VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky
a komunikačních technologií

DIPLOMOVÁ PRÁCE

Brno, 2023 Bc. Kristýna Mlčáková

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

ÚSTAV TELEKOMUNIKACÍ

DEPARTMENT OF TELECOMMUNICATIONS

ÚTOKY NA BIOMETRICKÉ HASHOVACÍ METODY POUŽITÍM OPTIMALIZAČNÍCH TECHNIK

ATTACKING BIOMETRIC HASHING VIA OPTIMIZATION TECHNIQUES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

AUTOR PRÁCE — Bc. Kristýna Mlčáková

AUTHOR

VEDOUCÍ PRÁCE — prof. Mgr. Pavel Rajmic, Ph.D.

SUPERVISOR

BRNO 2023

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

Diplomová práce

magisterský navazující studijní program **Informační bezpečnost**

Ústav telekomunikací

Studentka: Bc. Kristýna Mlčáková ***ID:*** 211323

Ročník: 2 ***Akademický rok:*** 2022/23

NÁZEV TÉMATU:

Útoky na biometrické hashovací metody použitím optimalizačních technik

POKYNY PRO VYPRACOVÁNÍ:

Biometrické hashování je dvoufaktorová autentizační metoda kombinující tajné heslo/klíč s biometrickým údajem. Nastudujte jak biohashovací algoritmy fungují. Na základě literatury [1] navrhněte několik typů útoků proti tomuto typu zabezpečení. Konkrétně se zaměřte na biometrii pomocí fotografie. Útoky budou založeny na řešení optimalizačních problémů, proto se seznamte se souvisejícími algoritmy [2]. Útoky implementujte v Matlabu či jiném dohodnutém programovacím jazyku, otestujte a kvantitativně vyhodnoťte. Učiňte závěr a doporučení týkající se bezpečnosti biohashování.

DOPORUČENÁ LITERATURA:

[1] Topcu, B. et al. Practical security and privacy attacks against biometric hashing using sparse recovery. EURASIP journal on advances in signal processing. Cham: Springer International Publishing, 2016(1), 1-20. ISSN 1687-6172. doi:10.1186/s13634-016-0396-1

[2] Candes, E.J., Wakin, M.B. An Introduction To Compressive Sampling. IEEE signal processing magazine. PISCATAWAY: IEEE, 2008, 25(2), 21-30. ISSN 1053-5888. doi:10.1109/MSP.2007.914731

Termín zadání: 6.2.2023 ***Termín odevzdání:*** 19.5.2023

Vedoucí práce: prof. Mgr. Pavel Rajmic, Ph.D.

doc. Ing. Jan Hajný, Ph.D.
předseda rady studijního programu

ABSTRAKT

Práce se zabývá biometrickým hashováním, což je dvoufaktorová autentizační metoda, která kombinuje tajný klíč s biometrickými daty za účelem pořízení bezpečné binární šablony (biohashe), která se poté používá při autentizaci. V práci je nejprve rozebrána tato problematika z teoretické stránky. Následně se věnujeme implementaci tohoto konceptu. V práci jsou také uvedeny útoky na biometrické hashovací systémy, které jsou následně prakticky realizovány. Při provádění útoku vycházíme z předpokladu, že útočník zná tajný klíč uživatele a biohashe uložené v databázi. Při útoku aplikujeme dvě metody, které dokážou rekonstruovat biometrická data oprávněného uživatele a biohashe. Metody jsou založeny na 1-bitové rekonstrukci signálu využívající koncept komprimovaného snímání. Cílem práce je vyhodnotit úspěšnost útoků na biometrické hashovací systémy a učinit závěr o bezpečnosti těchto systémů.

KLÍČOVÁ SLOVA

biometrické systémy, biometrika, biohash, Hammingova vzdálenost, komprimované snímání, náhodná projekční matice, vektor rysů, práh

ABSTRACT

The thesis deals with biometric hashing, which is a two-factor authentication method that combines a secret key with biometric data in order to create a secure biomeric template for authentication purposes. The thesis first discusses the theoretical aspects of this issue and then focuses on the implementation of this concept. The paper also describes attacks on biometric hashing systems that are subsequently implemented. Practically performed attacks are based on the assumption that the attacker knows the user's secret key and biohashes stored in the database. Two methods of attacks are applied. These methods are able to reconstruct the biometric data of an authorized user and related biohashes. Methods are based on 1-bit compressive sensing approach. The aim of the thesis is to evaluate the effectivity of the attacks on biometric hashing systems and to make a conclusion about the security of such systems.

KEYWORDS

biometric systems, biometrics, biohash, Hamming Distance, compressive sensing, Random Projection Matrix, feature vector, treshold

MLČÁKOVÁ, Kristýna. *Útoky na biometrické hashovací metody použitím optimalizačních technik*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, Ústav telekomunikací, 2023, 91 s. Diplomová práce. Vedoucí práce: prof. Mgr. Pavel Rajmic, Ph.D.

Prohlášení autora o původnosti díla

Jméno a příjmení autora:	Bc. Kristýna Mlčáková
VUT ID autora:	211323
Typ práce:	Diplomová práce
Akademický rok:	2022/23
Téma závěrečné práce:	Útoky na biometrické hashovací metody použitím optimalizačních technik

Prohlašuji, že svou závěrečnou práci jsem vypracovala samostatně pod vedením vedoucí/ho závěrečné práce a s použitím odborné literatury a dalších informačních zdrojů, které jsou všechny citovány v práci a uvedeny v seznamu literatury na konci práce.

Jako autorka uvedené závěrečné práce dále prohlašuji, že v souvislosti s vytvořením této závěrečné práce jsem neporušila autorská práva třetích osob, zejména jsem nezasáhla nedovoleným způsobem do cizích autorských práv osobnostních a/nebo majetkových a jsem si plně vědoma následků porušení ustanovení § 11 a následujících autorského zákona č. 121/2000 Sb., o právu autorském, o právech souvisejících s právem autorským a o změně některých zákonů (autorský zákon), ve znění pozdějších předpisů, včetně možných trestněprávních důsledků vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku č. 40/2009 Sb.

Brno ....

podpis autorky*

*Autor podepisuje pouze v tištěné verzi.

PODĚKOVÁNÍ

Ráda bych poděkovala vedoucímu diplomové práce panu prof. Mgr. Pavlu Rajmicovi, Ph.D. za odborné vedení, konzultace, trpělivost a podnětné návrhy k práci. Dále bych chtěla poděkovat své rodině za neutuchající podporu a pochopení.

Obsah

Seznam obrázků

Seznam tabulek

Seznam výpisů

Úvod

V současné době se rozmohlo využívání biometrických charakteristik pro autentizaci např. k mobilnímu telefonu, počítači. Řada uživatelů se tak autentizuje pomocí otisku prstu nebo snímku obličeje. Biometrické systémy se tak mohou jevit jako vhodnou náhradou pro klasické autentizační systémy, které využívají hesla. Biometrické systémy navíc běžným uživatelům poskytují řadu výhod – biometrické charakteristiky si nemusí „pamatovat“ jako hesla, není třeba vymýšlet dostatečně silné heslo pro autentizaci apod. Na druhé straně však biometrické systémy mají i svá negativa. Biometrickou charakteristiku, na rozdíl od hesla, nelze změnit. Tedy, pokud se útočníkovi podaří nějakým způsobem získat naši biometrickou charakteristiku a prolomit tak autentizaci systému, je toto trvalý problém. Z těchto důvodů začaly vznikat biometrické hashovací systémy. Ty kombinují biometrické charakteristiky s heslem či tokenem, čímž se vyvažuje výše zmíněný nedostatek.

Diplomová práce se zaměřuje právě na tyto systémy a má za cíl přiblížit tuto problematiku. Dále jsou diskutovány i možnosti útoků na tyto systémy – z tohoto důvodu je zde představen základní matematický aparát potřebný pro tento účel, konkrétně matematika využívaná komprimovaným snímáním. V práci jsou také popsány možné útoky na tyto systémy.

Cílem diplomové práce je seznámit se s teorií a technickými aspekty biometrického hashování a dále s matematikou týkající se rekonstrukce signálů z jejich komprimovaných pozorování a na základě takto získaných znalostí zrealizovat vytváření biohashe a provést útoky využitím optimalizačních metod. Na základě teoretických poznatků a výsledků získaných z implementace v praktické části práce je pak učiněn závěr o biometrickém hashování s využitím komprimovaného snímání a útocích na něj.

Práce je členěna do pěti kapitol, z čehož čtyři se věnují teoretickým aspektům a poslední se zabývá praktickou stránkou problematiky. V první kapitole jsou rozebrány obecně biometrické systémy jako takové. Na to navazuje druhá kapitola – biometrický hashing. Dále je popsána problematika komprimovaného snímání, jelikož principy tohoto konceptu budou využívány v praktické části práce. V závěru teoretické části jsou uvedeny možné útoky na biometrický hashing, které budou následně implementovány v praktické části. V poslední kapitole je přiblížena implementace a jsou zde diskutovány výsledky, kterých bylo v práci dosaženo.

1 Biometrické systémy

Tato kapitola stručně představí biometrické systémy jako takové, jelikož tvoří základ biometrických hashovacích systémů, kterým bude věnována stěžejní část práce.

Mezi nejznámější charakteristiky, které biometrické systémy využívají, řadíme otisky prstů, obličej, hlas či duhovku – tzv. biometrické charakteristiky. Biometrickou charakteristiku může představovat jakákoliv fyziologická či behaviorální vlastnost. Mezi fyziologické vlastnosti řadíme například duhovku, otisk prstu, obličej, otisk dlaně a mezi behaviorální řadíme např. podpis člověka, či styl jeho chůze, abychom takovou vlastnost mohli označit za biometrickou charakteristiku, musí splňovat následující požadavky:

- **Univerzalita** – tuto charakteristiku má každý člověk.
- **Rozlišitelnost** – tato charakteristika je u dvou různých osob jiná.
- **Trvalost** – charakteristika by měla být v čase neměnná.
- **Schopnost být shromažďován** – je možné charakteristiku kvantitativně měřit a zkoumat. [1, 2]

V praxi u biometrických systémů musíme však posuzovat také problémy s výkonem systému, přijatelnost pro subjekt poskytnout danou biometrickou charakteristiku a možnost obejití biometrického systému. V tabulce 1.1 je provedeno srovnání jednotlivých biometrických charakteristik na základě výše zmíněných požadavků.

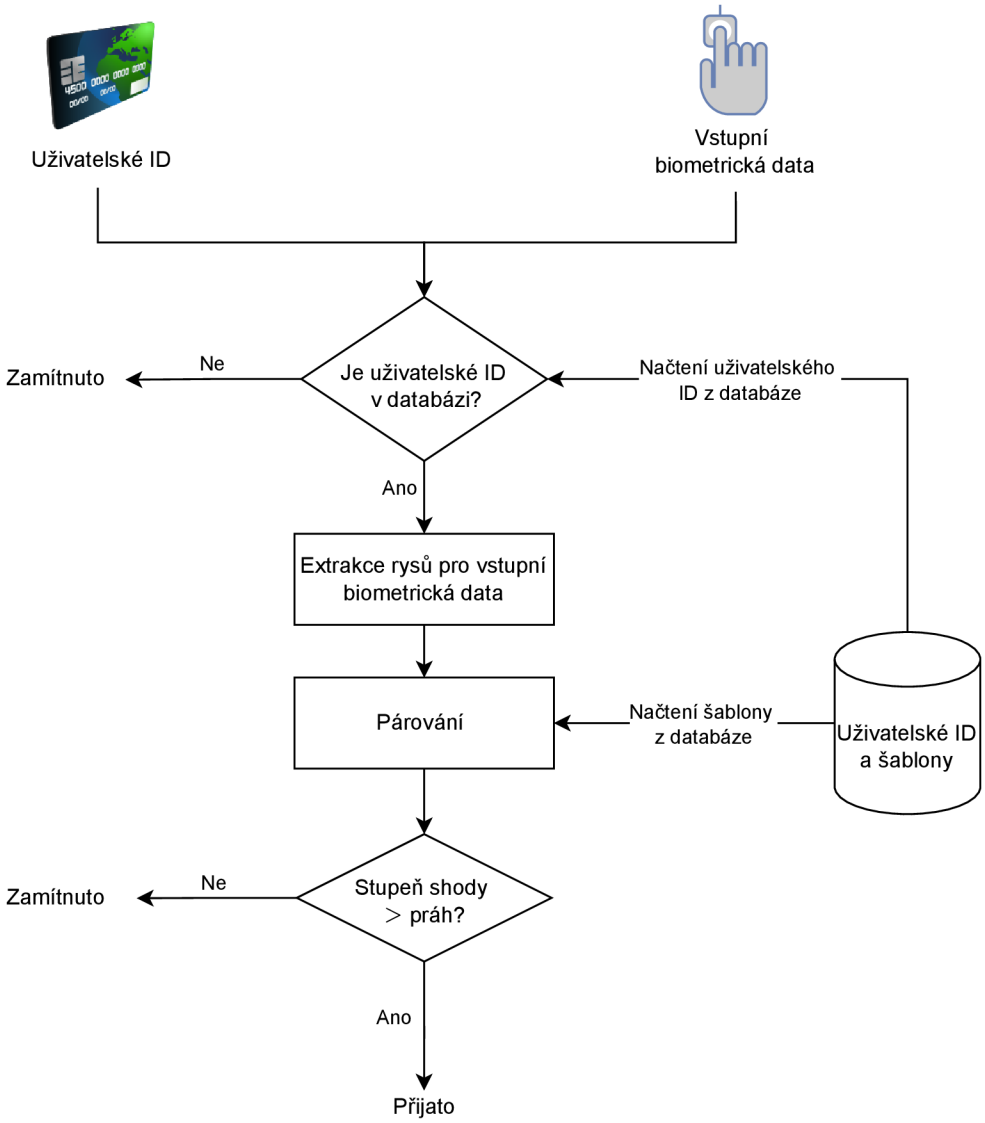
Tab. 1.1: Srovnání biometrických technik – převzato z [1]

	Otisk prstu	**Obličej**	**Geometrie ruky**	**Duhovka**	**Hlas**
Rozlišitelnost	vysoká	nízká	střední	vysoká	nízká
Trvalost	vysoká	střední	střední	vysoká	nízká
Shromažďovací schopnost	střední	vysoká	vysoká	střední	střední
Výkon	vysoký	střední	nízký	vysoký	nízký
Přijatelnost	střední	vysoká	střední	nízká	vysoká
Možnost obcházení/překonání	nízká	vysoká	střední	nízká	vysoká

Biometrický systém je systém rozpoznávání vzorů, který poznává osobu na základě vektoru rysů (*feature vector*), odvozeného od fyziologické či behaviorální vlastnosti, kterou daná osoba vlastní. Biometrický systém obvykle pracuje v jednom ze dvou režimů – identifikačním či verifikačním. Základní princip těchto režimů je zobrazen na obrázku 1.2.

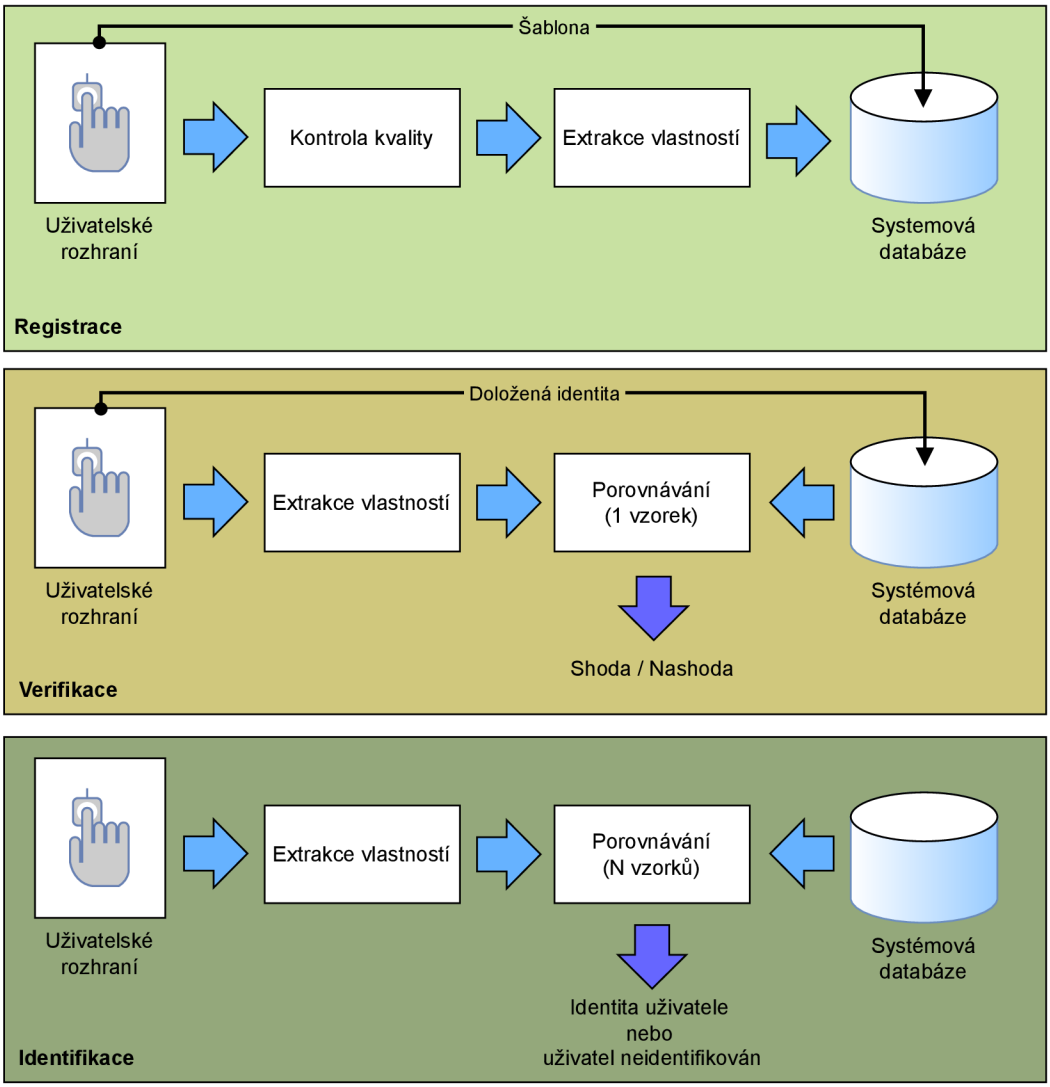
Systém pracující ve verifikačním režimu potvrzuje identitu osoby, tak že porovnává zachycené biometrické charakteristiky s biometrickou šablonou, kterou má uloženou v systémové databázi. Ověřování identity se typicky používá pro pozitivní rozpoznávání (*positive recognition*), jehož hlavním cílem je zabránit mnoha osobám v používání stejné identity. V tomto režimu se projevuje snaha nalézt odpověď např.

na otázku: *Je tato osoba Bob?* Pro názornost fungování verifikačního režimu viz obrázek 1.1.



Obr. 1.1: Biometrický verifikační systém – [2]

V identifikačním režimu systém rozpoznává osoby prohledáváním celé databáze se šablonami a hledá shodu. Na rozdíl od verifikačního režimu, který provádí porovnávání jedna ku jedné, v identifikačním režimu je provedeno porovnávání jedna ku mnoha, aby byla zjištěna identita osoby. Identifikační režim tak hledá odpověď na otázku: *Kdo je tato osoba?*. Identifikace tvoří stěžejní komponentu negativních rozpoznávacích aplikací (*negative recognition applications*). V těchto aplikacích systém prokazuje, že daná osoba je tím, kým tvrdí že není. Cílem negativního rozpoznávání je zabránit, aby se jedna osoba vydávala za více lidí. [1]



Obr. 1.2: Blokové schéma – registrace, verifikace, identifikace – [1]

1.1 Chyby biometrických systémů

U systémů využívajících biometrické charakteristiky je problematické, že tyto charakteristiky nikdy nebudou 100% shodné.[1] Tím se tyto systémy odlišují od těch, které využívají např. autentizaci pomocí hesel, které naopak musí vždy plně odpovídat uloženému heslu. Proto v biometrických systémech musíme nastavit určitou míru tolerance – za tímto účelem zavádíme tzv. práh resp. rozhodovací práh (*treshold*) t. Typickou odpovědí biometrického systému je tzv. stupeň shody (*matching score*) s. Stupeň shody nám udává míru shody mezi biometrickým vstupem a šablonovou reprezentací, která je uložena v databázi. Práh t reguluje rozhodování systému. Pokud

[1] Např. nikdy nepřiložíme prst ke čtečce otisků prstů stejným způsobem nebo kvůli nachlazení se nám změní hlas apod.

biometrické vzorky generují hodnotu $s \geq t$, pak je systém označí jako tzv. odpovídající si pár (*mate pairs*). Jako tzv. neodpovídající si páry (*nonmate pairs* označuje biometrické vzorky, pro které platí $s < t$. Rozložení hodnot, které nám generují odpovídající si páry, nazýváme rozložení oprávněného uživatele (*genuine distribution*) a rozložení, které nám generují neodpovídající si páry, označujeme jako rozložení narušitele.[2][1]

Při autentizaci osob se aplikují dva klasické přístupy – přístup založený na tokenech (využívá fyzická zařízení jako třeba čipové karty) a přístup využívající znalosti (spoléhá na soukromé znalosti – např. heslo). Oba přístupy mají své nevýhody, které se jsou eliminovány u biometrických prostředků autentizace, jako například: znalost i token mohou být zapomenuty, ztraceny, zcizeny, zkopírovány či sdíleny s neoprávněnou osobou. Autentizace využívající biometrii má však své vlastní nevýhody. U biometrické autentizace se nám projevují dva typy chyb:

1. **Falešné přijetí** (*false acceptance*) – chybné přiřazení charakteristik dvou různých osob jedné osobě.
2. **Falešné odmítnutí** (*false rejection*) – chybné přiřazení charakteristik jedné osoby dvěma různým osobám. [3] [2]

Spolehlivost biometrických systémů bývá běžně určována mírou falešných přijetí (*false acceptance rate* – FAR) a mírou falešných odmítnutí (*false rejection rate* – FRR). Tyto metriky mohou být přizpůsobeny změnou prahové hodnoty (*treshold*) – není ovšem možné využít práh k tomu, abychom paralelně snížili hodnoty FAR i FRR, jak je vidět na obrázku 1.3.

