešit nějaký rozhodovací problém v konstantním čase.

Lemma 10. *Vyhledávací problém 0-1 batohu umíme řešit pomocí orákula rozhodovacího problému 0-1 batohu.*

Důkaz. Mějme n-tici $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a $s \in \mathbb{N}$ zadání problému 0-1 batohu. Předpokládáme, že máme přístup k orákulu rozhodovacího problému 0-1 batohu, tedy funkci $\mathcal{F}$ takové, že

$$\mathcal{F}\Big(\{a_1,a_2,\ldots,a_n\}, s\Big) = \begin{cases} \text{ANO}, & \text{pokud existuje řešení problému 0-1 batohu,} \\ \text{NE}, & \text{pokud neexistuje řešení problému 0-1 batohu.} \end{cases}$$

Pomocí orákula $\mathcal{F}$ můžeme najít řešení $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ následujícím postupem.

Algoritmus 2: Vyhledávací problém 0-1 batohu pomocí orákula rozhodovacího problému 0-1 batohu

Vstup: n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n, s \in \mathbb{N}$, orákulum $\mathcal{F}$ rozhodovacího problému 0-1 batohu.
Výstup: vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$, nebo „řešení neexistuje“.

for $i := 1$ **to** n **do**
  if $\mathcal{F}\Big(\{a_i, a_{i+1} \ldots, a_n\} \setminus \{a_i\}, s - a_i\Big) = \text{ANO}$, **then**
    $x_i = 1$;
    $s = s - a_i$;
  else
    $x_i = 0$;
if $s \neq 0$ **then**
  return „řešení neexistuje“;
else
  return $(x_1, x_2, \ldots, x_n)$;

□

Poznámka. Vyhledávací problém batohu nemůže být $\mathcal{NP}$ úplný, jelikož se nejedná o rozhodovací problém. Analogicky se uvažuje o $\mathcal{NP}$ těžkosti a $\mathcal{NP}$ úplnosti jiných vyhledávacích a s nimi asociovaných rozhodovacích problémů. Například k známému vyhledávacímu problému faktorizace přirozených čísel, kdy pro zadané kladné číslo $N \in \mathbb{N}$ hledáme číslo $p \in \mathbb{N}$, které je vlastním dělitelem čísla N, je asociován rozhodovací problém, zda pro zadané číslo N a nějakou mez $k < N$ existuje číslo $p < k$, které je vlastním dělitelem N. Úvahy ohledně náležitosti do dané třídy složitosti se pak vážou k danému rozhodovacímu problému.

Poznámka. Pomocí rozhodovacího problému 0-1 batohu můžeme také zjistit, zda daný vyhledávací problém 0-1 batohu má jednoznačné řešení následujícím postupem.

Algoritmus 3: Jednoznačnost řešení vyhledávacího problému batohu pomocí rozhodovacího problému 0-1 batohu

Vstup: n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, $s \in \mathbb{N}$, orákulum $\mathcal{F}$ rozhodovacího problému 0-1 batohu
Výstup: vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že $\sum_{i=1}^{n} x_i a_i = s$ je jediné řešení nebo „NE“

$(x_1, x_2, \ldots, x_n) := (\overbrace{0, \ldots, 0}^{n})$;
for $i := 1$ **to** n **do**
    if $\mathcal{F}\Big(\{a_1, a_2, \ldots, a_n\} \backslash \{a_i\}, s\Big) = \text{NE}$, **then**
        $x_i = 1$;
if $\sum_{i=1}^{n} a_i x_i = s$ **then**
    return $(x_1, x_2, \ldots, x_n)$;
else
    return „NE“;

3. Kryptosystémy založené na problému batohu

V situaci, kdy Alice a Bob komunikují přes nezabezpečený kanál a chtějí, aby nikdo neoprávněný nemohl přečíst jejich zprávy, je potřeba, aby Alice i Bob posílané zprávy *šifrovali*. Podstata šifrování spočívá v tom, že je původní informace přetransformována tak, aby nikdo kromě Alice nebo Boba nebyl schopný posílanou zprávu odhalit. Naopak, pouze Alice nebo Bob mohou původní zprávu opět získat, nebo-li ji *dešifrovat* pomocí tzv. *kryptografického klíče*. Nezašifrované zprávě se říká *otevřený text* a zašifrované zprávě se říká *šifrový text*. Věda, která se zabývá metodami, jak danou informaci zašifrovat, dešifrovat a jak vyrobit kryptografické klíče se nazývá *kryptografie*.

Kryptografie se dělí na *symetrickou* a *asymetrickou* podle počtu a způsobu použití kryptografických klíčů. V případě symetrické kryptografie je použit ten samý klíč k šifrování i dešifrování. V případě asymetrické kryptografie má každá komunikující strana dva klíče - veřejný (značíme pk z anglického *public key*) a soukromý (značíme sk z anglického *secret key*). V situaci Alice a Boba bude Bob zprávy pro Alici šifrovat jejím veřejným klíčem a Alice následně zprávy dešifruje svým soukromým klíčem. Asymetrická kryptografie, nebo také kryptografie s veřejným klíčem, byla představen Martinem Hellmanem a Witfieldem Diffie roku 1976 v [15]. V této práci nás budou zajímat metody kryptografie asymetrické.

Asymetrická kryptografie je založena na principu *jednosměrných funkcí se zadními vrátky* pro které platí, že je jednoduché ze vstupu vypočítat výstup, ale naopak, z výstupu vypočítat vstup je obtížné, pokud nemáme přístup k další informaci tzv. *zadním vrátkům*. Nejčastěji se jako jednosměrná funkce se zadními vrátky použije nějaký problém, o kterém se domníváme, že je obtížné jej obecně vyřešit. Princip kryptosystémů postavených na těžkých problémech pak spočívá v tom, že by jakýkoliv útočník musel v případě, že nevlastní dešifrovací klíč (dešifrovací klíč zastává roli zadních vrátek), při odhalování zprávy vyřešit daný těžký problém. Nejznámější příklady takových konstrukcí mohou být například šifra RSA [50], která je postavena na problému faktorizace přirozených čísel nebo kryptosystém El Gamal [17], který pracuje na bázi problému diskrétního logaritmu.

Dalším příkladem těžkých problémů jsou $\mathcal{NP}$ úplné problémy, pro které zatím v nejhorších případech neznáme rychlejší algoritmy než ty s asymptotickou exponenciální složitostí (*worst-case complexity*). Nevýhodou $\mathcal{NP}$ úplných problémů je, že máme odhad pouze pro složitost v nejhorším případě, tedy je možné, že většina instancí z praxe je efektivně řešitelná. Toto je jeden z důvodů, proč $\mathcal{NP}$ úplné problémy nejsou nejvhodnějšími kandidáty pro tvorbu kryptografických aparátů, kde potřebujeme odhad asymptotické složitosti v průměrném případě (*average-case complexity*). Naopak výhodou $\mathcal{NP}$ úplných problémů je, že kryptografie na nich postavena může obstát útokům kvantovým počítačem, což neplatí například pro problém faktorizace přirozených čísel nebo pro problém diskrétního logaritmu [55].

V této práci se budeme zabývat kryptosystémy, které jsou postavené na problému batohu. O problému batohu z lemma 8 víme, že je $\mathcal{NP}$ úplný, a tedy

věříme, že je obtížné jej obecně vyřešit. Výhodou kryptosystémů postavených na problému batohu je především to, že šifrování i dešifrování je velice rychlé. Nevýhodou je jejich nízká bezpečnost. Ač jsou postavené na $\mathcal{NP}$ úplném problému, většina navrhovaných schémat byla prolomena.

V této kapitole nejprve popíšeme obecné schéma kryptosystému postaveném na problému batohu. Následně shrneme známé útoky na kryptosystémy tohoto typu a nastíníme útok založený na počítání redukované báze v mřížce, kterému se budeme podrobněji věnovat v závěru kapitoly. Poté popíšeme Merkleův-Hellmanův kryptosystém a ukážeme, že velice nevhodně zvolené parametry kryptosystému vedou k snadnému odhalení celého soukromého klíče a k následnému získání otevřeného textu. V poslední části se zaměříme na nový navržený koncept kryptosystému - kryptosystém založený na problému maticového 0-1 batohu. Ač byl tento kryptosystém vytvořen ve snaze předejít známým útokům, v závěru kapitoly dokážeme analogii důkazu z [35], že útok pomocí LLL algoritmu [39] pro výpočet redukované báze mřížky bude úspěšný pro téměř všechny kryptosystémy s hustotou problému maticového 0-1 batohu menší než 0,161.

3.1 Základní schéma kryptosystémů založených na problému batohu

Kryptosystémy založené na problému batohu využívají převážně toho, že existují lehké batohy postavené na rychle rostoucích posloupnostech, které jsou řešitelné v polynomiálním čase (viz algoritmus 1). Tyto lehké batohy pak tvoří soukromý klíč. Veřejný klíč je těžký batoh, který je ze soukromého klíče odvozen.

Základní postup využití problému batohu při tvorbě kryptosystémů vypadá následovně.

- Máme n-bitovou zprávu $m = (m_1, m_2, \ldots, m_n)$, kterou chceme šifrovaně poslat Alici.
- Alice najde rychle rostoucí posloupnost $(p_1, p_2, \ldots, p_n)$ lehkého batohu. Tato posloupnost tvoří její soukromý klíč sk.
- Alice transformuje $(p_1, p_2, \ldots, p_n)$ na posloupnost $(b_1, b_2, \ldots, b_n)$ těžkého batohu. Tato posloupnost tvoří její veřejný klíč pk, který Alice publikuje.
- Zašifrujeme zprávu m pomocí pk jako
$$\sum_{i=1}^{n} m_i b_i = c.$$
- Odešleme šifrový text c Alici.
- Alice transformuje šifrový text c pomocí soukromého klíče sk na
$$\sum_{i=1}^{n} m_i p_i = c'$$
a původní zprávu m najde jako řešení problému batohu založeném na rychle rostoucí posloupnosti algoritmem 1.

K problému batohu se váže důležitý pojem *hustoty* batohu, který přibližně vyjadřuje informační poměr mezi počtem bitů v otevřeném textu a průměrným počtem bitů šifrového textu. Dle [35] je hustota d problému 0-1 batohu definovaná jako

$$d = \frac{n}{\log_2(\max_i a_i)}.$$

3.2 Útoky na kryptosystémy založené na problému batohu

V případě, že obdržíme šifrový text c se nabízí možnost vyzkoušet všechny možné kombinace prvků $a_1, a_2, \ldots, a_n$, a tím najít otevřený text m. Těmto útokům se říká útoky hrubou silou a pro dostatečně velká vstupní data (počet bitů $n \geq 80$) nejsou tyto útoky proveditelné v reálném čase. To samé platí i pro útoky založené na space-time-trade-off, které původní asymptotickou složitost $\mathcal{O}(2^n)$ rozdělí na časovou složitost $\mathcal{O}\left(2^{\sqrt{n}}\right)$ a paměťovou složitost $\mathcal{O}\left(2^{\sqrt{n}}\right)$. V případě, že počet bitů zprávy bude $n \geq 160$, pak ani tento typ útoku nebude úspěšný.

Většina kryptosystémů založených a problému batohu byla prolomena útoky, které staví na tom, že transformace lehkého batohu na těžký batoh nedokáže lehký batoh skrýt natolik, aby jej nebylo možné odhalit. Merkleův-Hellmanův kryptosystém [20], což byl první kryptosystém postavený na problému batohu, prolomil v roce 1984 A. Shamir [53] Brand-and-bound algoritmem [10]. Obdoba tohoto algoritmu dále prolomila například i [63] a [30], což jsou moderní kryptosystémy představené v letech 2009 a 2010. Velká řada útoků využívá algoritmů pro výpočet redukované báze mřížky a řeší kryptosystémy postavené na problému batohu s malou hustotou [35, 6, 11] nebo využívá simultánních Diofantických aproximací [34, 22]. Detailnější popis útoků a průzkum je možno nalézt v [44, 8].

Útok založený na redukci báze mřížky. Nechť $n \in \mathbb{N}$. Pro problém 0-1 batohu zadaný n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a hodnotou $s \in \mathbb{N}$ probíhá útok založený na redukci báze mřížky podle [35] následovně.

Vektory

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, -\delta a_1)\,, \\ b_2 &= (0, 1, \ldots, 0, -\delta a_2)\,, \\ &\vdots \\ b_n &= (0, 0, \ldots, 1, -\delta a_n)\,, \\ b_{n+1} &= (0, 0, \ldots, 0, \delta s)\,, \end{aligned} \tag{3.1}$$

kde $\delta \in \mathbb{N}$, $\delta > \sqrt{n}$ je vhodně zvolená konstanta, tvoří bázi mřížky $\mathcal{L}$. Postup pro volbu konstanty δ je možno nalézt v [49]. Pomocí LLL algoritmu [39] spočítáme redukovanou bázi $b_1^*, b_2^*, \ldots, b_{n+1}^*$ mřížky $\mathcal{L}$. Pokud je vektor $(x_1, x_2, \ldots, x_n) \in \{0{,}1\}^n$ takový, že $\sum_{i=1}^n x_i a_i = s$ krátký vzhledem k Euklidově normě, pak jej s velkou pravděpodobností nalezneme jako nejkratší vektor v bázi $b_1^*, b_2^*, \ldots, b_{n+1}^*$.

Úspěch tohoto útoku se váže k hustotě problému batohu. V [35] autoři ukázali, že problémy 0-1 batohu s hustotu menší než 0,645 lze téměř vždy vyřešit pomocí této metody. Z [12] pak vidíme, že lze dosáhnout až meze 0,9408.

Místo LLL algoritmu můžeme použít i jiné definice redukované báze a jiné algoritmy, které redukované báze počítají [33].

3.3 Merkleův-Hellmanův kryptosystém

První kryptosystém založený na problému batohu je Merkleův-Hellmanův kryptosystém představený Martinem Hellmanem a Ralphem Merklem v roce 1978 v [20]. Soukromým klíčem je lehký batoh vytvořený pomocí rychle rostoucí posloupnosti a veřejným klíčem je těžký batoh, který je ze soukromého klíče odvozen pomocí modulárních operací. Výhodou tohoto kryptosystému, stejně jako u jiných kryptosystémů založených na problému batohu, je především jeho rychlost. Šifrování a dešifrování může probíhat až 100x rychleji než u šifry RSA (s modulem o velikosti přibližně 500 bitů) [44]. Naopak nevýhodou je nízká bezpečnost. Ukázalo se, že postup, při kterém se ze soukromého klíče vytvoří klíč veřejný, neskryje lehký batoh dostatečně. V roce 1982 představil A. Shamir útok na jednou iterovaný Merkleův-Hellmanův kryptosystém, který najde řešení v polynomiálním čase v [53]. Na více iterovanou variantu tohoto kryptosystému byl následně představen útok E.F. Brickellem v [7] a [13]. V případě, že útočník zná modulus N, představil A. Shamir a R. E. Zippel v [54] útok založený na řetězových zlomcích.

3.3.1 Popis Merkleova-Hellmanova kryptosystému

V této části popíšeme tři základní fáze - generování soukromého a veřejného klíče, zašifrování zprávy a dešifrování zprávy Merkleova-Hellmanova kryptosystému dle [20].

Mějme zprávu $m = (m_1, m_2, \ldots, m_n)$ o n bitech, kterou chceme zašifrovat.

Generování soukromého a veřejného klíče. Jako soukromý klíč sk zvolíme rychle rostoucí posloupnost $(p_1, p_2, \ldots, p_n) \in \mathbb{N}^n$, kde $p_1 \approx 2^n$ a $p_n \approx 2^{2n}$.

Zvolíme přirozené číslo N,

$$N > \sum_{i=1}^{n} p_i \tag{3.2}$$

a najdeme $e \in \mathbb{N}$ takové, že NSD(N,e) = 1. Spočítáme posloupnost $(b_1, b_2, \ldots, b_n)$ jako

$$b_i \equiv ep_i \mod N. \tag{3.3}$$

Potom soukromý a veřejný klíč je určen

$$\mathsf{sk} = (p_1, p_2, \ldots, p_n), N, e,$$
$$\mathsf{pk} = (b_1, b_2, \ldots, b_n).$$

Zašifrování zprávy. Zprávu $m = (m_1, m_2, \ldots, m_n)$ zašifrujeme veřejným klíčem pk jako

$$c = \sum_{i=1}^{n} b_i m_i.$$

Dešifrování zprávy. Z šifrového textu c získáme otevřený text následujícím postupem.

Nejprve najdeme $d \in \mathbb{N}$ takové, že $de \equiv 1 \bmod N$. Potom spočítáme

$$c' = dc = \sum_{i=1}^{n} d \cdot b_i m_i \equiv \sum_{i=1}^{n} de \cdot p_i m_i \mod N.$$

Dále

$$\sum_{i=1}^{n} de \cdot p_i m_i \equiv \sum_{i=1}^{n} p_i m_i \bmod N = \sum_{i=1}^{n} p_i m_i \tag{3.4}$$

z vlastnosti (3.2).

Vidíme, že v rovnici (3.4) je zpráva m řešením lehkého batohu, který umíme řešit v polynomiálním čase algoritmem 1.

3.3.2 Nevhodně zvolené parametry

V této části ukážeme, že je možné pro velice nevhodně zvolené parametry Merkleova-Hellmanova kryptosystému odhalit celý soukromý klíč sk a původní zprávu m. Klíčový pro tento postup je vztah (3.6).

Nechť p je prvočíslo. Předpokládejme, že jako soukromý klíč sk volíme rychle rostoucí posloupnost $(p_1, p_2, \ldots, p_n)$ takovou, že

$$p_i = p^i \quad \text{pro } 1 \leq i \leq n, \tag{3.5}$$

kde p je prvočíslo. Zbytek soukromého klíče N, e a veřejný klíč $\mathsf{pk} = (b_1, b_2, \ldots, b_n)$ vytvoříme stejným postupem jako v 3.3.1.

Takto zvolený soukromý klíč vede k tomu, že jednotlivé členy posloupnosti $(p_1, p_2, \ldots, p_n)$ jsou soudělné, konkrétně

$$\mathrm{NSD}(p_i, p_j) \neq 1 \quad \text{pro všechna } 1 \leq i < j \leq n.$$

Pokud pro $b_i, b_j, i > j$ platí

$$b_i = kb_j, \quad \text{kde } k \in \{2, 3, \ldots, N-1\}, \tag{3.6}$$

pak

$$ep_i \equiv k(ep_j) \mod N.$$

To nastane ve chvíli, kdy pro nějaké $i \in \{1, 2, \ldots, n-1\}$ platí, že $b_i < \frac{N}{p}$.

Dále pro NSD(e,N) = 1 máme

$$p_i \equiv kp_j \mod N, \quad \text{kde } k \in \{2, 3, \ldots, N-1\}. \tag{3.7}$$

Opačná implikace platit nemusí.

Pokud $p_i = kp_j$, pak $k = p^r$, pro $r \in \mathbb{N}$ a z (3.5) a (3.7) dále plyne, že $p_i = p^i$, pro $1 \leq i \leq n$.

Potom

$$b_i \equiv ep^i \mod N \quad \text{pro všechna } 1 \leq i \leq n$$

a z (3.2) dostáváme dolní odhad

$$N > \sum_{i=1}^{n} p^i.$$

Z každé kongruence

$$b_i p^j - b_j p^i \equiv 0 \mod N, \quad i \neq j \tag{3.8}$$

takové, že

$$b_i p^j - b_j p^i \neq 0 \tag{3.9}$$

získáme nějaké násobky N. Z největšího společného dělitele všech násobků obdržených ze vztahů (3.8) a (3.9) dostaneme hodnotu CN pro $C \in \mathbb{N}$. Konstanta C bude malá, tedy faktorizováním hodnoty CN odhalíme různé kandidáty N' modulu N. Pro tyto kandidáty N' spočítáme tajné číslo e ze vztahu

$$b_1 \equiv ep \mod N'$$

a ověříme platnost vztahu (3.2) a pro všechna $1 \leq i \leq n$ ověříme vztahy (3.3).

Původní zprávu m pak získáme pomocí algoritmu 1.

Příklad. Pomocí výše popsaného postupu nalezneme soukromý klíč a dešifrujeme zprávu z příkladu v [56, str 153]. Autoři na tomto příkladě ilustrují využití LLL algoritmu [39] k prolomení Merkleova-Hellmanova kryptosystému a jako ukázku uvádí kryptosystém s velice nevhodně zvolenými parametry.

