Gymnázium Matyáše Lercha, Brno, Žižkova 55

Bezpečnost kryptoměn a implementace kryptopeněženky

Závěrečná maturitní práce

Vedoucí práce:
Lukáš Malina

Tomáš Vondrák

Brno 2019

PODPORA SOČ

Jihomoravský kraj

Je mou povinností zde poděkovat zejména Lukáši Malinovi za jeho práci při seznamování mne s teorií i praxí, kterou jsem ve své práci využil. Dále pak mému učiteli informatiky, Lukáši Rýdlovi, který mi s mojí prací také významně pomáhal. Nemohu opomenout poděkovat JCMM a JMK, které mi poskytly školitele a finanční podporu.

Tato práce byla vypracována za finanční podpory JMK.

Prohlašuji, že jsem tuto práci vyřešil samostatně s použitím literatury, kterou uvádím v seznamu.

V Brně dne 21. února 2019 .....

Abstrakt

Vondrák T., Bezpečnost kryptoměn a implementace kryptopeněženky. Brno, 2019.

Práce pojednává o teorii bezpečnosti kryptoměn, příkladech kryptoměn s vyšším zabezpečením než je obvyklé a teorii týkající se hardwarových kryptopeněženek, zejména uložených na tzv. Java Card. Dále je zde popsána implementace samotné kryptopeněženky a aplikace jenž s touto peněženkou umí komunikovat a vypisovat do grafického prostředí její výkonnost.

Obsah

1 Úvod a cíl práce

1.1 Úvod do problematiky

Práce se bude v první řadě zabývat samotnými kryptoměnami. V této oblasti práce poskytne přehled základních principů kryptoměn a přehled kryptoměn se zvýšenou bezpečností.

Práce bude pokračovat kryptopeněženkami. Prozkoumá jejich typy a principy fungování. Zde bude také rozebraná implementace konkrétní kryptopeněženky.

Dále se práce bude zabývat čipovými kartami. Například komunikací mezi čipovou kartou a zabezpečením java karet. Čipové karty budou probrány spíše povrchově.

Následně bude popsána praktická část práce. Zprovoznění čipové karty. Nahrání implementace peněženky na kartu.

Posledním tématem bude implementace aplikace, která bude komunikovat s kryptopeněženkou na kartě. Výstupem bude grafické interaktivní prostředí, ze kterého půjde odečíst výkonnost jednotlivých funkcí kryptopeněženky.

1.2 Cíl práce

Za cíl práce jsem si nejdříve zvolil nastudovat a pochopit látku, která se týká kryptoměn a zejména pak kryptoměn se zvýšenou bezpečností, teorie ohledně kryptopeněženek. Za další milník jsem si zvolil zprovoznit čipovou kartu a poté na ni nahrát kryptopeněženku s názvem Eligibility ledger. Vrcholem celé práce by pak měla být vylepšená implementace programu v jazyku Java. Tento program by měl komunikovat s kartou prostřednictvím APDU protokolu přes čtečku čipových karet. Program by měl testovat rychlost funkcí prováděných při podpisu transakce Bitcoinu v kryptopeněžence.

2 Kryptoměny

2.1 Stručná historie kryptoměn

První pokusy zavést elektronickou měnu se odehrály v 20. století. Prvním známým vynálezcem, který experimentoval z touto formou peněz byl David Chaum v roce 1983. Tuto experimentální kryptoměnu Chaum nazval Ecash [18][19].

V roce 1995 poté David Chaum přišel s měnou Digicash [9].

Později se v devadesátých letech o kryptoměny experimentálně zajímala například NSA a podobně. [21]

První kryptoměna, která se později doopravdy ujala a přestala být měnou experimentální byl Bitcoin. Jejím zakladatelem byl člověk, či skupina lidí vystupující pod pseudonymem Satoshi Nakamoto. Tato měna se chlubila plnou decentralizací a nezávislostí na třetí straně. Využívala jako hashovací funkci SHA 512.

Později se nejznámější kryptoměnou stala právě kryptoměna Bitcoin. Nicméně do prvních příček se rval například Litecoin, Bitcoin cash nebo Altcoin.

V současnosti se do popředí dostávají právě měny se zvýšenou bezpečností uživatele a s větším soukromím.

2.2 Základní principy fungování kryptoměn

Blockchain

Jedná se vlastně o účetní knihu dané kryptoměny. Do fronty za sebe se do řetězce skládají provedené transakce.

Do Blockchainu se ve skutečnosti za sebe přímo nestaví samotné transakce, ale tzv. těžební bloky. Do těchto těžebních bloků se transakce uzavírají.