Hodnoty se vzájemně vyrovnávají – tzn. pokud snížíme hodnotu FAR, zároveň tím zvýšíme hodnotu FRR a naopak. FAR a FRR jsou funkcemi rozhodovacího prahu t. Je-li systém nastaven tak, aby byl tolerantnější, zároveň se tím zvýší počet falešných přijetí, tedy systém nebude tak bezpečný. A naopak nastavíme-li rozhodovací práh t moc přísně, může se systém stát nepřístupným i pro oprávněné uživatele. [1] Tuto skutečnost lze demonstrovat na obrázku 1.3 – chceme-li nastavit rozhodovací práh t přísněji, požadujeme vyšší hodnotu stupně shody s (přímka rozhodovacího prahu se nám v obrázku posune ve směru osy x), a tím se zvětší plocha vymezující oblast falešných odmítnutí. V případě benevolentnějšího systému postupujeme analogicky. Spolehlivost systému ve všech operačních bodech (prahových hodnotách t) můžeme znázornit pomocí operační charakteristiky přijímače (*receiver operating characteristic, ROC*).[4]

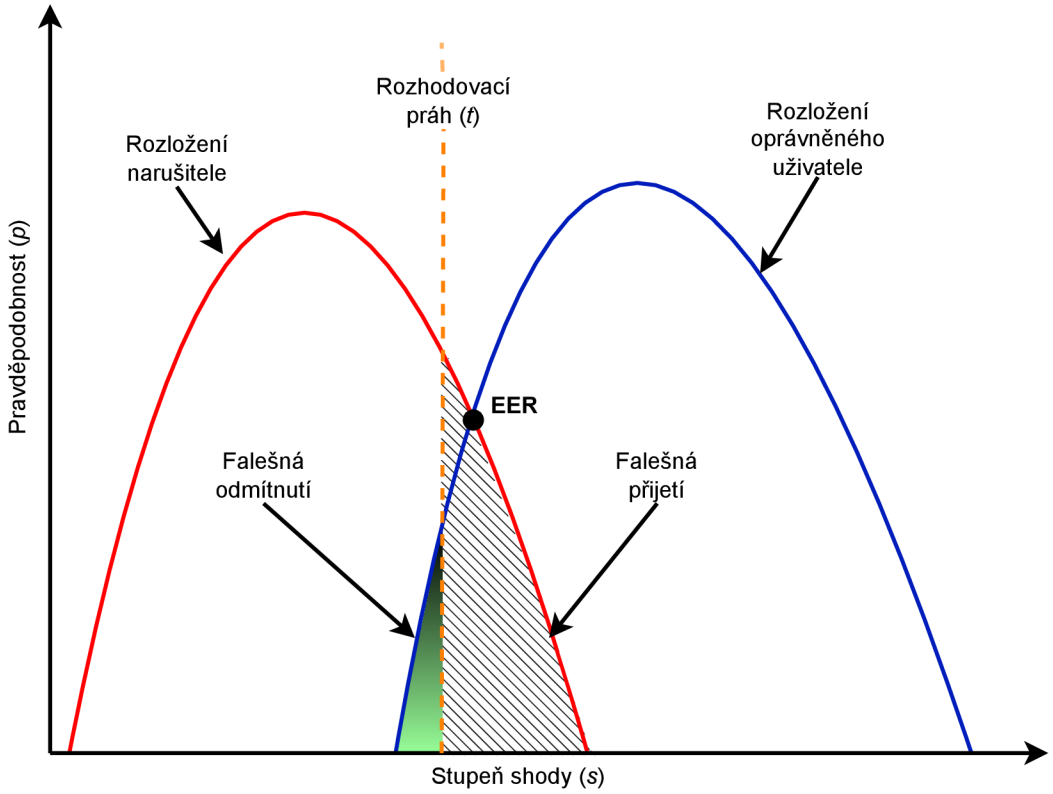
Dalším důležitým ukazatelem výkonu biometrického systému je míra ekvivalence chyby (*equal error rate* – EER), která je definována jako bod, kde se FAR a FRR

[2] K tomu viz obrázek 1.3

[3] Někdy také označováno jako falešné spárování a falešné nespárování – viz [1]

[4] Běžně používán pojem ROC křivka. K ukázce ROC křivek viz [2, s. 1363–1364].

rovnají. Systém, jehož hodnota EER by byla nulová, nazýváme dokonalým systémem z pohledu přesnosti identifikace.[1, 2]



Obr. 1.3: Biometrický systém – chybové hodnoty (inspirováno [1])

Kromě výše uvedených metrik se dále pro posuzování přesnosti biometrických systémů používají metriky neúspěšného zachycení (*failure to capture* – FTC) a neúspěšné registrace (*failure to enroll* – FTE). FTC nám udává procentuální hodnotu případů, kdy biometrické zařízení není schopno sejmout biometrický vzorek. FTE udává procentuální hodnotu případů, kdy se uživatel nemůže registrovat, typicky z důvodu odmítnutí nekvalitní šablony systémem.

1.2 Bezpečnost biometrických systémů

Pro dostatečné zabezpečení biometrických systémů je velmi důležité zajistit ochranu šablon legitimních uživatelů. Šablona je množinou vektorů extrahovaných z biometrických vlastnosti a používá se při párování s biometrickým vstupem během pokusu o autentizaci. Šablony mohou být uloženy buď v centrální databázi, a nebo na čipové kartě. Kompromitování biometrické šablony představuje závažnou bezpečnostní hrozbu a zásah do soukromí. Vzhledem ke specifické povaze biometrických

charakteristik totiž není možné pro oprávněného uživatele změnit své biometrické identifikátory a nahradit je jinými nekompromitovanými, jako je tomu v případě hesel. Pro zajištění ochrany biometrických šablon lze využít metodu transformace šablony (*transformation template approach*).

Transformace šablony spočívá v tom, že šablona, která sestává z rysů extrahovaných z biometrických vlastností, je transformována pomocí parametrů odvozených od uživatelem specifikovaného hesla či klíče. Ukládá se pouze takto transformovaná šablona a párování je prováděno přímo v transformovaném prostoru. Mezi dvě známé metody transformace šablony řadíme – biometrické hashování (viz kap. 2) a zrušitelné šablony otisků prstů (*cancelable fingerprint templates*). [3]

Biometrické systémy vzhledem k nezcizitelnému a osobitému charakteru biometrických charakteristik mají řadu výhod oproti tradičním autentizačním mechanismům. Existují však i určité nevýhody – systémy nejsou spolehlivé a závážnou zranitelnost představuje výše diskutovaná možnost kompromitace uložených šablon. Nelegálně získanou šablonu může útočník použít pro vytvoření biometrického podvrhu, se kterým lze získat přistup do systému. Vytvoření takového podvrhu bývá ovšem často náročné (k tomu viz [1, s. 38–40]). Zcizenou šablonu je možné napodobit a použít ji k překonání biometrického systému (útok vniknutím – *intrusion attack*). Vzhledem k tomu, že biometrické vlastnosti mají být trvalé a jedinečné, lze nelegálně získané šablony také využít k propojení uživatele napříč databázemi (útok propojením – *linkage atack*) nebo ke shromažďování dalších citlivých informací o uživatelích.[5] [1, 3]

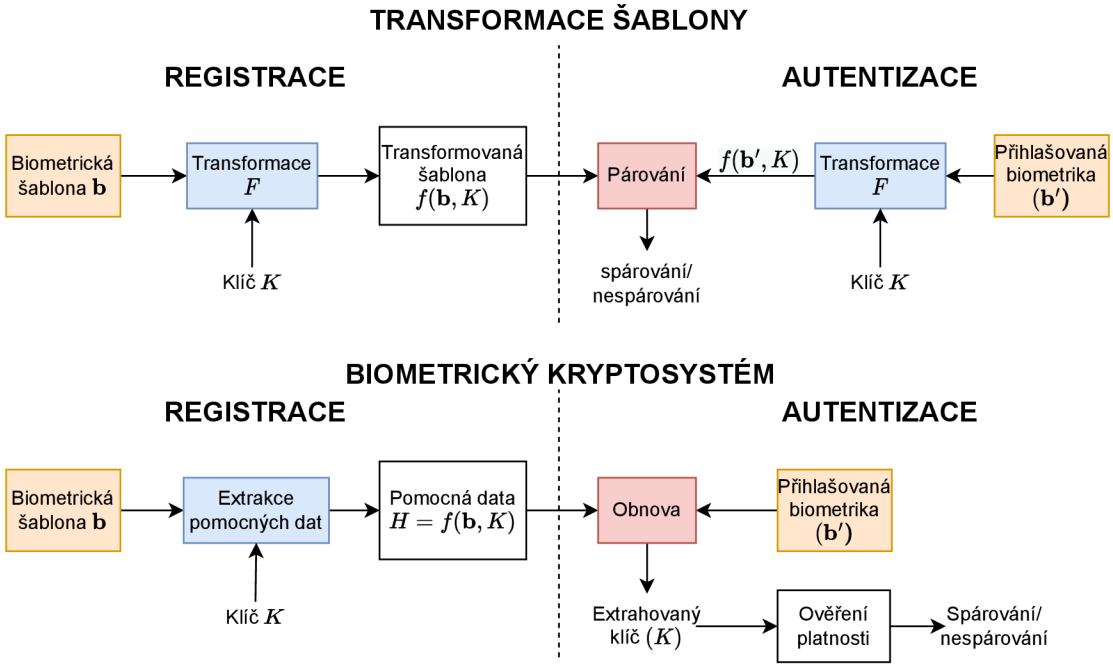
Pro lepší zabezpečení biometrických šablon byla navržena řada metod a technik – uplatňují se jak hardwarové, tak i softwarové přístupy. Hardwarový přístup zahrnuje návrh „uzavřeného“ rozpoznávacího systému, kde biometrická šablona nikdy neopustí fyzicky zabezpečený modul, např. čipová karta. Modul může obsahovat pouze šablonu a párovač (*Match-on-card*) nebo kompletní biometrický systém včetně senzoru, extraktoru rysů, šablony a párovače (*System-on-card*). Zařízení porovnává vstupní biometrické vlastnosti se šablonami uloženými v zařízení. V případě úspěšné autentizace uvolní klíč.

Softwarové řešení ukládá modifikovanou verzi šablony, která odhaluje pouze malé množství informace o původní biometrické vlastnosti, avšak i přesto může být úspěšně použita pro autentizaci. Rozlišujeme zejména dvě kategorie – transformaci šablony nebo rysů (*template or feature transformation*) a biometrický kryptosystém (*bimetric crytosystem*). Technika transformace šablony převádí biometrickou šablonu na základě parametrů odvozených od externí informace, jako heslo nebo klíč uživatele. V průběhu autentizace je použita stejná transformační funkce na přihlašo-

[5] Jako je rasa, pohlaví a určité zdravotní stavy.

vanou biometriku a spárování s uloženou šablonou v transformovaném prostoru. [3]

Biometrické kryptosystémy většinou spoléhají na teorii programové opravy chyb (*error correction coding theory*) a pokouší se získat informace k opravě chyb z biometrických vlastností (s použitím a nebo bez použití externího klíče) – známé jako pomocná data (*helper data*). Pomocná data neodhalují důležité informace o biometrice ani o klíči. Programy na opravu chyb se v těchto systémech běžně používají k obnově registrovaných biometrických vlastností nebo klíče při načtení daných biometrických dat. [3] Srovnání metod transformace šablony a biometrického kryptosystému je znázorněno na obrázku 1.4.



Obr. 1.4: Schématický diagram metody transformace šablony a biometrického kryptosystému – [3]

Obě metody mají svá pro a proti. Výhodou transformace šablony je, že šablony jsou snadno zrušitelné. Avšak poměrně složité je vyhodnotit míru zabezpečení technik transformace šablony. Biometrické kryptosystémy jsou založeny na teorii kódů pro opravu chyb, proto je relativně snadné míru zabezpečení vyhodnotit avšak zároveň omezuje užití sofistikovaného párování. Spolehlivost párování je však omezena schopností daného kódu opravit chyby.

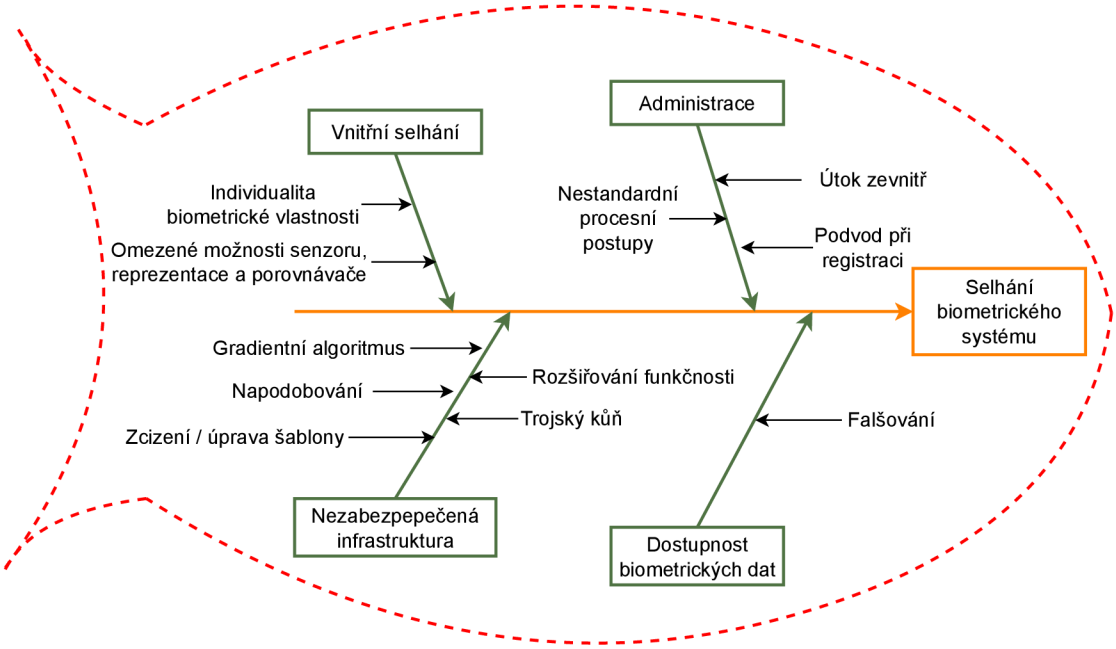
Pro transformaci šablon byla navržena řada technik – tyto techniky lze rozdělit do dvou skupin na základě použité reprezentace šablony: založené na vektorech a založené na zájmových bodech *interest point*).[6]

[6]Pro srovnání jednotlivých technik viz [3].

V prvním případě je šablona reprezentována jako vektor a rozdíly mezi vektory jsou obvykle počítány pomocí euklidovské vzdálenosti. Jedním z hlavních požadavků funkce transformace šablony na základě vektorů (vector based template transformation function) je zachování vzdáleností mezi vektory i po provedení transformace. Jednou z takových technik je biometrické hashování, které bude podrobněji rozebíráno v kap. 2. Díky zvýšené variaci mezi třídami a za současného zachování variací uvnitř tříd, biohashing výrazně zlepšuje výsledky porovnávání, avšak je-li útočníkovi znám klíč, spolehlivost porovnávání se typicky snižuje kvůli kvantizaci rysů a redukci dimenzí.[3]

Transformace šablony založená na zájmových bodech (*interest point based template transformation*) se využívá především u otisků prstů. Otisky prstů jsou nejčastěji reprezentovány pomocí množiny bodů, které se nazývají markanty (*minutia*). Mnoho technik transformace šablony otisků prstů je proto založeno na markantech jako jejich výchozí reprezentace. Navíc, aby bylo možné v transformovaném prostoru použít dostupné srovnávače otisků prstů založené na markantech, je žádoucí mít konečnou reprezentaci také ve formě množiny markantů.[3]

Možné útoky a zranitelnosti biometrického systému jsou znázorněny pomocí tzv. modelu rybí kosti (*fish-bone model*) na obrázku 1.5.



Obr. 1.5: Přehled možných selhání biometrických systémů – [4]

Na nejvyšší úrovni lze příčiny selhání biometrického systém rozdělit do dvou tříd: vnitřní selhání a selhání při útoku narušitele. K vnitřnímu selhání dochází díky vlastním nedostatkům při snímání, extrakci vlastností nebo ve srovnávacích

technologiích stejně jako omezenou rozlišitelností určité biometrické vlastnosti. Při útocích narušitele se vynalézavý hacker (nebo spíše organizovaná skupina) snaží obejít biometrický systém k osobnímu prospěchu. Dále dělíme útoky narušitele do tří typů podle faktorů, které útočníkovi umožňují napadnout bezpečnost systému. Mezi tyto faktory patří administrace, nezabezpečená infrastruktura a dostupnost biometrických dat.[4]

Vnitřní selhání představuje nedokonalosti vlastního biometrického systému, což může vést k nesprávným rozhodnutím tohoto systému. Biometrický systém může při rozhodování dělat dva typy chyb, a to falešné přijetí a falešné odmítnutí (viz kap. 1.1). K vnitřním poruchám může dojít i v případě, že protivník nevyvine žádnou výslovnou snahu systém obejít. Tento typ selhání je znám také jako útok s vynaložením nulového úsilí *(zero-effort attack)*. Představuje vážnou hrozbu, pokud jsou pravděpodobnosti falešného přijetí a falešného odmítnutí vysoké.

Útok administrátora (*administrator attack*), znám také jako útok zevnitř (*insider attack*), zahrnuje zranitelnost plynoucí z nedostatečného administračního zabezpečení biometrického systému. Sem patří integrita registračního procesu (např. platnost přihlašovacích údajů při registraci, tajné dohody mezi stranami – útočníkem a administrátorem systému, nebo legitimním uživatelem, nebo donucení. Dále je sem řazeno zneužití nestandardních procesních procedur.

Nezabezpečená infrastruktura biometrických systémů poskytuje útočníkovi možnost s ní různými způsoby manipulovat, a tím narušovat bezpečnost systému. Infrastruktura biometrického systému se skládá z hardwaru, software a komunikačních kanálů mezi různými moduly. [4]

Dostupnost biometrických dat představuje situaci, kdy útočník může skrytě získávat biometrické charakteristiky legitimního uživatele (například otisky prstů sejmuté z povrchu) a použít je k vytvoření fyzických artefaktů (gumových prstů) biometrických vlastností. Proto, když biometrický systém nemůže rozlišit mezi živou biometrickou prezentací a umělým podvrhem, může útočník obelstít systém poskytnutím falešných vlastností.

2 Biometrický hashing

V rámci této kapitoly bude představen biometrický hashing. Dále budou stručně představeny biohashovací systémy využívající vybrané biometrické charakteristiky. Zaměřujeme se na biometrické hashovací systémy pro rozpoznávání obličejů.

Biometrický hashing je dvoufaktorový autentizační systém, který využívá kromě biometrických charakteristik navíc jedinečný tajný klíč k vytvoření náhodných biometrických šablon pro každého uživatele. V průběhu autentizace je tak potřeba takovému systému předložit jak biometrickou charakteristiku, tak tajný klíč či heslo. Biometrický hashovací systém pak tyto prvky zkombinuje a vytvoří z nich binární biohashovací vektor.

Biometrický hashing[1] je jednou z metod založených na transformaci. Šablona uživatele je vynásobením pseudonáhodnou projekční maticí a kvantizací transformována na binární řetězec. Díky větší mezitřídní variaci (*inter-class variation*) a za současného zachování variace uvnitř tříd (*intra-cass variation*) biohashování významně zlepšuje přesnost při ověřování, oproti klasickým biometrickým systémům, za předpokladu, že tajný klíč je udržen v bezpečí před narušiteli, kterým není znám.[5]

Další výhodou biomerického hashování je jednodušší zrušení transformované šablony změnou příslušného tajného klíče příp. hesla. Tím je odstraněn jeden z bezpečnostních nedostatků biometrických systémů. V případě, že útočník získá biometrickou charakteristiku (vytvořením falešného otisku prstů, neoprávněným pořízením fotografie apod.), není pro oprávněného uživatele možné obstarat si novou. Uživatel pak nemá biometrickou charakteristiku, kterou by dále využíval k přihlašování do systému. Tedy pokud je biometrická charakteristika kompromitována je kompromitována navždy.[1] Navíc uživatel, který se prostřednictvím stejných biometrických dat přihlašuje k různým službám může být pokaždé autentizován různými biohashi, pokud používá jiná tajná hesla – pokud by tedy útočník prolomil biohash uživatele, nezíská tím automaticky přístup ke všem službám, ke kterým se uživatel autentizuje.

Biometrické hashovací systémy jsou jednoduché, a přesto silné metody ochrany biometrických šablon. Biohash je binární a pseudonáhodná reprezentace biometrické šablony. Biometrický hashovací systém má dva vstupy:

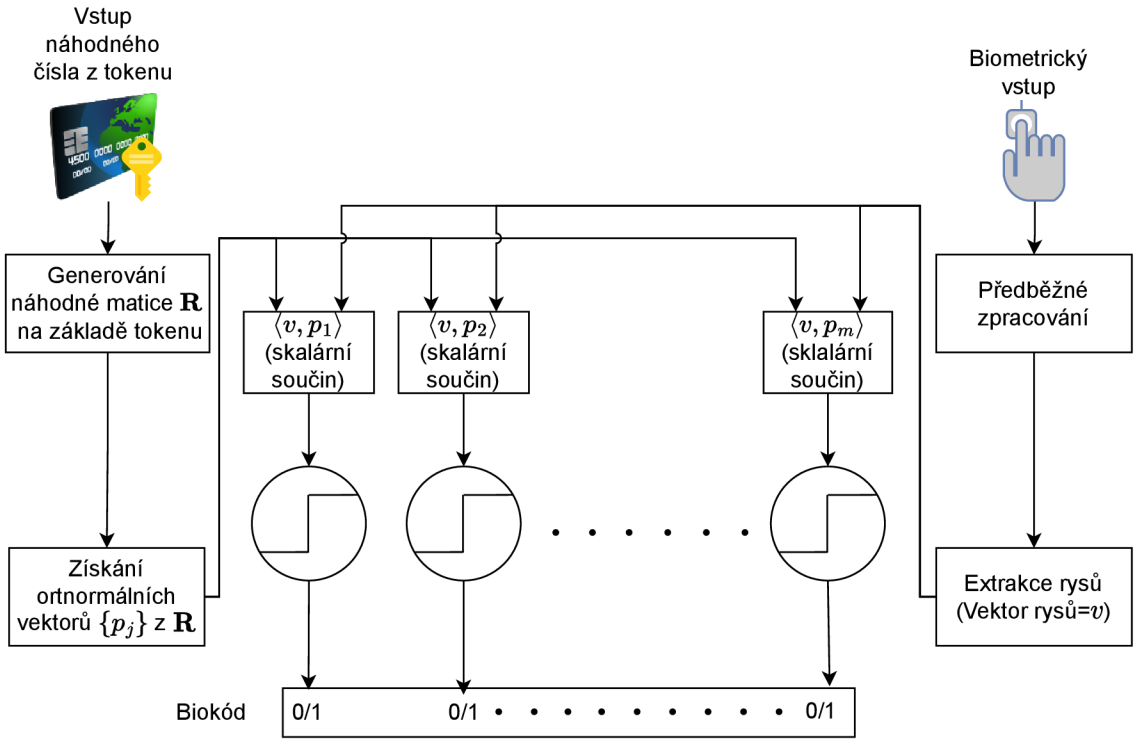
1. biometrickou šablonu a
2. uživatelem specifikovaný tajný klíč či heslo.

Biometrický vektor rysů je transformován do jiného prostoru s využitím pseudonáhodné množiny vektorů, která je generována na základě tajného uživatelského hesla či klíče. Výsledek je převeden do binární podoby, čímž je získán pseudonáhodný bitový řetězec, který je nazýván biohash.[2] Jedinečná a specifická pro každého uživatele

[1] Někdy také biohashing – v práci budou používány oba termíny jako synonyma.