Byl tedy vytvořen těžký batoh s veřejným klíčem pk:

$$\begin{array}{lll} b_1 = 1\,527\,086\,619\,781 & b_2 = 7\,635\,433\,098\,905 & b_3 = 14\,335\,307\,584\,368 \\ b_4 = 150\,964\,191\,369 & b_5 = 754\,820\,956\,845 & b_6 = 3\,774\,104\,784\,225 \\ b_7 = 18\,870\,523\,921\,125 & b_8 = 22\,827\,045\,875\,154 & b_9 = 18\,767\,797\,735\,142 \\ b_{10} = 22\,313\,414\,945\,239 & b_{11} = 16\,199\,643\,085\,567 & b_{12} = 9\,472\,641\,697\,364 \\ b_{13} = 23\,521\,350\,576\,663 & b_{14} = 22\,239\,321\,242\,687 & b_{15} = 15\,829\,174\,572\,807 \\ b_{16} = 7\,620\,299\,133\,564 & b_{17} = 14\,259\,637\,757\,663 & b_{18} = 23\,614\,472\,968\,001 \\ b_{19} = 22\,704\,933\,199\,377 & b_{20} = 18\,157\,234\,356\,257 & \end{array} \tag{3.10}$$

a šifrový text

$$c = 86\,175\,778\,454\,285.$$

Dále víme, že lehký batoh v soukromém klíči sk byl vytvořen jako

$$p_i = p^i \quad \text{pro všechna } 1 \leq i \leq 20.$$

Ačkoliv v [56] je přístup i k zprávě, my ukážeme, že je možné odhalit modulus N i soukromý klíč sk postupem nastíněným výše bez znalosti původní zprávy m.

Nejprve si všimneme, že $b_2 = kb_1$, pro $k \in \{2, 3, \ldots, N-1\}$. Konkrétně $b_2 = 5b_1$. Ze vztahu (3.7) vidíme, že pak $p_2 \equiv 5p_1 \mod N$, tedy zkusíme předpokládat, že

$$p_i = 5^i \quad \text{pro všechna } 1 \leq i \leq 20.$$

Pro veřejný klíč pk by pak platilo

$$b_i \equiv e \cdot 5^i \mod N \quad \text{pro všechna } 1 \leq i \leq 20.$$

Nyní najdeme soustavu dvou kongruencí typu (3.8), pro které platí vztah (3.9). V tomto příkladě se jedná o kongruence

$$\begin{aligned} b_1 &\equiv e \cdot 5^1 \mod N, \\ b_3 &\equiv e \cdot 5^3 \mod N, \end{aligned}$$

které splňují

$$b_1 \cdot 125 - b_3 \cdot 5 \neq 0,$$

konkrétně

$$b_1 \cdot 125 - b_3 \cdot 5 = 119\,209\,289\,550\,785 = 5 \cdot 7 \cdot 4111 \cdot 828503941.$$

Dostáváme kandidáta N' pro modulus N jako

$$N' = 7 \cdot 4111 \cdot 828503941 = 23\,841\,857\,910\,157.$$

Z vlastnosti (3.3) veřejného klíče pk pro N'

$$b_1 \equiv 5 \cdot e \mod N'$$

spočítáme

$$e = 9\,842\,160\,488\,019.$$

Pro tyto hodnoty soukromého klíče však neplatí vztah (3.2), tedy neplatí, že

$$23\,841\,857\,910\,157 = N > \sum_{i=1}^{20} 5^i = 119\,209\,289\,550\,780,$$

Pokud ale určíme prvky soukromého klíče $(p_1, p_2, \ldots, p_{20})$ jako $p_i = 5^{i-1}$, pro všechna $i \in \{1, 2, \ldots, 20\}$ a přepočítáme

$$e = 1\,527\,086\,619\,781,$$

pak je splněn vztah (3.2) a pro všechna $1 \leq i \leq 20$ jsou splněny všechny vztahy (3.3).

Nyní pomocí algoritmu 1 odhalíme zprávu m jako

$$(m_1, m_2, \ldots, m_{20}) = (1,0,0,0,1,1,0,0,0,1,1,0,0,0,1,1,0,0,0,1).$$

Vidíme, že s velice nevhodně zvolenými parametry kryptosystému je jednoduché odhalit soukromý klíč sk a získat původní zprávu m.

3.4 Kryptosystém postavený na problému maticového 0-1 batohu

V této části popíšeme konstrukci kryptosystému postaveném na problému maticového 0-1 batohu. Při tvorbě soukromého klíče vycházíme z rychle rostoucích posloupností, které uložíme jako část informace do dvou-dimenzionálních matic, které jsou dále transformovány pomocí invertibilních transformací a modulárních operací. Výsledkem těchto transformací je posloupnost dvou-dimenzionálních matic tvořící veřejný klíč pk kryptosystému. Jednotlivé bity otevřeného textu pak určují, které matice z veřejného klíče pk budou sečteny do šifrového textu. V případě, že by útočník chtěl z šifrového textu odhalit posílanou zprávu, musel by místo jednoho problému 0-1 batohu řešit čtyři navzájem provázané problémy 0-1 batohu.

Nejprve uvedeme definici problému maticového 0-1 batohu a jeho hustoty. Následně popíšeme konstrukci kryptosystému postaveném na tomto problému společně se stručnou kryptoanalýzou. V závěru kapitoly se budeme podrobně věnovat útoku založenému na algoritmu LLL [39] pro výpočet redukované báze mřížky. Uvedeme analogii důkazu z [35], čímž dokážeme, že útok pomocí LLL algoritmu bude pro tento kryptosystém úspěšný ve většině případů problémů maticového 0-1 batohu s hustotou menší než 0,161. Tento odhad může být vylepšen postupem z [12] na odhad 0,235. Uvidíme, že tento fakt povede k úspěšnosti útoku pro většinu kryptosystémů postavených na maticovém 0-1 batohu.

Problém maticového 0-1 batohu. Nechť $n \in \mathbb{N}$ a mějme danou n-tici dvoudimenzionálních matic $A_1, A_2, \ldots, A_n \in (\mathbb{Z}^+)^{2\times 2}$ a dvou-dimenzionální matici $C \in \mathbb{N}^{2\times 2}$. Pak *problém maticového 0-1 batohu* hledá vektor $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$ takový, že

$$\sum_{i=1}^{n} x_i A_i = C. \tag{3.11}$$

Označíme-li prvky matic $A_i, 1 \leq i \leq n$ jako

$$A_i = \begin{pmatrix} a_{11}^{(i)} & a_{12}^{(i)} \\ a_{21}^{(i)} & a_{22}^{(i)} \end{pmatrix}, \tag{3.12}$$

rozepsáním rovnice (3.11) dostaneme čtyři navzájem provázané problémy 0-1 batohu jako

$$\sum_{i=1}^{n} x_i a_{11}^{(i)} = c_{11}, \qquad \sum_{i=1}^{n} x_i a_{12}^{(i)} = c_{12},$$
$$\sum_{i=1}^{n} x_i a_{21}^{(i)} = c_{21}, \qquad \sum_{i=1}^{n} x_i a_{22}^{(i)} = c_{22}.$$

Hustota problému maticového 0-1 batohu je dle [28] dána jako

$$d = \frac{n}{\log_2 \max_i a_{11}^{(i)} + \log_2 \max_i a_{12}^{(i)} + \log_2 \max_i a_{21}^{(i)} + \log_2 \max_i a_{22}^{(i)}}.$$

Schéma kryptosystému. Nyní popíšeme fáze generování soukromého a veřejného klíče, šifrování a dešifrování pro kryptosystém postavený na maticovém problému 0-1 batohu.

Pro $n \in \mathbb{N}$ mějme zprávu $m = (m_1, m_2, \ldots, m_n)$ o n bitech, kterou chceme zašifrovat. Postup bude následující.

Generování soukromého a veřejného klíče. Zvolíme rychle rostoucí posloupnost $(p_1, p_2, \ldots, p_n)\mathbb{N}^n$ a číslo $N \in \mathbb{N}$ takové, že

$$N > \sum_{i=1}^{n} p_i.$$

Sestavíme matice $P_1, P_2, \ldots, P_n$ jako

$$P_i = \begin{pmatrix} p_i & \star \\ \star & \star \end{pmatrix}, \tag{3.13}$$

kde $\star$ představuje jakékoliv celé číslo řádově stejně velké jako p_i.

Dále zvolíme dvě matice $X, Y \in \mathrm{GL}(2, \mathbb{Z}_N)$. Pomocí matic X a Y vytvoříme matice $A_1, A_2, \ldots, A_n \in \mathbb{Z}_N^{(2\times 2)}$ jako

$$A_i \equiv X P_i Y \mod N.$$

Potom soukromý a veřejný klíč je

$$\mathsf{sk} = P_1, P_2, \ldots, P_n, X, Y, N,$$
$$\mathsf{pk} = A_1, A_2, \ldots, A_n.$$

Prvky veřejného klíče je možné propermutovat.

Zašifrování zprávy. Zprávu $m = (m_1, m_2, \ldots, m_n)$ zašifrujeme veřejným klíčem pk jako

$$C = \sum_{i=1}^{n} m_i A_i \tag{3.14}$$

a dostaneme šifrový text C.

Dešifrování zprávy. Z šifrového textu C získáme původní zprávu m za pomoci soukromého klíče sk následujícím postupem.

Spočítáme

$$X^{-1} C Y^{-1} \equiv X^{-1} \left(\sum_{i=1}^{n} m_i A_i \right) Y^{-1} \equiv \sum_{i=1}^{n} m_i P_i \mod N.$$

V levém horním rohu matice

$$\left(\sum_{i=1}^{n} m_i P_i \right) \mod N$$

je součet pro problém batohu postaveném na rychle rostoucí posloupnosti, který umíme řešit algoritmem 1 v polynomiálním čase.

3.4.1 Základní kryptoanalýza

V této části uvedeme odolnost kryptosystému založeném na problému maticového 0-1 batohu vzhledem k několika známým útokům na problém batohu. Nejprve zohledníme základní typy útoků založených na prohledávání pomocí hrubé síly a space-time-trade-off. Následně budeme zkoumat bezpečnost kryptosystému v jeho oslabenější verzi, a to v případě, že je v procesu generování soukromého a veřejného klíče vynechána operace modulo, tedy celé schéma pracuje bez modulu N. Na závěr této části se zaměříme na útok založený na redukci báze mřížky pomocí algoritmu LLL [39]. V souvislosti s tímto útokem dokážeme analogii postupu z [35], že tento útok uspěje pro téměř všechny kryptosystémy tohoto typu.

Útok hrubou silou a space-time-trade-off. První typ útoku, který je důležité zohlednit, je útok hrubou silou. V tomto případě by útočník musel vyzkoušet všech 2^n kombinací prvků A_i, aby zjistil původní otevřený text m. Pro volbu $n \geq 80$ nebude tento typ útoku proveditelný v reálném čase. Stejně tak útoky založené na principu space-time-trade-off, které celkovou časovou složitost 2^n rozdělí na časovou složitost $2^{\sqrt{n}}$ a paměťovou složitost $2^{\sqrt{n}}$ nebudou proveditelné v reálném čase pro $n \geq 160$.

Kryptosystém bez modulu *N*. Předpokládejme, že celý kryptosystém funguje stejně jako výše, ale operace nejsou moduleny číslem N. V tomto případě matice $P_1, P_2, \ldots, P_n$, prvky soukromého klíče sk, splňují (3.13) a pro matice X,$Y \in \mathrm{GL}(2, \mathbb{Z})$ vytvoříme prvky veřejného klíče pk jako

$$A_i = XP_iY \quad \text{pro všechna } 1 \leq i \leq n.$$

Nejprve zapíšeme prvky veřejného klíče pk jako sloupcové vektory, tedy

$$A_i = \begin{pmatrix} a_{11}^{(i)} \\ a_{12}^{(i)} \\ a_{21}^{(i)} \\ a_{22}^{(i)} \end{pmatrix} \in \mathbb{Z}^4 \quad \text{pro všechna } 1 \leq i \leq n.$$

Jelikož platí, že

$$X^{-1}A_iY^{-1} = P_i \quad \text{pro všechna } 1 \leq i \leq n,$$

kde v levých horních rozích matic $P_1, P_2, \ldots, P_n$ je propermutovaná rychle rostoucí posloupnost, pak víme, že existuje homomorfismus $\psi : \mathbb{Z}^4 \to \mathbb{Z}$ takový, že

1. $\psi(A_i) \in \mathbb{N}$, pro všechna $1 \leq i \leq n$,
2. existuje $j \in \{1, 2, \ldots, n\}$ takové, že $\psi(A_j) > \sum_{i \neq j} \psi(A_i)$

a že homomorfismus ψ je pro $z \in \mathbb{Z}^4, z = (z_1, z_2, z_3, z_4)^\top$ tvaru

$$\begin{pmatrix} z_1 \\ z_2 \\ z_2 \\ z_4 \end{pmatrix} \mapsto (u, v, x, y) \begin{pmatrix} z_1 \\ z_2 \\ z_2 \\ z_4 \end{pmatrix},$$

pro $(u, v, x, y) \in \mathbb{Z}^4$.

Hledáme tedy vektor $(u, v, x, y) \in \mathbb{Z}^4$ takový, že

1. $$(u, v, x, y) \begin{pmatrix} a_{11}^{(i)} \\ a_{12}^{(i)} \\ a_{21}^{(i)} \\ a_{22}^{(i)} \end{pmatrix} > 0 \quad \text{pro všechna } 1 \leq i \leq n,$$
2. existuje $j \in \{1, 2, \ldots, n\}$ takové, že
$$(u, v, x, y) \begin{pmatrix} a_{11}^{(j)} \\ a_{12}^{(j)} \\ a_{21}^{(j)} \\ a_{22}^{(j)} \end{pmatrix} - \sum_{i \neq j} (u, v, x, y) \begin{pmatrix} a_{11}^{(i)} \\ a_{12}^{(i)} \\ a_{21}^{(i)} \\ a_{22}^{(i)} \end{pmatrix} > 0.$$

O řešitelnosti výše popsaného systému (s opačnými znaménky) vypovídá Farkasova-Minkowského-Weylova věta, která vychází z [52, Věta 7.1a].

Věta 11 (Farkasova-Minkowského-Weylova věta)**.** *Pro jakoukoliv $A \in \mathbb{R}^{m\times n}$ existuje konečná množina X taková, že*

$$\left\{x = \sum_{i=1}^{n} x_i\lambda_i,\ x_i \in X,\ \lambda_i \geq 0\right\} = \{x,\ Ax \leq 0\}\,.$$

Pro konečné dimenze existuje polynomiální algoritmus, který nalezne množinu X [21, Věta 3.2].

3.4.2 Kryptoanalýza pomocí LLL algoritmu

Na začátku této kapitoly jsme viděli, že se algoritmy počítající redukovanou bázi mřížky dají použít při hledání řešení problému batohu. V případě kryptosystému postaveném na problému maticového 0-1 batohu nemusíme řešit pouze jeden problém 0-1 batohu, ale čtyři navzájem provázané. V roce 1985 ukázali J. C. Lagarais a A. M. Odlyzko jak aplikovat algoritmus LLL [39] při hledání řešení problému 0-1 batohu a dokázali, že jejich algoritmus SV [35, str 232], který pracuje v polynomiálním čase, najde téměř vždy hledané řešení jako nejkratší vektor v redukované bázi, pokud má problém 0-1 batohu hustotu $d \leq 0{,}645$. V roce 1991 pak M. J. Coster a spol. ukázali [12], že lze tento odhad vylepšit na 0,9408.

V následující části uvedeme analogii důkazu z [35] pro kryptosystém založený na problému maticového 0-1 batohu a dokážeme, že pomocí LLL algoritmu najdeme téměř vždy řešení jako nejkratší vektor redukované báze mřížky pro kryptosystémy s hustotou menší než 0,161 v polynomiální čase. Využitím postupu v [12] dostaneme odhad 0,235.

Dříve, než přistoupíme k důkazu, uvedeme horní odhad pro počet bodů celočíselné mřížky v kouli, který budeme dále potřebovat. K tomuto odhadu budeme také potřebovat definici *Jacobiho theta funkce*.

Počet bodů celočíselné mřížky v kouli.

Definice 24 (Jacobiho theta funkce)**.** *Nechť je $q \in \mathbb{C}, \tau \in \mathbb{C}$ takové, že $Im\ \tau > 0$ a $z = e^{i\pi\tau}$. Pak jsou první čtyři* Jacobiho theta funkce *definovány jako*

$$\begin{aligned}
\theta_1(q,z) &= \sum_{n=-\infty}^{\infty} (-1)^{n-1/2} z^{(n+1/2)^2} e^{(2n+1)iq},\\
\theta_2(q,z) &= \sum_{n=-\infty}^{\infty} z^{(n+1/2)^2} e^{(2n+1)iq},\\
\theta_3(q,z) &= \sum_{n=-\infty}^{\infty} z^{n^2} e^{2niq},\\
\theta_4(q,z) &= \sum_{n=-\infty}^{\infty} (-1)^n z^{n^2} e^{2niq}.
\end{aligned}$$

Definice 25 (Počet bodů celočíselné mřížky v kouli)**.** *Počet celočíselných řešení nerovnice*

$$\sum_{i=1}^{n} {x_i}^2 \leq R$$

značíme $S_n(R)$. Jedná se o počet bodů celočíselné mřížky, které leží uvnitř n-dimenzionální koule o poloměru $\sqrt{R}$ se středem v počátku.

Poznámka. Počet bodů celočíselné mřížky v množině $U \in \mathbb{R}^n$ můžeme většinou určit jako objem množiny U a chybovým faktorem, který je dán povrchem množiny U [16]. Tento postup však nebude fungovat v případě, kdy za množinu U volíme n-dimenzionální kouli o poloměru přibližně $\sqrt{n}$. Tuto skutečnost můžeme nahlédnout ze vzorců pro výpočet objemu V a povrchu S n-dimenzionální koule o poloměru $\sqrt{\mathrm{R}}$

$$V = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}\mathrm{R}^{\frac{n}{2}},$$
$$S = \frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)}n\mathrm{R}^{\frac{(n-1)}{2}}.$$

Dále z [43] víme, že počet prvků celočíselné mřížky v n-dimenzionální kouli s poloměrem $\sqrt{\alpha n}$ pro konstantu $\alpha \in \mathbb{N}$ silně závisí na umístění středu této koule. Průměrný počet bodů je roven objemu takové koule, ale minimální a maximální počet se pro kouli, jejíž střed se pohybuje v n-dimenzionální jednotkové krychli, liší od průměrné hodnoty až o násobek e^{cn}, kde c je konstanta odvozená od α. Dále se ukazuje, že maximální hodnoty se dosahuje právě pro koule se středem v počátku [43].

Následující věta z [35, Věta 3.2] uvádí odhad pro počet bodů celočíselné mřížky v n dimenzionální kouli a poloměrem $\sqrt{n/2}$ a se středem v počátku.

Věta 12. *Pro všechna $n \geq 1$, je $S_n(\frac{n}{2}) \leq 2^{1{,}54725n}$.*

Důkaz. Třetí Jacobiho theta funkci $\theta_3(0,z) = 1 + 2\sum_{i=1}^{\infty} z^{i^2}$ budeme zkráceně psát jako $\theta(z)$.

Počet celočíselných řešení rovnice

$$\sum_{i=1}^{n} x_i^2 = k$$

označíme $r_n(k)$.

Potom

$$[\theta(z)]^n = \sum_{k=0}^{\infty} r_n(k) z^k. \tag{3.15}$$

Rovnost (3.15) můžeme nahlédnout roznásobením výrazu

$$[\theta(z)]^n = \left(1 + 2\sum_{i=1}^{\infty} z^{i^2}\right)^n$$

člen po členu a sledováním koeficientu u z^k. V tomto momentě pracujeme s $\theta(z)$ jako s formální řadou.

$$\left(1 + 2\sum_{i=1}^{\infty} z^{i^2}\right)^n = \underbrace{\left(1 + 2\sum_{i=1}^{\infty} z^{i^2}\right) \cdot \left(1 + 2\sum_{i=1}^{\infty} z^{i^2}\right) \cdot \ldots \cdot \left(1 + 2\sum_{i=1}^{\infty} z^{i^2}\right)}_{n-\text{krát}} =$$
$$= \sum_{k=0}^{\infty} b_k z^k,$$

kde $b_k = |\{(i_1, i_2, \ldots, i_n) \in \mathbb{Z}^n,\, i_1^2 + \ldots + i_n^2 = k\}| = r_n(k)$.