Těžební bloky se jim říká z toho důvodu, že vznikají při procesu těžení a jsou jeho podstatou. O tom však až v sekci Těžba.

Výhoda Blockchainu spočívá v možnosti kontrolování jeho vývoje všemi uživateli. Myšlenková struktura blockchainu je, že všichni uživatelé jsou na stejné úrovni a všichni k němu mají přístup. Tedy do účetní knihy nahlížejí všichni uživatelé a kontrolují správnost jejího vývoje. Než tedy uživatel provede transakci, tak si všichni uživatelé zkontrolují, že mohl svoje peníze použít. Z toho vyplývá, že někdy se může stát, že se na konci účetní knihy udělá vidlice, kdy jedna z větví je špatně. Poté se musí rozhodnout, která větev je správná. Vidlice také mohou vznikat, pokud se komunita uživatelů neshodne na nějakém novém pravidle a blockchain dané měny se musí rozdělit do více větví, kde část uživatelů důvěřuje jedné větvi a část druhé. Toto je také jedna z vad kryptoměn, neboť pokud dojde k takovému „schizmatu“, u některé z kryptoměn, poté se musí nějak rozdělit peníze uživatelů, kteří založí novou větev v blockchainu na blockchainu stejné kryptoměny. Protože zde nefungují klasická pravidla obchodu s penězi, tak se může stát, že se zkrátka uživatelé domluví, že uživatelé důvěřující nové větvi z větve udělají vlastní kryptoměnu a jejich peníze z původní větve se zkopírují do nové kryptoměny, stejně tak uživatelům, kteří větvi

nedůvěřují, aby to bylo spravedlivé. Tím se nám najednou ovšem v oběhu objeví dvojnásobek „coinů“, což by se v klasickém bankovním systému nemohlo stát.

Systém blockchainu se stává nespolehlivým ve chvíli, kdy by nějaký těžař přesáhl 51 % celkové těžby (tedy celkového těžebního výkonu sítě) u této kryptoměny. Poté by mohl zpětně ovlivňovat větve blockchainu a tím zamezit uživatelům, aby mohli správně určovat, jestli jsou transakce korektní. Tomu se ale těžaři vyhýbají, protože kdyby některý z těžařů dosáhl takové těžby, poté by se přestalo měně důvěřovat a kryptoměna by okamžitě ztratila hodnotu a tím by přišli těžaři o své peníze.

Proof of work (PoW)

Jedná se o systém / protokol.

U měn, které využívají PoW, se hodnota platebních prostředků dokazuje pomocí vynaložené práce.

Jinak řečeno, aby byla transakce uznána, musí být podložena vytěženým blokem. O tom kdo blok vytěžil se musí hlasovat. Čím větší výpočetní výkon těžař má, tím větší má pravděpodobnost, že blok vytěží on.

Aby se transakce mohla provést musí být zaznamenána v nalezeném bloku. To se udělá tak, že těžař blok vytěží a transakce se poté do tohoto bloku uloží. Transakcí je potom v bloku celý balíček. Neposílá se pouze jedna v rámci jednoho bloku. Těžaři jsou poté odměněni za nalezení bloku výdělkem v podobě kryptoměny. To zabezpečuje motivaci těžařů těžit.

Průměrné časy vytěžení jednoho bloku se liší podle kryptoměny. (např. BTC má tento „block time“ cca 10 minut [22])

Proof of stake (PoS)

Jak systém PoS, tak systém PoW jsou systémy, které mají zabezpečit, aby se nemohla provést stejná transakce vícekrát. Tudíž aby se nemohlo při transakcích podvádět a uživatelé tak nemohli znásobovat své finanční prostředky.

Probíhá hlasování o tom, kdo objevil poslední blok. Čím více kryptoměny někdo vlastní a čím déle ji vlastní, tím má větší šanci na to, že odměnu z objeveného bloku dostane [23][24]. Proto jsou kryptoměny používající Proof-of-Stake vhodné pro investory, protože pomocí Proof-of-Stake lze vydělávat tím, že vlastní nějaké množství kryptoměny [24].

Možnost tzv. 51% útoku hrozí u obou systémů. Výhodou PoS systému je menší spotřeba elektrické energie než PoW, neboť zde nejsou potřeba žádné složité výpočty. To také implikuje, že nejsou třeba žádné investice do výkonného hardwaru (Nic se nepočítá.). Některým uživatelům nemusí vyhovovat nízká odměna za blok.

Těžba

Těžba nových bloků je proces, který se používá u kryptoměn, které fungují podle protokolu Proof of work.