[2] V [2, s. 1361] nazýván biokód. V práci jsou oba termíny používány jako synonyma..

je náhodná projekční matice. Ta může být vytvořena pomocí generátoru pseudonáhodných čísel na základě seedu[3] (tajný klíč specifický pro uživatele), který je uložený na USB tokenu či v mikroprocesoru čipové karty. [5]



Obr. 2.1: Schématický diagram vytvoření biohashe – [2]

Na obrázku 2.1 je zachycen proces vytvoření biohashe neboli biokódu. Probíhají zde dva základní procesy – proces extrakce biometrických rysů (pravá strana obrázku) a proces diskretizace (levá strana obrázku). Proces extrakce biometrických rysů se skládá z fáze akvizice signálu, předběžné zpracování a extrakce rysů. Proces diskretizace sestává ze čtyř kroků:

1. Využití vstupního tokenu pro generování množiny pseudonáhodných vektorů $\{r_i \in \Re^M | \ i = 1, \ldots, m\}$ na základě seedu.
2. Výpočet množiny ortonormálních vektorů $\{p_i \in \Re^M | \ i = 1, \ldots, m\}$ aplikováním Gram–Schmidtova ortogonalizačního procesu na vektory $\{r_i \in \Re^M | \ i = 1, \ldots, m\}$.
3. Výpočet skalárního součinu v, vektoru rysů s každým pseudonáhodným ortonormálním vektorem p_i, tedy $\langle v, p_i \rangle$.

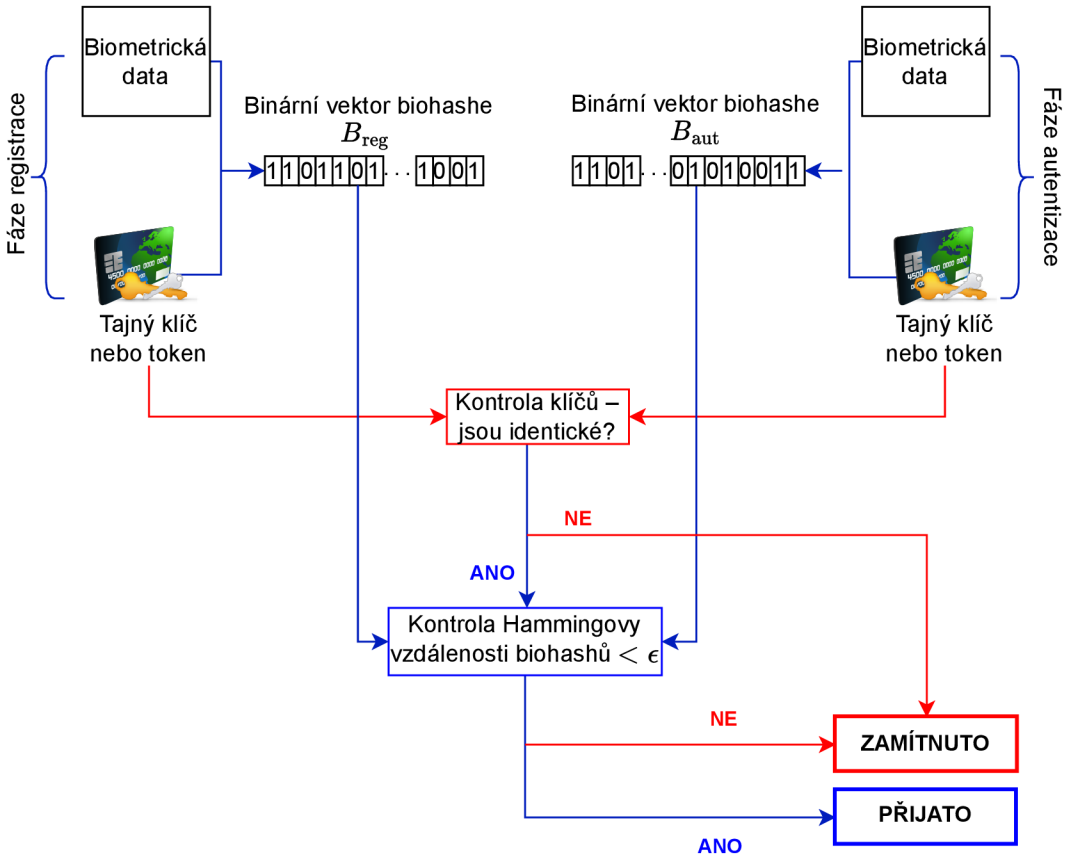
[3]Náhodné číslo, které se používá při inicializaci generátoru pseudonáhodných čísel.

4. Použití prahové hodnoty t pro vytvoření biokódu $\mathbf{b}$, jehož složky jsou získány následovně:

$$\begin{cases} 0, & \text{pokud } \langle v, p_i \rangle \leq t, \\ 1, & \text{pokud } \langle v, p_i \rangle > t, \end{cases} \tag{2.1}$$

kde i leží mezi 1 a m, kde m je délka biohashovacího vektoru $\mathbf{b}$.[2]

Dva biohashe jsou porovnávány pomocí Hammingovy vzdálenosti, jak lze vidět na obrázku 2.2. Hammingova vzdálenost je nejmenší počet pozic, na kterých se řetězce stejné délky daného kódu liší. Hammingova vzdálenost tak označuje počet změn, které je třeba provést pro změnu jednoho řetězce na druhý. U binárních řetězců pak tato veličina označuje počet bitů, ve kterých se daná slova liší a Hammingova vzdálenost je dána vztahem: $d_{\mathrm{H}}(\mathbf{a}, \mathbf{b}) = \sum_{i=1}^{n} |a_i - b_i|$.[6] V ideálním případě by Hammingova vzdálenost mezi biohashi příslušnými k biometrické šabloně jednoho uživatele měla být relativně malá (zde mluvíme o variaci uvnitř třídy). Naopak vzdálenost biohashů, které odpovídají biometrickým šablonám dvou různých uživatelů, by měla být dostatečně velká (označujeme jako variaci mezi třídami).



Obr. 2.2: Schéma biometrického hashovacího systému – [5]

Fáze registrace, jak je znázorněna na obrázku 2.2, zahrnuje tři části – extrakci rysů, náhodnou projekci a kvantizaci. Proces bude přiblížen na systému, který využívá obrázky obličeje a metodu PCA pro extrakci rysů, ačkoliv existují i další metody – Fisherův lineární diskriminant (*Fisher Linear Discriminant – FLD*), zobecnění této metody – lineární diskriminační analýza (*Linear Discriminant Analysis* – LDA), vlnková transformace (*Wavelet Transform* – WT), Fourier–Mellinova transformace apod.[4] Během extrakce rysů dochází k využití obrázků obličeje v trénovací množině, které jsou sbírány během registrační fáze. Množina obsahuje trénovací obrázky obličeje, které patří registrovaným uživatelům $\mathbf{I}_{i,j} \in \Re^{(m \times n)}$, kde $i = 1, \ldots, K$ a K značí počet uživatelů, $j = 1, \ldots L$ a L označuje počet trénovacích obrázků na uživatele. Každý obrázek je reprezentován jako vektor $\mathbf{y} \in \Re^{(mn) \times 1}$. Poté je aplikována analýza hlavních komponent (PCA – *principle component analysis*)[7] na obrázky obličeje v trénovací množině za účelem extrakce rysů:

$$\mathbf{x} = \mathbf{A}(\mathbf{y} - \mu), \tag{2.2}$$

kde $\mathbf{A} \in \Re^{k \times (mn)}$ je PCA maticí natrénovanou na obrázcích obličeje v trénovací množině, μ je průměrný vektor obrázků obličeje, a $\mathbf{x} \in \Re^{k \times 1}$ je vektor obsahující PCA koeficienty. [5]

Během fáze náhodné projekce je generována pseudonáhodná projekční matice $\mathbf{R} \in \Re^{\ell \times k}$, která je využita k transformaci vektorů PCA koeficientů. Koeficienty náhodné projekční matice jsou na sobě nezávislé a rovnoměrně rozložené. Tyto koeficienty jsou generovány na základě Gaussova normálního rozložení s nulovým průměrem a jednotkovým rozptylem s využitím generátoru pseudonáhodných čísel (PRNG – *pseudo-random number generator*) se seedem odvozeným od uživatelského tajného klíče či hesla. Náhodná projekční matice promítá PCA komponenty do ℓ-dimenzionálního prostoru:

$$\mathbf{z} = \mathbf{R}\mathbf{x}, \tag{2.3}$$

kde $\mathbf{z} \in \Re^{\ell \times 1}$ je pomocný biohashovací vektor. [5]

V průběhu kvantizační fáze jsou prvky přechodného biohashovacího vektoru $\mathbf{z}$ převedeny do binární podoby dle zvoleného prahu:

$$\mathbf{b}(k) = \begin{cases} 1, & \text{pokud } \mathbf{z}(k) \geq \beta, \\ 0, & \text{v ostatních případech,} \end{cases} \tag{2.4}$$

kde $\mathbf{b} \in \{0, 1\}^{\ell}$ značí biohashovací vektor uživatele a β označuje kvantizační práh. Po registrační fázi jsou biometrické hashe uloženy do databáze nebo na čipovou kartu.

[4] Je možné použít i kombinace jednotlivých metod. Pro srovnání různých metod použitých v systému na rozpoznávání obličejů viz [8].

V autentizační fázi zašle osoba žádající o přístup do systému obrázek obličeje $\tilde{\mathbf{I}} \in \Re^{m \times n}$ a svůj tajný klíč. Následně systém vypočítá testovací biohashovací vektor pro danou osobu s využitím stejného postupu jako v registrační fázi. Uživatel je autentizován, pokud je Hammingova vzdálenost (d_{H}) nižší než přednastavený práh vzdálenosti ϵ, tedy pokud je splněna následující nerovnice:

$$\sum_{k=1}^{n} \mathbf{b}_{\mathrm{reg}}(k) \oplus \mathbf{b}_{\mathrm{aut}}(k) \leq \epsilon, \tag{2.5}$$

kde $\oplus$ značí binární XOR (exkluzivní disjunkce) operátor. Práh vzdálenosti ϵ je celé číslo z rozsahu 0 až n (počet bitů v biohashi). Práh závisí na návrhu konkrétního systému a je vybrán tak, aby naplňoval požadované hodnoty FAR a FRR (k tomu viz kap. 1.1).

Systém spočítá Hammingovu vzdálenost mezi testovacím biohashovacím vektorem (z autentizační fáze) a biohashovacím vektor deklarovaného uživatele uloženým v databázi (z registrační fáze). Osoba, která žádá o přístup do systému je úspěšně autentizována pokud Hammingova vzdálenost splňuje nerovnici výše, jinak je autentizace zamítnuta.[5]

V následující části budou stručně přiblíženy biometrické hashovací systémy pracující s vybranými biometrickými charakteristikami – obecně lze vyslovit závěr, že systémy se liší především v metodách pro extrakci rysů, ale základní princip biohashovacích systémů zůstává zachován.

2.1 Otisky prstů

Otisky prstů představují preferovanou biometrickou charakteristiku v mnoha systémech, vzhledem k jejich vlastnostem shrnutým v tabulce 1.1 – zejména pak jejich rozlišitelnosti a výkonu systémů, které je používají, ale také poměrně nízké ceně zařízení pro snímání otisků prstů. Většina systémů pro rozpoznávání na základě otisků prstů je založena na párování otisků na základě markantů, což jsou koncové body a rozdvojení hřebenových linií (označujeme také markanty „ukončení“ a „rozdvojení“). Ty zůstávají po celý život neměnné a umožňují klasifikaci otisků prstů.[9]

Kombinace rozpoznávání otisků prstů s ochranou šablon představuje dvě důležitá požadavky na systémy rozpoznávání otisků prstů. Není možné přizpůsobit dva blízké otisky prstů (rotace, posunutí), kvůli zašifrovanému úložišti. Je vyžadována fixní délka vektoru rysů jako vstupu systémů ochrany šablon.

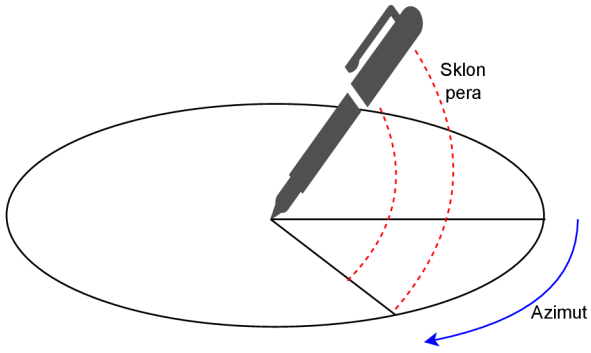
Problematika extrakce rysů a vytvoření vektoru rysů o fixní délce z otisku prstů je dlouho diskutována v odborných kruzích a byla za tímto účelem navržena řada metod. Např. FingerCode [10], který je založen na textuře otisku prstu – neposkytuje

však takovou rozlišitelnost jako markanty. Dále byl navržen kvantizační algoritmus založený na lokální orientaci hřebenů (*ridges*) [11], který je podobný metodě FingerCode. Kromě přístupů založených na textuře otisku byly navrženy také metody extrakce vektoru rysů o fixní délce z markantů otisku (např [12, 9]).

V práci [9] byla navržena metoda založená na umístění markantů a informaci o směru. Získaná šablona o fixní délce pro vzorek otisku prstu poté byla zkombinována s pseudonáhodnými daty, generovaných z klíče specifikovaného uživatelem, použitého jako seed, za účelem generování jedinečného kódu pro každou osobu – k tomu se používá biohashovací systém.

2.2 Rukopis

Rukopis představuje biometrickou behaviorální charakteristiku, tedy není založen na části lidského těla (fyziologické faktory), ale na chování člověka. Na rozdíl od většiny fyziologických charakteristik tak může být tedy relativně proměnný např. v čase, či vlivem jiných okolností (konkrétně u podpisu např. porovnání podpisu v dětství a v dospělosti, změna jména apod.). V biometrických systémech které pracují s rukopisem zkoumáme tato statistická data: vertikální a horizontální pozice tužky $x(t)$ a $y(t)$, tlak, který je vynakládán na hrot tužky $p(t)$, azimut – úhel měřený po směru hodinových ručiček od kolmého průmětu pera do roviny psaní $\Theta(t)$ (*azimuth*) a sklon tužky $\Phi(t)$ (*altitude*). Pro lepší ilustraci relevantních úhlů se kterými se pracuje viz obrázek 2.3. Z těchto statistických údajů vyplývá vektor rysů.



Obr. 2.3: Sklon pera a azimut

Biohashovací algoritmus vypočítá statistický vektor rysů, který obsahuje $k = 69$ prvků – ty vyplývají z měření statistických dat zmíněných výše. Vektor je transformován na hash pomocí funkce, která mapuje hodnoty na určitý interval (*interval mapping function*).[13]

2.3 Rozpoznávání obličejů

V případě biohashovacích systému, které rozpoznávají obrázky obličejů je opět zachován základní princip biometrického hashingu – je provedena extrakce biometrických rysů, je vygenerována pseudonáhodná matice s využitím tokenu, poté je na matici aplikován Gram–Schmidtův algoritmus a následně je provedena náhodná projekce a výsledek je převeden do binární podoby s využitím prahové hodnoty. Opět se však uplatňují jiné metody pro extrakci rysů (některé lze však uplatnit i pro jiné biometrické charakteristiky).

Metody extrakce rysů[5] z obrázků obličeje lze rozdělit do dvou skupin – popis lokálních rysů (LFD – *local feature description*) a analýza holistického podprostoru (HSA – *holistic subspace analysis*). HSA jsou efektivní pro redukci dimenze, zatímco LFD jsou vhodné pro extrakci lokálních rysů. Jako zástupce první skupiny lze uvést Gaborův filtr a lokální binární vzory (*local binary patterns*). Do druhé skupiny lze zařadit již zmíněné metody PCA a LDA.[6] [14]

Gaborův filtr [16] funguje na principu vyhledání určité frekvence v určitém směru okolo bodu nebo oblasti analýzy obrázku. Gaborův filtr je velmi citlivý na rozdíly v osvětlení.

PCA [7] je jednou z nejběžnějších metod určených k redukci dimenze. Základní myšlenkou PCA je najít takovou množinu hlavních komponent (*principal components*), aby bylo možné zohlednit co největší variabilitu původních dat. Redukce dimenze je dosaženo ponecháním pouze prvních pár hlavních komponent transformovaných dat.

LDA [17] hledá podprostor ortogonálních rysů, který odděluje třídy dat (dvě nebo více). LDA nám umožňuje namapovat vzorky dat C–tříd do m-dimenzionálního prostoru $C = m - 1$. Tím dochází k maximalizaci mezitřídní variace a naopak minimalizaci variace uvnitř tříd. Cílem LDA je nalézt projekční váhu W k projekci vysoce dimenzionálních vzorků dat y do proměnné y', která má malou dimenzi: $y' = W^T y$.

[5] Pro srovnání několika metod pro extrakci rysů viz [8].

[6] Srovnání těchto metod je provedeno např. v [15].

3 Komprimované snímání

V rámci této kapitoly bude představen matematický základ pro komprimované snímání a základní pojmy a fakta, na kterých samotné komprimované snímání stojí, jelikož princip komprimovaného snímání se využívá v útocích proti biometrickým hashovacím systémům (jak jsou představeny v [5]).[1] A právě tyto útoky budou realizovány v rámci praktické části diplomové práce. Definice a tvrzení odpovídají definicím obsaženým v [18, 19], příp. [20].

3.1 Základní pojmy a předpoklady

V této podkapitole budou popsány základní matematické nástroje používané při výpočtech komprimovaného snímání jako např. ℓ_p-norma, k-řídký vektor, jednotková koule v normě ℓ_p atd.

Stěžejním pojmem dané problematiky, je samotná ℓ_p norma, proto je vhodné daný termín nejprve definovat, abychom s ním nadále mohli pracovat.

Definice 3.1. *ℓ_p-norma vektoru $\mathbf{x} \in \mathbb{C}^N$ je definována jako*

$$\begin{aligned} \|\mathbf{x}\|_p &:= \left(\sum_{i=1}^{N} |x_i|^p \right)^{\frac{1}{p}} \text{ pro } 1 \leq p < \infty, \\ \|\mathbf{x}\|_p &:= \sum_{i=1}^{N} |x_i|^p \text{ pro } 0 < p < 1, \\ \|\mathbf{x}\|_\infty &:= \max_i |x_i|, \\ \|\mathbf{x}\|_0 &:= |\mathrm{supp}(x)|. \end{aligned} \tag{3.1}$$

Má-li se jednat o normu musí $p \in \langle 1; \infty)$. Případ, kdy se $p = 1$, tedy $\|\cdot\|_1$ představuje součet absolutních hodnot prvků daného vektoru. Případ $\|\cdot\|_0$ představuje počet nenulových složek vektorů, tedy velikost tzv. nosiče vektorů. Velikost nosiče vektoru označujeme $|\mathrm{supp}(\mathbf{x})|$. Nosič vektoru představuje množinu indexů vektoru, v nichž má vektor nenulové hodnoty, danou množinu značíme $\mathrm{supp}(\mathbf{x}) = \{i \mid x_i \neq 0\}$.[18]

Pro lepší představu a ilustraci ℓ_p norem zaveďme pojem jednotková koule v jednotlivých normách.

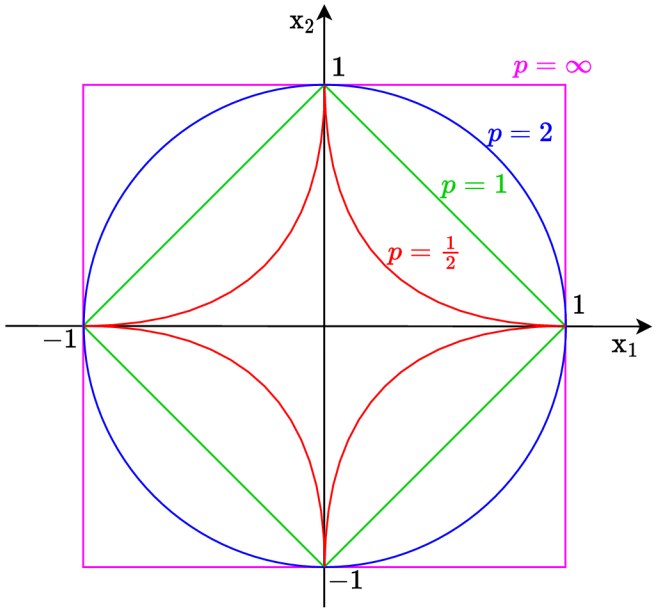
Definice 3.2. *Jednotková koule B_p^N v normě ℓ_p je definována jako*

$$B_p^N := \left\{ \mathbf{x} \in \mathbb{C}^N \middle| \|\mathbf{x}\|_p \leq 1 \right\}. \tag{3.2}$$

Na obrázku 3.1 je znázorněna jednotková koule v různých normách. Z obrázku je patrné, že v případě jednotkové koule v normě ℓ_0, koule kopíruje osy souřadného

[1] Jedná se však o spíše okrajové využití. K využití komprimovaného snímání viz 3.4.

systému. Jak se však p blíží nekonečnu, jednotková koule nabývá tvaru čtverce. Zatímco v případě, kdy se $p = 2$ má jednotková koule kulatý tvar.



Obr. 3.1: Ilustrace jednotkových koulí B_p^2 – inspirováno [18]

Dalším důležitým pojmem je k-řídký vektor. Pro k-řídký vektor platí, že čím větší je k, tím vyšší je počet nenulových složek (resp. tím nižší je počet nulových složek), jak vyplývá z definice 3.3.

Definice 3.3. *Vektor* $\mathbf{x} \in \mathbb{C}^N$ *nazveme* k*-řídkým (*k*-sparse), jestliže platí*

$$\|\mathbf{x}\|_0 \leq k, \ \textit{tedy} \ |\mathrm{supp}(\mathbf{x})| \leq k. \tag{3.3}$$

Pojem relativní řídkost vektoru $\mathbf{x}$ velikosti N, pak označuje poměr $\frac{k}{N}$. Množina $\sum_k := \sum_k^N := \{\mathbf{x} \in \mathbb{C}^N \mid \|\mathbf{x}\|_0 \leq k\}$ označuje všechny k-řídké vektory délky N. [18]

Reálné signály však místo nulových složek obsahují sice malé, ale stále nenulové hodnoty, z tohoto důvodu definujeme chybu aproximace.