Zvolme $\alpha > 0$, pak pro $x \geq 0$ máme

$$\mathrm{S}_n(\alpha n) = \sum_{k=0}^{\lfloor \alpha n \rfloor} r_n(k) \leq e^{n\alpha x} \sum_{k=0}^{\infty} r_n(k) e^{-kx} = e^{n\alpha x} \left[\theta(e^{-x})\right]^n, \tag{3.16}$$

jelikož pro $x \geq 0$ a $k \leq n\alpha$ máme

$$e^{n\alpha x} e^{-kx} \geq 1.$$

V rovnosti (3.16) zbývá ukázat, že výraz e^{-x} můžeme dosadit do třetí Jacobiho theta funkce. Ekvivalentně se ptáme, zda pro formální řadu $[\theta(z)]^n$ bude $[\theta(e^{-x})]^n$ konvergentní, pro $x > 0$.

Z rovnosti (3.15) víme, že

$$[\theta(z)]^n = \sum_{k=0}^{\infty} r_n(k) z^k.$$

Nechť φ_α je dosazovací homomorfismus pro $\alpha \in (0{,}1)$ (pro $x > 0$ je $e^{-x} < 1$) definovaný jako

$$\varphi_\alpha : \sum_{i=0}^{\infty} a_i z^i \mapsto \sum_{i=0}^{\infty} a_i \alpha^i,$$

kde pro každé $c \in (1, \infty)$ a pro skoro všechna k platí $a_k \leq c^k$.

Pak máme z $\sum_{i=n}^{m} u^i = \frac{u^{m+1} - u^n}{u-1}$

$$\sum_{i=n}^{m} a_i \alpha^i \leq c^m \sum_{i=n}^{m} \alpha^i = c^m \frac{\alpha^m \left(\alpha - \alpha^{n-m}\right)}{\alpha - 1},$$

pro všechna dostatečně velká n.

Pak pro $c\alpha < 1$ je

$$\lim_{m \to \infty} \frac{(c\alpha)^m \left(\alpha - \alpha^{n-m}\right)}{\alpha - 1} = 0$$

a řada $\sum_{i=0}^{\infty} a_i \alpha^i$ konverguje z Bolzano-Cauchyho podmínky.

Položme

$$\delta\left(\alpha{,}x\right) = \alpha x + \ln\, \theta(e^{-x}).$$

Z (3.16) plyne, že

$$\mathrm{S}_n\left(\alpha n\right) \leq e^{n\delta(\alpha,x)} = 2^{(\log_2 e)\delta(\alpha,x)n}.$$

Dosadíme $\alpha = \frac{1}{2}$ a označíme

$$g(x) = 2^{(\log_2 e)\delta\left(\frac{1}{2},x\right)}, \tag{3.17}$$

$$f(x) = x^n,$$

$$f(g(x)) = 2^{(\log_2 e)\delta\left(\frac{1}{2},x\right)n}. \tag{3.18}$$

Extrém funkce (3.17) byl nalezen pomocí vývojového prostředí Wolfram Mathematica 8 vypočítáním kořenu derivace funkce $g(x)$ jako

$$\frac{\partial g(x)}{\partial x} = 0 \text{ pro } x_0 \doteq 0{,}997\,994.$$

Funkce $f : \mathbb{R} \to \mathbb{R}$ definovaná vztahem $f(x) = x^n$ je rostoucí pro $n \in \mathbb{N}$ a $x > 0$ a extrém funkce (3.18) dostaneme jako

$$x_0 = \operatorname*{argmax}_x 2^{(\log_2 e)\delta\left(\frac{1}{2},x\right)n} \doteq 0{,}997\,994.$$

Potom

$$\delta\left(\frac{1}{2}, x_0\right) \leq 1{,}07247$$

a

$$(\log_2 e)\,\delta\left(\frac{1}{2}, x_0\right) \leq 1{,}54725$$

a

$$\mathrm{S}_n\left(\frac{n}{2}\right) \leq 2^{1{,}54725n}.$$

□

Odhad $\mathrm{S}_n(\mathrm{R})$, pro $\mathrm{R} = \frac{n}{2}$ je důležitý ze vztahu (3.21)

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2}.$$

V dalších aplikacích budeme potřebovat n-dimenzionální koule s poloměrem $\mathrm{R} = \sqrt{\frac{n}{2}}$ a se středem v počátku.

Útoku pomocí LLL algoritmu a jeho analýza. V této části uvedeme analogii důkazu z [35]. Nejprve popíšeme obecné schéma útoku v případě aplikace na kryptosystém založený na problému maticového 0-1 batohu. Následně uvedeme algoritmus KV (nejkratší vektor), který řeší rovnici (3.14) pomocí LLL algoritmu. Poslední část věnujeme analogii důkazu úspěšnosti algoritmu KV z [35] a dokážeme, že řešení (3.14) bude v polynomiálním čase nalezeno téměř vždy, pokud je hustota d problému maticového 0-1 batohu menší než 0,161. Na závěr poznamenáme, že využitím modifikace z [12] můžeme dosáhnout meze 0,235.

Obecné schéma útoku. Předpokládáme, že máme kryptosystém popsaný výše. Pro všechna $i \in \{1, 2, \ldots, n\}$ označíme prvky veřejného klíče $A_1, A_2, \ldots, A_n$ stejně jako v (3.12) a prvky šifrového textu označíme

$$C = \begin{pmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{pmatrix}. \tag{3.19}$$

Vektory

$$\begin{aligned}
b_1 &= \left(1, 0, \ldots, 0, 0, -a_{11}^{(1)}, -a_{12}^{(1)}, -a_{21}^{(1)}, -a_{22}^{(1)}\right) \\
b_2 &= \left(0, 1, \ldots, 0, 0, -a_{11}^{(2)}, -a_{12}^{(2)}, -a_{21}^{(2)}, -a_{22}^{(2)}\right) \\
&\vdots \\
b_n &= \left(0, 0, \ldots, 0, 1, -a_{11}^{(n)}, -a_{12}^{(n)}, -a_{21}^{(n)}, -a_{22}^{(n)}\right) \\
b_{n+1} &= \Big(0, 0, \ldots, 0, 0, c_{11}, c_{12}, c_{21}, c_{22}\Big),
\end{aligned} \tag{3.20}$$

tvoří bázi celočíselné mřížky $\mathcal{L}(\mathbf{A},C)$ s hodností $(n+1)$ a dimenzí $(n+4)$, tedy

$$\mathcal{L}(\mathbf{A},C) = \mathbb{Z}b_1 \oplus \mathbb{Z}b_2 \oplus \ldots \oplus \mathbb{Z}b_{n+1} \subseteq \mathbb{Z}^{n+4}.$$

Algoritmus, který řeší (3.14) pomocí LLL algoritmu nazýváme ***algoritmus KV***. Algoritmus KV je analogií Algoritmu SV (shortest vector) [35] a funguje následovně.

1. Pro bázi $b_1, b_2, \ldots, b_{n+1}$ určenou vztahy (3.20) najde redukovanou bázi $b_1^*, b_2^*, \ldots, b_{n+1}^*$ mřížky $\mathcal{L}$ pomocí LLL algoritmu z [39].

2. Ověří, zda pro nějaké $b_j^* = (b_{j,1}^*, b_{j,2}^*, \ldots, b_{j,n+4}^*)$ platí, že $b_{j,k}^* = 0$ nebo $b_{j,k}^* = \lambda$ pro nějaké fixní $\lambda \in \mathbb{N}$ a pro všechna $1 \leq k \leq n$. Pro takové b_j^* ověříme, zda vektor $x = (\lambda^{-1}b_{j,1}^*, \lambda^{-1}b_{j,2}^*, \ldots, \lambda^{-1}b_{j,n}^*)$ řeší rovnici (3.14) a pokud ano, pak jsme obdrželi řešení a algoritmus KV skončí.

3. Zopakuje kroky 1) a 2) s $C' = \sum_{i=1}^{n} A_i - C$ a skončí.

Asymptotická časová složitost je dle [35, Lemma 2.3] daná jako $\mathcal{O}(n^6(\log n\mathrm{B})^3)$ bitových operací, kde B je horní odhad pro prvky v maticích $A_1, A_2, \ldots, A_n$.

Víme, že rovnice (3.14) má řešení. Nějaké takové řešení označíme $e = (e_1, e_2, \ldots, e_n)$, tedy

$$\sum_{i=1}^{n} A_i e_i = C$$

a

$$\sum_{i=1}^{n} a_{11}^{(i)} e_i = c_{11}, \qquad \sum_{i=1}^{n} a_{12}^{(i)} e_i = c_{12},$$
$$\sum_{i=1}^{n} a_{21}^{(i)} e_i = c_{21}, \qquad \sum_{i=1}^{n} a_{22}^{(i)} e_i = c_{22}.$$

Poznámka. Pro každý takový vektor $e \in \{0,1\}^n$ a jeho doplněk $e^* \in \{0,1\}^n$, tedy vektor $e_i^* = 1 - e_i$, pro $i \in \{1, 2, \ldots, n\}$ platí, že buď $\sum_{i=1}^{n} e_i \leq \frac{n}{2}$, nebo $\sum_{i=1}^{n} e_i^* \leq \frac{n}{2}$. Také nebudeme uvažovat triviální případ, kdy C je nulová matice a tedy $e = (0, \ldots, 0)$. Vidíme, že můžeme bez újmy na obecnosti předpokládat, že

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2}. \tag{3.21}$$

Poznámka. Bez újmy na obecnosti předpokládáme, že jsou všechny prvky veřejného klíče $a_{\mathrm{H}}^{(i)}$, kde $1 \leq i \leq n$ a $\mathrm{H} \in \{11, 12, 21, 22\}$, označené stejně jako v (3.12), nenulové. V případě, že by se ve veřejném klíči pk vyskytovaly matice s nulovými prvky, pak můžeme provést transformaci $A_i + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ pro všechny matice $A_i, 1 \leq i \leq n$, které mají nějaký prvek nulový. Označme tedy $K \in \{1, 2, \ldots, n\}$ počet matic $A_1, A_2, \ldots, A_n$, které mají nějaký prvek nulový. Ke všem těmto K maticím přičteme matici $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$. Jelikož neznáme zašifrovanou zprávu m, pak nevíme, kolikrát je nyní potřeba matici $\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ přičíst k šifrovému textu C. Útok pomocí algoritmu KV tedy provedeme postupně pro všechny matice

$$C, C + \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, C + 2\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, \ldots, C + K\begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}.$$

Maximálně budeme muset celý postup provést n-krát a tedy dostaneme výslednou složitost útoku $\mathcal{O}(n^7(\log n\mathrm{B})^3)$.

Označme vektor $\mathbf{e} = (e_1, e_2, \ldots, e_n, 0, 0, 0, 0) \in \mathcal{L}(\mathbf{A}, C)$. Ze vztahu (3.21) plyne

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2}.$$

Z vektoru $\mathbf{e}$ a meze $\mathrm{B} \in \mathbb{R}$ vytvoříme prostor $\Lambda(\mathrm{B}, \mathbf{e})$ všech možných vstupů pro algoritmus KV. Pomocí tohoto prostoru budeme analyzovat práci algoritmu KV.

Prostor $\Lambda(\mathrm{B}, \mathbf{e})$ je množina určená jako

$$\Lambda(\mathrm{B}, \mathbf{e}) = \\ = \left\{\mathbf{L}(\mathbf{A}, \mathrm{C}), 1 \leq a_{\mathrm{H}}^{(i)} \leq \mathrm{B}, 1 \leq i \leq n, \mathrm{H} \in \{11, 12, 21, 22\}, C\right\},$$

kde $C = \sum_{i=1}^{n} A_i e_i$. Potom $\mathbf{L}(\mathbf{A},\mathrm{C}) = \mathbb{Z}b_1 \oplus \mathbb{Z}b_2 \oplus \ldots \oplus \mathbb{Z}b_{n+1}$, vektory $b_1, b_2, \ldots, b_{n+1}$ jsou určeny vztahy (3.20) a matice C a $A_i, 1 \leq i \leq n$ jsou dány vztahem (3.14). Vidíme, že každý vstup $\mathbf{L}(\mathbf{A},\mathrm{C})$ jednoznačně určuje mřížku $\mathcal{L}(\mathrm{A},\mathrm{C})$ s bází $b_1, b_2, \ldots, b_{n+1}$ danou vztahy (3.20). Velikost tohoto prostoru je $|\Lambda(\mathrm{B}, \mathbf{e})| = \mathrm{B}^{4n}$.

Při použití útoků založených na redukci báze mřížky jsou úspěšné ty případy, kdy je vektor $\mathbf{e}$ nejkratším vektorem v redukované bázi. V následující části ukážeme, že pro většinu vstupů algoritmu KV tomu tak doopravdy bude. Intuitivní důvod toho je, že algoritmus LLL ve většině případů najde lepší aproximaci nejkratšího nenulového vektoru, než jaká je garantovaná lemmatem 3 [39].

Pomocí prostoru $\Lambda(\mathrm{B}, \mathbf{e})$ a Algoritmu KV nyní můžeme formulovat problém, který chceme vyřešit. Jak často najde algoritmus KV vektor $\mathbf{e}$ jako nejkratší vektor v redukované bázi, když jej aplikujeme na všechny vstupy z prostoru $\Lambda(\mathrm{B}, \mathbf{e})$? O tom, kolik vstupů z prostoru $\Lambda(\mathrm{B}, \mathbf{e})$ obsahuje krátký vektor, který je různý od vektoru $\mathbf{e}$ nebo od jeho násobku, vypovídá následující věta.

Věta 13. *Nechť $n \in \mathbb{N}$, $R \in \mathbb{N}, R \geq 4$, pak počet vstupů $\boldsymbol{L}(\boldsymbol{A},C) \in \Lambda(B,\boldsymbol{e})$ takových, že $\boldsymbol{L}(\boldsymbol{A},C)$ obsahuje vektor w takový, že*

1. *$w \neq k\boldsymbol{e}, k \in \mathbb{Z}$*
2. *$\|w\|^2 \leq R$*

je méně než

$$64 \cdot R^4 B^{4(n-1)} S_{(n+4)}(R).$$

Důkaz. Označme $T = T(\mathrm{R}, \mathrm{B}, \mathbf{e})$ počet takových vstupů $\mathbf{L}(\mathbf{A},C)$.

Ze vztahu (3.21) předpokládáme, že $\mathbf{e} \neq (0, \ldots, 0)$. Pokud $\mathbf{e} = (0, \ldots, 0)$, pak pracujeme s nulovou maticí C. Prostor $\Lambda(\mathrm{B},(0,\ldots,0)) = \Lambda(\mathrm{B}, \mathbf{e})$, pro $\mathbf{e} = (0,\ldots,0)$ a věta 13 platí, jelikož v tomto případě je vektorů $w \neq (0,\ldots,0)$, $\|w\|^2 \leq \mathrm{R}$ méně, než $\mathrm{S}_{n+4}(\mathrm{R})$.

Mějme vektor $w \in \mathbf{L}(\mathbf{A},C)$, který splňuje podmínky 1 a 2. Spočítáme, kolik vstupů $\mathbf{L}(\mathbf{A},C) \in \Lambda(\mathrm{B}, \mathbf{e})$ obsahuje vektor w.

Označme složky $w = (w_1, w_2, \ldots, w_n, r_{11}, r_{12}, r_{21}, r_{22})$. Potom z podmínky 2 máme

$$\|w\|^2 = \sum_{i=1}^{n} w_i^2 + r_{11}^2 + r_{12}^2 + r_{21}^2 + r_{22}^2 \leq \mathrm{R}. \tag{3.22}$$

Odtud platí

$$|r_{\mathrm{H}}| \leq \sqrt{\mathrm{R}}, \quad \text{pro každé H} \in \{11, 12, 21, 22\}, \tag{3.23}$$

$$|w_i| \leq \sqrt{\mathrm{R}}, \quad \text{pro všechna } 1 \leq i \leq n. \tag{3.24}$$

Jelikož $w \in \mathbf{L}\,(\mathbf{A},C)$, můžeme w vyjádřit jako kombinaci prvků báze $\mathbf{L}\,(\mathbf{A},C)$ jako

$$w = \sum_{i=1}^{n} w_i b_i + \lambda b_{n+1}, \tag{3.25}$$

pro $\lambda \in \mathbb{Z}$.

Potom pro poslední 4 souřadnice vektoru w platí

$$\begin{aligned}
r_{11} &= -\sum_{i=1}^{n} w_i a_{11}^{(i)} + \lambda c_{11} = -\sum_{i=1}^{n} w_i a_{11}^{(i)} + \lambda \sum_{i=1}^{n} a_{11}^{(i)} e_i = \sum_{i=1}^{n} \left(-w_i + \lambda e_i\right) a_{11}^{(i)} \\
r_{12} &= -\sum_{i=1}^{n} w_i a_{12}^{(i)} + \lambda c_{12} = -\sum_{i=1}^{n} w_i a_{12}^{(i)} + \lambda \sum_{i=1}^{n} a_{12}^{(i)} e_i = \sum_{i=1}^{n} \left(-w_i + \lambda e_i\right) a_{12}^{(i)} \\
r_{21} &= -\sum_{i=1}^{n} w_i a_{21}^{(i)} + \lambda c_{21} = -\sum_{i=1}^{n} w_i a_{21}^{(i)} + \lambda \sum_{i=1}^{n} a_{21}^{(i)} e_i = \sum_{i=1}^{n} \left(-w_i + \lambda e_i\right) a_{21}^{(i)} \\
r_{22} &= -\sum_{i=1}^{n} w_i a_{22}^{(i)} + \lambda c_{22} = -\sum_{i=1}^{n} w_i a_{22}^{(i)} + \lambda \sum_{i=1}^{n} a_{22}^{(i)} e_i = \sum_{i=1}^{n} \left(-w_i + \lambda e_i\right) a_{22}^{(i)}
\end{aligned} \tag{3.26}$$

a dále

$$\begin{aligned}
r_{11} + r_{12} + r_{21} + r_{22} = \\
&= -\sum_{i=1}^{n} w_i a_{11}^{(i)} + \lambda c_{11} - \sum_{i=1}^{n} w_i a_{12}^{(i)} + \lambda c_{12} - \sum_{i=1}^{n} w_i a_{21}^{(i)} + \lambda c_{21} \\
&- \sum_{i=1}^{n} w_i a_{22}^{(i)} + \lambda c_{22}.
\end{aligned}$$

Tedy pro každé $a_{\mathrm{H}}^{(j)} \leq \mathrm{B}, j \in \{1, 2, 3, 4\}, \mathrm{H} \in \{11, 12, 21, 22\}$ dostáváme

$$\begin{aligned}
&|\lambda \cdot (c_{11} + c_{12} + c_{21} + c_{22})| \leq \\
&\leq |r_{11} + r_{12} + r_{21} + r_{22}| + \left| \sum_{i=1}^{n} w_i a_{11}^{(i)} + \sum_{i=1}^{n} w_i a_{12}^{(i)} + \sum_{i=1}^{n} w_i a_{21}^{(i)} + \sum_{i=1}^{n} w_i a_{22}^{(i)} \right| \leq \\
&\leq \mathrm{B} \left(4 \sum_{i=1}^{n} |w_i| + |r_{11} + r_{12} + r_{21} + r_{22}| \right) \leq 4\mathrm{B} \left(\sum_{i=1}^{n} w_i^2 + r_{11}^2 + r_{12}^2 + r_{21}^2 + r_{22}^2 \right) \leq \\
&\leq 4\mathrm{RB}.
\end{aligned} \tag{3.27}$$

Protože $c_{11} + c_{12} + c_{21} + c_{22} \neq 0$, máme

$$|\lambda| \leq 4\mathrm{RB}. \tag{3.28}$$

Bez újmy na obecnosti předpokládejme, že $e_1 \neq 0$, tedy $e_1 = 1$. Potom pro každé $\mathrm{H} \in \{11, 12, 21, 22\}$ platí

$$c_{\mathrm{H}} = \sum_{i=1}^{n} a_{\mathrm{H}}^{(i)} e_i \geq a_{\mathrm{H}}^{(1)} e_1 = a_{\mathrm{H}}^{(1)}. \tag{3.29}$$

Z (3.22) a (3.26) pro každé $a_H^{(j)} \leq \mathrm{B}$, $\mathrm{H} \in \{11, 12, 21, 22\}$, $j \in \{1, 2, 3, 4\}$ platí, že

$$|\lambda| a_{\mathrm{H}}^{(1)} \leq |\lambda| c_{\mathrm{H}} \leq |r_{\mathrm{H}}| + \left| \sum_{i=1}^{n} w_i a_{\mathrm{H}}^{(i)} \right| \leq \sqrt{\mathrm{R}} + \mathrm{RB}$$

a pro $\lambda \neq 0$ platí

$$a_{\mathrm{H}}^{(1)} \leq \frac{\sqrt{\mathrm{R}} + \mathrm{RB}}{|\lambda|} \tag{3.30}$$

pro každé $\mathrm{H} \in \{11, 12, 21, 22\}$.