U každé kryptoměny se jedná o nějaký náročný výpočet, kde je z hardwarového hlediska zatížená zejména grafická karta.

Pokuď chce těžař těžit efektivně, tak musí zhodnotit náklady a zisky, které z těžení plynou. Zejména tedy nákup drahého hardwaru, poplatky za elektrickou energii, poplatky spojené s chlazením strojů na straně výdajů a zisky z těžení bloků na straně druhé.

Po světě se zakládají i tzv. mining pooly, což jsou místa, kde si může člověk pronajmout hardware a těžit na něm. Výhoda tohoto systému spočívá v tom, že pravděpodobnost, že celý mining pool vytěží blok je mnohem větší než že jej vytěží těžař v jednom člověku, který nemá prostředky pro tolik drahého hardwaru. Finanční výdělek si poté rozdělí rovnoměrně nájemci hardwaru mining poolu.

Pokud se podaří vytěžit blok nějakému těžaři, poté se mu přičtou za jeho zásluhy odměny v podobě měny, u které těží. Do bloku se uloží transakce, o které bylo požádáno a blok se po skontrolování uloží do blockchainu.

Šifrování

Asynchroní šifry (Digitální podpis transakce)

Jejich základním principem jsou dva klíče. Jeden z klíčů je privátní a druhý je veřejný. Uschování privátních klíčů je pro nás klíčové. Privátní klíče můžeme vědět jenom my, protože by se dalo říci, že jsou to klíče k našim financím. Pro naši práci jsou pak důležité z toho hlediska, že kryptopeněženky slouží právě k bezpečnému uchovávání privátních klíčů.

Máme tedy veřejný klíč, kteří mohou znát všichni uživatelé a k němu vždy klíč soukromý. Pokud budeme chtít komunikovat s jiným uživatelem, (v našem případě provádět transakce) tak zakódujeme zprávu podle veřejného klíče uživatele, kterému chceme zprávu poslat a jediný způsob, jak je možné tuto zprávu rozkódovat je použít soukromý klíč uživatele.

To se může dělat například násobením velkých prvočísel. Protože vynásobit dvě prvočísla je jednoduchá operace, ale z jednoho čísla získat dvě prvočísla, jejichž vynásobením mohlo toto číslo vzniknout je operace velmi náročná. To znamená, že útočníci jsou postaveni před tak složitý výpočet, že se jim nevyplatí čekat na prolomení. Tomuto šifrování se říká RSA. Dalšími šifrovacími algoritmy jsou např. ECSDA, což je algoritmus, který využívá eliptických křivek.

Hashování

Hashování je proces, kdy z nějakého čísla dostanu číslo jiné, které v nějakém systému je adresou tohoto čísla, či toto číslo jinak definuje.

Pro hashování se využívá mnoho různých hashovacích algoritmů. Do nich patří např. SHA12, RIPEMD160.

2.3 Kryptoměny se zvýšenou bezpečností

Základní charakteristika

Jedná se o kryptoměny s větší mírou zabezpečení uživatele (tzn. nemožnost vystopování totožnosti uživatele a zabezpečení jeho financí) a jeho soukromých údajů.

Například nelze vystopovat původ transakce (nebo to jde složitěji než u běžných kryptoměn), nebo nelze zjistit email majitele účtu.

Srovnání kryptoměn se zvýšenou bezpečností

Zde je vytvořená tabulka, ve které jsou porovnané měny, které vynikají svojí privátností a vyšší bezpečností. [1][3][9][11][10][12]

KRYPTOMĚNA	ROK	POPIS
Monero (XMR)	2014 [14]	Momentálně nejoblíbenější kryptoměna. (2018) Zachovává maximální anonymitu uživatel. Spíše stálejší kurz. Značně škálovatelná kryptoměna - dobře reaguje na velkou zátěž a rychle reaguje na změny. (https://getmonero.org/)
Navcoin (NAV)	2014 [14]	Postavena na Bitcoin Core (přidává změny). Velmi rychlá (Transakce během 30 sekund). Silný marketing.Kompletně anonymní a decentralizovaná. Objevili se stížnosti investorů na soukromí. (https://navcoin.org/)
ZCash (ZEC)	2016 [14]	Méně známá. Užívá systém zk-snark (Dobře postavená konstrukce). Spolupracuje s jednou z největších bankovních společností na světě - JPMorgan Chase. Přispívá charitám. Těžko stopovatelná. (https://z.cash/)
PIVX (PIVX)	2016 [14]	Open-source kryptoměna. Horší marketingový tým. Silná komunita fanoušků. Panují obavy o růstu kryptoměny.
Spectrecoin (XSPEC)	2017 [14]	Unikátní díky efektivnímu algoritmu. Rychlé transakce. Tajné adresy - soukromí. Velký vývojářský tým.
Sumokoin (SUMO)	2017 [14]	Velmi mladá na trhu kryptoměn. Postavená na Moneru. Někteří kritici mají pochyby o anonymitě.
Verge (XVG)	2014 [14]	Open-souce software. Uživatelé mohou přepínat mezi soukromými a veřejnými transakcemi. Nepředvídatelné chování. Velký pokles hodnoty.
DeepOnion (ONION)	2017 [14]	Běží přes TOR síť. Dobře šifrované transakce. Těžko stopovatelné transakce.
CloakCoin (CLOAK)	2014 [14]	Anonymní transakce. Vysoká rychlost transakcí.
Enigma (ENG)	2017 [14]	Problém se škálováním.