Definice 3.4. *Chyba nejlepší aproximace vektoru* $\mathbf{x} \in \mathbb{C}^N$ k*-řídkým vektorem v normě* ℓ_p *(best k-term approximation error) je definována jako*

$$\sigma_k(\mathbf{x})_p := \sigma_k^N(\mathbf{x})_p := \inf_{\mathbf{z} \in \sum_k^N} \|\mathbf{x} - \mathbf{z}\|_p. \tag{3.4}$$

Pro $T \subset \{1, \ldots, N\}$ označíme $\mathbf{x}_T \in \mathbb{C}^N$ vektor odvozený z $\mathbf{x} \in \mathbb{C}^N$ tak, že prvky na pozicích, které patří do množiny T zachováme a ostatní vynulujeme. Komplement

T označíme $T^c = \{1, \ldots, N\} \setminus T$. Chybu nejlepší aproximace $\sigma_k(\mathbf{x}_p)$ lze vyjádřit jako p-normu vektoru, který vznikne z $\mathbf{x}$ odstraněním k složek o největší velikosti, lze tedy psát $\sigma_k(\mathbf{x}_p) = \min_{T \subset \{1,\ldots,N\}, |T| \leq k} \|\mathbf{x}_{T^c}\|_p$. [18]

3.2 Řídká řešení systému lineárních rovnic

Výše uvedené definice využijeme pro řídká řešení systémů lineárních rovnic. Využíváme je tedy, pokud chceme nalézt co možná nejřidší vektor $\mathbf{x}$ (tedy obsahující co možná nejvíc nulových složek) splňující rovnici $\mathbf{Ax} = \mathbf{y}$. V takovém případě řešíme úlohu:

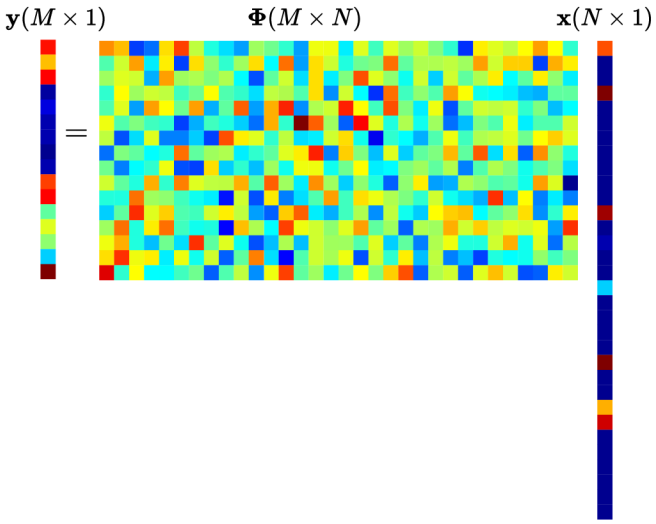
$$\begin{aligned} &\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{Ax} = \mathbf{y}, \text{neboli} \\ &\min_{\mathbf{x}} |\text{supp}(\mathbf{x})| \text{ vzhledem k } \mathbf{Ax} = \mathbf{y}, \end{aligned} \tag{3.5}$$

známe vektor $\mathbf{y} \in \mathbb{C}^m$ (signál, měření) a matici $\mathbf{A} \in \mathbb{C}^{m \times N}$ (nazýváme měřicí matice). Ilustrativní schéma úlohy lze vidět na obrázku 3.2.

Požadavek $\mathbf{Ax} = \mathbf{y}$ není striktní a povoluje se malá odchylka δ (např. kvůli zašumění signálu), tedy:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \|\mathbf{Ax} = \mathbf{y}\|_p \leq \delta, \tag{3.6}$$

přičemž většinou uvažujeme $p = 2$.[18]



Obr. 3.2: Schéma nedourčeného systému rovnic $\mathbf{Ax} = \mathbf{y}$ s řídkým vektorem $\mathbf{x}$ – tmavě modré prvky reprezentují nulové složky – [18]. Barvy kódují absolutní hodnoty elementů matic a vektorů.

Pojem přípustná řešení, příp. přípustné reprezentace vektoru y zahrnuje všechna $\mathbf{x}$, která splňují rovnici $\mathbf{Ax} = \mathbf{y}$. Aby bylo možné zrekonstruovat původní $\mathbf{x}$ ze $\mathbf{z}$, musí matice $\mathbf{A}$ splňovat určité požadavky.

3.2.1 Požadavky na měřicí matici A

Zde se zaměříme na požadavky, které jsou kladeny na měřicí matici $\mathbf{A}$, aby byla zajištěna jednoznačnost řešení rovnice 3.5.

Definice 3.5. *Číslo* spark($\mathbf{A}$) *(doslova „jiskra“) je nejmenší počet sloupců matice* $\mathbf{A}$*, které jsou lineárně závislé (tedy netvoří bázi):*

$$\text{spark}(\mathbf{A}) = \min_{\mathbf{z} \in \ker \mathbf{A}, \mathbf{z} \neq 0} \|\mathbf{z}\|_0. \tag{3.7}$$

Čím menší je číslo spark, tím řidší musí být vektor $\mathbf{x}$, aby byla zajištěna jedinečnost řešení.

Tvrzení 3.1. *Má-li soustava* $\mathbf{Ax} = \mathbf{y}$ *řešení* $\mathbf{x}$ *splňující*

$$\|\mathbf{x}\|_0 < \frac{\text{spark}(\mathbf{A})}{2}, \tag{3.8}$$

pak $\mathbf{x}$ *je nejřidší možné řešení a jiné řešení se stejnou řídkostí neexistuje.*

Vzhledem ke skutečnosti, že nalezení spark($\mathbf{A}$) je výpočetně náročné. Pro řešení problému 3.5, využíváme spíše tzv. vzájemnou koherenci (*mutual coherence*).[18]

Definice 3.6. *Vzájemná koherence matice* $\mathbf{A}$ *je definována jako největší absolutní normalizovaný skalární součin dvou různých sloupců matice* $\mathbf{A}$*,*

$$\mu(\mathbf{A}) = \max_{1 \leq j,k \leq N, j \neq k} \frac{\left| \mathbf{a}_j^\top \mathbf{a}_k \right|}{\|\mathbf{a}_j\|_2 \cdot \|\mathbf{a}_k\|_2}, \tag{3.9}$$

$\mathbf{a}_j$ *označuje j-tý sloupec matice* $\mathbf{A}$

Vzájemná koherence nám určuje míru lineární závislosti dvou sloupců matice. Pro matice $m \times N$, kde $m < N$ plné hodnosti se vzájemné koherence pohybuje v rozmezí:

$$\sqrt{\frac{N-m}{m(N-1)}} \leq \mu(\mathbf{A}) \leq 1. \tag{3.10}$$

Tvrzení 3.2. *Tvrzení 3.1 lze upravit způsobem níže, jelikož pro libovolnou matici* $\mathbf{A}$ *platí:*

$$\text{spark}(\mathbf{A}) \geq 1 + \frac{1}{\mu(\mathbf{A})} \Rightarrow \|\mathbf{x}\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{A})}\right), \tag{3.11}$$

pak $\mathbf{x}$ *je nejřidší možné řešení a je jediné takové. Řešení je možné dosáhnout* ℓ_1*-minimalizací.*

3.3 ℓ_1 relaxace

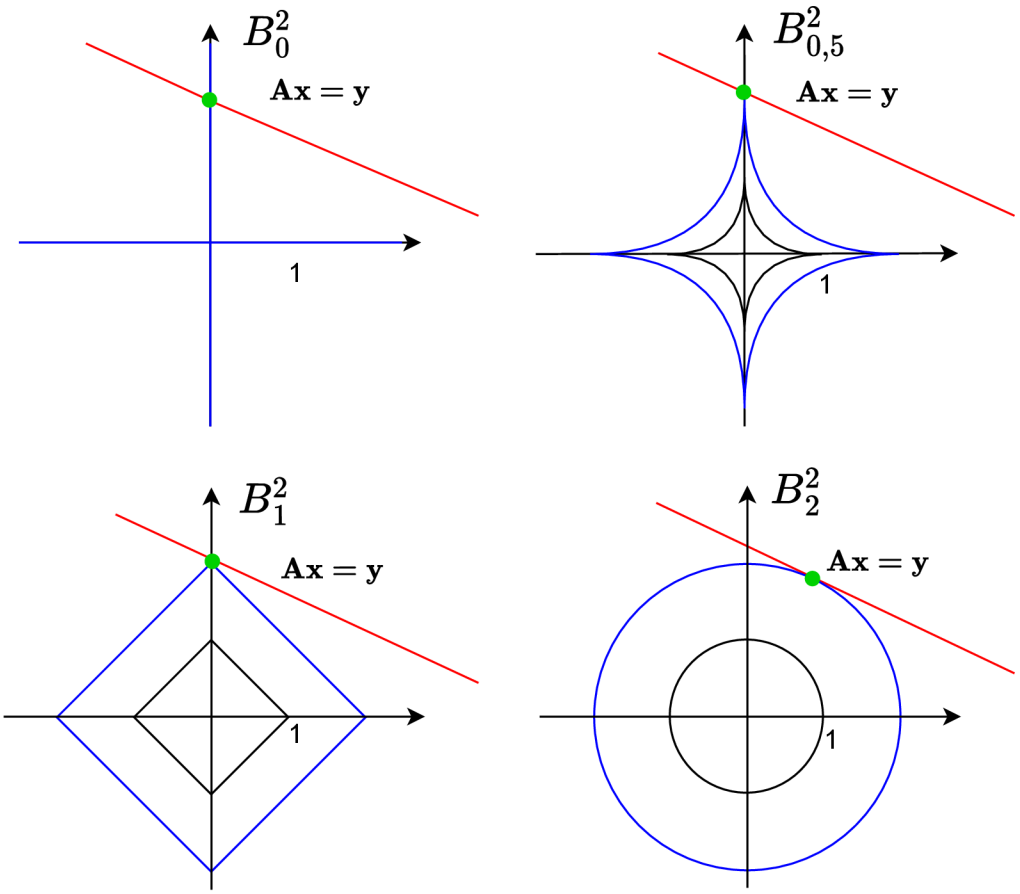
ℓ_0-norma není konvexní, tato skutečnost má za následek, že pro řešení problému 3.5 nelze využít algoritmy konvexní optimalizace [21]. Normy ℓ_p však konvexní jsou pro $p \geq 1$, jak lze vidět i na obrázku 3.3. Za určitých podmínek (viz 3.3.1) je pak možné pro řešení problému 3.5 využít normu ℓ_1 (jakožto nejbližší konvexní normu). [18]

Poté tedy řešíme úlohu:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{Ax} = \mathbf{y}. \tag{3.12}$$

Pro zašuměná data, pak stejně jako v případě ℓ_0-normy povolujeme odchylku od přesného měření:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \|\mathbf{Ax} = \mathbf{y}\|_2 \leq \delta.$$



Obr. 3.3: Postupné zvětšování jednotkových koulí B_p^2 pro nalezení řešení rovnice $\mathbf{Ax} = \mathbf{y}$ v různých normách – [18]

3.3.1 Podmínky ekvivalence ℓ_0-minimalizace a ℓ_1-minimalizace

Mezi podmínky, které je nutné splnit, abychom mohli prohlásit ℓ_0- a ℓ_1-minimalizaci za ekvivalentní řadíme vlastnost nulového prostoru (NSP – null space property) [22] a vlastnost zeslabené isometrie (RIP – restricted isometry property) [23].

Nulový prostor je definován jako $N = \{\mathbf{x} \in \mathbb{R}^n \mid \mathbf{Ax} = \mathbf{0}\}$ a jedná se vlastně o jádro matice $\mathbf{A}$, tedy ker$\mathbf{A}$.

Definice 3.7. *Matice $\mathbf{A} \in \mathbb{C}^{m\times N}$ splňuje NSP řádu k s konstantou $\gamma \in (0,1)$, platí–li*

$$\|\boldsymbol{\eta}_T\|_1 \leq \gamma \|\boldsymbol{\eta}_{T^c}\|_1, \tag{3.13}$$

pro všechny množiny $T \subset \{1,\ldots,N\}$ a pro všechny vektory $\boldsymbol{\eta} \in \ker\mathbf{A}, |T| \leq k$.

Isometrie je lineární zobrazení, které zachovává délku vektorů. Pro definici vlastnosti zeslabené isometrie RIP, využíváme konstantu zeslabené isometrie δ_k.

Definice 3.8. *Konstanta omezené isometrie δ_k matice $\mathbf{A} \in \mathbb{C}^{m\times N}$ je nejmenší číslo takové, že platí:*

$$(1-\delta_k) \leq \frac{\|\mathbf{Az}\|_2^2}{\|\mathbf{z}\|_2^2} \leq (1+\delta_k), \tag{3.14}$$

pro všechny vektory $\mathbf{z} \in \sum_k^N$. Matice $\mathbf{A}$ splňuje RIP řádu k s konstantou δ_k, pokud $\delta_k \in (0,1)$.

Zeslabení isometrie je dvojí:

1. Omezujeme se na všechny podmatice $\mathbf{A}$ o k sloupcích.
2. Nepožadujeme přesnou isometrii, nýbrž povolujeme malou odchylku δ_k. [18]

3.4 Komprimované snímání

Základní myšlenkou komprimovaného snímání (*compressed sensing* i *compressive sampling*) je teorie, že je možné opětovně zrekonstruovat určité signály příp. obrázky s využitím menšího počtu vzorků nebo měření, než kolik využívají tradiční metody rekonstrukce signálu. Komprimované snímání tak přichází s jiným přístupem – ve vhodné reprezentaci signál rovnou snímá lineárně a neadaptivně (za předpokladu přibližné řídkosti), a to pouze tolikrát, kolikrát je třeba.[19] Tradiční metody totiž uplatňují Nyquistův–Shannonův teorém, který říká, že vzorkovací frekvence musí být alespoň dvojnásobkem maximální frekvence daného signálu (tzv. Nyquistova frekvence).[24] Komprimované snímání stojí na dvou základních principech – řídkosti (*sparsity*) a inkoherenci (*incoherence*). Řídkost se týká signálů zájmu (*signal of interest*) a inkoherence se týká modality snímání (*sensing modality*).

Hlavní myšlenkou řídkosti je, že informační hodnota signálu spojitého v čase může být výrazně menší než šířka pásma. U diskrétního signálu, který závisí na stupních volnosti, pak těchto stupňů může být také výrazně méně než (konečná) délka tohoto signálu. Kompresní snímání využívá skutečnosti, že mnoho přirozených signálů je řídkých nebo komprimovatelných ve smyslu, že mají stručné reprezentace (*concise representation*), jsou-li vyjádřeny ve vhodně volené bázi Ψ.

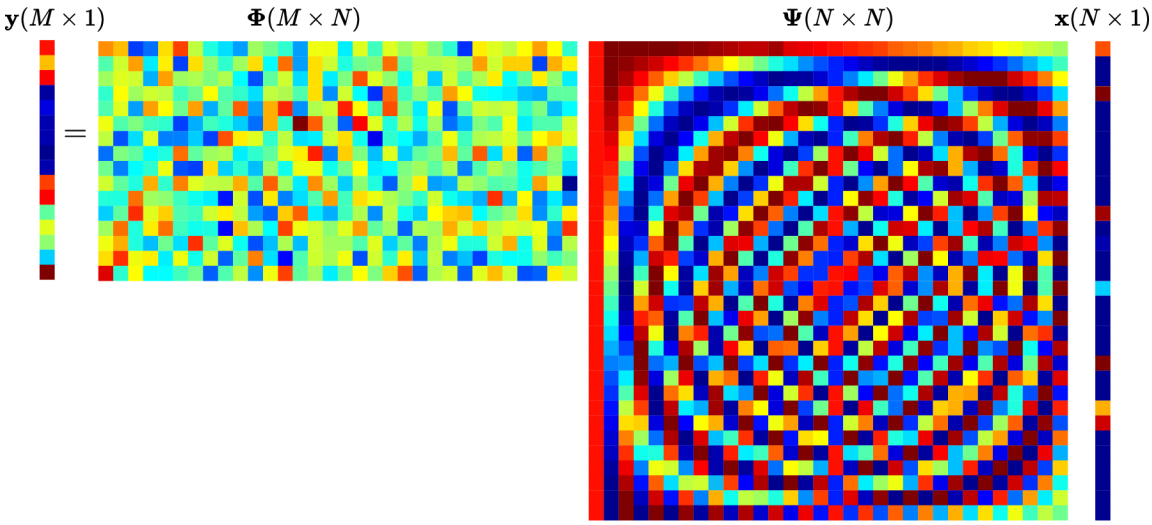
Inkoherence stojí na myšlence, že objekty, mající řídkou reprezentaci v bázi Ψ, musí být možné rozšířit v prostoru, ve kterém byly pořízeny. Inkoherence tak říká, že na rozdíl od signálů zájmu, mají vzorkovací průběhy signálu extrémně husté reprezentace v Ψ. [20]

Cílem komprimovaného snímání je nalezení tzv. řídkého řešení soustavy lineárních rovnic:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_0 \text{ vzhledem k } \mathbf{A}\mathbf{x} = \mathbf{y}. \tag{3.15}$$

V praxi je však potřeba sáhnout po určité formě aproximace – např. s využitím hladových (*greedy*) algoritmů[25] nebo pomocí ℓ_1-minimalizace:

$$\min_{\mathbf{x}} \|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{A}\mathbf{x} = \mathbf{y}. \tag{3.16}$$



Obr. 3.4: Ilustrace situace při komprimovaném snímání – [19]. Matice Φ je náhodná matice, zatímco matice Ψ je nejčastěji uspořádaná matice, v našem případě se jedná o DCT matici. Vektor $\mathbf{x}$ je řídký vektor a jeho tmavě modré složky reprezentují nulové prvky.

Předpokládejme signál $\mathbf{z}$, který můžeme vyjádřit jako $\mathbf{z} = \mathbf{\Psi x}$, přičemž $\mathbf{x}$ představuje k-řídký signál. Provedeme malý počet neadaptivních měření, měření budou mít charakter skalárních součinů se signálem: $\mathbf{y} = \mathbf{Pz} = \mathbf{P\Psi x}$. $\mathbf{P}$ představuje snímací matici o rozměrech $m \times N$. Jednotlivé složky vektoru $\mathbf{y}$ vznikají jako lineární kombinace vzorků signálů a představují výsledky měření. Počet měření $m \ll N$.

Rekonstrukce signálu je již nelineární a tedy výrazně časové náročnější. Průběh rekonstrukce technikami ℓ_1-minimalizace je následující:

$$\mathbf{x}^1 := \text{argmin}\|\mathbf{x}\|_1 \text{ vzhledem k } \mathbf{y} = \mathbf{P\Psi x}. \tag{3.17}$$

Samotný signál $\mathbf{z}$ následně získáme jako $\mathbf{z} = \mathbf{\Psi x^1}$.[19]

Měřicí matice $\mathbf{P}$ však musí být vybrána tak, aby $\mathbf{A} := \mathbf{P\Psi}$ vyhovovala vlastnosti omezené isometrie (RIP – *restricted isometry property*) [23] s vhodnou konstantou δ. RIP je totiž podmínkou úspěšné ℓ_1-rekonstrukce. Vhodné měřicí matice se uvažují ve tvaru $\mathbf{P} = \mathbf{R\Phi}$, kde $\mathbf{\Phi}$ je matice $N \times N$. $\mathbf{R}$ je matice, která vznikne ponecháním m náhodně vybraných řádků z jednotkové matice $N \times N$. Matice $\mathbf{R}$ tak tedy funguje jako podvzorkování $\mathbf{\Phi}$ a náhodný výběr řádků z $\mathbf{\Phi}$ se řídí rovnoměrným rozdělením pravděpodobnosti. Pro matici $\mathbf{A}$ tak tedy platí: $\mathbf{A} = \mathbf{R\Phi\Psi}$.

Pro vhodný výběr $\mathbf{\Phi}$ a m tak, aby $\mathbf{A}$ vyhovovala RIP, lze využít koherenci μ. Pro případ, kdy matice $\mathbf{A} = \mathbf{\Psi\Phi}$ a je tedy součinem dvou ortonormálních bází platí:

$$\mu(\Phi, \Psi) = \sqrt{n} \cdot \max_{1 \leq k,j \leq n} |\langle \phi_k, \psi_j \rangle|, \tag{3.18}$$

a platí že $\mu(\Phi, \Psi) \in [1, \sqrt{n}]$. [20]

Tvrzení 3.3. *Nechť je dán signál* $\mathbf{z}$*, který má v* Ψ *k-řídkou reprezentaci* $\mathbf{x}$*. Pak řešení* ℓ_1*-minimalizace, kde* $\mathbf{y}$ *jsou měření, je současně s vysokou pravděpodobností nejřidší možné, pokud je zvoleno*

$$m \geq C \cdot \mu^2([\Phi, \Psi]) \cdot k \cdot N \cdot \log n, \tag{3.19}$$

pro určitou kladnou konstantu C*. Počet měření na řídkosti závisí lineárně. [19, 20]*

V [20] jsou formulovány tři závěry:

1. Úloha koherence je zcela zřejmá – čím menší je koherence, tím méně vzorků je zapotřebí k rekonstrukci signálu.
2. Změřením pouze libovolné množiny m koeficientů, přičemž množina může být mnohem menší, než by odpovídalo velikosti signálu, nedojde k žádné ztrátě informace. Pokud je koherence $\mu(\Phi, \Psi)$ rovna nebo blízká jedné, potom postačí $k \log n$ vzorků místo n.

3. Signál f je možné přesně obnovit na základě redukované množiny dat pomocí minimalizace konvexní funkce, která nepředpokládá žádnou znalost počtu nenulových souřadnic x, jejich lokaci ani jejich amplitudy, u kterých předpokládáme, že jsou zcela neznámé. Pouze aplikujeme algoritmus a je-li signál dostatečně řídký, dojde k jeho přesnému obnovení.

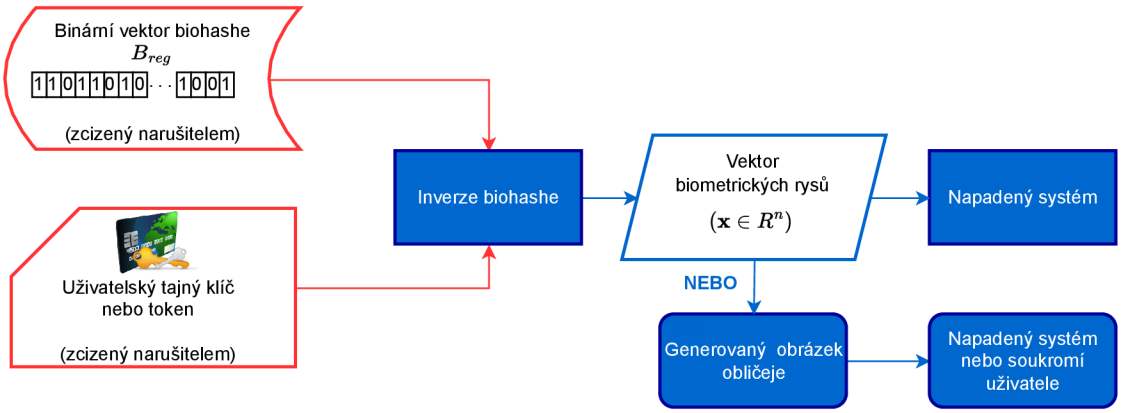
Princip komprimovaného snímání má řadu využití, jak je uvedeno v [20], zejména pak v oblasti komprese dat, kódování kanálu, řešení inverzních problémů a akvizice dat. Využívá se např. pro zajištění rychlého snímání obrazu v magnetické rezonanci, dále pro účely odstranění šumu ze signálu, odstranění rozmazání apod.

4 Útoky na biometrický hashing

V rámci této kapitoly budou představeny útoky na biometrický hashing využívající pro autentizaci obrázky obličeje. Kapitola obsahuje obecný popis útoků a jejich algoritmů a představuje základ pro následující kapitolu věnovanou samotné implementaci a realizaci jednotlivých útoků. V článku [5] byly představeny čtyři metody, které umožňují zrekonstruovat biometrické rysy anebo celý obrázek z biohashe a jedna metoda založená na hledání nejbližších biometrických dat (tedy obrázků obličeje) v databázi (duhový útok – *rainbow attack*). Dvě z rekonstrukčních metod jsou založeny na rekonstrukci signálu pomocí 1-bitového komprimovaného snímání. Autoři článku se snaží poukázat na výrazné bezpečnostní riziko biohashovacích systémů, které spočívá v možnosti provedení útoku s využitím komprimovaného snímání, jak bylo představeno v kapitole 3.