Označme $N(w, \lambda)$ počet vstupů $\mathbf{L}(\mathbf{A}, C)$, pro které $w \in \mathbf{L}(\mathbf{A}, C)$ splňuje podmínky 1 a 2 a λ splňuje rovnici (3.25). Potom z (3.28) platí

$$T \leq \sum_{\substack{\|w\|^2 \leq \mathrm{R} \\ w \neq k\mathbf{e}}} \left(\sum_{\lambda=-4\mathrm{RB}}^{4\mathrm{RB}} N(w, \lambda) \right), \tag{3.31}$$

pro $k \in \mathbb{Z}$.

K určení horního odhadu pro (3.31) zvolíme pomocný vektor

$$z = (w_1 - \lambda e_1, w_2 - \lambda e_2, \ldots, w_n - \lambda e_n)$$

a budeme uvažovat čtyři případy podle hodnoty vektoru z a λ. Tato konkrétní volba pomocného vektoru z je patrná z (3.26).

Případ 1. $z = 0$

Pro $z = 0$ máme $(w_1 - \lambda e_1, w_2 - \lambda e_2, \ldots, w_n - \lambda e_n) = (0,0,\ldots,0)$. Potom

$$w = (\lambda e_1, \lambda e_2, \ldots, \lambda e_n, N_1, N_2, N_3, N_4),$$

kde $N_1, N_2, N_3, N_4 \in \mathbb{Z}$.

Potom vektor

$$w - \lambda\mathbf{e} = (0, \ldots, 0, N_1, N_2, N_3, N_4)$$

leží v $\mathbf{L}(\mathbf{A},C)$ a $(N_1, N_2, N_3, N_4) = k(c_{11}, c_{12}, c_{21}, c_{22})$, pro $k \in \mathbb{Z}$. Pokud by bylo $k = 0$, potom $w = \lambda\mathbf{e}$ a to je ve sporu s předpokladem 1, tedy $|k| \geq 1$.

Dále

$$||w||^2 = \sum_{i=1}^{n} \lambda^2 {e_i}^2 + k^2 c_{11}^2 + k^2 c_{12}^2 + k^2 c_{21}^2 + k^2 c_{22}^2 \geq c_{11}^2 + c_{12}^2 + c_{21}^2 + c_{22}^2$$

a

$$||w|| \geq \sqrt{c_{11}^2 + c_{12}^2 + c_{21}^2 + c_{22}^2}.$$

Z lemma 2 máme

$$2\sqrt{c_{11}^2 + c_{12}^2 + c_{21}^2 + c_{22}^2} \geq |c_{11}| + |c_{12}| + |c_{21}| + |c_{22}|$$

a dostáváme

$$2\,||w|| \geq 2\sqrt{c_{11}^2 + c_{12}^2 + c_{21}^2 + c_{22}^2} \geq |c_{11}| + |c_{12}| + |c_{21}| + |c_{22}|\,.$$

Z (3.22) a (3.29) pak máme

$$2\sqrt{\mathrm{R}} \geq 2\,||w|| \geq a_{11}^{(1)} + a_{12}^{(1)} + a_{21}^{(1)} + a_{22}^{(1)},$$

tedy

$$2\sqrt{\mathrm{R}} \geq a_{11}^{(1)} + a_{12}^{(1)} + a_{21}^{(1)} + a_{22}^{(1)}$$

a pro každé $\mathrm{H} \in \{11, 12, 21, 22\}$ platí

$$a_{\mathrm{H}}^{(1)} \leq 2\sqrt{\mathrm{R}}. \tag{3.32}$$

Pro pevně zvolené w a λ můžeme odhadnout

$$\mathrm{N}(w, \lambda) \leq 2^4 \mathrm{R}^{\frac{4}{2}} \mathrm{B}^{4(n-1)} = 2^4 \mathrm{R}^2 \mathrm{B}^{4(n-1)}.$$

Jelikož $w \in \mathbf{L}(\mathbf{A}, C) \subseteq \mathbb{Z}^{n+4}$, pak máme nejvýše $\mathrm{S}_{(n+4)}(\mathrm{R})$ možností [1], jak volit vektor w.

Dále každé w již jednoznačně určí λ z předpokladu $z = 0$, tedy

$$\sum_{\text{Případ 1}} \mathrm{N}(w, \lambda) \leq 2^4 \cdot \mathrm{R}^2 \mathrm{S}_{(n+4)}(\mathrm{R})\, \mathrm{B}^{4(n-1)}. \tag{3.33}$$

Případ 2. $w_1 - \lambda e_1 \neq 0, w_j - \lambda e_j = 0$, pro každé $2 \leq j \leq n$.

V tomto případě máme z (3.26)

$$\begin{aligned} r_{11} &= \sum_{i=1}^{n} (-w_i + \lambda e_i)\, a_{11}^{(i)} = (-w_1 + \lambda e_1)\, a_{11}^{(1)} \\ r_{12} &= (-w_1 + \lambda e_1)\, a_{12}^{(1)} \\ r_{21} &= (-w_1 + \lambda e_1)\, a_{21}^{(1)} \\ r_{22} &= (-w_1 + \lambda e_1)\, a_{22}^{(1)} \end{aligned} \tag{3.34}$$

a společně s (3.23) máme

$$1 \leq a_{\mathrm{H}}^{(1)} \leq \sqrt{\mathrm{R}}, \tag{3.35}$$

pro každé $\mathrm{H} \in \{11, 12, 21, 22\}$.

Můžeme tedy pro w a λ pevně zvolené odhadnout

$$N(w, \lambda) \leq \mathrm{R}^{\frac{4}{2}} \mathrm{B}^{4(n-1)} \leq \mathrm{R}^2 \mathrm{B}^{4(n-1)}.$$

Z (3.24) máme

$$|w_1| \leq \sqrt{\mathrm{R}}. \tag{3.36}$$

Z (3.23) a (3.34) platí

$$|r_{11}| = |(-w_1 + \lambda e_1) a_{11}^{(1)}| \leq \sqrt{\mathrm{R}}.$$

Dále máme

$$|\lambda e_1| - |w_1| \leq |-w_1 + \lambda e_1| \leq \frac{\sqrt{\mathrm{R}}}{a_{11}^{(1)}}$$

[1] Autoři v [35] uvažovali počet možností, jak volit vektor $w \in \mathbb{Z}^{(n+1)}$ pouze jako $\mathrm{S}_{(n)}(\mathrm{R})$ místo očekávaného $\mathrm{S}_{(n+1)}(\mathrm{R})$. Vzhledem k tomu, že se v následujících důkazech pracuje s $\mathrm{R} = \frac{n}{2}$, pak se v [35] tato nepřesnost neprojevila.

a pro $e_1 = 1$ z (3.36) platí

$$|\lambda| \leq \frac{\sqrt{\mathrm{R}}}{a_{11}^{(1)}} + |w_1| \leq \sqrt{\mathrm{R}} \left(\frac{1}{a_{11}^{(1)}} + 1 \right) \leq 2\sqrt{\mathrm{R}}.$$

Zbylé hodnoty $(w_2, \ldots, w_n)$ jsou jednoznačně určeny vztahem

$$w_j = \lambda e_j,$$

tedy máme nejvýše $\left(4\sqrt{\mathrm{R}} + 1\right)\left(2\sqrt{\mathrm{R}} + 1\right) = 8\mathrm{R} + 6\sqrt{\mathrm{R}} + 1$ možností voleb dvojic (w, λ).

Potom

$$\sum_{\text{Případ 2}} N(w, \lambda) \leq \left(8\mathrm{R} + 6\sqrt{\mathrm{R}} + 1\right) \cdot \mathrm{R}^2 \mathrm{B}^{4(n-1)} = \left(8\mathrm{R}^3 + 6\mathrm{R}^{\frac{5}{2}} + \mathrm{R}^2\right) \mathrm{B}^{4(n-1)}. \tag{3.37}$$

Případ 3. Existuje $j \in \{2, 3, \ldots, n\}$ takové, že $w_j - \lambda e_j \neq 0, \lambda \neq 0$.

Zvolme náhodně pevně w, λ a jeden ze sloupců $\mathrm{H} \in \{11, 12, 21, 22\}$. Ze vztahu (3.30) máme nejvýše $\frac{\sqrt{\mathrm{R}}+\mathrm{RB}}{|\lambda|}$ možností pro každé $a_H^{(1)}$ a B^{n-2} pro $a_H^{(i)}, i \neq j, i \neq 1$. Pro každou takovou volbu $a_H^{(i)}$ máme nejvýše jednu volbu pro $a_H^{(j)}$, jelikož $a_H^{(j)}$ je pro $w_j - \lambda e_j \neq 0$ jednoznačně určeno vztahem (3.26).

Pro všechny 4 sloupce tedy dostáváme

$$\mathrm{N}(w, \lambda) \leq \left(\frac{\sqrt{\mathrm{R}} + \mathrm{RB}}{|\lambda|} \cdot \mathrm{B}^{n-2} \right)^4 = \frac{\left(\sqrt{\mathrm{R}} + \mathrm{RB}\right)^4 \mathrm{B}^{4(n-2)}}{|\lambda|^4}.$$

A celkem máme

$$\sum_{\text{Případ 3}} N(w, \lambda) \leq \sum_{\substack{\|w\|^2 \leq \mathrm{R} \\ w \neq k\mathbf{e}}} \sum_{\substack{\lambda = -4\mathrm{RB} \\ \lambda \neq 0}}^{4\mathrm{RB}} \left(\frac{\left(\sqrt{\mathrm{R}} + \mathrm{RB}\right) \mathrm{B}^{n-2}}{\lambda} \right)^4 \leq$$

$$\leq \mathrm{S}_{(n+4)}(\mathrm{R})\, 2 \cdot \left(\left(\sqrt{\mathrm{R}} + \mathrm{RB}\right)^4 \mathrm{B}^{4(n-2)} \right) \sum_{\lambda=1}^{4\mathrm{RB}} \frac{1}{\lambda^4}.$$

Jelikož

$$\sum_{\lambda=1}^{4\mathrm{RB}} \frac{1}{\lambda^4} \leq \frac{\pi^4}{90}$$

a

$$\left(\sqrt{\mathrm{R}} + \mathrm{RB}\right)^4 = \sum_{k=0}^{4} \binom{4}{k} \mathrm{R}^{\frac{4-k}{2}} (\mathrm{RB})^k \leq 16 \cdot \mathrm{R}^4 \mathrm{B}^4,$$

máme

$$\sum_{\text{Případ 3}} N(w, \lambda) \leq \frac{16\pi^4}{90} \cdot 2 \cdot \mathrm{R}^4 \mathrm{B}^4 \mathrm{S}_{(n+4)}(\mathrm{R}) \mathrm{B}^{4(n-2)} = \frac{32\pi^4}{90} \cdot \mathrm{R}^4 \mathrm{S}_{(n+4)}(\mathrm{R}) \mathrm{B}^{4(n-1)}. \tag{3.38}$$

Případ 4. Existuje $j \in \{2, 3, \ldots, n\}$ takové, že $w_j - \lambda e_j \neq 0, \lambda = 0$.

Zvolme náhodně pevně w a jeden sloupec $\mathrm{H} \in \{11, 12, 21, 22\}$. Máme B^{n-1} možností pro všechna $a_H^{(i)}$ kromě $a_H^{(j)}$. Ze vztahu (3.23) máme nejvýše $2\sqrt{\mathrm{R}}+1$ možností volby pro r_{H}, $\mathrm{H} \in \{11, 12, 21, 22\}$, tedy máme nejvýše $2\sqrt{\mathrm{R}}+1$ možností volby pro $a_H^{(j)}$, které musí splňovat (3.26). Tedy pro všechny čtyři sloupce dostáváme

$$N(w,0) \leq \left(\left(2\sqrt{\mathrm{R}}+1\right)\mathrm{B}^{n-1}\right)^4.$$

Jelikož

$$\left(2\sqrt{\mathrm{R}}+1\right)^4 = \sum_{k=0}^{4}\binom{4}{k}\left(2\sqrt{\mathrm{R}}\right)^{(4-k)} \leq 16\cdot 2^4\mathrm{R}^2,$$

pak pro všechny volby w, dostáváme

$$\sum_{\text{Případ 4}} N(w,0) \leq \left(\left(2\sqrt{\mathrm{R}}+1\right)\mathrm{B}^{n-1}\right)^4 \mathrm{S}_{(n+4)}(\mathrm{R}) \leq 256\cdot \mathrm{R}^2\mathrm{S}_{(n+4)}(\mathrm{R})\,\mathrm{B}^{4(n-1)}. \tag{3.39}$$

Spojením odhadů (3.33), (3.37), (3.38) a (3.39) dostáváme

$$\begin{aligned} T &\leq 2^4\cdot \mathrm{R}^2\mathrm{S}_{(n+4)}(\mathrm{R})\,\mathrm{B}^{4(n-1)} + \left(8\mathrm{R}^3 + 6\mathrm{R}^{\frac{5}{2}} + \mathrm{R}^2\right)\mathrm{B}^{4(n-1)} + \\ &+ \frac{32\pi^4}{90}\cdot \mathrm{R}^4\mathrm{S}_{(n+4)}(\mathrm{R})\mathrm{B}^{4(n-1)} + 256\cdot \mathrm{R}^2\mathrm{S}_{(n+4)}(\mathrm{R})\mathrm{B}^{4(n-1)} \leq \\ &\leq 64\cdot \mathrm{R}^4\mathrm{S}_{(n+4)}(\mathrm{R})\mathrm{B}^{4(n-1)}, \end{aligned}$$

pro $\mathrm{R} \geq 4$.

□

Následující věta dokazuje, že téměř všechny vstupy $\mathbf{L}(\mathbf{A},C) \in \Lambda(\mathrm{B},\mathbf{e})$ mají vektor $\mathbf{e}$ jako nejkratší nenulový vektor, pro fixní vektor $\mathbf{e}$. Konkrétně pro $\mathrm{B} = 2^{\beta n}$ z definice hustoty d problému maticového 0-1 batohu dostaneme

$$d \leq \frac{n}{4\log_2 2^{\beta n}} = \frac{n}{4\beta} = \frac{n}{4\cdot 1{,}54725} \doteq 0{,}161.$$

Z toho plyne, že v případě, že máme polynomiální algoritmus, který téměř vždy najde nejkratší nenulový vektor ve vstupu $\mathbf{L}(\mathbf{A},C)$.

Věta 14. *Nechť $n \in \mathbb{N}, n \geq 8$ a nechť $\boldsymbol{e} \in \{0,1\}^{(n+4)}$ je vektor takový, že*

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2}.$$

Potom pro mez $B = 2^{\beta n}$, kde $\beta > 1{,}54725$ platí, že počet vstupů $\boldsymbol{L}(\boldsymbol{A},C) \in \Lambda(B,\boldsymbol{e})$, pro které je vektor $\boldsymbol{e}$ nejkratší nenulový vektor vzhledem k Euklidovské normě je alespoň

$$B^{4n} - 4\cdot 73\cdot n^4 B^{4n-c_1(\beta)},$$

kde $c_1(\beta) = 4 - \frac{1{,}54725}{\beta} > 0$.

Důkaz. Z věty 13 plyne, že počet takových vstupů je pro $\mathrm{R} = \frac{n}{2}, \mathrm{R} \geq 4$ alespoň

$$\mathrm{B}^{4n} - \frac{64}{2^4}\cdot n^4\mathrm{S}_{(n+4)}\left(\frac{n}{2}\right)\mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 4\cdot n^4\mathrm{S}_{(n+4)}\left(\frac{n}{2}\right)\mathrm{B}^{4(n-1)}.$$

Z věty 12 máme

$$\mathrm{S}_n\left(\frac{n}{2}\right) \leq 2^{1{,}54725n}$$

a

$$\mathrm{S}_{(n+4)}\left(\frac{n}{2}\right) \leq 2^{1{,}54725(n+4)} \leq 73 \cdot 2^{1{,}54725n}.$$

Dále

$$2^{1{,}54725n+4} = 73 \cdot \mathrm{B}^{4-c_1(\beta)},$$

tedy počet vstupů je alespoň

$$\mathrm{B}^{4n} - 4 \cdot 73 \cdot n^4 \mathrm{B}^{4n-c_1(\beta)}.$$

□

O případu, kdy vektor **e** není fixní, vypovídá následující věta.

Věta 15. *Nechť $n \in \mathbb{N}, n \geq 8$. Mějme mez $B = 2^{\beta n}$, $\beta > 2.54725$. Potom počet vektorů*

$$\begin{aligned} a_1 &= \left(a_{11}^{(1)}, a_{11}^{(2)}, \ldots, a_{11}^{(n)}\right), \\ a_2 &= \left(a_{12}^{(1)}, a_{12}^{(2)}, \ldots, a_{12}^{(n)}\right), \\ a_3 &= \left(a_{21}^{(1)}, a_{21}^{(2)}, \ldots, a_{21}^{(n)}\right), \\ a_4 &= \left(a_{22}^{(1)}, a_{22}^{(2)}, \ldots, a_{22}^{(n)}\right), \end{aligned}$$

kde $1 \leq a_H^i \leq B$, $1 \leq i \leq n$, $H \in \{11, 12, 21, 22\}$, pro které je vektor $\boldsymbol{e}$ splňující

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2} \tag{3.40}$$

nejkratší nenulový vektor ve vstupu $\boldsymbol{L}(\boldsymbol{A}, C)$, kde $C = \sum_{i=1}^{n} e_i A_i$ je alespoň

$$B^{4n} - 4 \cdot 73 \cdot n^4 B^{4n-c_2(\beta)},$$

kde $c_2(\beta) = 4 - \frac{2.54725}{\beta} > 0$.

Důkaz. Odhad z věty 13 sečteme přes všechny vektory **e**, které splňují (3.40) (nejvýše jich je 2^n) a pro $\mathrm{R} = \frac{n}{2}, \mathrm{R} \geq 4$ dostaneme

$$\mathrm{B}^{4n} - \frac{64}{2^4} \cdot n^4 2^n \mathrm{S}_{(n+4)}\left(\frac{n}{2}\right) \mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 4 \cdot n^4 2^n \mathrm{S}_{(n+4)}\left(\frac{n}{2}\right) \mathrm{B}^{4(n-1)}.$$

Dále máme

$$2^n \mathrm{S}_{(n+4)}\left(\frac{n}{2}\right) \leq 73 \cdot 2^{2.54725n} = 73 \cdot \mathrm{B}^{4-c_2(\beta)},$$

tedy dostaneme odhad

$$\mathrm{B}^{4n} - 4 \cdot 73 \cdot n^4 \mathrm{B}^{4n-c_2(\beta)}.$$

□

Věta 16. *Nechť $n \in \mathbb{N}, n \geq 10$ a nechť $B \geq 2^{(2+\beta)n^2}$ pro $\beta > 0$. Potom počet vektorů*

$$\begin{aligned} a_1 &= \left(a_{11}^{(1)}, a_{11}^{(2)}, \ldots, a_{11}^{(n)}\right), \\ a_2 &= \left(a_{12}^{(1)}, a_{12}^{(2)}, \ldots, a_{12}^{(n)}\right), \\ a_3 &= \left(a_{21}^{(1)}, a_{21}^{(2)}, \ldots, a_{21}^{(n)}\right), \\ a_4 &= \left(a_{22}^{(1)}, a_{22}^{(2)}, \ldots, a_{22}^{(n)}\right), \end{aligned}$$

pro které algoritmus KV uspěje pro všechny vektory $\boldsymbol{e}$ je alespoň

$$B^{4n} - B^{4n-c_3(\beta)-3+\log n/n},$$

kde $c_3(\beta) = \frac{\beta}{2+\beta} > 0$.