KRYPTOMĚNA	ROK	POPIS
Hcash (HSR)	2018 [15]	Užívá tzv. dual sidechain. (může anonymně kontrolovat správnost transakcí mezi systémy, které využívají blockchain metody a systémy nevyužívající žádný block systém.) Pro těžení užívá jak Proof of work systémy, tak Proof of stake systémy.
StealthCoin (XST)	2014 [14]	Jedna z prvních kryptoměn kombinující Proof of stake systém s TOR sítí.
Crave Project (CRAVE)	2015 [14]	Používá blockchain kryptoměny DASH. Uživatelsky příjemné rozhraní.
InnovaCoin (INN)	2017 [14]	Těžení je přístupné více lidem, protože na těžení této kryptoměny není nutná tak velká výpočetní energie jako u jiných kryptoměn.
Zoin (ZOI)	2016 [14]	Těžení probíhá pouze skrz procesor. Čili nejsou kladeny požadavky na grafické karty apod.
Phore (PHR)	2017 [14]	Demokratická struktura komunity. Soukromí na trhu. Zabezpečení transakcí a anonymity uživatelů.
Grin	2019 [14]	Ranná fáze vývoje. Zatím spíše experimentální. Snaží se podporovat projekty investorů Investoři se můžou přidat svými projekty k vývojářské skupině.
Aeon (AEON)	2015 [14]	Postavená na Moneru. Snaží se být více mobilní než Monero. Na rozdíl od Monera si uživatel může vybrat, zda chce transparentní účet, nebo soukromý.

3 Kryptopeněženky

3.1 Základní rozdělení

Jedná se o programy, které umí komunikovat s blockchainem dané kryptoměny a dokáží schraňovat privátní klíče uživatele. V oblasti kryptopeněženek je velmi důležitá bezpečnost a kódování, aby se k transakcím a hlavně k samotným privátním klíčům nikdo nezvaný nedostal.

Objevují se různé implementace kryptopeněženek. Především se jedná o programy, které fungují jako klasické desktopové aplikace. Dále pak může být kryptopeněženka pro snazší použití a mobilnost ve formě programu na flash disku, Smart card a Java card, atd..

Placení z kryptopeněženek funguje na takovém principu, že soukromé klíče uložené v kryptopeněžence odblokují část uživatelových financí z blokchainu a provede se transakce. Resp. požadavek na transakci se odešle do systému, kde čeká na vykonání samotné operace.

Cold wallet – peněženky nepřipojené k síti (hardwarové a papírové – dražší, více bezpečné)

Hot wallet – peněženky připojené k síti (online, desktopové – bývají zdarma, intuitivní ovládání)

3.2 Příklady hardwarových peněženek

Kryptopeněženka uložená na paměťovém zařízení, které dokáže komunikovat s počítačem. Jsou na ni uložené privátní klíče uživatele v zašifrované podobě. Je žádoucí, aby do zařízení s kryptopeněženkou byl přístup možný pouze pro majitele. To znamená zabezpečený vstup (Možnosti jsou zabezpečení heslem, pinem, apod.).

Hardwarové peněženky se zavádí pro usnadnění přístupu uživatele ke své peněžence. Uživatel může mít peněženku, a tedy své finance stále u sebe. Navíc v případě rozšíření terminálů pro tyto zařízení v obchodech by se s hardwarovými kryptopeněženkami dalo platit jako například obyčejnou debetní kartou fyzicky (za fyzické přítomnosti majitele peněženky).

Příklady přenosných uložišť kryptopeněženek:

Flash disk

- Program nahraný na flash disku.