Největší hrozbou pro schopnost biometrických hashovacích systémů chránit šablony je útok nalezením předlohy (*pre-image attack* [26]). Útok spočívá v rekonstrukci dostatečně podobného vektoru rysů, který poskytuje biohash blízký původnímu biohashi. Biometrické hashování není odolné vůči tomuto útoku, pokud je útočníkovi znám tajný klíč, který je použit pro generování biohashe.

Na obrázku 4.1 je ilustrován inverzní útok na biohash (k problematice vytváření biohashe odkazujeme na obrázky 2.1 a 2.2). Útočník, který má k dispozici biohashovací vektor uživatele a odpovídající tajný klíč, je schopen provést inverzi biohashe a získat vektor rysů s reálnými hodnotami. Takový vektor rysů lze použít k provedení přímého útoku na systém a získat tak neoprávněný přístup k systému. Kromě toho si útočník může vygenerovat obrázek obličeje (či jiné biometrické charakteristiky), který lze použít k provedení útoku na systém, ale také k narušení soukromí uživatele.



Obr. 4.1: Inverze biohashe – [5]

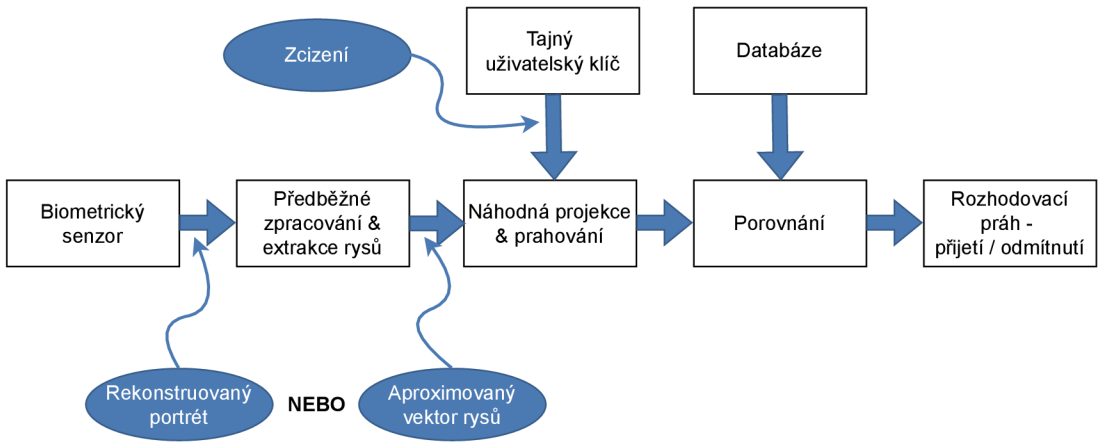
4.1 Navržené metody pro aproximaci rysů z biohashů

V této části budou popsány metody vniknutí[1] útočníků do systému pomocí rekonstrukce vektoru biometrických rysů z biohashů. Zde zavedeme významy jednotlivých symbolů, které budou dále používány:

- $\mathbf{b}$ představuje biohashovací vektor legitimního uživatele, který je získán útočníkem za účelem provedení neoprávněného vniknutí pomocí aproximace rysů,
- $\mathbf{R}$ je náhodná projekční matice specifická pro každého uživatele, útočník matici zná,
- $\mathbf{x}$ je původní vektor biometrických rysů, ze kterého vznikl vektor $\mathbf{b}$ – vektor není útočníkovi znám,
- $\hat{\mathbf{x}}$ představuje vektor rysů (nebo vzor – *preimage*), který je aproximován z inverze $\mathbf{b}$.

Přičemž platí, že $\hat{\mathbf{x}}$ nemusí nutně korespondovat s platným vektorem biometrických rysů $\mathbf{x}$. Ovšem s jeho využitím je možné vyprodukovat biohashovací vektor, který útočníkovi umožní neoprávněný přístup do biometrického systému. Předpokládáme, že k vniknutí do sytému může dojít dvěma způsoby ještě před fází náhodné projekce. Útočník buď systému poskytne digitální obrázek obličeje (zrekonstruovaný obrázek), a to ještě před fází extrakce rysů, nebo použije aproximovaný vektor rysů jako vstup do fáze náhodné projekce (viz obrázek 4.2).

K výpočtu pravděpodobnosti úspěchu takového útoku na biohashovací systém lze formálně použít vzorec $P(d(\text{sign}(\mathbf{R}\hat{\mathbf{x}}), \mathbf{b}) < \epsilon)^1$, kde $d(\cdot)$ značí Hammingovu vzdálenost mezi dvěma biohashi. [5]



Obr. 4.2: Útoky na biohashovací systém – [5]

[1] Vniknutím rozumíme získání přístupu do biometrického systému předložením falsifikovaných autentizačních dat.

V následujících podkapitolách jsou představeny jednotlivé metody k získání vektoru rysů $\hat{\mathbf{x}}$, tak jak jsou popsány v [5]. Vektor rysů $\hat{\mathbf{x}}$ útočníkovi umožňuje nelegitimní přístup do biometrického systému, pokud zároveň útočník má k dispozici biohash $\mathbf{b}$ a transformační parametry.

4.1.1 1-bitové komprimované snímání

1-bitové komprimované snímání představuje efektivní akvizici řídkých signálů pomocí lineárních měřicích systémů, kde pro každou měřenou hodnotu se ukládá pouze 1 bit. Na biometrické hashování založené na náhodné projekci lze pohlížet podobně jako na 1-bitové komprimované snímání. Je-li použitý práh při kvantizaci projektovaného signálu roven 0 (tak že je zachována hodnota kvantizační funkce sign$(\cdot)$ signálu), potom každý bit biohashe odpovídá hodnotě kvantizační funkce sign$(\cdot)$ skalárního součinu vektoru vlastností $\mathbf{x}$ a vektoru měření (v biometrickém hashování se jedná o každý řádek náhodné projekční matice $\mathbf{R}$):

$$b_i = \text{sign}(R_i\mathbf{x}), \text{ nebo také lze psát } \mathbf{b} = \text{sign}(\mathbf{R}\mathbf{x}), \tag{4.1}$$

z naměřených hodnot je tak ponecháno pouze znaménko.

Při konzistentní rekonstrukci z 1-bitových měření, se na jednotlivá měření pohlíží jako na omezení daná kvantizační funkcí sign$(\cdot)$, které je nutné respektovat při rekonstrukci signálu. Aby bylo možné získat řídké řešení, je norma ℓ_1 vektoru rysů při rekonstrukci minimalizována (viz kap. 3.2). Při analýze PCA koeficientů obrázků obličejů bylo zjištěno, že většina koeficientů nabývá malých velikostí a pouze okolo 25 % z nich postačuje k získání 70 % celkové energie (k tomu viz [5, s. 7–8]).

Při provádění rekonstrukcí u netriviálních řešení je třeba se vypořádat s nejednoznačností hodnoty (*amplitude ambiguity*), právě proto, že známe pouze znaménko. Z tohoto důvodu bylo zavedeno omezení pro energii (*energy constraint*), tak že rekonstruovaný signál musí ležet na jednotkové ℓ_2-kouli:

$$\|\mathbf{x}\|_2 = \left(\sum_i x_i^2\right)^{\frac{1}{2}} = 1. \tag{4.2}$$

Signál na jednotkové kouli, který se blíží naměřeným hodnotám lze získat takto:

$$\begin{aligned} &\hat{\mathbf{x}} = \arg\min_x \|\mathbf{x}\|_1, \text{ tak že} \\ &\text{sign}(\mathbf{R}\hat{\mathbf{x}}) \equiv \mathbf{b}, \text{ a} \\ &\|\mathbf{x}\|_2 = 1. \end{aligned} \tag{4.3}$$

Jelikož jsou měření komprimovaného snímání kvantována do jednotlivých bitů, je jasné, že rozsah signálu je ztracen a není tedy hned zřejmé, že zbývající informace postačí k rekonstrukci signálu, přestože tomu tak je (viz [27]).

1-bitové komprimované snímání pomocí lineárního programování [28] a binární iterace s konstantním prahem (*binary iterative hard tresholding* – BIHT) [29] jsou dvě rekonstrukční metody, které byly v [5] implementovány za účelem pořízení inverzních obrazů biohashů a nalezení vektorů biometrických vlastností, které umožní získat biohashovací vektory přijatelné pro ověřovací systém (tj. s menší vzdáleností od původního biohashového vektoru, než je hodnota prahu).

Rekonstrukce z 1-bitového komprimovaného snímání pomocí lineárního programování

Studie [28] ukázala, že $\mathbf{x}$ lze přesně odhadnout z extrémně kvantovaného vektoru měření v rovnici 4.1. Je důležité poznamenat, že vektor $\mathbf{b}$ neobsahuje žádné informace o velkosti vektoru $\mathbf{x}$ a je možné obnovit pouze normalizovaný vektor $\frac{\mathbf{x}}{\|\mathbf{x}\|_2}$. Signál může být přesně obnoven vyřešením následujícího konvexního minimalizačního algoritmu (*convex minimization program*):

$$\begin{aligned} &\min \|\hat{\mathbf{x}}\|_1, \text{ tak že} \\ &\mathbf{BR}\hat{\mathbf{x}} \geq \mathbf{0} \text{ a} \\ &\|\mathbf{R}\hat{\mathbf{x}}\|_1 = m, \text{ kde } \mathbf{B} = \text{diag}(\mathbf{b}). \end{aligned} \tag{4.4}$$

První podmínka $\mathbf{BR}\hat{\mathbf{x}} \geq \mathbf{0}$ zajišťuje konzistenci řešení s původním biohashovacím vektorem. Je definována vztahem $\langle R_i, \hat{\mathbf{x}} \rangle \geq 0$ pro $i = 1, 2, \ldots m$, kde R_i je i-tý řádek náhodné projekční matice $\mathbf{R}$.

Druhá podmínka v původní definici problému 4.3 obsahuje ℓ_2 normu, která je kvadratickou podmínkou a může být nahrazena ℓ_1-normou, čímž se optimalizace stane lineárním problémem. Podmínka $\|\mathbf{R}\hat{\mathbf{x}}\|_1 = m$ má zabránit tomu, aby program vracel nulové řešení. Podmínka je lineární, proto může být definována vztahem $\sum_{i=1}^{m} b_i \langle R_i, \hat{\mathbf{x}} \rangle$, kde m je délka biohashovacího vektoru. Proto je 4.4 konvexní minimalizační problém a lze jej řešit s využitím lineárního programování (viz algoritmus 4.1). [5]

Algoritmus 4.1 Aproximace vektoru biometrických rysů $\hat{\mathbf{x}}$ s využitím lineárního programování

Vstup: $\mathbf{b}, \mathbf{R}$
Výstup: $\hat{\mathbf{x}}$
vypočítej $\mathbf{A}$, tak že $\mathbf{A}\hat{\mathbf{x}} \geq \mathbf{0}$ s použitím $\mathbf{b}$ a $\mathbf{R}$
vypočítej $\mathbf{A}_{\text{eq}}$, tak že $\mathbf{A}_{\text{eq}}\hat{\mathbf{x}} = m$ s použitím $\mathbf{R}$
nastav f pro výpočet $\hat{\mathbf{x}}$ v ℓ_1 normě

Binární iterace s konstantním prahem

Binární iterace s konstantním prahem představuje modifikaci iterace s pevným prahem (*iterative hard tresholding* – IHT) [30], což je algoritmus pro zpracování reálných hodnot navržený pro komprimované snímání k rekonstrukci k-řídkých signálů. IHT se skládá ze dvou kroků

1. gradientní sestup s cílem dosáhnout nejmenších čtverců $\frac{\|\mathbf{y}-\mathbf{R}\mathbf{x}\|_2^2}{2}$, přičemž v každé iteraci IHT se stanoví $\mathbf{a}^{l+1} = \mathbf{x}^l + \mathbf{R}^T(\mathbf{y} - \mathbf{R}\mathbf{x})$
2. druhý krok využívá modelu řídkého signálu při výběru K prvků o největší velikosti z $\mathbf{a}^{l+1}$

BIHT modifikuje první krok tohoto algoritmu a minimalizuje požadavek na konzistenci. S daným inicializačním odhadem $\mathbf{x}^0 = \mathbf{0}$ a 1-bitovým měřením $\mathbf{b}$ v každé iteraci l, BIHT počítá:

$$\begin{aligned} \mathbf{a}^{l+1} &= \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T\left(\mathbf{b} - \operatorname{sign}\left(\mathbf{R}\mathbf{x}^l\right)\right) \\ \mathbf{x}^{l+1} &= \eta_K\left(\mathbf{a}^{l+1}\right), \end{aligned} \tag{4.5}$$

kde τ je skalár, který určuje velikost kroku gradientního sestupu a funkce η_K počítá nejlepší K-řídkou aproximaci $\mathbf{a}^{l+1}$. [5]

Algoritmus 4.2 Aproximace vektoru biometrických rysů $\hat{\mathbf{x}}$ s využitím BIHT

Vstup: $\mathbf{b}, \mathbf{R}, K$

Výstup: $\hat{\mathbf{x}}$

```
inicializace x^0 samými nulami
while |b − sign(Rx^l)|_1 > 0 do
    a^{l+1} = x^l + (τ/2) R^T (b − sign(Rx^l))
    setřiď prvky a^{l+1} a nastav všechny na nulu, kromě největších K prvků
end while
nastav x̂ ← a^{l+1}
```

4.1.2 Rekonstrukce obrázku obličeje

Pokud je k extrakci rysů použita ortogonální transformační matice, je možné provést inverzi procesu extrakce rysů jednoduše s využitím pseudoinverze transformační matice a obrázek obličeje lze jednoduše zrekonstruovat. PCA využívá ortogonální transformaci – sloupce PCA matice jsou na sebe navzájem kolmé – a lze tak zrekonstruovat obrázek obličeje $\hat{\mathbf{y}}$ z $\hat{\mathbf{x}}$ s využitím vlastnosti ortogonální matice $\mathbf{A}^\dagger = \mathbf{A}^T$:

$$\hat{\mathbf{y}} = \mathbf{A}^T\hat{\mathbf{x}} + \mu, \tag{4.6}$$

kde $\mathbf{A} \in \Re^{k\times(mn)}$ je PCA matice, $\mathbf{A}^\dagger$ je pseudo inverze $\mathbf{A}$ a μ je průměrný obličejový vektor. [5]

5 Praktická část práce

V této kapitole bude popsána a přiblížena implementace problematiky probírané v předcházejících kapitolách teoreticky. Nejprve bude popsána implementace vytváření biohashe a následně metody pro aproximaci rysů z biohashů, které budou následně využity k provedení útoku na biometrický hashovací systém. K implementaci byl využit programovací jazyk MATLAB verze R2022b.

5.1 Vybraná databáze fotografií

Pro potřeby následné implementace a testování bylo třeba zvolit dostupnou databázi obrázků. Databází dostupných na internetu je poměrně velké množství. Při výběru jsme se zaměřili na databáze z akademického prostředí[1] či na známou databázi FERET (přístupnou prostřednictvím organizace NIST), se kterou pracuje řada dosavadních odborných článků. Vzhledem k tomu, že zpřístupnění FERET databáze vyžaduje odeslání e-mailu konkrétního znění a instalaci programu pro stažení databáze – bylo od ní upuštěno. Navíc je databáze příliš veliká, cca 8,5 GB a obrázky v ní nejsou nijak tříděny.

Pro implementaci byla použita databáze [33]. Databáze byla zvolena zejména proto, že je spravována vyučujícím z ČVUT na rozdíl od zbytku databází, které pochází ze zahraničí. Databáze je také jednoduše přístupná a pro účely práce dostačující. Databáze obsahuje fotografie obličejů 72 osob (z toho dvanáct žen), přičemž pro každou osobu je pořízeno dvacet fotografií. Fotografie mají rozměry 180×200 pixelů a liší se zejména v těchto aspektech – umístění osoby na obrázku, osvětlení, případně velmi drobné změny ve výrazu či např. natočení hlavy, vzdálenost subjektů od kamery apod. Z databáze bylo vybráno dvacet subjektů, tak aby zastoupení mužů a žen bylo rovnoměrné (tj. deset mužů a deset žen). Dvacet obrázků příslušných konkrétní osobě bylo rozděleno na tzv. trénovací a testovací množinu (vždy po deseti obrázcích). Trénovací se využívá pro potřeby výpočtu biohashe ve fázi registrace a testovací pro potřeby výpočtu biohashe ve fázi autentizace.

5.2 Vytváření biohashe

V této sekci bude popsáno vytváření biohashe ve fázi registrace a následně autentizace. Pro teoretický základ viz kap. 2. Velikost biohashů je možno volit až do

[1] Například MIT má na svém webu řadu dostupných databází viz `https://web.mit.edu/emeyers/www/face_databases.html`.

velikosti vektoru rysů minus jedna (tedy do 198), autorka však po vzoru článku [5] velikost volí vždy jako mocninu dvou (tedy aktuálně maximálně 128 bitů).

5.2.1 Fáze registrace – enrollment

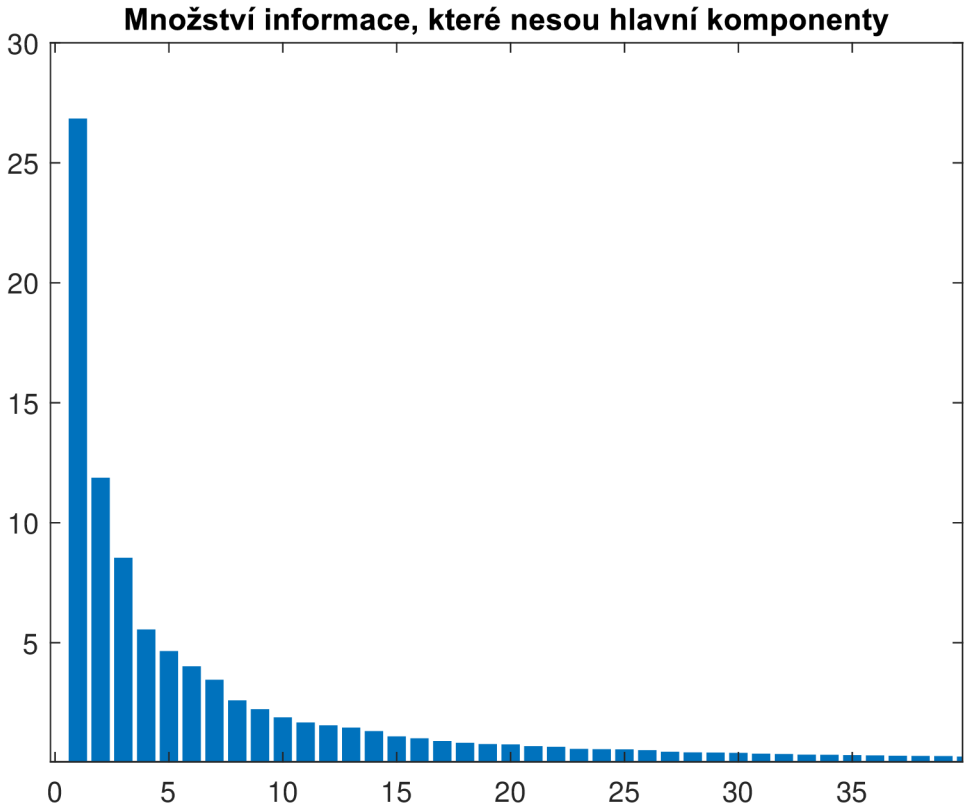
V této kapitole bude popsána fáze registrace, ta je implementována v souboru `bhEnroll.m`. Implementace v tomto souboru představuje databázi biohashů, které byly vytvořeny při registraci uživatele.

Nejprve jsou načteny obrázky ze složky `train2`. Obrázky jsou načítány v cyklu v abecedním pořadí. Každý obrázek je nejprve převeden na typ double, poté je převeden na černobílý a následně je pomocí funkce `reshape` zkonvertován na dlouhý vektor (vektor má délku $180 \times 200 = 36000$. Jakmile je obrázek takto upraven a převeden na vektor, je uložen do matice, kam jsou postupně ukládány všechny vektory reprezentující obrázky ze složky `train2`. Výsledkem této fáze je matice `ORIMAT`, kde každý řádek reprezentuje jeden obrázek. Matice má rozměry 200×36000. V této části kódu jsou také uloženy důležité hodnoty do proměnných, konkrétně – `noImPerUser`, `noUs`, `bh_no`.

Výpis 5.1: bhEnroll.m – načtení obrázků do matice

```
%% Reading data from file ...\faces\

myFolder = '.\facesDat\train2';
filePattern = fullfile(myFolder,'*.jpg');
jpegFiles = dir(filePattern);

noImPerUser=10;                          %no. of img per user
noUs = size(jpegFiles,1)/noImPerUser;    %no. of users
bh_no = size(jpegFiles,1);          %no. of images = no. of BH

ORIMAT(bh_no,36000)=zeros;       % allocation of matrix

for k = 1:length(jpegFiles)
  baseFileName = jpegFiles(k).name;
  fullFileName = fullfile(myFolder,baseFileName);
  img = imread(fullFileName);
  % creating a matrix from vectors representing each image
  % transfering orig. img to type double and black and white
  ORIMAT(k,:)=reshape(im2gray(im2double(img)),1,[]);
end
```

Za účelem získání vektoru rysů je potřeba na matici původních obrázků `ORIMAT` aplikovat PCA metodu (k tomu viz vzorec 2.2). V MATLABu je přímo k dispozici funkce `pca`, která nám v matici `coeff` vrací v jednotlivých sloupcích hlavní komponenty. Matice `score`, která je také výstupem funkce `pca`, představuje reprezentaci matice `ORIMAT` v prostoru PCA. Matice `explained` nám vrací množství informace, které nesou jednotlivé hlavní komponenty (viz obr. 5.1).[2] Pro účely ověření správnosti spočítaných komponent byla provedena zpětná rekonstrukce obrázku z PCA pomocí matic `coeff` a `score`.



Obr. 5.1: Množství informace v prvních 40 hlavních komponentách

Rekonstrukce se provádí vynásobením matice `score` a transponované matice `coeff'` a přičtením průměrného obrázku (matice `mu`) získaného z matice `ORIMAT`. Při implementaci rekonstrukce je uživatel vyzván k zadání čísla obrázku, který má být rekonstruován a uživatelem zvolený obrázek je následně rekonstruován. K tomu, abychom takovýto obrázek mohli zobrazit, je třeba jej zpětně převést z vektoru na matici o rozměrech původního obrázku (tedy 200×180) – k tomuto účelu slouží metoda `getReshaped`. Metoda převede vybraný řádek (odpovídá číslu obrázku zvole-

[2]Pro názornost je na obrázku zobrazeno pouze prvních čtyřicet komponent z dvou set, další komponenty již nesou zanedbatelné množství informace.