Důkaz. Ze vztahu (3.21) bez újmy na obecnosti předpokládáme, že

$$1 \leq \sum_{i=1}^{n} e_i \leq \frac{n}{2}. \tag{3.41}$$

Odtud pro $\|\mathbf{e}\|$ máme

$$\begin{aligned} \|\mathbf{e}\| = \sqrt{\sum_{i=1}^{n+4} e_i^2} &\leq \sqrt{\frac{n}{2}}, \\ 2^n \|\mathbf{e}\|^2 &\leq n 2^{n-1}. \end{aligned} \tag{3.42}$$

Nejprve předpokládejme, že pro vstup $\mathbf{L}(\mathbf{A},C)$ platí, že pro každý vektor $w \in \mathbf{L}(\mathbf{A},C)$ takový, že $w \neq k\mathbf{e}, k \in \mathbb{Z}$ platí ze vztahu (3.42)

$$\|w\|^2 > n 2^{n-1} \geq 2^n \|\mathbf{e}\|^2.$$[2]

Z lemma 3 pro redukovanou bázi $b_1^*, b_2^*, \ldots, b_{n+1}^*$ vstupu $\mathbf{L}(\mathbf{A},C)$ plyne, že nějaký vektor $k\mathbf{e}$ musí být v redukované bázi $\mathbf{L}(\mathbf{A},C)$, a tedy algoritmus v tomto případě uspěje. V případech, kdy toto nenastane, pak z věty 13 pro $\mathrm{R} = n2^{n-1}, \mathrm{R} \geq 4$ sečteno přes všechna $\mathbf{e}$ splňující (3.41) dostáváme odhad

$$\mathrm{B}^{4n} - 64 \cdot n^4 2^{4(n-1)} 2^n \mathrm{S}_{(n+4)}\left(n2^{n-1}\right) \mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 4 \cdot n^4 2^{5n} \mathrm{S}_{(n+4)}\left(n2^{n-1}\right) \mathrm{B}^{4(n-1)}. \tag{3.43}$$

Pro $n \geq 3$ platí

$$\mathrm{S}_{(n+4)}(\mathrm{R}) \leq \left(2\sqrt{\mathrm{R}} + 1\right)^n \leq 3^n \mathrm{R}^{\frac{n}{2}}.$$

Dále pro $n \geq 10$ (pak je $2^{\frac{n^2}{2}} \geq 2^{5n}$ a $2^{\frac{n^2}{2}} \geq 3^n$) máme

$$\begin{aligned} &\mathrm{B}^{4n} - 4 \cdot n^4 2^{5n} 3^n n^{\frac{n}{2}} 2^{(n-1)\frac{n}{2}} \mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 2^{\frac{n^2}{2}+5n-\frac{n}{2}+2} 3^n n^{\frac{n}{2}+4} \mathrm{B}^{4(n-1)} \geq \\ &\geq \mathrm{B}^{4n} - 2^{4\frac{n^2}{2}} n^{2\frac{n}{2}} \mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 2^{2n^2} n^n \mathrm{B}^{4(n-1)}. \end{aligned}$$

[2] Autoři v [35] v tomto kroku uvažují počet bázových vektorů jako n místo očekávaných $n+1$, tedy dostávají odhad $2^n\|\mathbf{e}\|$ místo odhadu $2^{n-1}\|\mathbf{e}\|$ viz [35, Věta 3.5].

Dále

$$\mathrm{B}^{4n} - 2^{2n^2} n^n \mathrm{B}^{4(n-1)} = \mathrm{B}^{4n} - 2^{2n^2 + n(\log_2 n)} \mathrm{B}^{4(n-1)}.$$

Pro $\mathrm{B} \geq 2^{2n^2+\beta n^2}$ a $c_3(\beta) = \frac{\beta}{2+\beta}$ máme

$$\mathrm{B}^{\left(1-c_3(\beta)+\frac{\log_2 n}{n}\right)} \geq 2^{\left(2n^2+\beta n^2\right)\left(1-\frac{\beta}{2+\beta}+\frac{\log_2 n}{n}\right)} =$$

$$= 2^{2n^2+\beta n^2-\frac{2n^2\beta}{2+\beta}-\frac{\beta^2 n^2}{2+\beta}+n\log_2 n(2+\beta)} = 2^{2n^2+n\log_2 n(2+\beta)} \geq 2^{2n^2+(n\log_2 n)},$$

tedy

$$2^{2n^2+n(\log_2 n)} \leq \mathrm{B}^{\left(1-c_3(\beta)+\frac{\log_2 n}{n}\right)}$$

a

$$\mathrm{B}^{4n} - 2^{2n^2} n^n \mathrm{B}^{4(n-1)} \geq \mathrm{B}^{4n} - \mathrm{B}^{4n-c_3(\beta)-3+\log_2 n/n}.$$

□

V předchozím postupu by bylo možné využít jiný algoritmus pro výpočet redukované báze mřížky a tím dosáhnout lepších odhadů. Podrobnější diskuzi je možno nalézt v [35, Část 3].

Diskuze. V roce 1992 ukázal M. J. Coster a spol. [12], že odhad 0,646 z [35] lze vylepšit na odhad 0,9408 volbou posledního bázového vektoru b_{n+1} v (3.1) jako

$$b_{n+1} = \left(\frac{1}{2}, \ldots, \frac{1}{2}, s\right).$$

Tato modifikace může být požita i v našem případě, kdy zvolíme poslední bázový vektor b_{n+1} v (3.20) jako

$$b_{n+1} = \left(\frac{1}{2}, \ldots, \frac{1}{2}, c_{11}, c_{12}, c_{21}, c_{22}\right).$$

Obdobným postupem jako byl popsán výše, avšak s tímto posledním bázovým vektorem, dojdeme k odhadu 0,235.

4. Moderní kryptosystémy založené na problému batohu

Od doby prvního Merkleova-Hellmanova kryptosystému založeném na problému batohu v roce 1978 vznikla celá řada kryptosystémů, které staví na problému batohu (průzkum je možno nalézt v [44] nebo [8]). Ač byla naprostá většina těchto kryptosystémů prolomena, zájmem o problém batohu z hlediska kryptografie přetrvává. Důvodem je především rychlé šifrování a dešifrování a případná odolnost vůči útokům pomocí kvantového počítače, kterou nabízí $\mathcal{NP}$ úplnost problému batohu.

V této kapitole uvedeme souhrn řady kryptosystémů postavených na různých variantách problému batohu, které vznikaly v letech 2000-2017. I u těchto kryptosystémů platí, že naprostá většina z nich byla prolomena.

První kryptosystém, který uvedeme, byl představen v roce 2009 [23]. Toto schéma je postaveno na problému kvadratického batohu ve snaze vyhnout se útokům založeným na redukci báze mřížky. Následně uvidíme, že je možné napadnout tento kryptosystém heuristickým útokem [62] využívajícím stereotypních otevřených textů nebo heuristickým útokem [37] využívajícím omezení, která jsou kladena na parametry kryptosystémů.

V další části popíšeme kryptosystém, uvedený v práci [28] v roce 2011, který v konstrukci nepoužívá rychle rostoucí posloupnost a šifruje zprávy pravděpodobnostním šifrováním.

Poslední kryptosystém, který podrobněji popíšeme, byl představen v roce 2008 [57]. Konstrukce je postavena na lineárně posunutém problému batohu a pracuje nad grupou bodů eliptické křivky.

V další části stručně představíme další kryptosystémy a v závěru kapitoly uvedeme několik málo kryptosystémů založených na problému batohu, pro které doposud nebyla nalezena úspěšná kryptoanalýza.

4.1 Příprava

V této části uvedeme ty varianty problému batohu, které budeme dále potřebovat a základní značení z teorie eliptických křivek.

4.2 Problém batohu a jeho varianty

Nechť $n \in \mathbb{N}$ a nechť je dána n-tice $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslo $s \in \mathbb{N}$. Pak *problém 0-1 batohu* hledá vektor $(x_1, x_2 \ldots, x_n) \in \{0,1\}^n$ takový, že

$$\sum_{i=1}^{n} x_i a_i = s.$$

Problém kompaktního batohu hledá vektor $(x_1, x_2 \ldots, x_n)$, kde $0 \leq x_i \leq 2^b - 1$, pro mez $b \in \mathbb{N}, b \geq 1$ takový, že

$$\sum_{i=1}^{n} x_i a_i = s.$$

Problém pokrytí batohu maticí hledá vektor $X = (x_1, x_2 \ldots, x_n) \in \{0,1\}^n$ takový, že

$$s = XAX^\top,$$

kde $A \in \mathbb{N}^{n \times n}$ je n-dimenzionální čtvercová matice. V případě, že je matice A diagonální, pak se jedná o *problém kvadratického batohu*, který hledá vektor $(x_1, x_2 \ldots, x_n) \in \{0,1\}^n$ takový, že

$$\sum_{i=1}^{n} x_i^2 a_i = s.$$

Eliptické křivky. Algebraické struktury, které jsou vhodné k použití při tvoření kryptografických aparátů by měly obsahovat operaci, která je pro její prvky lehce počítatelná a nějaký problém, který je naopak obecně těžce řešitelný. Takovým příkladem vhodné algebraické struktury jsou *eliptické křivky*. Konkrétně se jedná o grupu bodů eliptické křivky společně s operací sčítání + a bodem v nekonečnu ∞. V této algebraické struktuře je jednoduché spočítat součet dvou bodů, avšak pro dané body P a Q je obecně obtížné spočítat číslo m takové, že $Q = mP$. Tomuto problému se říká *problém diskrétního logaritmu nad eliptickými křivkami*.

V dalším textu budeme uvažovat *eliptické křivky nad konečným tělesem* $\mathbb{F}_q$, kde $q > 3$ je prvočíslo. Pro čísla $a, b \in \mathbb{Z}^+, a, b < q$ je eliptická křivka nad konečným tělesem $\mathrm{E}(\mathbb{F}_q)$ definovaná jako

$$\mathrm{E}(\mathbb{F}_q) = \Big\{[x,y] \ : \ y^2 = x^3 + ax + b \bmod q, 4a^3 + 27b^2 \not\equiv 0 \bmod q\Big\} \cup \{\infty\}. \quad (4.1)$$

Na eliptické křivce $\mathrm{E}(\mathbb{F}_q)$ můžeme nadefinovat operaci + společně s bodem ∞ následovně. Pro dva body $P = [x_1, y_1], Q = [x_2, y_2] \in \mathrm{E}(F_q)$ platí:

- $P + \infty = \infty + P = P$.
- Pokud $Q = [x_1, -y_1]$, pak $P + Q = \infty$.
- Pokud $P \neq Q$, potom součet $P + Q = [x_3, y_3]$ je dán jako

$$\begin{aligned} x_3 &\equiv \lambda^2 - x_1 - x_2 \mod q, \\ y_3 &\equiv \lambda(x_1 - x_3) - y_1 \mod q, \end{aligned}$$

  kde

$$\lambda = \begin{cases} (y_2 - y_1)/(x_2 - x_1) & \text{pro všechna } x_2 \neq x_1, \\ (3x_1^2 + a)/(2y_1) & \text{pro všechna } x_1 = x_2, y_1 \neq 0. \end{cases}$$

Potom $(\mathrm{E}(\mathbb{F}_q), +)$ je grupa. Například $3P$ můžeme spočítat jako

$$3P = 2P + P = (P + P) + P.$$

4.3 Kryptosystém založený na problému kvadratického batohu

4.3.1 Úvod

Baocang Wang a Yupu Hu představili v roce 2009 [23] novou koncepci lehkého problému batohu a to *simultánní problém kvadratického batohu*. Tento problém, který je řešitelný v polynomiálním čase, použili pro vytvoření soukromého klíče a pomocí Čínské věty o zbytcích získali těžký problém batohu, který pak zastává funkci veřejného klíče. Díky konstrukci popsané níže uvidíme, že takto vytvořený kryptosystém má šifrovací funkci, která není lineární v otevřeném textu, a který má velkou hustotu, čímž se stává odolným vůči přímému použití základních útoků na kryptosystémy postavené na problému batohu. Ač autoři v [23] předpokládali, že tento kryptosystém bude odolný vůči přímému použití běžných útoků, v roce 2011 představil Moon Sung Lee úspěšný útok, který je založený na stereotypních zprávách [62] a v tom samém roce pak Amr M. Youssef představil další úspěšný útok, který využívá omezení, která jsou kladena na parametry kryptosystému [37]. Tyto dvě práce ukazují, že kryptosystém bezpečný není.

V první části popíšeme postup pro vytvoření lehkého simultánního kvadratického batohu, postup pro generování soukromého a veřejného klíče, šifrovací funkci a dešifrovací funkci a následně představíme oba dva útoky z [62] a [37].

4.3.2 Popis kryptosystému

V následující části popíšeme postup pro vytvoření simultánního kvadratického batohu a postup pro vytvořený kryptosystému postaveném na problému simultánního kvadratického batohu jako v [23].

Simultánní problém kvadratického batohu. Mějme množiny $I \subset \mathbb{Z}$ a $J = \{j = (j_1, j_2), j_1 j_2 \in \mathbb{Z}^+\}$. Množina J^T značí množinu $\{(j_2, j_1), (j_1, j_2) \in J\}$. Pro $j = (j_1, j_2) \in J$ máme $I \bmod j = \{(i \bmod j_1, i \bmod j_2), i \in I\}$. Řekneme, že množina I je *rozlišitelná* množinou J, pokud $\forall j \in J$ platí, že $|I \bmod J| = |I|$.

Pro množinu W, která obsahuje dvojice (1, 31),(1,34),(1,37),(1,38),(1,41), (1,43),(1,46),(1,47),(1,53),(1,58),(1,59),(1,61),(1,62),(1,67),(1,68),(1,71),(1,73), (1,74),(1,76),(1,78),(1,79),(1,82),(1,83),(1,86),(1,87),(1,89),(1,92),(1,93),(1,94), (1,97),(2,17),(2,19),(2,23),(2,29),(2,31),(2,34),(2,37),(2,38),(2,39),(2,41),(2,43), (2,46),(2,47),(3,26),(3,29), (3,31),(4,17),(4,19),(4,23),(6,13) definovali autoři [23] množiny I a J jako

$$I = \{i^2, i = 0, 1, \ldots, 15\}, \tag{4.2}$$

$$J = W \cup W^T. \tag{4.3}$$

Platí, že I je rozlišitelná množinou J.

Simultánní problém kvadratického batohu je dán následující větou.

Věta 17. *Nechť jsou dány dvě posloupnosti* $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$

a $(b_1, b_2, \ldots, b_n) \in \mathbb{N}^n$ *a dvě čísla* $s_1 \in \mathbb{N}$, $s_2 \in \mathbb{N}$. *Dále nechť*

$$\begin{aligned} c_i &= NSD(a_n, a_{n-1}, \ldots, a_{n-i+1}) \quad \textit{pro všechna } 1 \leq i \leq n-1, \\ d_i &= NSD(b_n, b_{n-1}, \ldots, b_{n-i+1}) \quad \textit{pro všechna } 1 \leq i \leq n-1, \\ c_n &= d_n = 1. \end{aligned}$$

Nechť $G = \{g_i = (g_{1i}, g_{2i}) = (c_{i-1}/c_i, d_{i-1}/d_i), i = 2, \ldots, n\}$. Simultánní problém kvadratického batohu *je dán vztahy*

$$\sum_{i=1}^{n} x_i^2 a_i = s_1, \qquad \sum_{i=1}^{n} x_i^2 b_i = s_2, \quad x_i^2 \in I. \tag{4.4}$$

Pokud $G \subset J$*, pak může být simultánní problém kvadratického batohu vyřešen v polynomiálním čase vzhledem k* n*. Dále existuje nejvýše jedno řešení* $(x_1, x_2 \ldots, x_n)$ *takové, že* $x_i \in \{0, 1, \ldots, 15\}$.

Důkaz předchozí věty najdeme v [23, Věta 1]. Postup pro řešení (4.4) nalezneme jako [23, Algoritmus 1].

Posloupnosti $(a_1, a_2, \ldots, a_n)$ a $(b_1, b_2, \ldots, b_n)$, které splňují předpoklady věty 17 vygenerujeme následujícím postupem:

1. Náhodně zvolíme $n-1$ různých dvojic $g'_j = \left(g'_{1j}, g'_{2j}\right) \in J$, kde $2 \leq j \leq n$.
2. Náhodně zvolíme $2(n-1)$ přirozených čísel $s_1, s_2, \ldots, s_{n-1}$ a $t_1, t_2, \ldots, t_{n-1}$, která splňují
   (a) $\mathrm{NSD}(s_i, g'_{1j}) = 1$ pro všechna $1 \leq i \leq n-1$, $2 \leq j \leq n$,
   (b) $\mathrm{NSD}(t_i, g'_{2j}) = 1$ pro všechna $1 \leq i \leq n-1$, $2 \leq j \leq n$,
   (c) $\mathrm{NSD}(s_i, s_{i+1}) = 1$ pro všechna $1 \leq i \leq n-2$,
   (d) $\mathrm{NSD}(t_i, t_{i+1}) = 1$ pro všechna $1 \leq i \leq n-2$.
3. Položíme $s_n = t_n = 1$.
4. Položíme $a_1 = s_1$, $b_1 = t_1$ a spočítáme
$$a_i = s_i \prod_{j=n-i+2}^{n} g'_{1j}, \qquad b_i = t_i \prod_{j=n-i+2}^{n} g'_{2j}, \quad i = 2, 3, \ldots, n.$$
5. Získáme posloupnosti $(a_1, a_2, \ldots, a_n)$ a $(b_1, b_2, \ldots, b_n)$.

Důkaz toho, že posloupnosti $(a_1, a_2, \ldots, a_n)$ a $(b_1, b_2, \ldots, b_n)$ vytvořené postupem výše splňují předpoklady věty 17, je možno nalézt v [23, Věta 2].

Generování soukromého a veřejného klíče. Generování soukromého a veřejného klíče probíhá následujícím postupem:

1. Zvolíme dvě náhodné posloupnosti $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a $(b_1, b_2, \ldots, b_n) \in \mathbb{N}^n$, které splňují větu 17.
2. Zvolíme náhodnou matici $C \in \mathrm{GL}(2,\mathbb{Z})$ s kladnými koeficienty, pro jejíž prvky c_{ij} platí $|c_{ij}|_2 \leq k$, pro $k \in \mathbb{N}$, kde $|\star|_2$ značí binární délku $\star$.

3. Spočítáme
$$C \cdot \begin{pmatrix} a_1 & \dots & a_n \\ b_1 & \dots & b_n \end{pmatrix} = \begin{pmatrix} \hat{a}_1 & \dots & \hat{a}_n \\ \hat{b}_1 & \dots & \hat{b}_n \end{pmatrix}$$

4. Náhodně zvolíme dvě prvočísla p a q taková, že $p > 225 \sum_{i=1}^{n} \hat{a}_i$ a $q > 225 \sum_{i=1}^{n} \hat{b}_i$ a spočítáme modulus $N = pq$.

5. Pomocí Čínské věty o zbytcích najdeme vektor $(e_1, e_2, \dots, e_n)$, který splňuje
$$e_i \equiv \hat{a}_i \mod p, \qquad e_i \equiv \hat{b}_i \mod q.$$

6. Náhodně zvolíme $v \in \mathbb{Z}_N^*$.

7. Spočítáme vektor $(f_1, f_2, \dots, f_n)$ jako
$$f_i \equiv e_i v \mod N.$$

Soukromý a veřejný klíč je určen následovně:

$$\mathsf{sk} = N, p, q, C, v;$$
$$\mathsf{pk} = (f_1, f_2, \dots, f_n).$$

Složky veřejného klíče je možné propermutovat.

Zašifrování zprávy. Zpráva $m = (m_1, m_2, \dots, m_n) \in \{0, 1, \dots, 15\}^n$ je zašifrována jako
$$c = \sum_{i=1}^{n} m_i^2 f_i.$$

Dešifrování zprávy. Šifrový text c dešifrujeme následujícím postupem.

1. Označíme $t \equiv cv^{-1} \equiv \sum_{i=1}^{n} m_i^2 f_i v^{-1} \equiv \sum_{i=1}^{n} m_i^2 e_i \mod N$.

2. Spočítáme $t_p \equiv t \mod p$, $t_q \equiv t \mod q$

3. Spočítáme $(s_A, s_B)^\top = C^{-1} (t_p, t_q)^\top$.

4. Postupem v [23, Algoritmus 1] vyřešíme simultánní problém kvadratického batohu daný vztahy
$$s_A = \sum_{i=1}^{n} m_i^2 a_i, \quad s_B = \sum_{i=1}^{n} m_i^2 b_i, \quad m_i^2 \in I$$
a obdržíme původní zprávu $m = (m_1, m_2, \dots, m_n)$.

Celkem je výpočetní složitost celého kryptosystému $\mathcal{O}(n^2)$ (viz [23]).