Smart card

- Může obsahovat navíc displej nebo klávesnici.
- Java card (JC)

- Vývojové prostředí JCIDE
- Klasická Java upravená pro karty
- Např. Ledger Nano

Mobilní telefon

- Mobilní aplikace
- Např. Green Address

Paper wallet

- Fyzicky zapsané privátní klíče (např. QR kód)

Online

- Peněženka uložena na cloudovém uložišti
- Např. Copay

Desktopová aplikace

- Klasická aplikace pro stolní, nebo přenosný počítač
- Např. Atomic wallet

3.3 Příklady softwarových peněženek

Atomic wallet

- uložení a správa BTC, ETH, XLM, XRP, LTC a více než 300 dalších coinů
- pravidelné aktualizace
- Aplikace pro počítače je k dispozici pro Windows, MacOS, Ubuntu, Debian a Fedoru.
- Snaha přesunout i na telefony. (říjen 2018)
- funkce Atomic Swaps – zmenšení poplatků pro uživatele
- nevýhodou je, že Atomic swap zatím není podporován všemi měnami

Bread wallet

- jednoduché odesílání bitcoinů
- volně na App Storu, nebo Google play

- vlastní klient – žádné nebezpečí třetí strany
- uživatelsky přívětivé, open source, zdarma
- žádné webové, nebo desktopové prostředí

Mycelium

- mobilní peněženka
- pro Iphone + Android
- integrovaný skener QR kódů
- bezpečný chat mezi uživateli
- dobré soukromí, pokročilé zabezpečení, bohatá na funkce, open source software, zdarma

Exodus

- nová a neznámá digitální peněženka
- pouze na Desktop
- podpora: Bitcoin, Ethereum, Litecoins, Dogecoins a Dash
- jednoduchý průvodce pro zálohování peněženky

Copay

- Desktop, telefon, online
- Dobré zabezpečení
- Vícenásobné ukládání peněz
- Může být pomalá a nereagující, omezená podpora uživatelů

Jaxx

- Podpora: Ethereum, Ethereum Classic, Dash, DAO, Litecoin, REP, Zcash

Rootstock

- k dispozici na různých platformách a zařízeních (Windows, Linux, Chrome, Firefox, OSX, mobilní zařízení a tablety pro Android, mobilní zařízení i tablety iOS) a propojuje se s webovými stránkami prostřednictvím rozšíření prohlížeče Firefox a Chrome
- převod mezi mincemi Ethereum, Bitcoin a DAO

Armory

- pro zkušené uživatele
- open source
- funkce chlazení, transakce s více podpisi
- jednorázové tiskové zálohy
- import klíčů
- šifrování odolné proti GPU
- odstranění použitých klíčů a zahlazení stop

Trezor

- por velké množství bitcoinů
- transparentní opensource
- podpora Windows, OS X, Linux
- velké množství bitcoinů vysoce zabezpečené
- vhodnější pro delší držení a spoření
- cold wallet

Ledger Nano

- hardwarová deterministická peněženka
- pro uživatele Bitcoinu
- druhá vrstva zabezpečení
- kompaktní zařízení USB založené na čipové kartě
- zařízení chráněné kovovým otočným krytem
- cold wallet

Green Address

- pro začátečníky
- Desktop, online, mobilní telefon
- Dvoufaktorové ověřování
- Zálohování papírových peněženek

- Všechny transakce musí nejprve Green Addres schválit - ne zcela plná kontrola nad financemi
- hot wallet

Blockchain.info

- Blockchain.info je jedna z nejoblíbenějších kryptopeněženek
- Podpora Bitcoin
- Přístup z libovolného prohlížeče, nebo smartphonu
- Aplikace pro mobilní telefony vyžaduje PIN kód, zatímco přístup přes prohlížeč umožňuje dvou faktorové ověření
- Došlo k výpadkům, nutná podpora třetí strany
- Bezpečí, důvěryhodná třetí strana
- Hot wallet

3.4 Rozbor implementace kryptopeněženky Bitcoin wallet

Popis fungování

Kryptopeněženka začne být aktivní při připojení se k počítači. Základní firmware karty se pomocí sady základních bytových příkazů zkontaktuje s počítačem. Ověří se bezpečnostní klíče a pokračuje se k vykonávání dalších operací.

Peneženka umí získat z počítače privátní klíče uživatele. Privátní klíče se šifrují a hashují, aby nebyli dohledatelné a ukládají se do paměti karty.

Dále si peněženka umí ukládat data z blockchainu dané kryptoměny. To potřebuje, aby měla celkový obraz transakcí a mohla rozpoznat, jestli jsou platby v pořádku. S tím souvisí i analýza dat. Říká se tomu kontext transakcí.