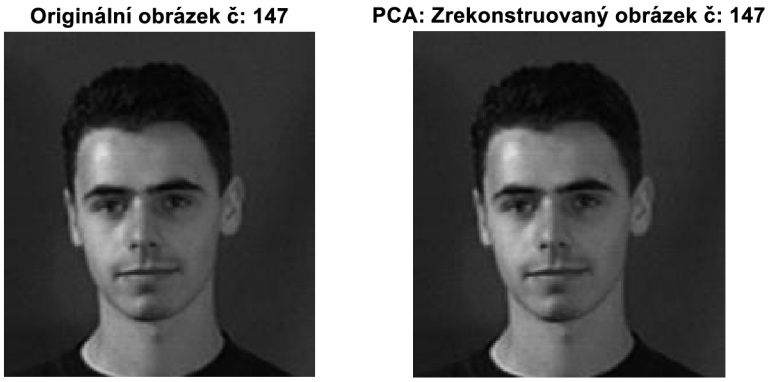
ného uživatelem) zadané matice na matici o rozměrech původního obrázku. Výsledky jsou poté zobrazeny pomocí standardní funkce `imshow` – viz obrázek 5.2.

Výpis 5.2: bhEnroll.m – PCA metoda a rekonstrukce

```
%% PCA function - uses SVD by default
[coeff, score, ~, ~, explained, mu] = pca(ORIMAT);

%% Reconstruction from PCA
REC = score*coeff' + mu;

prompt = "Zadejte číslo obrázku (1-200): ";
nIm = input(prompt);

RECIMG = getReshaped(REC,nIm,m,n);
ORIIMG = getReshaped(ORIMAT,nIm,m,n);
```

Výpis 5.3: getReshaped.m

```
%% Function for reshaping chosen vector to image
function [IMAGE] = getReshaped(MATRIX,nIm,m,n)
%MATRIX - matrix from which we choose a row to reshape
% nIm - which picture (row) of matrix we reshape
% m,n - dimensions to which we want to reshape
    IMAGE = reshape(MATRIX(nIm,:),m,n);
end
```



Obr. 5.2: Původní obrázek a zrekonstruovaný pomocí metody PCA s využitím všech 199 hlavních komponent

Řádky matice `score` reprezentují vektory rysů pro jednotlivé obrázky – proto je matice `score` uložena pod názvem `FEATUREVEC` – pro jednodušší práci dále v kódu je uložena délka vektoru rysů, tj. druhý rozměr matice `FEATUREVEC`.

Uživatel je dále vyzván k zadání velikosti biohashe (ta může být max. 128). Pokud uživatel zadá číslo menší než dva proměnná `bh_size` se přepočítá na dva a v případě, že uživatel zadá číslo větší než 128 je jeho hodnota přepočítána na 128. Hodnota se uloží do proměnné `bh_size`. Následně uživatel zadá požadovanou procentní shodu biohashů – v tomto kroku se nastavuje práh systému. Ten je získán přepočtem hodnoty zadané uživatelem na maximální Hammingovu vzdálenost, kterou mohou mít dva biohashe. Tento parametr se však uplatní až v další fázi implementace (fáze autentizace a aproximace vektoru rysů).

Výpis 5.4: bhEnroll.m – nastavení velikosti biohashe a prahu systému

```
1  %% Saving the biohash vector size as variable
2  prompt = "Počet složek biohashe (2, 4, 8, 16, 32, 64, 128)?";
3  bh_size = input(prompt);
4
5  if bh_size < 2
6      bh_size = 2;
7  elseif bh_size > 128
8      bh_size = 128;
9  end
10
11 %% Setting treshold
12 prompt = "Zadejte požadovanou procentní shodu bh (0-100): ";
13 % percentage match of bh_enroll a bh_aut
14
15 tres=fix(bh_size*((100-input(prompt))/100));
16 % recalculate the percentage match to get the no. of bits
17 % in which bh can differ (HD) - rounded towards zero
```

Dále je vygenerována náhodná projekční matice – ta se vytvoří ze seedu (tokenu) specifického pro každého uživatele. V našem případě token odpovídá pořadí uživatele. Pomocí funkce `rng` řídíme funkce pro generování náhodných čísel (tímto způsobem získáme vždy pro konkrétní token stejnou matici). Parametr `"twister"` určuje použitý generátor náhodných čísel – v tomto případě se jedná o *Mersenne Twister*. Náhodná projekční matice každého uživatele má rozměry `bh_size`×`FEATUREVEC_dim` (tedy pro případ, kdy má biohash 128 bitů jsou rozměry 128×199). Náhodné projekční matice jednotlivých uživatelů jsou opět ukládány do jedné jediné matice

pod sebe. Matice náhodných projekčních matic všech uživatelů má poté rozměry ((počet uživatelů · velikost biohashe) × dimenze vektoru rysů).

Výpis 5.5: bhEnroll.m – výpočet náhodné projekční matice

```
%% Creating random proj. matrix based on the seed (token)
% noUS*bh_size x feature_vec_dim
for j = 1:noUs
  rng(j,"twister");
  R((j-1)*bh_size+1:j*bh_size,:) = randn(bh_size,...
        FEATUREVEC_dim);
end
```

Vynásobením projekční matice s příslušným vektorem rysů získáme přechodný biohash (viz rovnice 2.3). Je však třeba zajistit, aby vektor rysů konkrétního uživatele byl násoben odpovídající náhodnou projekční maticí tohoto uživatele. Takto vzniklé přechodné biohashe jsou ukládány do sloupců.

Na matici obsahující dočasné biohashe ve sloupcích je poté aplikována funkce `sign0Mat`. Funkce pracuje obdobně jako standardní funkce `sign`, avšak místo aby vracela jedničky, nuly a minus jedničky vrací pouze nuly a jedničky. Pokud je vstup menší než zadaný práh (v našem případě nula), vrací nulu, jinak vrátí jedničku (viz rovnice 2.4). Výsledkem je matice obsahující finální biohashe ve sloupcích. Takto vzniklých biohashů je dvě stě (tj. počet uživatelů · počet obrázků na uživatele).

Výpis 5.6: bhEnroll.m – výpočet biohashe

```
%% Intermediate biohash vector z=Rx
for j = 1:noUs
  intBH(:,(j-1)*(bh_no/noUs)+1:j*bh_no/noUs) = (...
            R((j-1)*bh_size+1:j*bh_size,:)*...
            FEATUREVEC((j-1)*(bh_no/noUs)+1:j*bh_no/noUs,:)');
end

%% Binarization of biohash vector - sign function
bhENROLL=sign0Mat(intBH,0);      % treshold = 0
```

Aby nebylo nutné tento kód opakovaně spouštět, jsou na konci kódu vypočítané veličiny uloženy (funkce `save`) a v dalších souborech vždy načteny pomocí funkce `load`. Tímto způsobem ukládáme proměnné – `intBH`, `bhENROLL`, `R`, `FEATUREVEC`, `coeff`, `ORIMAT`, `mu`, `m`, `n`, `tres`.

5.2.2 Fáze autentizace – authentication

Fáze autentizace je řešena pomocí kódu uloženého v souboru `bhAuth.m`. Fáze autentizace probíhá obdobně jako fáze registrace. Při autentizaci se však využívají obrázky ze složky `test2`. Nejprve jsou načteny proměnné vytvořené v `bhEnroll.m` – matice `R`, matice `bhENROLL` a hodnota prahu systému `tres`. Následně jsou načteny obrázky ze složky `test2`, tyto obrázky jsou upraveny a uloženy do matice `TESTMAT`. Proces vytváření biohashe je obdobný jako ve výše popsaném případě. Obrázek 5.3 znázorňuje rekonstrukci původní fotografie pomocí PCA metody.

Obr. 5.3: Původní obrázek a zrekonstruovaný PCA– autentizační fáze

Jakmile jsou vypočítány autentizační biohashe (uložené v `bhAUTH`), jsou porovnávány s již uloženými biohashi spočítanými ve fázi registrace (uložené v `bhENROLL`). Porovnávány jsou vždy odpovídající 10-tice (tzn. deset biohashů, které náleží jednomu uživateli). Při porovnávání se počítá Hammingova vzdálenost mezi dvěma vektory (viz vzorec 2.5). V rámci jedné skupiny deseti vektorů se vždy porovnává jeden vektor z matice `bhAUTH` se všemi deseti z matice `bhENROLL`[3] – uložena je pouze nejnižší Hammingova vzdálenost. Výsledkem tohoto procesu je vektor Hammingových vzdáleností, který má délku dvě stě bitů (vektor bude dále využit pro výpočet FRR, kdy budou hodnoty porovnávány s nastaveným prahem).

[3]K procesu porovnávání viz obrázek v příloze A.1 – obrázek odpovídá velikosti biohashe 64 a počtu uživatelů deset.

Výpis 5.7: bhAUTH.m – počítání Hamminovy vzdálenosti mezi uloženými a vypočítanými biohashi

```
%% Hamming distance
hd (bh_no,1)=zeros;

for i = 1:bh_no
    hd(i,:)=bh_size;
    for j = (floor((i-1)/noUs)*noUs)+1:(floor(..
            (i-1)/noUs)+1)*noUs
        hdj=nnz(bhAUTH(:,i)-bhENROLL(:,j));
            % function nnz - returns no. of nonzero elements
        if hd(i,:)>hdj
            hd(i,:)=hdj;
        end
    end
end
```

5.3 Aproximace vektoru rysů

V této části budou popsány útoky na biometrický hashovací systém. Při útoku jsou aproximovány vektory rysů, které lze využít dvěma způsoby:

1. aproximovaný vektor je předložen systému a vypočítá se nový biohash nebo,
2. aproximovaný vektor je použit k rekonstrukci portrétu oprávněného uživatele.

Při útoku předpokládáme, že útočník zná náhodnou projekční matici i uložené biohashe – k tomu viz kap. 4. Implementace útoků je obsažena v souboru `bhAtt.m`. Nejprve jsou načteny potřebné proměnné a dimenze, které budou dále využívány. Poté je provedena aproximace vektoru rysů, a to pomocí dvou metod – BIHT a lineárního programování. U obou metod je třeba vždy počítat aproximaci z biohashů a jim odpovídající části matice (tzn. počítat s biohashi a maticí jednoho uživatele).

5.3.1 Metoda BIHT – Binary Iterative Hard Tresholding

V této části bude popsána implementace metody BIHT. Implementace vychází z algoritmu 4.2. Po načtení potřebných proměnných je provedena metoda BIHT, tato metoda je v cyklu aplikována na veškeré biohashe uložené v systému. Uživatel je nejprve vyzván k zadání řídkosti, tj. počtu nenulových hodnot. Řídkost může být maximálně 199 (velikost vektoru rysů). Samotná metoda BIHT je implementována v souboru `bihtDP.m`.

Výpis 5.8: bhAtt.m – aplikace metody BIHT

```
%% BIHT
prompt="Zadejte požadovanou řídkost vektoru (0-199): ";
% sparsity = no. of nonzero values
k=input(prompt);
BIHTX (bh_no,x_dim) = zeros;    %allocations
HDBIHT (bh_no, 1) = zeros;
noIt (bh_no,1) = zeros;

for j = 1:noUs
  for i = (j-1)*(bh_no/noUs)+1:j*bh_no/noUs
    Rbiht = R(((j-1)*bh_size+1):j*bh_size,:);
            [BIHTX(i,:),HDBIHT(i,:),noIt(i,:)]=bihtDP(...
                bhENROLL(:,i), Rbiht,k);
  end
end
```

Nejprve jsou nastaveny základní parametry využívané touto metodou (dimenze). Poté je vytvořena funkce (pomocí tzv. *function handleru*), která vrací hodnotu funkce `signOMat` součinu matice $\mathbf{R}$ s jakýmkoliv vektorem.

Stěžejní část programu představuje cyklus `while`, který představuje vlastní metodu BIHT. Nejprve je nutné si nastavit vstupní parametry. Nastavíme si proto iteraci na hodnotu 0, maximální počet iterací je nastaven na 100. Poté v souladu s algoritmem 4.2 inicializujeme vektor $\mathbf{x}$ tím, že jej naplníme nulami. Práh τ nastavíme na hodnotu 0,001. V každé iteraci cyklu zkoumáme dvě podmínky. Nejprve jestli je norma rozdílu vektoru $\mathbf{b}$ a součinu $\mathbf{Rx}$ větší než nula. Současně musí být splněna i druhá podmínka – že počet iterací nepřesáhl maximální počet iterací. V rámci každé iterace je spočítán gradientní krok $\mathbf{a}$, ten se spočítá jako: $\mathbf{a}^{l+1} = \mathbf{x}^l + \frac{\tau}{2}\mathbf{R}^T\left(\mathbf{b} - \text{sign}\left(\mathbf{R}\mathbf{x}^l\right)\right)$. Absolutní hodnoty prvků vektoru $\mathbf{a}$ jsou seřazeny sestupně, dále je ponecháno pouze k největších prvků a zbytek je nastaven na hodnotu nula. Nakonec se $\mathbf{x}$ nastaví na $\mathbf{a}$, spočítá se Hammingova vzdálenost a algoritmus přejde do další iterace. Výsledkem BIHT je aproximovaný vektor, vektor Hammingových vzdáleností a počet provedených iterací.

Výpis 5.9: Metoda BIHT – cyklus while

```
%% BIHT while cycle
while (norm(b-RsignOM(xAprox))>0)&&(i<maxiter)
    % gradient step a
    a=xAprox+(tau/2)*R'*(b-RsignOM(xAprox));

    % Best K-term = function eta_n
    [trash, aidx] = sort(abs(a), 'descend');
        %sorts absolute values of a in descending order;
        %aidx = coresponding index to the value
        a(aidx(k+1:end)) = 0; %keeps only largest k components

    % Update xAprox
    xAprox=a;

    %Measure HD to original 1bit measurements
    hdBIHT=nnz(b-RsignOM(xAprox));
    i=i+1;  %next iteration
end
```

5.3.2 Metoda lineárního programování

Nyní bude popsána aproximace vektoru rysů pomocí lineárního programování. Implementace vychází z algoritmu 4.1. Metoda je opět volána z hlavního programu útočníka `bhAtt.m`.

Výpis 5.10: bhAtt.m – aplikace lineárního programování

```
%% LINPROG
LPX (bh_no,x_dim) = zeros;  %allocation
HDLP (bh_no, 1) = zeros;

for j = 1:noUs
  for i = (j-1)*(bh_no/noUs)+1:j*bh_no/noUs
    Rlp = R(((j-1)*bh_size+1):j*bh_size,:);
    [LPX(i,:),HDLP(i,:)]=linprogDP(bhENROLL(:,i),Rlp);
  end
end
```

Samotná metoda `linprogDP` je implementována ve stejnojmenném souboru s příponou `.m`. Metoda nám vrací aproximované vektory rysů do matice `LPX`, kde každý

řádek obsahuje jeden aproximovaný vektor rysů. Dále metoda vrací vektor vypočítaných Hammingových vzdáleností mezi uloženým biohashem a biohashem spočítaným z aproximovaného vektoru rysů.

Úloha, kterou řešíme lineárním programováním je popsána v rovnici 4.4. Tuto úlohu je však potřeba přeformulovat do takové podoby, abychom mohli využít standardní matlabovskou funkci `linprog`. Funkce je definována jako:

$$\min_x f^T \cdot x \text{ tak, že} = \begin{cases} A \cdot x \leq b, \\ A_{eq} \cdot x = b_{eq}, \\ lb \leq x \leq ub. \end{cases} \tag{5.1}$$

Aby bylo možné tuto funkci použít, bylo tudíž nutné přeformulovat problém v rovnici 4.4. Norma ℓ_1 představuje součet absolutních hodnot prvků vektoru – k tomu viz rovnici 3.1. Tedy v našem případě formulujeme problém takto:

$$\arg\min_x \sum_{i=1}^{N} |x_i| \text{ takové, že} = \begin{cases} \mathbf{BR}\hat{\mathbf{x}} \geq \mathbf{0}, \\ \mathbf{1}^T \cdot \mathbf{BR}\hat{\mathbf{x}} = m, \text{ kde } m \text{ je velikost biohashe.} \end{cases} \tag{5.2}$$

Sumu lze realizovat jako násobení vektoru s transponovaným vektorem jedniček. Absolutní hodnotu lze vyjádřit jako rozdíl dvou nenulových hodnot. Vycházíme z předpokladu, že $\forall x \in \mathbb{R}$ platí, že $x = x' - x''$ kde, $x', x'' \geq 0$. Zavedeme vektor $\mathbf{y}$, který je dvojnásobné velikosti jako vektor $\hat{\mathbf{x}}$ a odečtením druhé poloviny tohoto vektoru od první získáme hledaný vektor $\hat{\mathbf{x}}$. Součin $\mathbf{BR}$ představuje matici $\mathbf{A}$ z definice funkce `linprog` (viz rovnice 5.1). Součin $\mathbf{1}^T \cdot \mathbf{BR}$ pak představuje matici $\mathbf{A}_{eq}$. Dále je třeba tyto matice adekvátně prodloužit, jelikož používáme vektor $\mathbf{y}$ dvojnásobné velikosti než původní $\hat{\mathbf{x}}$ – budeme tedy pracovat s maticí $\mathbf{A}$, která bude složena z matic $\mathbf{A}'$ a $-\mathbf{A}'$ (analogicky pro $\mathbf{A}_{eq}$). Navíc je třeba omezující podmínku ve tvaru $\geq$ vynásobit -1, jelikož funkce linprog je definována pouze pro podmínky ve tvaru $\leq$ nebo $=$. Finální podoba našeho problému tak vypadá následovně[4]

$$\arg\min_y(\mathbf{1}^T \cdot \mathbf{y}) \text{ takové, že} = \begin{cases} -\left[\mathbf{A}'\right| - \mathbf{A}'] \, \mathbf{y} \leq \mathbf{0}, \\ \mathbf{1}^T \left[\mathbf{A}'\right| - \mathbf{A}'] \, \mathbf{y} = m \\ \mathbf{y} \geq 0. \end{cases} \tag{5.3}$$

Nejprve jsou nachystány základní parametry funkce. Je vypočítána matice `B`, což je čtvercová matice, která obsahuje na diagonále biohashovací vektor. Dále jsou uloženy potřebné dimenze bihashe a matice `R` a jako účelová funkce (*objective function*) `f` je uložen vektor $\mathbf{1}^T$.

[4]Srovnej s definicí funkce `linprog` 5.1.

Dále jsou stanoveny potřebné proměnné, se kterými bude volána funkce `linprog` – tedy veličiny `A, b, Aeq` a `beq`. Navíc aby bylo zajištěno, že dílčí x (tj. obě poloviny vektoru `y`) budou větší než nula, je k matici `A` přidána čtvercová jednotková matice, která nám toto zajistí. A parametr `b` musí tedy být adekvátně prodloužen o další nulové prvky.

Výpis 5.11: linprogDP.m - nastavení parametrů

```
function [xApprox, hdLP] = linprogDP (bh,R)
% bh - biohash, R - random projection matrix

options = optimoptions('linprog','Display','none');
% options - enables to stop linprog from displaying msg.

% Calculating matrix B = diag(b)
B=diag(bh);  % B [bh size x bh size]

%Saving dim. of vector x as param. (= no. of columns of R)
n=size(R,2);

%Saving dim. of vector bh as param
m=size(bh,1);

%% Objective function f
f=ones(2*n,1);
```

Výpis 5.12: linprogDP.m – omezující podmínky

```
%% 1st condition BRx>=0
preA=B*R;   %preA [bh size x feature vector size]
% modify matrix preA  to be able to apply linprog
A = -[preA -preA; eye(2*n)]; %sets lower bound to 0
b=zeros(2*n+m,1);

%% 2nd condition ||Rx||1 = m - can be written as
%sum i=1,m (bi<Ri,x> = m) => sum i=1,m <bi*Ri, x> = m
% => 1' * BRx = m
preAeq = ones(1,m)*B*R;
Aeq = [preAeq -preAeq];
beq=m;      % m = size of bh
```

Na konci funkce `linprogDP.m` je volána matlabovská funkce `linprog` se všemi nadefinovanými proměnnými a je proveden výpočet finálního vektoru (tj. odečtení druhé poloviny vektoru od první). Proces je ukončen výpočtem Hammingových vzdáleností od původního biohashe a uložením těchto hodnot do vektoru Hammingových vzdáleností.

Výpis 5.13: linprogDP.m – volání funkce linprog a získání hledaného vektoru

```
% using linprog
[yApprox] = linprog(f,A,b,Aeq,beq,[],[],options);

% first half of the vector
yFin1=yApprox(1:(size(yApprox,1)/2),1);

% second half of the vector
yFin2=yApprox((size(yApprox,1)/2)+1:size(yApprox,1),1);

% subtraction
xApprox = yFin1 - yFin2;

%% Computing Hamming distance from original bh
RsignOM=@(inp) signOMat((R*inp),0);
hdLP = nnz (bh-RsignOM(xApprox));
```

5.3.3 Rekonstrukce biohashů a obrázků uživatelů

Rekonstrukce biohashů

Aproximované vektory je možné použít dvěma způsoby – k výpočtu nového biohashe a k rekonstrukci obrázku. Nejprve jsou v `bhAtt.m` rekonstruovány biohashe z aproximovaných vektorů rysů metodami BIHT a lineárního programování. Biohashe jsou počítány ze všech aproximovaných vektorů a ukládány do matice, kde každý sloupec odpovídá jednomu biohashi.

Systém počítá Hammingovu vzdálenost mezi biohashi, aby bylo možné vyhodnotit, zda bude útočník s takto vygenerovaným vektorem do biohashovacího systému vpuštěn či nikoli. Rozhodování je prováděno na základě prahu systému. Pokud je práh nižší než Hammingova vzdálenost útočníkem vygenerovaného biohashe od biohashe uloženého, útočník do systému není vpuštěn, v opačném případě ano.

Výpis 5.14: bhAtt.m – rekonstrukce biohashů

```
% BIHT biohashes
for j = 1:noUs
  BIHTINTBH(:,(j-1)*(bh_no/noUs)+1:j*bh_no/noUs) = (R...
            ((j-1)*bh_size+1:j*bh_size,:)*BIHTX((...
            j-1)*(bh_no/noUs)+1:j*bh_no/noUs,:)');
end

BIHTBH = sign0Mat(BIHTINTBH,0);

% LP biohashes
for j = 1:noUs
  LPINTBH(:,(j-1)*(bh_no/noUs)+1:j*bh_no/noUs) = (R...
            ((j-1)*bh_size+1:j*bh_size,:)*LPX((...
            j-1)*(bh_no/noUs)+1:j*bh_no/noUs,:)');
end

LPBH = sign0Mat(LPINTBH,0);
```

Rekonstrukce obrázků

Rekonstrukce obrázků se provádí podle vzorce 4.6, kde matice $\mathbf{A}$ je pro nás transponovaná matice `coeff` z funkce `pca`. Vektor $\hat{\mathbf{x}}$ je aproximovaný vektor rysů (tj. výstup z výše popsaných metod) a μ je průměrný vektor obrázku obličeje (tento průměrný vektor počítáme z části matice `ORIMAT`, která obsahuje vektory obrázků konkrétního uživatele.

Výsledné matice `ximgBIHT` a `ximgLP` mají stejné rozměry jako původní matice `ORIMAT` a v každém řádku obsahují vektor reprezentující jeden obrázek. Pro potřeby zobrazení rekonstruovaných obrázků je pak aplikována funkce `getReshaped` a jsou nastaveny rozměry původního obrázku, tj. 200×180.