Parametry kryptosystému. Baocang Wang a Yupu Hu uvedli doporučená rozmezí a hodnoty parametrů kryptosystému v [23]. Konkrétně doporučili hodnotu $n = 100$, pro které má veřejný klíč pk přibližně 6157 bitů. Prvky matice C náleží do $\mathcal{O}(1)$, konkrétně se jedná o volbu

$$C_1 = \begin{pmatrix} 2 & 1 \\ 3 & 2 \end{pmatrix} \quad \text{nebo} \quad C_2 = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}.$$

Proto, aby čísla $a_1, a_2, \ldots, a_n$ měla téměř stejnou binární délku a aby čísla $b_1, b_2, \ldots, b_n$ měla téměř stejnou binární délku je potřeba zvolit s_i a t_i, pro $1 \leq i \leq n-1$ tak, aby

$$|s_i|_2 = \left| \prod_{j=2}^{n} g'_{1j} \right|_2 - \left| \prod_{j=n-i+2}^{n} g'_{1j} \right|_2,$$
$$|t_i|_2 = \left| \prod_{j=2}^{n} g'_{2j} \right|_2 - \left| \prod_{j=n-i+2}^{n} g'_{2j} \right|_2.$$

Potom

$$|a_1|_2 \approx |a_2|_2 \approx \ldots \approx |a_n|_2 = \left| \prod_{j=2}^{n} g'_{1j} \right|_2,$$
$$|b_1|_2 \approx |b_2|_2 \approx \ldots \approx |b_n|_2 = \left| \prod_{j=2}^{n} g'_{2j} \right|_2$$

a s_i a t_i mají binární délku danou

$$\begin{aligned} |s_i|_2 &= \left| \prod_{j=2}^{n-i+1} g'_{1j} \right|_2 + \epsilon, \\ |t_i|_2 &= \left| \prod_{j=2}^{n-i+1} g'_{2j} \right|_2 + \epsilon', \text{ kde } \epsilon, \epsilon' \in \{0,1\}. \end{aligned} \tag{4.5}$$

Později uvidíme, že tato omezení na parametry kryptosystému povedou k úspěšnému útoku z [37].

4.3.3 Kryptoanalýza systému

V případě, že bychom chtěli odhalit zprávu m pomocí útoku hrubou silou, potřebovali bychom provést přibližně $n16^{n/2}$ operací. Tento typ útoku není efektivní v případě, že bude parametr n dostatečně velký.

Mnoho kryptosystémů založených na problému batohu je prolomeno útoky, které využívají algoritmy pro hledání redukované báze mřížky (například LLL algoritmus viz [35]). Tyto algoritmy hledají krátké řešení lineární rovnice vzhledem k Euklidově normě. Výhodou kryptosystému založeném na simultánním problému kvadratického batohu je, že šifrovací funkce není lineární vzhledem k zprávě m, tedy není možné přímo použít útok pomocí redukce báze mřížky. Pokud by útočník v rovnici (4.3.2) použil substituci $y_i = m_i^2$ pro všechna $1 \leq i \leq n$, pak by výsledná rovnice

$$c = \sum_{i=1}^{n} y_i f_i, \quad \text{pro všechna } y_i \in \mathbb{Z}_{226}, 1 \leq i \leq n$$

byla lineární, avšak ani v tomto případě nenajdeme algoritmem pro výpočet redukované báze požadované řešení. Algoritmy, které hledají redukovanou bázi mřížky pracují na tom principu, že jsme schopni najít polynomiálně mnoho koulí s „malým“ poloměrem, do kterých můžeme uzavřít body mřížky. V případně rovnice (4.3.3) však v daných koulích leží exponenciálně mnoho řešení vzhledem k n a řešení, které odpovídá otevřenému textu, nemusí být nejkratší vzhledem k Euklidově normě, jelikož nepracujeme s binárním zprávou m, ale $m \in \{0, 1, \ldots, 15\}^n$. Navíc požadavek, který na otevřený text klademe, je aby y_i byl čtverec pro všechna $i \in \{1, 2, \ldots, n\}$, a to je požadavek, který pomocí redukce báze mřížky neumíme vystihnout. Můžeme tedy předpokládat, že algoritmus založený na redukci báze mřížky nalezne náhodný vektor, který řeší rovnice (4.3.3). Abychom našli správné řešení, museli bychom podle [23] prohledat pro $n = 100$ přibližně 2^{151} možných řešení, a to není proveditelné v reálném čase. Předpoklad, že by kryptosystém mohl být odolný vůči algoritmům, které hledají redukovanou bázi mřížky, je podpořen také tím, že tento kryptosystém má hustotu přibližně 1,3 [23] a zatím je ukázáno viz [12], že tyto algoritmy jsou úspěšné u kryptosystémů, které mají hustotu $d \leq 0{,}9408$. Můžeme se tedy domnívat, že tento kryptosystém bude odolný vůči přímé aplikaci útoků založených na redukci báze mřížky. Důkaz tohoto tvrzení však Baocang Wu a Yupu Hu v [23] neuvedli.

V případě, že by útočník znal modulus N, pak by mohl faktorizovat N (což by bylo možné vzhledem k tomu, že N má přibližně 616 bitů), najít hodnotu v^{-1} mod N a následně odhalit celý soukromý klíč.

Ač se kryptosystém v [23] zdá být odolný vůči přímému použití známých útoků na kryptosystémy založené na problému batohu, Amr M. Youssef představil úspěšný heuristický útok pomocí stereotypních zpráv v roce 2011 [62]. Dále Moon Sung Lee představil v roce 2011 úspěšný útok, který využívá omezení, která jsou kladena na parametry kryptosystému [37]. Oba dva tyto útoky si v následující části představíme.

Heuristický útok založený na stereotypních zprávách. Útoky na kryptosystémy, které využívají stereotypních otevřených textů, jsou postavené na tom, že útočník zná nějakou část otevřeného textu. Toto se může stát například v situaci, kdy útočník ví, že každá zpráva bude formátu „Vaše uživatelské jméno je ***** a Vaše heslo je *****“. Amr M. Youssef v [62] ukázal, že pro $n = 100$ je možné s 90-ti procentní pravděpodobnostní odhalit celý otevřený text do dvou hodin, pokud je 60 % otevřeného textu známo útočníkovi. Tento útok je založen na dvojím použití algoritmu pro redukci báze mřížky.

Předpokládejme, že z otevřeného textu m známe $l \in \{1, 2, \ldots, n\}$ složek $m' = (m_{i_1}, m_{i_2}, \ldots, m_{i_l})$. Zpráva m je zašifrovaná pomocí veřejného klíče $\mathsf{pk} = (f_1, f_2, \ldots, f_n)$ stejně jako v (4.3.2) na šifrový text

$$c = \sum_{i=1}^{n} m_i^2 f_i \tag{4.6}$$

Vektory

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, 0, -\lambda f_1)\,, \\ b_2 &= (0, 1, \ldots, 0, 0, -\lambda f_2)\,, \\ &\vdots \\ b_n &= (0, 0, \ldots, 1, 0, -\lambda f_n)\,, \\ b_{n+1} &= (0, 0, \ldots, 0, 1, \lambda c) \end{aligned}$$

tvoří bázi mřížky $\mathcal{L}$.

Dostatečně velkou volbou konstanty $\lambda \in \mathbb{N}$ [49] můžeme zaručit, že redukovaná báze $b_1^*, b_2^*, \ldots, b_{n+1}^*$ bude tvaru

$$\begin{aligned} b_1^* &= (a_{11}, a_{12}, \ldots, a_{1n}, 0)\,, \\ b_2^* &= (a_{21}, a_{22}, \ldots, a_{2n}, 0)\,, \\ &\vdots \\ b_n^* &= (a_{n1}, a_{n2}, \ldots, a_{nn}, 0)\,, \\ b_{n+1}^* &= (w_1, w_2, \ldots, w_n, \lambda)\,. \end{aligned}$$

Na prvky a_{ij}, $i, j \in \{1, 2, \ldots, n\}$ budeme nahlížet jako na matici A a její řádky označíme $a_1, a_2, \ldots, a_n$, tedy

$$A = \begin{pmatrix} a_{11} & a_{12} & \ldots & a_{1n} \\ a_{21} & a_{22} & \ldots & a_{2n} \\ & & \vdots & \\ a_{n1} & a_{n2} & \ldots & a_{nn} \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \end{pmatrix}.$$

Ze struktury bázových vektorů $b_1, b_2, \ldots, b_{n+1}$ a $b_1^*, b_2^*, \ldots, b_{n+1}^*$ víme, že existuje celočíselná lineární kombinace řádků matice A, která se rovná $(m_1^2, m_2^2, \ldots, m_n^2)$. Konkrétně z toho, že vektory redukované báze jsou krátké a téměř ortogonální a vektor $(m_1^2, m_2^2, \ldots, m_n^2)$ je také poměrně krátký, víme, že existuje krátký celočíselný vektor $(s_1, s_2, \ldots, s_n) \in \mathbb{Z}^n$ takový, že

$$(m_1^2, m_2^2, \ldots, m_n^2) = (s_1, s_2, \ldots, s_n)A.$$

Tento vektor můžeme nalézt druhým aplikováním algoritmu pro redukci báze mřížky pro známé koeficienty bitů zprávy m.

Z matice A vybereme ty sloupce, které odpovídají koeficientům známých bitů zprávy a z těchto sloupců vytvoříme matici A', tedy $A'[i][j] = A[i][i_j]$, pro $1 \leq i \leq n, 1 \leq j \leq l$. Nyní jsou vektory

$$\begin{aligned} c_1 &= (1, 0, \ldots, 0, 0, -\kappa a'_{11}, -\kappa a'_{12}, \ldots, -\kappa a'_{1l})\,, \\ c_2 &= (0, 1, \ldots, 0, 0, -\kappa a'_{21}, -\kappa a'_{22}, \ldots, -\kappa a'_{2l})\,, \\ &\vdots \\ c_n &= (0, 0, \ldots, 1, 0, -\kappa a'_{n1}, -\kappa a'_{n2}, \ldots, -\kappa a'_{nl})\,, \\ c_{n+1} &= \left(0, 0, \ldots, 0, 1, \kappa m_{i_1}^2, \kappa m_{i_2}^2, \ldots, \kappa m_{i_l}^2\right) \end{aligned}$$

bázové vektory mřížky $\mathcal{L}_2$.

Můžeme volit konstantu $\kappa \in \mathbb{N}$ [49] tak velkou, aby matice C^*, jejíž řádky tvoří prvky redukované báze $c_1^*, c_2^*, \ldots, c_{n+1}^*$, byla tvaru

$$C^* = \begin{pmatrix} S_{(n+1-l)\times n} & \mathbf{0}_{(n+1-l)\times l} \\ Z_{l\times n} & -\kappa\Pi_{l\times l} \end{pmatrix},$$

kde Π je permutační matice.

Důležitý je nulový blok $\mathbf{0}_{(n+1-l)\times l}$ v pravém horním rohu matice C^*, který zaručuje, že se sloupce $i_1, \ldots, i_l$, kde indexy $i_1, \ldots, i_l$ odpovídají odhaleným bitům zprávy m, matice $S \cdot A$ rovnají $k \cdot m'$, pro $k \in \mathbb{Z}$. Najdeme takový řádek s matice $S \cdot A$, který obsahuje prvky dané vektorem $\pm m'$. Pro tento řádek s platí, že $s \cdot A = \pm(m_1^2, m_2^2, \ldots, m_n^2)$. Pokud takový řádek nalezen není, algoritmus skončí neúspěchem. Pravděpodobnost, že algoritmus skončí neúspěchem je tím menší, čím větší počet bitů zprávy je odhalen. V [62] autoři ukázali, že algoritmus uspěje již v případech, kdy je známo alespoň 60 % otevřeného textu.

Heuristický útok založený na restrikcích parametrů kryptosystému. Moon Sung Lee v roce 2011 představil [37] útok, který najde řešení v polynomiálním čase vzhledem k n na výše představený kryptosystém. V tomto útoku bylo využito toho, že v tajných posloupnostech $(a_1, a_2, \ldots, a_n)$ a $(b_1, b_2, \ldots, b_n)$ jsou prvky a_i a b_i vytvořeny jako součin prvků z nějaké malé množiny (viz (4.3.2)), tedy je možné najít vztahy mezi hodnotami veřejného klíče $(f_1, f_2, \ldots, f_n)$. Tyto vztahy vedou na systém lineárních rovnic s několika neznámými a s malými řešeními. Vyřešením tohoto systému získáme seznam kandidátů pro modulus N, pomocí kterého pak algoritmem pro redukci báze najdeme celý soukromý klíč sk.

Stěžejním pro daný útok je následující lemma, které vystihuje vztahy mezi prvky veřejného klíče $(f_1, f_2, \ldots, f_n)$.

Lemma 18. *Nechť $(f_1, f_2, \ldots, f_n)$ je veřejný klíč a nechť g'_{ij}, s_k a t_k, kde $1 \leq i \leq n, 2 \leq j \leq n$ a $1 \leq k \leq n$ jsou parametry kryptosystému generované algoritmem pro generování klíče v části 4.3.2. Potom existují celá čísla $W_i \approx 2s_{n-3}t_{n-3}$ taková, že*

$$W_0 f_n + W_1 f_{n-1} + W_2 f_{n-2} + W_3 f_{n-3} = 0 \tag{4.7}$$

a navíc

$$W_0 g'_{12} g'_{13} g'_{14} + W_1 g'_{13} g'_{14} s_{n-1} + W_2 g'_{14} s_{n-2} + W_3 s_{n-3} = 0, \tag{4.8}$$
$$W_0 g'_{22} g'_{23} g'_{24} + W_1 g'_{23} g'_{24} t_{n-1} + W_2 g'_{24} t_{n-2} + W_3 t_{n-3} = 0. \tag{4.9}$$

Důkaz předchozího lemmatu je možno nalézt v [37, Lemma 1].

Samotný útok předpokládá znalost pouze veřejného klíče $(f_1, f_2, \ldots, f_n)$ a omezení, která jsou kladena na parametry kryptosystému. Útok zahrnuje i variantu, že jsou složky veřejného klíče permutovány.

K nalezení čísel W_0, W_1, W_2 a W_3, které splňují lemma 18, byl využit Blockwise-Korkine-Zolotarev (BKZ) algoritmus z [51] pro redukci báze mřížky s velikostí bloku 4 a parametrem $\delta = 0{,}99$. Nalezneme redukovanou bázi mřížky

$\mathcal{L}$, která má bázové vektory dané jako

$$\begin{aligned} b_1 &= (1,0,0,0,\lambda f_n)\,, \\ b_2 &= (0,1,0,0,\lambda f_{n-1})\,, \\ b_3 &= (0,0,1,0,\lambda f_{n-2})\,, \\ b_4 &= (0,0,0,1,\lambda f_{n-3})\,, \end{aligned}$$

kde $\lambda = 10^{10}$ je vhodně zvolená konstanta [49]. Označíme nejkratší vektor redukované báze jako b^*. Předpokládáme, že $b^* = (W_0, W_1, W_2, W_3, 0)$, a pokud další postup nepovede ke správnému soukromému klíči pro žádnou z 4! permutací vektoru b^*, pak vezmeme vektor s druhou nejkratší normou v redukované bázi jako $(W_0, W_1, W_2, W_3, 0)$.

Jelikož neznáme permutaci veřejného klíče, je výše uvedený postup potřeba provést pro všech $\binom{n}{4} \approx n^4$ možných kombinací prvků veřejného klíče pk.

Po tomto kroku předpokládáme, že známe hodnoty $f_n, f_{n-1}, f_{n-2}, f_{n-3}$, W_0, W_1, W_2 a W_3. V dalším kroku využijeme toho, že všechna $s_i, 1 \leq i \leq n$ mají binární délku danou (4.5) a nalezneme hodnoty splňující vztahy v (4.8) a (4.9).

Najdeme množinu

$$L = \{(g_2, g_3, g_4, u_1, u_2, u_3) : W_0 g_2 g_3 g_4 + W_1 g_3 g_4 u_1 + W_2 g_4 u_2 + W_3 u_3 = 0\},$$

kde $g_i \in K = \{g, (g, g') \in J\}, 0 < u_1 < U_1 = 2^{\|g_2\|_2+1}, 0 < u_2 < U_2 = 2^{\|g_2 g_3\|_2+1}$ a $0 < u_3 < U_3 = 2^{\|g_2 g_3 g_4\|_2+1}$, kde rozsah hodnot $u_i, 1 \leq i \leq 3$ je dán binární délkou s_{n-i} z (4.5).

Pro všechny možné kombinace $g_2, g_3, g_4 \in K$ a $u_1, 0 < u_1 < U_1$ najdeme hodnoty u_2 a u_3 jako řešení diofantické lineární rovnice

$$Au_2 + Bu_3 + C = 0 \quad \text{pro všechna } A = W_2 g_4, B = W_3, C = W_0 g_2 g_3 g_4 + W_1 g_3 g_4 u_1,$$

kde $0 < u_2 < U_2$ a $0 < u_3 < U_3$. Postup pro řešení takových rovnic využívá rozšířeného Euklidova algoritmu a je popsán v [37, odstavec 2.3]. Množinu L pak máme danou jako množinu šestic $(g_2, g_3, g_4, u_1, u_2, u_3)$. Z množiny L obdržíme kandidáty pro hodnoty g'_{ij}, s_{n-k} a t_{n-k} pro $i \in \{1,2\}, j \in \{2,3,4\}$ a $k \in \{1,2,3\}$.

Dále vytvoříme množinu L' jako

$$\begin{aligned} &L' = \\ &\{(l_1, l_2) = ((g_2, g_3, g_4, u_1, u_2, u_3),(g'_2,g'_3, g'_4, u'_1, u'_2, u'_3)) : l_1, l_2 \in L, (g_i, g'_i) \in J\}. \end{aligned}$$

Pomocí množiny L' určíme hodnoty pro modulus N a g'_{ij} tak, že pro každou dvojici $(l_1, l_2) \in L$ spočítáme kandidáta na N a následně vypočítáme hodnoty g'_{ij}, s_{n-k} a t_{n-k}, které budou splňovat podmínky dané (4.5). Špatný kandidát pro modulus N nepovede na hodnoty, které by splňovaly meze v (4.5). Se znalostí těchto hodnot je možné určit celý soukromý klíč sk.

V [37, Odstavec 5.2] je uvedeno, že výše uvedený útok má asymptotickou časovou složitost $\mathcal{O}(n^6)$.

Moon Sung Lee navrhl, aby byly prvky s_n a t_n voleny jako náhodná celá čísla $s_n \approx t_n \approx 2^n$, což by znemožnilo použití tohoto útoku.

4.4 Kryptosystém bez zadních vrátek s pravděpodobnostním šifrováním

4.4.1 Úvod

V předchozích kapitolách jsme viděli, že kryptosystémy založené na problému batohu ve většině případů používají lehký problém batohu vytvořený z rychle rostoucí posloupnosti ke konstrukci soukromého klíče, a ten pak transformují na těžký problém batohu, který slouží jako veřejný klíč. Doposud byla naprostá většina těchto kryptosystémů prolomena a jedna ze spekulací, proč tomu tak je, tvrdí, že je to způsobené nedostatečným ukrytím vlastností rychle rostoucí posloupnosti. Tomu předešli Yasuyuki Murakami a Masao Kasahara, kteří v [28] představili kryptosystém postavený na problému batohu, který nepoužívá ke konstrukci soukromého klíče rychle rostoucí posloupnost. Soukromý klíč je dán problémem batohu s takovými parametry, že je možné jej vyřešit v reálném čase například pomocí útoku hrubou silou. Také k vytvoření soukromého a veřejného klíče autoři využívají *pravděpodobnostního šifrování*, kdy se při šifrování ukryjí zprávu pomocí náhodných binárních sekvencí. Tento druh šifrování mimo jiné vede k tomu, že dvojí zašifrování té samé zprávy vyprodukuje dva různé šifrové texty.

4.4.2 Popis kryptosystému

Generování soukromého a veřejného klíče.