S peněženkou se dají i provádět platby (čili transakce). Na to peněženka potřebuje umět digitální podpis transakce. K tomu použije privátní klíče, které má uživatel u sebe (s těmito klíčemi jsou spojené finance v dané kryptoměně) a poté některý z druhů algoritmů, které má k dispozici. Tedy např. ECSDA (Eliptické křivky).

Peněženka s počítačem komunikuje v řeči bytů, neboť jiná možnost skrz čipovou kartu není. Má tedy funkce pro převody funkcí do bytové řeči. Tato základní řeč se určuje firmwarem karty.

Peněženka dále poskytuje třídy, které mají speciální matematické funkce, které se využívají při výpočtech hashů a digitálních podpisů.

Peněženka má funkce pro získání zůstatku na kartě.

Třídy

BCDUtils.java

Převede množství bitcoinů do zobrazitelné podoby pro druhou fázi ověření.

Base58.java

Provádí kódování a dekódování. (převádí mezi anglickou abecedou (ALPHABET) včetně čísel do bytů v BASE58TABLE)

Bip32.java

Systém hierarchie deterministických peněženek. (https://github.com/bitcoin/bips/blob/master/bip-0032.mediawiki)

Bip32Cache.java

Zde se nacházejí různé metody, jak pracovat s cache pamětí při používání BIP.

Crypto.java

Metody a nástroje šifrování použité v aplikaci. Tedy příprava digitálních podpisů, ověřování, ...

GenericBEHelper.java

Základní operace s velkými čísly. (Porovnání bytů, ...)

HmacSha512.java

Hashovací systém HMAC. Tvoří základ podepisování transakcí.

Keycard.java

Implementace zabezpečení peněženky na JAVA kartě. Ověřování klíče uživatele.

LWNFCForumApplet.java

NFC je typ bezdrátového spojení, kdy mohou komunikovat obě strany. Tedy u nás čtečka a karta mezi sebou (čtečka je připojena k počítači).

LedgerWalletApplet.java

public LedgerWalletApplet()

Inicializace potřebných tříd aplikace Ledger wallet applet. Nastavení pinů a klíčů peněženky.

private static reset ()

Vygeneruje náhodné pole bytů a udělá z něj klíč.

protected static boolean isContactless()

Vrátí TRUE, jestliže je peněženka na kartě připojená přes APDU protokol.

private static void checkAccess()

Kontrola přístupu. Vyhazuje výjimky, pokud karta není správně připojená, nebo pokud nebyl úspěšně přijat pin.

private static void checkInterfaceConsistency()

Opět kontrola připojení.

private static void verifyKeyChecksum()

Ověřuje kontrolní součet klíče. Vyhazuje výjimku v případě, že součet neodpovídá.

private static void signTransientPrivate()

Podepisuje krátkodobý privátní klíč.

private static void checkAirgapPersonalizationAvailable()

Vyhazuje podmínku, pokud klíč nebyl schválen.

private static void handleGetAttestation()

APDU nastaví odcházející data a odešle.

private static void handleAirgapKeyAgreement()

APDU nastaví příchozí data a přijme. Vyhazuje podmínky v případě špatné délky dat v bufferu. Vytváří nový pár klíčů (generuje veřejný podle soukromého), podepisuje data a odesílá je.

private static void handleSetAttestationPublic(APDU apdu)

Kontroluje přijímaná data.

private static void handleHasCachedPublicKey(APDU apdu)

Opět kontroluje přijímaná data podle velikosti derivace. Dívá se na první byte v bufferu a porovnává jej s proměnou result.

private static void handleStorePublicKey(APDU apdu)

Ukládá privátní i veřejný klíč.

private static void handleGetHalfPublicKey()

Podílí se na generování veřejného klíče.

private static void handleGetFeatures()

Nastavuje funkce proprietaryAPI.

private static void handleTrustedInput()

Pracuje s transakcemi, pokud lze důvěřovat vstupním datům.

private static void handleHashTransaction()

Kontroluje a pouští transakce.

private static short writeAmount()

Vrací celkovou částku v zobrazitelné podobě.

private static void handleHashOutputFull()

Pracuje s celým výstupem. Na konci odesílá data skrz APDU.

private static void handleHashSign()

Pracuje s hashem podpisu.

private static void handleSignMessage()

Kontroluje, zda jsou zprávy, které chce aplikace odeslat, podepsané.

private static void handleSetUserKeycard()

Nastavuje uživatele karty. Pracuje s daty, které slouží k párování.

private static void handleSetup()

Provádí základní nastavení (setup).

private static void handleVerifyPin()