Výpis 5.15: bhAtt.m – rekonstrukce obrázků

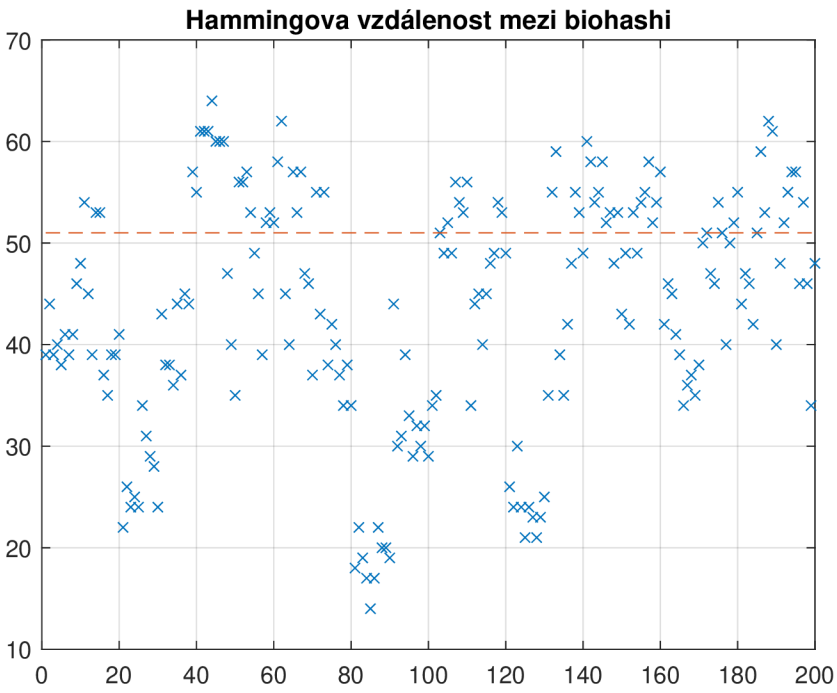
```
%% Reconstruction of images
% BIHT images
ximgBIHT(m*n,bh_no)=zeros;
for j = 1:noUs
  for i = (j-1)*(bh_no/noUs)+1:j*bh_no/noUs
    ximgBIHT(:,i) = coeff * BIHTX(i,:)' + mean(ORIMAT...
            ((j-1)*(bh_no/noUs)+1:j*bh_no/noUs,:))';
  end
end

% LP images
ximgLP(m*n,bh_no)=zeros;
for j = 1:noUs
  for i = (j-1)*(bh_no/noUs)+1:j*bh_no/noUs
    ximgLP(:,i) = coeff * LPX(i,:)' + mean(ORIMAT...
            ((j-1)*(bh_no/noUs)+1:j*bh_no/noUs,:))';
  end
end
```

5.4 Výsledky

V této části budou prezentovány výsledky, kterých bylo dosaženo. Nejprve budou prezentovány obrázky vztahující se k rekonstrukci biohashů popsaných v kap. 5.3.3 a následně zrekonstruované obrázky. Na závěr bude proveden výpočet FRR, FAR a EER popsaných v kap. 1.1.

Nejprve si zobrazíme výsledky autentizace uživatele. Na obrázku 5.4 lze vidět velikost vypočítaných Hammingových vzdáleností a jejich dostribuci vzhledem k nastavenému prahu systému (na obrázku je práh nastaven na hodnotu 51, tj. 60% shoda biohashů). Veškeré biohashe, jejichž Hammingova vzdálenost od uložených biohashů je menší než práh (tedy na obrázku se bod nachází pod přerušovanou čarou) budou úspěšně přihlášeny. Naopak body nad přerušovanou čarou představují neúspěšné pokusy o přihlášení.

Na obrázku 5.5 je možné vidět Hammingovy vzdálenosti mezi biohashi vypočítanými z aproximovaných vektorů rysů metodou BIHT od uložených biohashů. Obrázek odpovídá výsledkům pro práh $t = 51$ a řídkost vektoru $k = 2$. Na obrázku 5.6 jsou vidět Hammingovy vzdáleností mezi vypočítanými biohashi z aproximovaných vektorů rysů metodou lineárního programování od uložených biohashů, taktéž je zde práh $t = 51$.

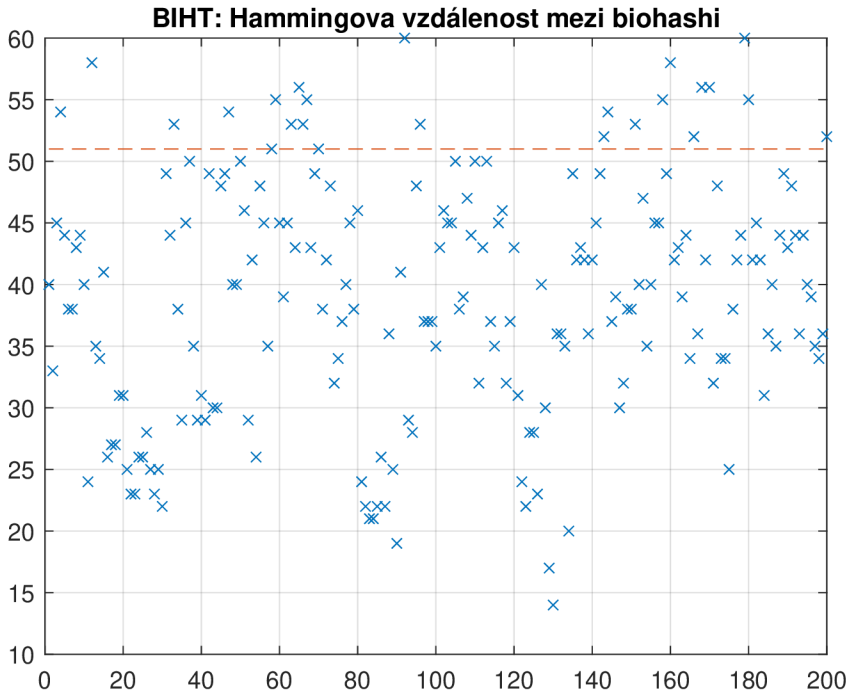


Obr. 5.4: Hammingova vzdálenost mezi biohashi a treshold – autentizace

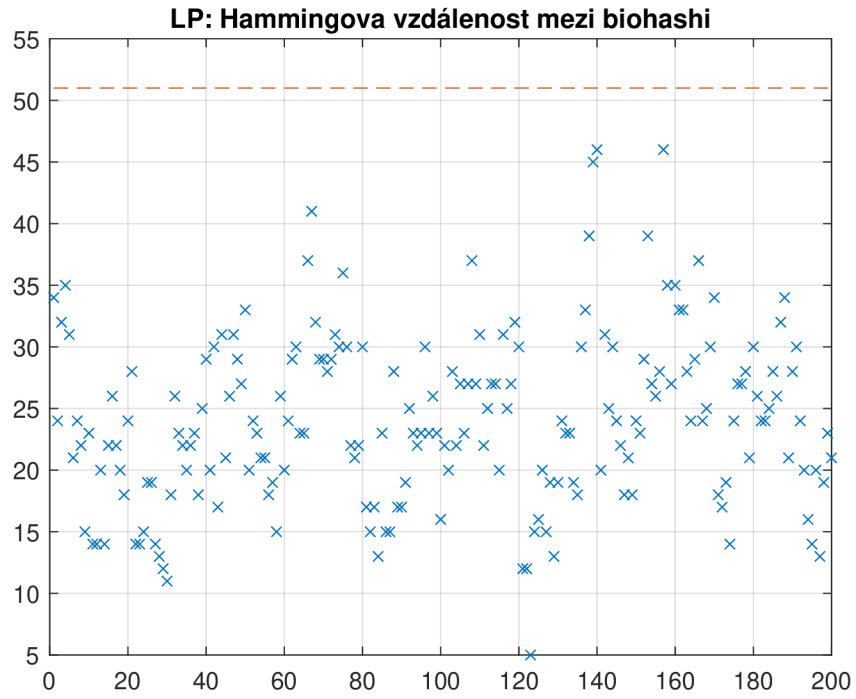
Při srovnání obrázků 5.6 a 5.5 (Hammingovy vzdálenosti biohashů útočníka) s obr. 5.4 je vidět, že útočník dosahuje lepších výsledků než uživatel, který se chce do systému přihlásit.

Zrekonstruované obrázky (vždy je zobrazen pouze první obrázek z deseti příslušných jednomu uživateli) je možné vidět na obrázku 5.7 a 5.8. Při rekonstrukci velmi záleží na uložené databázi obrázků. Pokud se jednotlivé obrázky od sebe liší (například subjekt je blíže kameře a poté dále nebo je posunutý více ke straně apod.) pak rekonstrukce není příliš zdařilá, ačkoliv i zde jsou identifikovatelné základní rysy osoby. Naopak, pokud subjekt má obrázky konzistentní (tj. s minimálními rozdíly), pak je rekonstrukce zdařilejší a osoba je dobře identifikovatelná (např. třetí a čtvrtý obrázek na prvním řádku). Pro možnost srovnání zrekonstruovaných obrázků s obrázky původními zobrazujeme také originální obrázky – viz obr. 5.9.

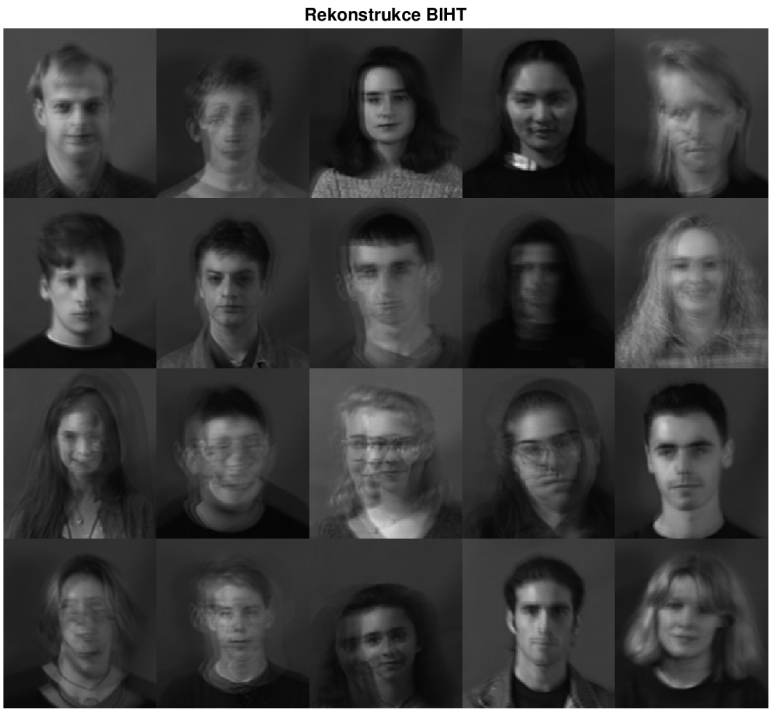
Tedy pro potřeby co možná nejúspěšnější autentizace je dobré mít v uložené databázi zachyceno co nejvíce variant portrétu osoby, poté je autentizace úspěšnější, než když máme uloženy obrázky, které jsou si vzájemně velmi blízké, ale poté se chceme autentizovat s portrétem, který se od uložených velmi liší. Proto je dobré mít uloženy různé varianty fotografie, aby byla pravděpodobnost úspěšného přihlášení větší. V případě rekonstrukce obrázků vykazují lepší výsledky osoby, které mají uloženy v databázi portréty s minimálními odchylkami mezi sebou – v těchto případech je rekonstrukce výrazně zdařilejší.



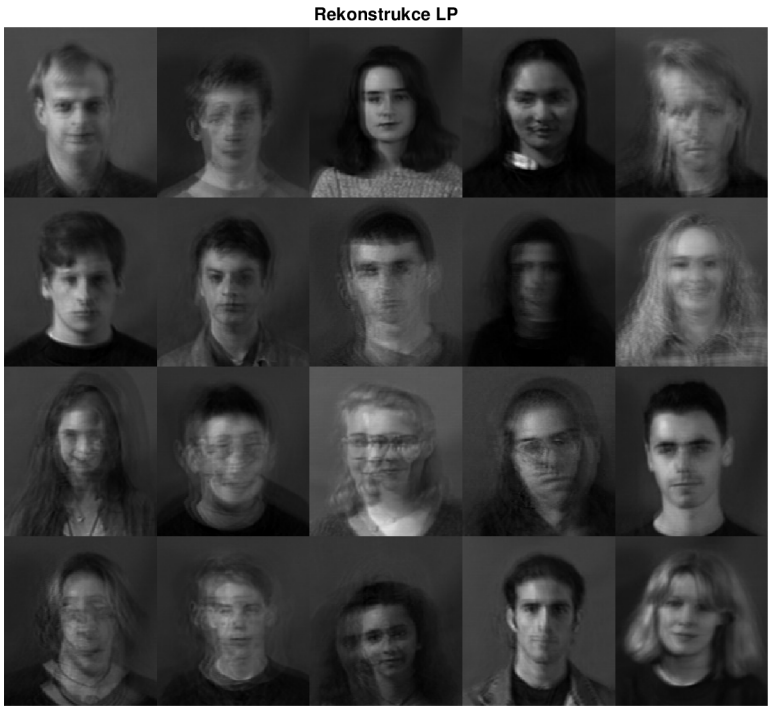
Obr. 5.5: Hammingova vzdálenost mezi biohashi a treshold – BIHT



Obr. 5.6: Hammingova vzdálenost mezi biohashi a treshold – lineární programování



Obr. 5.7: Rekonstrukce obrázků – BIHT



Obr. 5.8: Rekonstrukce obrázků – lineární programování

Obr. 5.9: Uložené originální obrázky

V tabulkách 5.2, 5.3 a 5.4 je možné vidět vypočítané hodnoty FRR (falešného odmítnutí) pro různé délky biohashů (32, 64 a 128 bitů). V případě FRR jde o situaci, kdy se oprávněný uživatel chce autentizovat, ale není do systému vpuštěn, jelikož je Hammingova vzdálenost nově vygenerovaného biohashe vyšší než nastavený práh systému. V tabulce jsou vypočítány hodnoty nejprve pro jednotlivé uživatele a poté celkově pro všechny. FRR je vypočítáno pomocí vzorce[5]

$$FRR = \frac{\text{počet falešných odmítnutí}}{\text{počet pokusů o přihlášení}} \cdot 100.$$

Z tabulek je patrné, že pro biohashe menších velikostí dosahuje FRR nižších hodnot. Dále hodnota FRR závisí na nastaveném prahu systému – čím nižší práh je nastaven, tím vyšší je i FRR. Uživatelé kteří jsou vždy nebo téměř vždy úspěšně autentizováni mají velmi konzistentní portréty nebo v obou množinách (testovací a trénovací) mají všechny různé varianty portrétů (blíže kameře, dále od kamery apod.), naopak osoby jejichž portréty v testovací množině vykazují poměrně velké odlišnosti od testovací množiny mají úspěšnost přihlášení výrazně nižší.

Hodnoty FAR jsou zobrazeny v tabulce 5.5 a 5.6. V tabulkách jsou hodnoty pro různé délky bihashů (32, 64 a 128). Hodnota FAR vzrůstá když se podaří útočníkovi

[5]K tomu viz [34, s. 8].

dostat do systému (tedy jím zrekonstruovaný biohash má Hammingovu vzdálenost od uloženého biohashe nižší, než je nastavený práh systému. Při výpočtu hodnoty FAR jsme vycházeli ze vzorce[6].

$$FAR = \frac{\text{počet falešných přijetí}}{\text{počet pokusů o přihlášení}} \cdot 100.$$

Úspěšnost metody BIHT závisí na nastavené řídkosti vektoru k. Je-li tento parametr nastaven na hodnotu pět a vyšší, mají biohashe vypočítané na základě této metody velmi nízkou Hammingovu vzdálenost od uložených biohashů. Z tabulky 5.5 je patrné, že čím nižší velikost biohashe, tím nižší je hodnota FAR. Jak můžeme vidět z tabulek u metody BIHT více záleží na nastavené řídkosti, než na prahu systému – od $k = 5$ a více jsou hodnoty FAR větší než 90 % a pro $k = 10$ bývají Hammingovy vzdálenosti téměř vždy nulové (tedy vygenerovaný biohash je shodný s uloženým).

Metoda lineárního programování má hodnoty FAR v případě procentní shody biohashů nastavené na 60 % téměř 100 %. Ovšem jak lze vidět z tabulky 5.6, FAR při zvyšování procentní shody od 60 % do 90 % poměrně rapidně klesají. Opět platí, že čím nižší je dimenze biohashe, tím nižší je hodnota FAR.

Z výsledků vypočítáme hodnotu EER podle následujícího vzorce[7]

$$EER = \frac{FAR + FRR}{2}.$$

Tato hodnota se zvyšuje s rostoucí procentní shodou a s klesající velikostí biohashe se snižuje.

Tab. 5.1: EER

	BH_dim = 128				
	FAR LP [%]	FAR BIHT [%], k=2	FRR [%]	EER LP [%]	EER BIHT [%]
t=51; shoda 60%	100	89	32	66	60,5
t=49; shoda 61%	100	85,5	35	67,5	60,25
t=48; shoda 62%	100	81,5	39	69,5	60,25
t=47; shoda 63%	100	78,5	42	71	60,25
t=46; shoda 64%	100	77,5	44	72	60,75
t=44; shoda 65%	98,5	69	51	74,75	60

	BH_dim = 64				
	FAR LP [%]	FAR BIHT [%], k=2	FRR [%]	EER LP [%]	EER BIHT [%]
t=25; shoda 60%	99,5	88,5	24,5	62	56,5
t=24; shoda 61%	98	79,5	30,5	64,25	55
t=23; shoda 63%	95	74,5	34	64,5	54,25
t=22; shoda 65%	92	69	40,5	66,25	54,75

	BH_dim = 32				
	FAR LP [%]	FAR BIHT [%], k=2	FRR [%]	EER LP [%]	EER BIHT [%]
t=12; shoda 60%	92,5	83,5	16,5	54,5	50
t=11; shoda 65%	87	73,5	28,5	57,75	51

[6]K tomu viz [34, s. 8].
[7]K tomu viz [34, s. 8].

Tab. 5.2: FRR pro biohash délky 32 bitů – autentizace

		BH_dim = 32					
		t=12; shoda=60%			t=11; shoda=65%		
		FRR [%]	FR	TA	FRR [%]	FR	TA
1	ben	0	0	10	0	0	10
2	dan	0	0	10	40	4	6
3	ema	10	1	9	10	1	9
4	eva	0	0	10	0	0	10
5	iva	50	5	5	80	8	2
6	ivo	10	1	9	20	2	8
7	jan	10	1	9	20	2	8
8	joe	10	1	9	10	1	9
9	joy	0	0	10	0	0	10
10	kim	0	0	10	0	0	10
11	lin	40	4	6	50	5	5
12	max	10	1	9	20	2	8
13	meg	0	0	10	0	0	10
14	mia	20	2	8	50	5	5
15	oto	80	8	2	90	9	1
16	sam	40	4	6	60	6	4
17	sid	0	0	10	20	2	8
18	sue	10	1	9	20	2	8
19	tom	40	4	6	70	7	3
20	zoe	0	0	10	10	1	9
celkem		16,5	33	167	28,5	57	143

Tab. 5.3: FRR pro biohash délky 64 bitů – autentizace

		BH_dim = 64											
		t=25; shoda=60%			t=24; shoda=61%			t=23; shoda = 63%			t=22; shoda=65%		
		FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA
1	ben	0	0	10	0	0	10	0	0	10	0	0	10
2	dan	40	4	6	40	4	6	40	4	6	40	4	6
3	ema	0	0	10	0	0	10	0	0	10	0	0	10
4	eva	20	2	8	20	2	8	20	2	8	20	2	8
5	iva	80	8	2	80	8	2	80	8	2	80	8	2
6	ivo	30	3	7	30	3	7	30	3	7	50	5	5
7	jan	50	5	5	60	6	4	60	6	4	60	6	4
8	joe	0	0	10	20	2	8	20	2	8	20	2	8
9	joy	0	0	10	0	0	10	0	0	10	0	0	10
10	kim	0	0	10	0	0	10	0	0	10	0	0	10
11	lin	30	3	7	60	6	4	70	7	3	70	7	3
12	max	30	3	7	40	4	6	50	5	5	70	7	3
13	meg	0	0	10	0	0	10	0	0	10	0	0	10
14	mia	50	5	5	50	5	5	50	5	5	70	7	3
15	oto	90	9	1	100	10	0	100	10	0	100	10	0
16	sam	40	4	6	60	6	4	70	7	3	80	8	2
17	sid	0	0	10	0	0	10	0	0	10	30	3	7
18	sue	0	0	10	0	0	10	20	2	8	40	4	6
19	tom	20	2	8	30	3	7	30	3	7	30	3	7
20	zoe	10	1	9	20	2	8	40	4	6	50	5	5
celkem		24,5	49	151	30,5	61	139	34	68	132	40,5	81	119

Tab. 5.4: FRR pro biohash délky 128 bitů – autentizace

		BH_dim = 128																	
		t=51; shoda=60%			t=49; shoda=61%			t=48; shoda = 62%			t=47; shoda=63%			t=46; shoda=64%			t=44; shoda=65%		
		FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA	FRR [%]	FR	TA
1	ben	0	0	10	0	0	10	0	0	10	10	1	9	10	1	9	20	2	8
2	dan	30	3	7	30	3	7	30	3	7	30	3	7	30	3	7	40	4	6
3	ema	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10
4	eva	20	2	8	20	2	8	20	2	8	20	2	8	20	2	8	30	3	7
5	iva	70	7	3	70	7	3	70	7	3	70	7	3	80	8	2	80	8	2
6	ivo	70	7	3	70	7	3	80	8	2	80	8	2	80	8	2	90	9	1
7	jan	50	5	5	50	5	5	50	5	5	50	5	5	60	6	4	80	8	2
8	joe	20	2	8	20	2	8	20	2	8	20	2	8	20	2	8	20	2	8
9	joy	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10
10	kim	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10
11	lin	50	5	5	60	6	4	80	8	2	80	8	2	80	8	2	80	8	2
12	max	20	2	8	20	2	8	40	4	6	50	5	5	50	5	5	70	7	3
13	meg	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10
14	mia	40	4	6	40	4	6	50	5	5	60	6	4	60	6	4	60	6	4
15	oto	80	8	2	80	8	2	80	8	2	90	9	1	90	9	1	90	9	1
16	sam	70	7	3	70	7	3	90	9	1	90	9	1	90	9	1	90	9	1
17	sid	0	0	10	0	0	10	0	0	10	0	0	10	0	0	10	20	2	8
18	sue	30	3	7	70	7	3	70	7	3	70	7	3	80	8	2	90	9	1
19	tom	40	4	6	50	5	5	50	5	5	50	5	5	60	6	4	70	7	3
20	zoe	50	5	5	50	5	5	50	5	5	70	7	3	70	7	3	90	9	1
celkem		32	64	136	35	70	130	39	78	122	42	84	116	44	88	112	51	102	98

Tab. 5.5: FAR pro metodu BIHT (biohashe délky 32, 64, 128 bitů)

	BH_dim=128														
	k=1			k=2			k=3			k=4			k=5		
	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR
t=51; shoda 60%	78,5	157	43	89	178	22	95	190	10	98	196	4	99	198	2
t=49; shoda 61%	71,5	143	57	85,5	171	29	93	186	14	96	192	8	98	196	4
t=48; shoda 62%	65	130	70	81,5	163	37	92,5	185	15	95,5	191	9	97,5	195	5
t=47; shoda 63%	58,5	117	83	78,5	157	43	89,5	179	21	95	190	10	97,5	195	5
t=46; shoda 64%	51	102	98	77,5	155	45	86,5	173	27	94,5	189	11	97	194	6
t=44; shoda 65%	41,4	83	117	69	138	62	82,5	165	35	93,5	187	13	96,5	193	7

	BH_dim=64														
	k=1			k=2			k=3			k=4			k=5		
	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR
t=25; shoda 60%	66,5	133	67	88,5	171	29	92,5	185	15	99	198	2	100	200	0
t=24; shoda 61%	59,5	119	81	79,5	159	41	89,5	179	21	98	196	4	100	200	0
t=23; shoda 63%	49	98	102	74,5	149	51	88,5	171	29	98	196	4	99	198	2
t=22; shoda 65%	43,5	87	113	69	138	62	81	162	38	93,5	187	13	98,5	197	3

	BH_dim=32														
	k=1			k=2			k=3			k=4			k=5		
	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR	FAR [%]	FA	TR
t=12; shoda 60%	62	124	76	83,5	167	33	96,5	193	7	100	200	0	100	200	0
t=11; shoda 65%	46,5	93	107	73,5	147	53	94,5	189	11	99	198	2	100	200	0

Tab. 5.6: FAR pro metodu lineárního programování (biohashe délky 32, 64, 128 bitů)

	BH_dim=128		
	FAR [%]	FA	TR
t=51; shoda 60%	100	200	0
t=44; shoda 65%	98,5	197	3
t=40; shoda 68%	98	196	4
t=35; shoda 72%	95	190	10
t=30; shoda 76%	84,5	169	31
t=28; shoda 78%	74,5	149	51
t=25; shoda 80%	60,5	121	79
t=20; shoda 84%	31,5	63	137
t=15; shoda 88%	12,5	25	175
t=12; shoda 90%	2,5	5	195

	BH_dim=64		
	FAR [%]	FA	TR
t=25; shoda 60%	99,5	199	1
t=24; shoda 61%	98	196	4
t=23; shoda 63%	95	190	10
t=22; shoda 65%	92	184	16
t=20; shoda 68%	81,5	163	37
t=17; shoda 72%	65	130	70
t=15; shoda 76%	52	104	96
t=12; shoda 80%	29	58	142
t=10; shoda 84%	20	40	160
t=5; shoda 91%	1,5	3	197

	BH_dim=32		
	FAR [%]	FA	TR
t=12; shoda 60%	92,5	185	15
t=11; shoda 65%	87	174	26
t=10; shoda 68%	76	152	48
t=9; shoda 70%	60,5	121	79
t=8; shoda 75%	44,5	89	111
t=7; shoda 78%	28	56	144
t=6; shoda 80%	17	98	102
t=5; shoda 84%	8,5	17	183
t=4; shoda 85%	3,5	7	193
t=3; shoda 90%	2	4	196

Závěr

Diplomová práce se převážně věnuje problematice biometrického hashování se zaměřením na biohashování obrázků obličeje. Práce je rozdělena do pěti kapitol, přičemž první čtyři popisují teoretický rámec problematiky – biometrické systémy, biometrický hashing, komprimované snímání a útoky na biometrický hashing.