1. Zvolíme parametry k, n a u takové, že $n < u < n+k$ a $n \ll k$. Je doporučeno zvolit $n \leq 60$.
2. Náhodně vygenerujeme n čísel $s_1, s_2, \ldots, s_n$, $s_i \in \mathbb{Z}^+, |s_i|_2 = u$.
3. Náhodně vygenerujeme k čísel $t_1, t_2, \ldots, t_k$, $t_i \in \mathbb{Z}^+, |t_i|_2 = u$.
4. Vygenerujeme prvočíslo p takové, že $p > \sum_{i=1}^{n} s_i + \sum_{j=1}^{k} t_j > p/2$. Existence takového prvočísla p plyne z věty 1.
5. Vygenerujeme liché číslo v takové, že $0 < v < p$.
6. Vygenerujeme prvočíslo q takové, že $q > \sum_{j=1}^{k} t_j > q/2$. Existence takového prvočísla q plyne z věty 1.
7. Vygenerujeme liché číslo w takové, že $0 < w < q$.
8. Spočítáme hodnoty a_i, b_j, c_j následovně:

$$\begin{aligned} a_i &\equiv v s_i \mod p \quad \text{pro všechna } 1 \leq i \leq n, \\ b_j &\equiv v t_j \mod p \quad \text{pro všechna } 1 \leq j \leq k, \\ c_j &\equiv w t_j \mod q \quad \text{pro všechna } 1 \leq j \leq k. \end{aligned}$$

Soukromý a veřejný klíč jsou pak dány jako

$$\begin{aligned} \mathsf{sk} &= s_i, t_j, v, p, w, q \quad \text{pro všechna } 1 \leq i \leq n,\ 1 \leq j \leq k, \\ \mathsf{pk} &= a_i, b_j, c_j \quad \text{pro všechna } 1 \leq i \leq n,\ 1 \leq j \leq k. \end{aligned}$$

Šifrování zprávy. Při šifrování n bitové zprávy $m = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ vznikne dvojice šifrových textů (c_1, c_2) následovně:

$$c_1 = \sum_{i=1}^{n} m_i a_i + \sum_{j=1}^{k} r_j b_j, \tag{4.10}$$

$$c_2 = \sum_{j=1}^{k} r_j c_j, \tag{4.11}$$

kde $r = (r_1, r_2, \ldots, r_k)$ je posloupnost náhodných bitů. Vidíme, že se doopravdy jedná o pravděpodobnostní šifrování.

Dešifrování zprávy. Po obdržení dvojice šifrových textů (c_1, c_2) získáme původní zprávu m následujícím postupem. Spočítáme

$$m' \equiv v^{-1} c_1 \bmod p \equiv \sum_{i=1}^{n} s_i m_i + \sum_{j=1}^{k} t_j r_j,$$

$$n \equiv w^{-1} c_2 \bmod q \equiv \sum_{j=1}^{k} t_j r_j$$

a

$$m = m' - n = \sum_{i=1}^{n} s_i m_i. \tag{4.12}$$

Poslední rovnice (4.12) definuje problém batohu, jehož řešením je původní zpráva m.

V prvním kroku generování veřejného a soukromého klíče jsme viděli, že parametr n se zvolil tak, aby $n \leq 60$. Důvodem této volby je jistota, že problém batohu daný rovnicí (4.12) bude řešitelný. K tomu, abychom obdrželi původní zprávu m můžeme použít jakýkoliv z algoritmů pro řešení problému batohu. Příkladem může být použití hrubé síly (v tomto případě se však doporučuje parametr $n \leq 32$), algoritmy založené na space-time-tradeoff nebo útoky založené na algoritmech pro redukci báze mřížky.

4.4.3 Kryptoanalýza systému

Doposud nebyla nalezena žádná práce, která by snížila bezpečnost tohoto kryptosystému. Autoři v [28] uvádí bezpečnost proti přímému použití běžných kryptoanalytických aparátů jako je útok hrubou silou, útoky založené na space-time-tradeoff nebo útoky založené na algoritmech pro redukci báze mřížky.

Je zřejmé, že bezpečnost kryptosystému závisí na následujícím lemmatu.

Lemma 19. *Nechť $s_1, s_2, \ldots, s_n \in \mathbb{Z}^+$ jsou náhodná čísla. Nechť $a_i \equiv v s_i \bmod p$, $1 \leq i \leq n$, kde $p > \sum_{i=1}^{n} s_i$ a $v \in \mathbb{Z}^+$ je náhodné číslo takové, že $NSD(v, p) = 1$. Potom je posloupnost $a_1, a_2, \ldots, a_n$ nerozlišitelná od náhodné posloupnosti té samé délky.*

Ač je toto lemma stěžejní pro odvození bezpečnosti daného kryptosystému, autoři v [11] neuvedli důkaz.

Bezpečnost posloupnosti $s_1, s_2, \ldots, s_k$ je založena na problému, kdy pro dvě dané posloupnosti $b_1, b_2, \ldots, b_k$ a $c_1, c_2, \ldots, c_k$ takové, že

$$\begin{aligned} b_j &\equiv v s_j \mod p \quad \text{pro všechna } 1 \leq j \leq k \\ c_j &\equiv w s_j \mod q \quad \text{pro všechna } 1 \leq j \leq k, \end{aligned}$$

kde p a q jsou prvočísla a v a w jsou náhodná čísla, hledáme posloupnost $s_1, s_2, \ldots, s_k$. V [11] je toto ponecháno jako otevřený problém.

Nyní shrneme předpoklady o bezpečnosti kryptosystému, ke kterým Yasuyuki Murakami a Masao Kasahara došli v [11].

Pokud se zvolí parametr $k \geq 80$, pak z výpočetních důvodů nebude možné použít útok hrubou silou. Navíc, pokud se zvolí $k \geq 160$, pak nebude efektivní ani útok založený na space-time-tradeoff.

V případě použití útoků založených na redukci báze mřížky, můžeme na rovnice (4.10) a (4.11) pohlížet jako na dva oddělené problémy batohu, kde pro každý z nich sestavíme matici z bázových vektorů a spočítáme redukovanou bázi. Také na ně můžeme nahlížet jako na *dvojitý problém batohu*. Dvojitý problém batohu je definován tak, že se pro kladná celá čísla $a_1, a_2, \ldots, a_n$, $a'_1, a'_2, \ldots, a'_n, s_1$ a s_2 hledá vektor $x = (x_1, x_2 \ldots, x_n) \in \{0,1\}^n$ takový, že $s_1 = \sum_{i=1}^{n} x_i a_i$ a zároveň $s_2 = \sum_{i=1}^{n} x_i a'_i$.

Nejprve vytvoříme mřížky $\mathcal{L}_1$ a $\mathcal{L}_2$ pro každou z rovnic (4.10) a (4.11) zvlášť a potom pro obě dvě dohromady.

Pro rovnici (4.10) budeme mít bázové vektory $b_1, b_2, \ldots, b_{n+k+1} \in \mathbb{Z}^{n+k+1}$ mřížky $\mathcal{L}_1$ dané jako

$$\begin{aligned} b_1 &= (1, 0, \ldots, 0, -\lambda a_1)\,, \\ b_2 &= (0, 1, 0, \ldots, 0, -\lambda a_2)\,, \\ &\vdots \\ b_n &= (\overbrace{0, \ldots, 0}^{n-1}, 1, 0, \ldots, 0, -\lambda a_n), \\ b_{n+1} &= (\overbrace{0, \ldots, 0}^{n}, 1, 0, \ldots, 0, -\lambda b_1), \\ b_{n+2} &= (\overbrace{0, \ldots, 0}^{n+1}, 1, 0, \ldots, 0, -\lambda b_2), \\ b_{n+k} &= (0, \ldots, 0, 1, -\lambda b_k), \\ b_{n+k+1} &= (-1/2, \ldots, \ldots, \ldots, -1/2, \lambda c_1)\,, \end{aligned}$$

pro vhodně zvolenou konstantu $\lambda \in \mathbb{Z}$ takovou, že $\lambda > \sqrt{n}$. Hustota d_1 problému batohu zadaného rovnicí (4.10) může být dle [11] a [28] určena jako

$$d_1 = \frac{n+k}{u + \log_2(n+k)}.$$

Dále pro rovnici (4.11) budeme mít bázové vektory $b'_1, b'_2, \ldots, b'_{k+1} \in \mathbb{Z}^{k+1}$

mřížky $\mathcal{L}_2$ dané jako

$$\begin{aligned} b'_1 &= (1, 0, \ldots, 0, 0, -\lambda c_1)\,, \\ b'_2 &= (0, 1, \ldots, 0, 0, -\lambda c_2)\,, \\ &\vdots \\ b'_k &= (0, 0, \ldots, 0, 1, -\lambda c_k)\,, \\ b'_{k+1} &= (-1/2, \ldots, -1/2, \lambda s_2)\,, \end{aligned}$$

pro vhodně zvolenou konstantu $\lambda \in \mathbb{Z}$ takovou, že $\lambda > \sqrt{n}$. Hustotu d_2 můžeme opět dle [11] a [28] určit jako

$$d_2 = \frac{k}{u + \log_2 k}.$$

Pro rovnice (4.10) a (4.11) budeme mít bázové vektory $\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_{n+k+1} \in \mathbb{Z}^{n+k+1}$ mřížky $\mathcal{L}$ dané jako

$$\begin{aligned} \bar{b}_1 &= (1, 0, \ldots, 0, -\lambda a_1, 0)\,, \\ \bar{b}_2 &= (1, 0, \ldots, 0, -\lambda a_2, 0)\,, \\ &\vdots \\ \bar{b}_n &= (\overbrace{0, \ldots, 0}^{n-1}, 1, 0, \ldots, 0, -\lambda a_n, 0), \\ \bar{b}_{n+1} &= (\overbrace{0, \ldots, 0}^{n}, 1, 0, \ldots, 0, -\lambda b_1, -\lambda c_1), \\ &\vdots \\ \bar{b}_{n+k} &= (0, \ldots, 0, 1, -\lambda b_k, -\lambda c_k)\,, \\ \bar{b}_{n+k+1} &= (-1/2, \ldots, -1/2, \lambda s_1, \lambda s_2)\,, \end{aligned}$$

pro vhodně zvolenou konstantu $\lambda \in \mathbb{Z}$ takovou, že $\lambda > \sqrt{n}$. Hustotu D můžeme opět dle [11] a [28] určit jako

$$D = \frac{n + k}{2(u + \log_2(n + k))}.$$

Z [11] víme, že problém batohu je těžce řešitelný, pokud je jeho hustota $d > 1$. Zajisté je možné dosáhnout hustot $d_1 > 1, d_2 > 1$ a $D > 1$, pokud $n + k > 2(u + \log_2(n + k))$ správnou volbou parametrů $n > 60, k \geq 3n$ a $u = k/2$ dle [28]. Společně s lemma 19 pak víme, že odpovídající problémy batohu budou těžce řešitelné. Protože autoři v [11] neuvedli stěžejní důkaz lemma 19, nemůžeme s jistotou nic předpokládat o odolnosti daného kryptosystému proti útokům založeným na redukci báze mřížky.

4.5 Kryptosystém postavený na hybridním modelu

4.5.1 Úvod

Pin-Chang Su a Chien-Hua Tsai představili [57] kryptosystém, jehož bezpečnost je postavena jak na těžkém problému batohu, tak na problému diskrétního

logaritmu nad eliptickými křivkami. Spojení dvou silných kryptografických aparátů mělo vést k vytvoření odolného kryptosystému, jehož prolomení bude tak obtížné, jako by bylo vyřešení obou problémů zároveň. Z [26] uvidíme, že tento předpoklad nebyl správný a bezpečnost kryptosystému je postavena pouze na obtížnosti nalezení řešení těžkého problému batohu.

4.5.2 Popis kryptosystému

Jako těžký problém batohu [57] je brán lineárně posunutý těžký problém batohu z [36]. Pro vytvoření takového problému batohu je nejprve náhodně vybraná soukromá rychle rostoucí posloupnost p, ze které je vytvořena veřejná posloupnost b stejným postupem jako v případě Merkleova-Hellmanova kryptosystému v (3.3). Z posloupnosti b je vypočítaná posloupnost b' s velkou hustotou, která je dále transformovaná pomocí lineárního posunutí na šifrovací klíč $\bar{b}$. Pro takto vytvořený šifrovací klíč bylo dokázáno [36], že $\bar{b}$ nemůže vzniknout pomocí modulárních transformací rychle rostoucích posloupností a tedy v jeho případě nebudou úspěšné dosud známé útoky. Společně s šifrovacím klíčem $\bar{b}$ je vytvořen i dešifrovací klíč $\bar{p}$.

Daný postup nyní uvedeme.

Lineárně posunutý problém těžkého batohu.

- Problém batohu jako v Merkleově-Hellmanově kryptosystému
  1. Zvolíme rychle rostoucí posloupnost $p = (p_1, p_2, \ldots, p_n) \in \mathbb{N}^n$ a zvolíme dvě čísla $e, N \in \mathbb{N}$ taková, že $\mathrm{NSD}(e, N) = 1$ a $N > \sum_{i=1}^{n} p_i$.
  2. Spočítáme posloupnost $b = (b_1, b_2, \ldots, b_n)$ jako $b_i \equiv p_i e \bmod N$. Z této posloupnosti bude vytvořen klíč s velkou hustotou.
- Problém batohu s velkou hustotou
  1. Transformujeme posloupnost b na $b' = (b'_1, b'_2, \ldots, b'_n)$ s velkou hustotou tak, že položíme $b'_i \equiv b_i \bmod e$. Z toho plyne, že $b_i = b'_i + e\lfloor \frac{b_i}{e} \rfloor$.
  2. V následujících krocích pro posloupnost b' vytvoříme odpovídající rychle rostoucí posloupnost $p' = (p'_1, p'_2, \ldots, p'_n)$ takovou, že $b'_i \equiv p'_i e \bmod N$ pro všechna $1 \leq i \leq n$.
  3. Položíme $v = \lfloor \frac{N}{e} \rfloor$.
  4. Spočítáme posloupnost $c = (c_1, c_2 \ldots, c_n)$, kde $c_i = \lfloor b_i / e \rfloor$. Potom $0 \leq c_i \leq v$, pro všechna $0 \leq i \leq n$.
  5. Spočítáme posloupnost $p' = (p'_1, p'_2, \ldots, p'_n)$ jako $p'_i = p_i - c_i$, pro všechna $1 \leq i \leq n$.

Z volby hodnot c_i pro všechna $1 \leq i \leq n$ vidíme, že posloupnost p' bude rychle rostoucí. Dále z postupu výše vidíme, že pro všechna $1 \leq i \leq n$ platí, že $b'_i \equiv p'_i e \bmod N$, protože $p'_i e \equiv (p_i - c_i)e \equiv b_i - e\lfloor \frac{b_i}{e} \rfloor \bmod N$ a $b'_i = b_i - e\lfloor \frac{b_i}{e} \rfloor$. Je zřejmé, že volbou hodnoty e můžeme kontrolovat hustotu batohu.

Nyní z posloupností b' a p' vytvoříme lineárně posunutý problém batohu $\bar{b}$, $\bar{p}$.

- Lineárně posunutý problém batohu

1. Zvolíme náhodou binární posloupnost $t = (t_1, t_2, \ldots, t_n) \in \{0,1\}^n$.
2. Zvolíme $k \in \mathbb{N}$ takové, že
$$0 < k < \min_{i:\ t_i=1} b'_i.$$
3. Spočítáme veřejný klíč $\bar{b} = (\bar{b}_1, \bar{b}_2, \ldots, \bar{b}_n)$ pomocí lineárního posunutí jako $\bar{b}_i = b'_i - kt_i$, pro všechna $1 \leq i \leq n$.
4. Položíme $\bar{p} = p'$.

- Soukromý a veřejný klíč jsou dány jako
$$\mathsf{sk} = k, e, N, \bar{p},$$
$$\mathsf{pk} = \bar{b}.$$

Tento postup pro vytvoření lineárně posunutého problému těžkého batohu bude využit k tvorbě schéma [57].

Generování soukromého a veřejného klíče. Na začátku komunikace mezi stranami A a B je potřeba stanovit eliptickou křivku $\mathrm{E}(\mathbb{F}_q)$ a veřejné a soukromé klíče obou stran. Uvedeme si případ, kdy strana A přijímá zprávu od strany B. Strana A určí eliptickou křivku $\mathrm{E}(\mathbb{F}_q)$ jako v (4.1) a zvolí na ní bod $P \neq \infty$. Dále vybere jednosměrnou hašovací funkci
$$f : (\mathrm{E}(\mathbb{F}_q), \mathrm{E}(\mathbb{F}_q)) \to \mathbb{F}_q.$$
Parametry $N, e, k, \bar{b}, \bar{p}$, určí postupem z odstavce 4.5.2 pro $N < q$. Dále vybere $r_A \in \mathbb{F}_q$ a spočítá $Q_A = r_A P$. Strana B vybere $r_B \in \mathbb{F}_q$ a spočítá $Q_B = r_B P$.

Soukromé a veřejné klíče obou stran jsou
$$\mathsf{sk}_A = e, N, k, \bar{p}, r_A,$$
$$\mathsf{pk}_A = \bar{b}, \mathrm{E}(\mathbb{F}_q), P, Q_A$$
$$\mathsf{sk}_B = r_B,$$
$$\mathsf{pk}_B = Q_B.$$

Šifrování zprávy. Strana B zašifruje zprávu $m = (m_1, m_2, \ldots, m_n) \in \{0,1\}^n$ dle [57] následovně.

1. Spočítá $k_1 = \sum_{i=1}^{n} m_i \bar{b}_i$.
2. Spočítá $k_2 = f(Q_A, Q_B)$.
3. Odešle šifrový text $C = ([k_1, k_2] + r_B Q_A) \in \mathbb{F}_q \times \mathbb{F}_q$.

Dešifrování zprávy. Po obdržení šifrového textu C získá strana A otevřený text m následujícím postupem.

1. Spočítá $C - r_A Q_B = [k_1, k_2]$.
2. Ověří $k_2 = f(Q_A, Q_B)$.

3. Spočítá

$$\begin{aligned} k_1 e^{-1} &\equiv \left(\sum_{i=1}^{n} m_i \bar{b}_i\right) e^{-1} \equiv \left(\sum_{i=1}^{n} m_i \left(b'_i - k t_i\right)\right) e^{-1} \equiv \\ &\equiv \left(\sum_{i=1}^{n} m_i b'_i\right) e^{-1} - \left(\sum_{i=1}^{n} m_i k t_i\right) e^{-1} \equiv \\ &\equiv \sum_{i=1}^{n} m_i p'_i - k e^{-1} \sum_{i=1}^{n} m_i t_i \equiv \\ &\equiv \sum_{i=1}^{n} m_i \bar{p}_i - k e^{-1} \sum_{i=1}^{n} m_i t_i \mod N. \end{aligned}$$

Odtud je součet rychle rostoucí posloupnosti určen jako

$$\sum_{i=1}^{n} m_i \bar{p}_i \equiv k_1 e^{-1} + k e^{-1} \sum_{i=1}^{n} m_i t_i \mod N.$$

Jedinou neznámou hodnotou na pravé straně tohoto vztahu je $\sum_{i=1}^{n} m_i t_i$. My však víme, že $\sum_{i=1}^{n} t_i \leq n$, tedy hodnotu $\sum_{i=1}^{n} m_i t_i$ nalezneme po vyzkoušení maximálně $n+1$ možností. Pro každý tip vyzkoušíme, zda jsme obdrželi součet rychle rostoucí posloupnosti $\bar{p}$ pomocí algoritmu 1.

4.5.3 Kryptoanalýza systému

Pin-Chang Su a Chuen-Hua Tsai [57] předpokládali, že bezpečnost daného kryptosystému je založená na složitosti vyřešení těžkého problému batohu a zároveň na složitosti vyřešení problému diskrétního logaritmu nad eliptickými křivkami. V [26] však bylo pod $\mathcal{CPA}$ útokem (z anglického *chosen plaintext attack*) ukázáno, že je bezpečnost tohoto kryptosystému založena pouze na bezpečnosti problému lineárního posunutého batohu.

V případně útoku $\mathcal{CPA}$ má útočník přístup k šifrovacímu orákulu, tedy pro jakýkoliv otevřený text m může obdržet jemu příslušný šifrový text $c = \mathrm{enc}(m)$, kde *enc* je šifrovací funkce popsaná výše. Předpokládejme tedy, že pro zprávu m máme šifrový text $c = \mathrm{enc}(m) = [k_1, k_2] + r_B Q_A$. Za předpokladu, že máme orákulum pro vyřešení problému lineárního posunutého batohu, pak umíme určit k_1, tedy z šifrového textu c můžeme spočítat

$$c - [k_1, k_2] = r_B Q_A.$$

Potom pro jakýkoliv další otevřený text $m^\star$ a jemu odpovídající šifrový text $c^\star = \mathrm{enc}(m^\star) = [k_1^\star, k_2^\star] + r_B Q_A$ může útočník spočítat

$$c^\star - r_B Q_A = [k_1^\star, k_2^\star]$$

a v případě, že má přístup k orákulu pro vyřešení lineárně posunutého problému batohu, může obdržet otevřený text $m^\star$.