Ověřuje správnost pinu.

private static void handleSetContactlessLimit()

Nastavuje bezkontaktní limit.

private static void handleGetFirmwareVersion()

Získává verzi firmwaru. (základního softwarového vybavení karty)

private static void handleGetOperationMode()

Do bufferu na adresu 0 ukládá jaký se používá operační mód. (nastavuje momentální mód, nebo NFC)

private static void handleAdmSetKeycardSeed()

Nastavuje seed klíče karty. Seed je náhodné číslo, které se přimíchává do výpočtu, aby se do výpočtu přimíchala náhoda.

public static void clearScratch()

Vyčistí scratch. (Scratch je pole bytových instrukcí deklarované v public LedgerWalletApplet())

public void process()

Plní veškeré instrukce uložené v bufferu přes výše zmíněné metody třídy LedgerWalletApplet.java.

public static void install()

Metoda install vytvoří novou instanci třídy LedgerWalletApplet() a zadá jí parametry nutné ke spuštění.

private static final byte FIRMWARE_VERSION

Verze firmwaru. V tomto poli bytů jsou uloženy veškeré konstanty, které se v třídě LedgerWalletApplet.java používají.

MathMod256.java

Matematické funkce pro operování s byty. Porovnávání čísel. Modulární funkce.

ProprietaryAPI.java

Další základní funkce, které nejsou obsaženy v rozhraní Java Card API nebo v optimalizacích.

TC.java

Schraňování historie transakcí. Aby aplikace měla potřebný kontext transakcí, se kterými pracuje.

Transaction.java

Analýza bitcoinových transakcí.

Uint32Helper.java a Uinit64Helper.java

Opět pomocné třídy pro základní operace s 64 bitovými a 32 bitovými čísly.

3.5 Rozbor implementace kryptopeněženky Eligibility wallet

Tato kryptopeněženka je převzata z GitHubu [17]. Jedná se pouze o testovací peněženku. To znamená, že některé její funkce jsou osekány a peněženka se pouze snaží realisticky simulovat operace na reálné kryptopeněžence.

Pro svoji testovací kryptopeněženku jsem si zvolil Eligibility wallet zejména proto, že obsahuje naprogramované funkce, díky kterým se dá s peněženkou lehce komunikovat a lze z ní odebírat informace, které jsem pro svoji práci potřeboval. Z těchto funkcí zejména pak zmíním možnost otestovat rychlost procesů jako je šifrování, hashování apod..

4 Java Card

4.1 Komunikace s kartou

Komunikace z hardwarového hlediska

Čipové karty obsahují malý výpočetní přístroj, který se skládá s tištěných spojů. Většinou nemá žádnou vlastní baterii (Karta s níž jsem pracoval neměla vlastní baterii.) a tudíž nemůže pracovat sám bez energie zvenčí.

Limitními hardwarovými parametry karet jsou přenosová rychlost a velikost uložiště, které je možné zaplnit nahráním aplikace. Za základní parametr můžeme také považovat verzi firmwaru.

Počítač pak komunikuje s čipovou kartou prostřednictvím čtečky karet.

Komunikace ze softwarového hlediska

Firmware čipové karty

Firmware je obecně základní software výpočetního stroje, který řeší start a spuštění systému. Zde u java karet to je seznam bytových příkazů, které závisí na druhu karty.

APDU - komunikační protokol

Aplikační protokolová datová jednotka (APDU) je komunikační formát mezi kartou a aplikacemi mimo kartu. Při odesílání velkých bytových polí jako dat odpovědí poskytuje třída APDU speciální metodu sendBytesLong (), která spravuje vyrovnávací paměť APDU. APDU je vlastně vyrovnávací paměť, která má podobu globálního pole. Hardware karty má totiž příliš malý buffer, takže musí informace posílané na kartu a z karty rozdělovat na menší oddíly a přenos musí být formou bytů. To zajišťuje objekt APDU, jeho instance a metody. Jestli to správně chápu, tak APDU je nějaký způsobem hierarchicky navázáno na JCRE, což je Java Card Runtime Environment. To ale asi není nic překvapivého, protože na tomhle principu bude fungovat vše spojené s Java Card.