V teoretické části byly nejprve diskutovány biometrické systémy jako takové, základní požadavky na tyto systémy, také bylo provedeno jejich základní srovnání. Následně byly uvedeny chyby biometrických systémů, jelikož ty se často používají při jejich srovnávání. Závěrečná sekce první kapitoly se věnuje problematice bezpečnosti biometrických systémů. Druhá kapitola plynule navazuje na předcházející a věnuje se biometrickému hashingu. Rozebírá princip vytváření biohashe a následně stručně popisuje biometrické hashovací systémy využívající různé biometrické charakteristiky. Následně je rozebrána problematika komprimovaného snímání včetně základní matematiky potřebné pro pochopení této techniky. V závěru teoretické části práce jsou uvedeny možné útoky na biometrický hashing. Při těchto útocích se využívá již dříve představené metody komprimovaného snímání.

V praktické části bylo implementováno vytváření biohashe. Byla zde vytvořena simulace jak registrační fáze, tak i autentizační. Dále zde byly naprogramovány útoky, konkrétně jde o metodu BIHT a lineární programování. Tyto metody provádějí aproximaci vektoru rysů a takto vypočítaný vektor využijí buď k rekonstrukci biohashe, který je porovnán s uloženým, nebo k rekonstrukci portrétu uživatele. Na závěr praktické části byly uvedeny výsledky, kterých bylo dosaženo.

Z dosažených výsledků lze dospět k následujícím závěrům. Zaměříme-li se na autentizační fázi procesu, velmi záleží na sadě obrázků, které jsou v systému uloženy. Jsou-li tyto konzistentní s obrázkem, se kterým se uživatel přihlašuje, je uživatel bez problému přihlášen. Pokud se ovšem obrázek použitý k autentizaci velmi odlišuje od uložených obrázků, pak hrozí vysoké riziko, že uživatel není úspěšně autentizován. Dalším přístupem pro účely co možná nejúspěšnější autentizace, je mít v uložené databázi zachyceny všechny možné varianty portrétu osoby, poté je autentizace úspěšnější. Úspěšnost autentizace byla vyhodnocována na základě hodnoty FRR (počítaná na základě Hammingovy vzdálenosti a prahu systému). Pro biohashe menších velikostí dosahuje FRR nižších hodnot. Dále hodnota FRR závisí na nastaveném prahu systému – čím vyšší práh je nastavený, tím vyšší je i FRR.

Rekonstrukce obrázků prováděná útočníkem vykazuje nejlepší výsledky, pokud se portréty uložené v databázi od sebe příliš neliší. Ovšem i pokud jsou rozdíly mezi jednotlivými obrázky větší, jsou základní rysy osoby patrné.

Úspěšnost rekonstrukce biohashů byla vyhodnocována na základě hodnoty FAR (taktéž počítána na základě Hammingovy vzdálenosti a prahu systému). Úspěšnost

metody BIHT závisí na nastavené řídkosti vektoru. Je-li tento parametr nastaven na hodnotu pět a vyšší, mají biohashe vypočítané na základě této metody velmi nízkou Hammingovu vzdálenost od uložených biohashů. Dále platí, že čím nižší velikost biohashe, tím nižší je hodnota FAR. Metoda lineárního programování má hodnoty FAR, v případě nastaveného prahu na hodnotu 51 (ze 128), téměř 100 %. Může se tak na první pohled zdát, že lineární programování vykazuje lepší výsledky než BIHT, ovšem úspěšnost BIHT se výrazně odvíjí od nastavené hodnoty řídkosti. Pro vyšší hodnoty procentní shody (tj. menší práh) totiž hodnoty FAR pro lineární programování poměrně rapidně klesají. Pro BIHT i lineární programování platí že čím nižší je dimenze biohashe, tím nižší je hodnota FAR. Z vypočítaných hodnot FAR a FRR byla spočítána hodnota EER – hodnota se zvyšuje s rostoucí procentní shodou a s klesající velikostí biohashe se snižuje.

Závěrem lze říci, že spolehlivost těchto systémů je relativně nízká, jak je také patrné z dosažených výsledků. Naopak pro oprávněného uživatele není tento systém zcela komfortní, neboť je třeba dbát na konzistenci autentizačního obrázku s portréty pořízenými ve fázi registrace, které jsou v systému uloženy.

Literatura

[1] PRABHAKAR, S.et al. *Biometric recognition: security and privacy concerns.* IEEE Secur. Priv. 1(2), 33–42 (2003).

[2] KONG, A. et al. *An analysis of biohashing and its variants.* Pattern Recognit.39(7), 1359–1368 (2006). doi:10.1016/j.patcog.2005.10.025.

[3] NAGAR, A. et al., *Biometric Template Transformation: A Security Analysis*, in Media Forensics and Security, 7541, ed. by ND Memon, J Dittmann, AM Alattar, and EJ Delp. SPIE Proceedings (SPIEUnited States, 2010), p. 75410, doi:10.1117/12.839976.

[4] JAIN, AK. et al., *Biometric Template Security.* EURASIP J. Adv. Signal Process.2008:, 113–117 (2008). doi:10.1155/2008/579416.

[5] TOPCU, B. et al. *Practical security and privacy attacks against biometric hashing using sparse recovery.* EURASIP journal on advances in signal processing. Cham: Springer International Publishing, 2016(1), 1-20. ISSN 1687-6172. doi:10.1186/s13634-016-0396-1

[6] BOOKSTEIN, A. et al. *Generalized Hamming Distance.* Information Retrieval 5, 353–375 (2002). Dostupné z: <https://doi.org/10.1023/A:1020499411651>

[7] ABDI, H.; WILLIAMS, L.J. (2010), *Principal component analysis.* WIREs Comp Stat, 2: 433-459. Dostupné z: <`https://doi.org/10.1002/wics.101`>

[8] NGO, D. C. L. et al., *Biometric hash: high-confidence face recognition*, in IEEE Transactions on Circuits and Systems for Video Technology, vol. 16, no. 6, pp. 771-775, June 2006, doi: 10.1109/TCSVT.2006.873780.

[9] TOPCU, B. et al. *Biohashing with fingerprint spectral minutiae*, 2013 International Conference of the BIOSIG Special Interest Group (BIOSIG), 2013, pp. 1-12.

[10] JAIN, A. K. et al. *Filterbank-based fingerprint matching*, Trans. Img. Proc., vol. 9, no. 5, pp. 846-859, May 2000.

[11] PARK, Chul-Hyun Park et al. *Fingerprint matching using the distribution of the pairwise distances between minutiae*, Proceedings of the 5th international conference on Audio-and Video-Based Biometric Person Authentication AVBPA'05, pp. 693-701, 2005.

[12] TUYLS, P. et al. *Practical Biometric Authentication with Template Protection*, 5th Int. Conf. on Audio- and Video-Based Personal Authentication (AVBPA) Rye Brook New York, pp. 436-446, July 2005.

[13] SCHEIDAT, T. et al. *Biometric hash generation and user authentication based on handwriting using secure sketches*, 2009 Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, 2009, pp. 89-94, doi: 10.1109/ISPA.2009.5297688.

[14] ZHENG, Y. et al. *Facial biohashing based user-device physical unclonable function for bring your own device security*, 2018 IEEE International Conference on Consumer Electronics (ICCE), 2018, pp. 1-6, doi: 10.1109/ICCE.2018.8326074.

[15] MARTINEZ, A. M.; KAK, A. C., *PCA versus LDA*, in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, no. 2, pp. 228-233, Feb. 2001, doi: 10.1109/34.908974.

[16] LI, X. et al. *Facial expression recognition with local Gabor filters*, IEEE 10th INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING PROCEEDINGS, 2010, pp. 1013-1016, doi: 10.1109/ICOSP.2010.5655855.

[17] JI, S.; YE, J. *Generalized Linear Discriminant Analysis: A Unified Framework and Efficient Model Selection*, in IEEE Transactions on Neural Networks, vol. 19, no. 10, pp. 1768-1782, Oct. 2008, doi: 10.1109/TNN.2008.2002078.

[18] HRBÁČEK, R. et al. *Řídké reprezentace signálů: úvod do problematiky.* Elektrorevue – Internetový časopis, 2011: s. 1–10. Dostupné z: <http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/0/ridke-reprezentace-signalu--uvod-do-problematiky/>

[19] HRBÁČEK, R. et al. *Řídké reprezentace signálů: komprimované snímání.* Elektrorevue - Internetový časopis, 2011. Dostupné z: <http://www.elektrorevue.cz/cz/clanky/zpracovani-signalu/0/ridke-reprezentace-signalu--komprimovane-snimani/>

[20] CANDES, E.J.; WAKIN, M.B. *An Introduction To Compressive Sampling.* IEEE signal processing magazine. PISCATAWAY: IEEE, 2008, 25(2), 21-30. ISSN 1053-5888. doi:10.1109/MSP.2007.914731

[21] BERTSEKAS, D. *Convex Optimization Theory.* Athena Scientific, 2009, ISBN 978-1-886529-31-1.

[22] GAO,Yi et al. *On the Null Space Property of ℓ_q-Minimization for* $0 < q \leq 1$ in Compressed Sensing, Journal of Function Spaces, vol. 2015, Article ID 579853, 10 pages, 2015. Dostupné z: <https://doi.org/10.1155/2015/579853>

[23] CHARTRAND, Rick; STANEVA, Valentina. 2008, *Inverse Problems*, 24 035020. DOI 10.1088/0266-5611/24/3/035020

[24] LANDAU, H. J. *Sampling, data transmission, and the Nyquist rate.* In Proceedings of the IEEE, vol. 55, no. 10, pp. 1701-1706, Oct. 1967, doi: 10.1109/PROC.1967.5962.

[25] BOUCHET, A. *Greedy algorithm and symmetric matroids.* Mathematical Programming 38, 147–159 (1987). Dostupné z <https://doi.org/10.1007/BF02604639>

[26] LACHARME, P. et al. *Preimage attack on BioHashing*, 2013 International Conference on Security and Cryptography (SECRYPT), 2013, pp. 1-8.

[27] BOUFOUNOS, PT; BARANIUK, RG. *42nd Annual Conference on Information Sciences and Systems, CISS.* 1-bit Compressive Sensing, (2008), pp. 16–21, doi:10.1109/CISS.2008.4558487.

[28] PLAN, Y; VERSHYNIN, R. *One-bit compressed sensing by linear programming.* Commun. Pur. Appl. Math.66(8), 1275–1297 (2013). doi:10.1002/cpa.21442.

[29] JACQUES, L et al. *Robust 1-bit compressive sensing via binary stable embeddings of sparse vectors.* IEEE Trans. Inf. Theory.59(4) (2013). doi:10.1109/TIT.2012.2234823.

[30] BLUMENSATH, T; DAVIES, ME. *Iterative hard thresholding for compressed sensing.* Appl. Comput. Harmon. Anal.27(3), 265–274 (2009). doi:10.1016/j.acha.2009.04.002.

[31] BARATA, J.C.A.; HUSSEIN, M.S. *The Moore–Penrose Pseudoinverse: A Tutorial Review of the Theory.* Braz J Phys 42, 146–165 (2012). Dostupné z: <https://doi.org/10.1007/s13538-011-0052-z>

[32] KARA, O.; ATALAY, A. *Preimages of hash functions through rainbow tables*, 2009 24th International Symposium on Computer and Information Sciences, 2009, pp. 304-309, doi: 10.1109/ISCIS.2009.5291831.

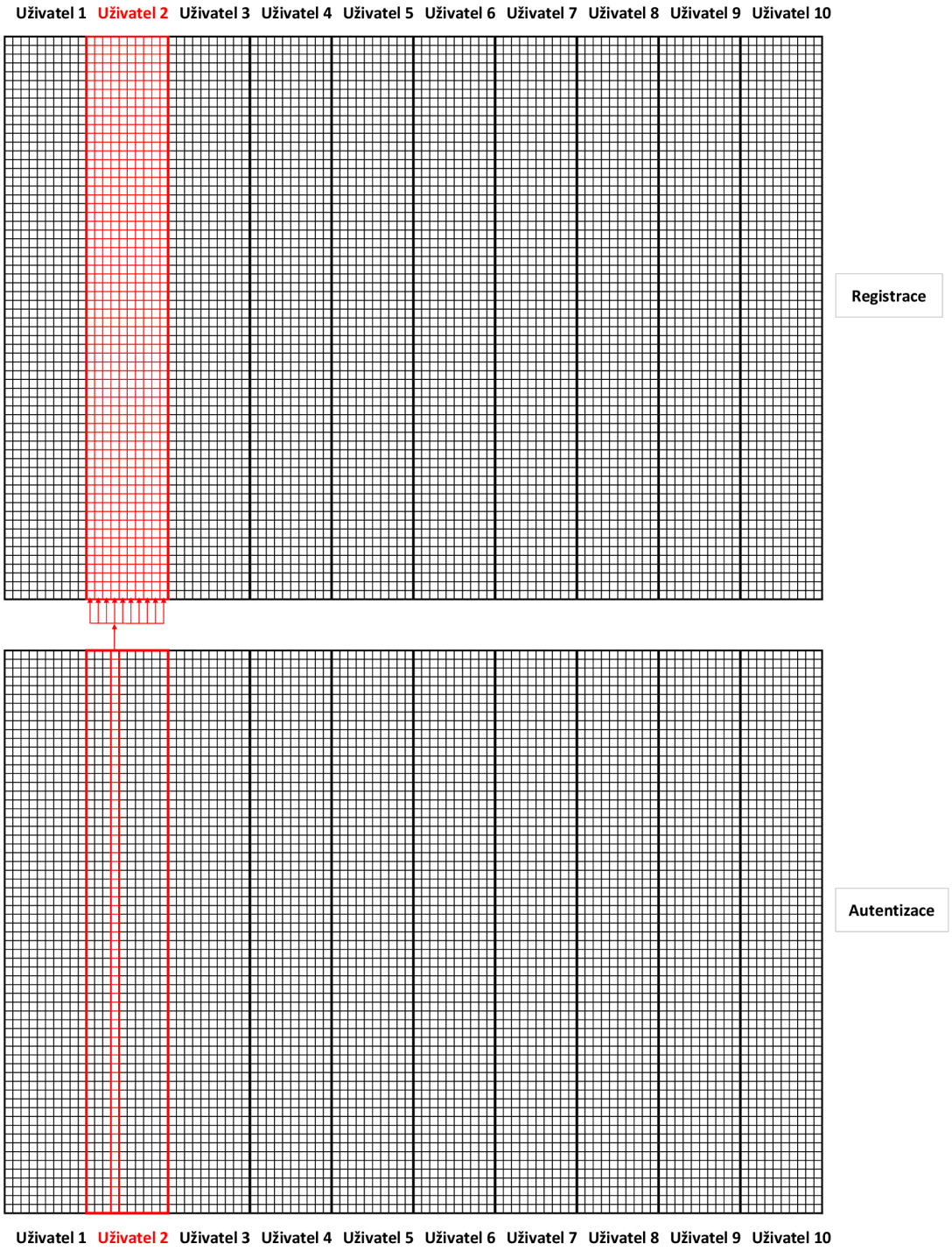
[33] Facial Images: Faces95. Libor Spacek's Facial Images Databases [online]. 2009 [cit. 2022-12-08]. Dostupné z: <https://cmp.felk.cvut.cz/~spacelib/faces/faces95.html>

[34] ZAIRI, Ahmad Zairi et al. *Touch-based continuous mobile device authentication: State-of-the-art, challenges and opportunities*. Journal of Network and Computer Applications, Volume 191, 2021, ISSN 1084-8045. Dostupné z: <`https://doi.org/10.1016/j.jnca.2021.103162`>.

Seznam symbolů a zkratek

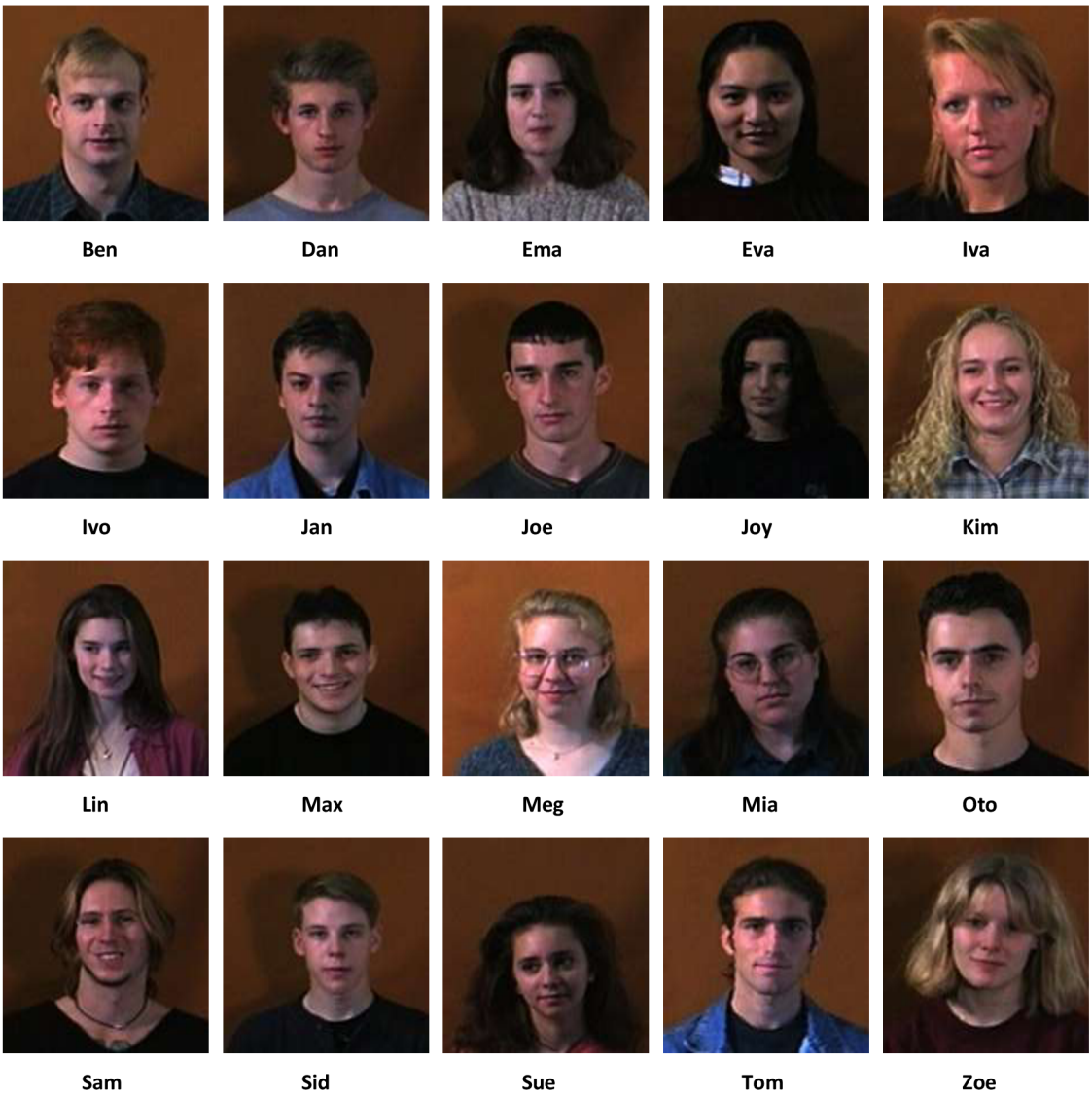
FAR	False Acceptance Rate
FRR	False Rejection Rate
EER	Equal Error Rate
ROC	Receiver Operating Characteristic
FTC	Failure To Capture
FTE	Failure To Enroll
TRN	Tokenized Pseudorandom Number
PCA	Principle Component Analysis
FLD	Fisher Linear Discriminant
LDA	Linear Discriminant Analysis
WT	Wavelet Transform
LFD	Local Feature Description
HSA	Holistic Subspace Analysis
NSP	Null Space Property
RIP	Restricted Isometry Property
PRNG	Pseudo-Random Number Generator
BIHT	Binary Iterative Hard Tresholding
IHT	Iterative Hard Tresholding
BH	Biometric Hashing (uplatňováno v kódech)
LP	Linear Programming (uplatňováno v kódech)

A Proces porovnávání biohashů v autentizační fázi



Obr. A.1: Proces porovnávání biohashů – autentizační fáze

B Přehled uživatelů



Obr. B.1: Přehled uživatelů