Vidíme, že bezpečnost daného kryptosystému je pod útokem $\mathcal{CPA}$ založena pouze na složitosti problému těžkého batohu, tedy se nejedná o schéma, které by zároveň využívalo obou bezpečnostních předpokladů.

4.6 Další kryptosystémy založené na problému batohu

Bylo navrženo mnoho dalších kryptosystémů založených na problému batohu.

V roce 2007 představili Baocang Wang, Qianhong Wu a Yupu Hu pravděpodobnostní kryptosystém [24], který je postaven na lehkém problému kompaktního batohu. Na tento kryptosystém můžeme pohlížet jako na analogii schéma z části 4.3 představené v [23], ale s vynecháním Čínské věty o zbytcích v konstrukci. Lehký problém kompaktního batohu je definován následujícím lemmatem.

Lemma 20. *Nechť $n \in \mathbb{N}$ a problém batohu je dán n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ a číslem $s \in \mathbb{N}$. Dále, nechť $D = (d_1, d_2, \ldots, d_n)$, kde $d_1 = a_1, d_i = NSD(a_i, d_{i-1})$, $2 \leq i \leq n-1$ a $d_n = 1$. Potom je pro všechna $i = 2, \ldots, n$ a $k \leq d_{i-1}/d_i$ problém kompaktního batohu lehký, pokud $0 \leq x_i \leq k-1$.*

Tento lehký problém batohu zastává roli soukromého klíče a z něj je pak odvozen problém těžkého batohu jako veřejný klíč. Kryptosystém z [24] má šifrovací funkci, která není lineární v otevřeném textu a problém těžkého batohu má hustotu vyšší než 1.06. Autoři [24] uvedli, že je kryptosystém odolný vůči přímému použití známých útoků. Hlubší kryptoanalýza byla provedena Amr. M. Youssefem v roce 2009 [61], ukázal, že omezení, která jsou kladena na parametry kryptosystému, vedou k snížení jeho bezpečnosti. Jedná se o heuristický útok, který využívá omezení kladená na parametry kryptosystému a metod z [1] pro řešení lineárních Diofantických rovnic k tomu, aby nalezl krátký list kandidátů na soukromý klíč. Z tohoto seznamu je pak možné nalézt ten správný soukromý klíč pomocí dešifrování zašifrované zprávy. Celý útok má asymptotickou časovou složitost $\mathcal{O}(n^7)$, kde n je počet bitů zprávy m. Následně byl tento útok vylepšen Moon Sung Leem v roce 2013 [38] pomocí řešení modulárních rovnic na útok s asymptotickou časovou složitost $\mathcal{O}(n^3)$.

Sílu eliptických křivek v kryptografii chtěl využít také Pin-Chang Su a spol. v roce 2005 [58]. V této práci je představen kryptosystém, který propojuje těžký problém batohu a problém diskrétního logaritmu nad eliptickými křivkami. Správnou volbou parametrů kryptosystému je možné docílit problému batohu s vysokou hustotou a eliptické křivky, pro kterou bude těžké vyřešit problém diskrétního logaritmu. Ač autoři v [58] předpokládali, že kryptosystém je odolný vůči přímému použití známých útoků, v roce 2009 [5] a v roce 2010 [2] byl představen ten samý útok, který snižuje bezpečnost kryptosystému z [58], a který jeho bezpečnost převede na bezpečnost Merkleova-Hellmanova kryptosystému [20], a kryptosystém následně prolomí analogií Shamirova útoku [53].

Obdoba Shamirova útoku může být použita i na kryptosystém navržený v [25] v roce 2008. Tento kryptosystém, který může docílit vysoké hustoty, je založený na permutačním kombinačním algoritmu. Hlubší kryptoanalýza, která vyvrací bezpečnost navrženého schéma, byla provedena v roce 2009 [5] a ten samý útok byl uveden i v roce 2011 [3].

Ve snaze předejít Shamirovu útoku z [53] navrhli Weidong Zhang a spol. v roce 2009 [63] kryptosystém, který není postavený na rychle rostoucí posloupnosti. Jako lehký problém batohu je zde představen problém batohu daný n-ticí $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$, kde $a_i = 2^{i-1}c_i$ a c_i jsou lichá čísla pro $1 \leq i \leq n$ a číslem $s \in \mathbb{N}$. V [63, Věta 1] je ukázáno, že problém nalezení vektoru $(x_1, x_2, \ldots, x_n) \in \{0,1\}^n$

takového, že $\sum_{i=1}^{n} x_i a_i = s$ je řešitelný v polynomiálním čase. Konstrukce kryptosystému je stejná jako v případě Merkleova-Hellmanova kryptosystému [20]. Další schéma, které má za cíl předejít Shamirovu útoku bylo navrženo v roce 2010 [30]. Zde soukromý klíč tvoří trojice problémů batohu, z nichž jeden je postaven na rychle rostoucí posloupnosti a zbylé dva ne. Tvorba veřejného klíče a proces šifrování a dešifrování zprávy je obdobný jako v Merkleově-Hellmanově kryptosystému. Ač jsou obě dvě práce [63] a [30] cíleny na to, aby vytvořily kryptosystém, který bude odolný vůči Shamirově útoku, v roce 2012 [19] byl představen úspěšný útok na oba dva kryptosystémy, který ukazuje, že rychle rostoucí posloupnost v soukromém klíči sk není nutným požadavkem pro aplikaci Shamirova útoku, ale stačí, aby dostatečně mnoho prvků v soukromém klíči bylo mnohem menších než modulus N.

4.7 Doposud neprolomené kryptosystémy založené na problému batohu

Ač byla většina kryptosystémů založených na problému batohu prolomena, stále zůstává několik návrhů, pro které dosud nebyla nalezena úspěšná kryptoanalýza. V tomto odstavci si uvedeme některé z nich.

V roce 1985 představili B. Chor a R. Rivest [9] Chorův-Rivestův kryptosystém, který není, na rozdíl od většiny kryptosystému založených na problému batohu, postavený na modulárních operacích, ale pracuje nad konečnými tělesy $\mathrm{GF}(p^{24})$, pro p prvočíslo a $\mathrm{GF}(256^{25})$. Sergey Vaydenay představil [60] v roce 2001 útok, který prolomí kryptosystém pro konečná tělesa $\mathrm{GF}(p^{24})$, pro p prvočíslo a $\mathrm{GF}(256^{25})$, avšak tento útok není použitelný pro jiná konečná tělesa.

Jukka A. Koskinen v roce 2001 [32] představil dva neinjektivní kryptosystémy založené na problému 0-1 batohu. Oba tyto kryptosystémy mají velkou hustotu a veřejný klíč je těžký problém batohu, který vznikl modulárními operacemi s malým modulem. To vede k tomu, že jsou vytvořeny neinjektivní kryptosystémy a při dešifrování příjemce získá více otevřených textů, mezi kterými pak musí vybrat ten správný. Správnost otevřeného textu může zjistit například tak, že se společně se zprávou zašifruje i její délka. Nevýhodou těchto kryptosystémů je, že dešifrování je pomalé.

Dosud také zůstává neprolomen kvantový kryptosystém navržený T. Okamotou a spol. roce 2000 v [45].

Závěr

Problém batohu je $\mathcal{NP}$ úplný problém, který je studován již mnoho let. Díky svému charakteru má mnohá uplatnění ve finančnictví a průmyslu a intenzivně je zkoumáno i jeho uplatnění v kryptografii. Silou kryptosystémů založených na $\mathcal{NP}$ úplném problému je, že jsou instance, pro které je velice nepravděpodobné, že by existoval (nyní i v budoucnu) nějaký algoritmus, který by je uměl řešit v polynomiálním čase. Zatím není známo, jak všechny takové instance generovat, ale je to vlastnost, kterou nemohou nabídnout současné kryptosystémy založené na problému faktorizace přirozených čísel nebo na problému diskrétního logaritmu. Od roku 1978, kdy byl představen první kryptosystém (Merkleův-Hellmanův kryptosystém [20]) založený na problému batohu, vznikla celá řada kryptosystémů tohoto typu. Drtivá většina z nich byla prolomena, a proto výzkum hledající bezpečné kryptosystémy založené na problému batohu neustává.

V této práci jsme se zabývali problémem batohu jak z pohledu teorie složitosti, tak z pohledu kryptografie. Po definování potřebných pojmů z teorie algebry a složitosti v první kapitole, jsme v druhé kapitole definovali problém 0-1 batohu. Pro problém 0-1 batohu jsme ukázali, že existují instance - problémy batohu založené na rychle rostoucích posloupnostech, které jsou řešitelné v polynomiálním čase. Ve druhé části této kapitoly jsme dokázali, že rozhodovací problém 0-1 batohu je $\mathcal{NP}$ úplný pomocí posloupnosti redukcí rozhodovacích problémů. Z této posloupnosti jsme dokázali ty redukce, které jsou méně známé a v literatuře se často nevyskytují. Konkrétně se jedná o redukce: rozhodovací problém k-obarvení grafu $\leq_p$ rozhodovací problém přesného pokrytí $\leq_p$ rozhodovací problém 0-1 batohu. Následně jsme dokázali redukci rozhodovací problém 0-1 batohu $\leq_p$ rozhodovací problém dvou loupežníků. Ve třetí kapitole jsme se věnovali kryptosystémům založeným na problému batohu. Nejprve jsme popsali jejich obecné schéma pomocí Merkleova-Hellmanova kryptosystému a uvedli jsme útok založený na redukci báze mřížky. Poté jsme pro Merkleův-Hellmanův kryptosystém ukázali, že pro nevhodně zvolené parametry kryptosystému bude možné snadno obdržet celý soukromý klíč a dešifrovat šifrový text. V druhé části této kapitoly jsme se zabývali novým navrženým konceptem kryptosystému - kryptosystém postavený na maticovém 0-1 batohu. Tento kryptosystém byl navržen ve snaze předejít známým útokům, avšak v závěru kapitoly bylo dokázáno analogií důkazu J. C. Lagariase a A. M. Odyzlka [35], že útok založený na redukci báze mřížky bude úspěšný ve většině případů šifrování pomocí kryptosystému postaveném na maticovém 0-1 batohu. V poslední kapitole jsme shrnuli moderní kryptosystémy postavené na problému batohu v letech 2000-2017. Tato kapitola mimo jiné ilustruje fakt, že jsou kryptosystémy tohoto typu stále studovány i přes to, že byla většina z nich prolomena.

Seznam použité literatury

[1] Aardal, K., Hurkens, C. a Lenstra, A. (2000). Solving a system of linear Diophantine equations with lower and upper bounds on the variables. *Mathematics of Operations Research*, **25 (3)**, 427–442.

[2] Aboud, S. J. (2010). Criticism of knapsack encryption scheme. *Journal of computing*, **2**.

[3] Aboud, S. J. (2011). Attack knapsack public key encryption scheme. *Information Society (i-Society)*.

[4] Bellman, R. E. (1957). *Dynamic programming*. Princeton University Press, New York. ISBN 978-0691146683.

[5] Bi, J., Meng, X. a Han, L. (2009). Cryptanalysis of two knapsack public-key cryptosystems. *Computer Application and System Modelling (ICCASM)*.

[6] Brickell, E. F. (1983). Solving low density knapsacks. *Advances in cryptology-CRYPTO*, pages 24–37.

[7] Brickell, E. F. (1985). Breaking iterated knapsacks. *Advances in cryptology. Proceedings of CRYPTO*, **84**, 342–358.

[8] Brickell, E. F. a Odlyzko, A. M. (1988). Cryptanalysis: A survey of recent results. *IEEE*, **76**, 578–593.

[9] Chor, B. a Rivest, R. (1988). A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, **IT-34**, 901–909.

[10] Clausen, J. (1999). Branch and bound algorithms-principles and examples. Technical report. University of Copenhagen.

[11] Coster, M. J., Joux, A., La Macchia, B. A., Odlyzko, A. M., Schnorr, C. a Stern, J. (1992). An improved low-density subset sum algorithms. *Computational Complexity*, **2**, 97–186.

[12] Coster, M. J., Joux, A., LaMacchia, B. A., Odlyzko, A. M., Schnorr, C. P. a Stern, J. (1992). An improved low-density subset-sum algorithm. *Computational Complexity*, **2**, 111–128.

[13] Cox, D. R. (1972). Regression models and life-tables (with Discussion). *Journal of the Royal Statistical Society, Series B*, **34**(2), 187–220.

[14] Dantzig, G. B. (1957). Discrete-variable extremum problems. *Operations Research*, **5 (2)**, 266–277.

[15] Diffie, W. a Hellman, M. (2010). New directions in cryptography. *IEEE Transaction on Information Theory*, **59 (1)**, 644–654.

[16] Erdös, P., Gruber, P. a Hammer, J. (1998). *Lattice points*. Pitman Monographs and Surveys in Pure and Applied Mathematics (book 39). Longman Sci. and Tech., John Wiley and Sons. ISBN 0-582-01478-6.

[17] Gamal, T. (1985). A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, **31(4)**, 469–472. conference version appeared in CRYPTO'84, pp. 10–18.

[18] Garey, M. R. a Johnson, D. S. (1979). *Computers and intractability: a guide to the theory of NP-completeness*. Dvacáté šesté vydání. W. H. Freeman and Company, New York. ISBN 978-0-7167-1044-8.

[19] Gottfried, H. a Meurer, A. (2012). New attacks for knapsack based cryptosystems. *International Conference on Security and Cryptography for Networks*, **7485**.

[20] Hellman, M. a Merkle, R. (1978). Hiding information and signatures in trapdoor knapsacks. *IEEE Transactions on Information Theory*, **24**, 525–530.

[21] Henk, M. a Weismantel, R. (1997). On Hilbert bases of polyhedral cones. *Results in Mathematics*, **32**, 298—-303.

[22] Hu, Y. a Wang, B. (2006). Diophantine approximation attack on a fast public key cryptosystem. *International Conference on Information Security Practice and Experience*, **3903**, 25–32.

[23] Hu, Y. a Wang, B. (2010). Quadratic compact knapsack public-key cryptosystem. *Computers and Mathematics with Applications*, **59**, 194–206.

[24] Hu, Y., Wu, Q. a Wang, B. (2007). A knapsack-based probabilistic encryption scheme. *Information Science*, **177**(19), 3981–3994.

[25] Hwang, M., Lee, C. C. a Tzeng, S. F. (2008). A new knapsack public-key cryptosystem based on permutation combination algorithm. *Int'l Journal of Mathematical and Computer Sciences*, **5**(1), 33–38.

[26] Jin, Z., Zhang, H. a Li, Z. (2017). Security on a knapsack-type encryption scheme based upon hybrid-model assumption. *International Journal of Network Security*, **19**, 644–647.

[27] Karp, M. R. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103.

[28] Kasahara, M. a Murakami, Y. (2011). A differential knapsack scheme with no trapdoor sequence. *SCIS 2011*, pages 25–28.

[29] Kellerer, H., Pisinger, D. a Pferschy, U. (2004). *Knapsack Problems*. Springer, Berlin. ISBN 978-3-540-40286-2.

[30] Kobayashi, K., Tadaki, K., Kasahara, M. a Tsujii, S. (2010). A knapsack cryptosystem based on multiple knapsacks. *International Symposium on Information Theory and its Applications*.

[31] Kolesar, P. J. (1967). A branch and bound algorithm for the knapsack problem. *Management Science*, **13**, 723–735.

[32] Koskinen, J. A. (2001). Non-injective knapsack public-key cryptosystems. *Theoretical Computer Science*, **255**, 401–422.

[33] Kunihiro, N. (2008). New definition of density on knapsack cryptosystems. *Progress in Cryptology AFRICACYPT*, pages 156–173.

[34] Lagarias, J. C. (1984). Knapsack public key cryptosystems and Diophantine approximation. *Proceedings of Crypto*, pages 3–23.

[35] Lagarias, J. C. a Odlyzko, A. M. (1985). Solving low-density subset sum problems. *Journal of the Association for Computing Machinery*, **32**(1), 229–246.

[36] Laih, C. S., Lee, J. Y., Harn, L. a Su, Y. K. (1989). Linearly shift knapsack public-key cryptosystem. *IEEE Journal on Selected Areas in Communications*, **7 (4)**, 534–539.

[37] Lee, M. S. (2011). Cryptanalysis of a quadratic compact knapsack public-key cryptosystem. *Computers and Mathematics with Applications*, **62(9)**, 3614–3621.

[38] Lee, M. S. (2013). Improved cryptanalysis of a knapsack-based probabilistic encryption scheme. *Information Science*, **222**(10), 779–783.

[39] Lenstra, A., Lenstra, H. a Lovász, L. (1982). Factoring polynomials with rational coefficients. *Math. Annalen*, **261**, 515–534.

[40] Martello, S. a Toth, P. (1988). A new algorithm for the 0-1 knapsack problem. *Management Science*, **34**, 633–644.

[41] Martello, S., Pisinger, D. a Toth, P. (1987). New trends in exact algorithms for the 0-1 knapsack problem. *European Journal of Operational Research*, **123 (2)**, 325–332.

[42] Martello, S., Pisinger, D. a Toth, P. (1999). Dynamic programming and strong bounds for the 0-1 knapsack problem. *Management Science No. 3, March 1999*, **45 (3)**, 414–424.

[43] Mazo, J. E. a Odlyzko, A. M. (1990). Lattice points in high dimensional spheres. *Monatsh. Math.*, **17**, 47–61.

[44] Odlyzko, A. M. (1990). The rise and fall of knapsack cryptosystems. *Cryptology and Computational Number Theory*, **42**, 75–88.

[45] Okamoto, T., Tanaka, K. a Uchiyama, S. (2000). Quantum public key cryptosystems. *Advances in Cryptology - CRYPTO'00 (LNCS 1880)*, pages 147–165.

[46] Papadimitriou, C. H. a Steiglitz, K. (1982). *Combinatorial optimization: algorithms and complexity.* Reprint of the Prentice-Hall, New Jersey. ISBN 978-0486402581.

[47] Pisinger, D. (1995). Algorithms for knapsack problems. *PhD thesis, University of Copenhagen, Dept. of Computer Science*.

[48] Pisinger, D. (2005). Where are the hard knapsack problems? *Computers Operations Research*, **32 (9)**, 2271–2284.

[49] Pohst, M. (1987). A modification of the LLL reduction algorithm. *J. Symbolic Computation*, **4**, 123–127.

[50] Rivest, R., Shamir, A. a Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, **21(2)**, 120–126. Previously released as an MIT „Technical Memo“ in April 1977.

[51] Schnorr, C. a Euchner, M. (1994). Lattice basis reduction: improved practical algorithms and solving subset sum problems. *Math. Programming*, **66**, 181–199.

[52] Schrijver, A. (1999). *Theory of linear and integer programming*. John Wiley and Sons, Amsterdam. ISBN 0-471-98232-6.

[53] Shamir, A. (1984). A polynomial-time algorithm for breaking the basic Merkle-Hellman cryptosystem. *IEEE Transactions on Information Theory*, **IT-30**, 699–704.

[54] Shamir, A. a Zippel, R. (1980). On the security of the Merkle-Hellman cryptographic scheme. *IEEE Trans. Informat. Theory*, **23 (3)**, 339–340.

[55] Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on wuantum computer. *SIAM Journal of Computing*, **26**, 1484–1509.

[56] Smart, N. P. (1989). *The algorithmic resolution of Diophantine equations*. London Mathematical Society Student Texts 41. Cambridge University Press. ISBN 0 521 64633 2.

[57] Su, P. C. a Tsai, C. H. (2009). New cryptosystem design based on hybrid-mode problems. *Computers and Electrical Engineering*, **35**, 478–484.

[58] Su, P. C., Lu, E. H. a Chang, H. K. C. (2005). A knapsack public-key cryptosystem based on elliptic curve discrete logarithm. *Applied Mathematics and Computation*, **168**, 40–46.

[59] Turing, A. M. (1936). On computable numbers with an application to the Entscheidungsproblem. *Proc. London Math. Soc.*, **42**, 230–265.

[60] Vaudenay, S. (2001). Cryptanalysis of the Chor-Rivest cryptosystem. *Advances in Cryptology CRYPTO'98*, **1462**, 243–256.

[61] Youssef, A. M. (2009). Cryptanalysis of a knapsack-based probabilistic encryption scheme. *Information Science*, **179**(18), 3116–3121.

[62] Youssef, A. M. (2011). Cryptanalysis of a quadratic knapsack cryptosystem. *Computers and Mathematics with Applications*, **61(4)**, 1261–1265.

[63] Zhang, W., Wang, B. a Hu, Y. (2009). A new knapsack public-key cryptosystem. *Fifth International Conference on Information Assurance and Security*, **2**, 53–56.