4.2 Zabezpečení Java karet

Zablokování karty

Většina čipových karet obsahuje nějaký přístupový klíč. Karty mívají od výroby sadu defaultních klíčů, které bývají pro všechny karty stejné. Když se snažíme dorozumět s kartou, musíme při komunikaci používat její klíč. Klíč karty se dá změnit, ovšem poté musíme při komunikaci znát tento klíč.
Problém nastává ve chvíli, kdy se pokusíme s kartou spojit přes neplatný resp. špatný klíč. V takovém případě se karta zablokuje a nejde s ní komunikovat, dokud se neodblokuje. [1]

5 Zprovoznění testovací kryptopeněženky na Java kartě (Eligibility wallet)

5.1 Aplikace GlobalPlatformPro

Pro komunikaci s kartou budeme využívat aplikaci GlobalPlatformPro [12]. Tato aplikace poskytuje sadu základních příkazů pro komunikaci s kartou. Aplikace nemá grafické prostředí, ale lze s ní pracovat skrz počítačovou konzoli.

5.2 Nahrání kryptopeněženky na kartu

V případě, že chceme nahrát program na kartu, musíme si vytvořit z programu spustitelný soubor kompatibilní s kartou. Takový soubor je např. typu Wireshark capture file (soubor *.cap). Spuštěním aplikace gp.jar, nebo gp.exe (GlobalPlatformPro) se dostaneme do konzole (pokud je správně připojena karta) a zadáme příkaz: java -jar gp.jar -install *.cap. Soubor se nahraje na kartu. Karta odpoví, jestli byla instalace úspěšná, či neúspěšná.

5.3 Připojení karty k počítači

Čtečka karet

K počítači se karta připojuje prostřednictvím čtečky karet. Já ke své práci využíval čtečku Microsoft Usbccid Smartcard Reader (WUDF). Čtečka se k počítači připojovala přes USB konektor.

6 Aplikace komunikující s kartou

6.1 Rozbor aplikace

Aplikace se skládá z dvou tříd.

Dump.java

Třída, která má na starost převod odpovědi peněženky do řeči, které rozumí naše aplikace a zároveň převod zpět našich příkazů vyslané k peněžence.

LedgerEligibilityReportGUI.java

Toto je hlavní třída aplikace.
Třída rozšiřuje třídy starající se o grafické prostředí. (extends JFrame) Aby bylo možné jednoduše vytvářet GUI nad aplikací.
Hlavní třída ještě obsahuje podtřídu APDUResponse, což je statická třída, která využívá třídu Dump. Jak už vyplývá z názvu třída se stará o dopovědi peněžence skrz APDU protokol. Obsahuje tedy v proměnných uložené příkazy pro peněženku převedené právě přes třídu Dump.
Aplikace kontroluje, jestli je karta správně připojená.
Aplikace se spojuje s peněženkou na kartě prostřednictvím metody Exchange
Metoda Exchange vytváří instanci třídy APDUResponse a také ji poté vrací. Stará se o výměnu dat s peněženkou.
Dále pak aplikace provádí postupně test vyjmenovaných funkcí, které peněženka umí. Z tohoto testování získává čas, jak dlouho trval výpočet testované funkce. Tento čas vypisuje do konzole a ukládá si jej, aby jej mohla později vypsat do grafického prostředí aplikace.
Nakonec aplikace vykreslí grafické prostředí, které je udělané nástroji z vývojového prostředí NetBeans 8.2 a vypíše do něj, po interakci uživatele, výsledky testování.

6.2 Grafické prostředí

Grafické prostředí (GUI) je zpracované v programu NetBeans IDE 8.2, který poskytuje možnosti pro snadnou tvorbu grafických nástaveb.
GUI obsahuje interaktivní tlačítka. Spuštěním těchto tlačítek se vypíše text do textových polí. Stiskem prvního tlačítka (Test 1) dostaneme výpis do textového pole jedna, který se týká testování jednotlivých funkcí peněženky na kartě. Zatímco stiskem druhého tlačítka získáme výpis do druhého textového pole. Druhý výpis bude obsahovat pouze rychlost kompletní provedené transakce na testované peněžence, podle metody podepisování transakce.

7 Výsledky práce

7.1 Výpis testovací aplikace

Test šifrovacích metod kryptopeněženky

Po spuštění desktopové aplikace jsem dostal výpis rychlostí jednotlivých metod. Nejdříve pouze do konzole [Obrázek 1], poté i do grafického prostředí [Obrázek 2], které jsem vytvořil.

V grafickém prostředí poté již vidíme i implementovanou funkci pro součet a výpis času ověření celé transakce peněženkou. k tomu dál v sekci Časová výkonnost kryptopeněženky při podpisu jedné transakce.

Časová výkonnost kryptopeněženky při podpisu jedné transakce

Pro co nejpřesnější zjištění výkonnosti peněženky při podpisu jedné transakce jsem se rozhodl udělat měření, které zkoumá jednotlivé časy ověřování transakce.

Měření jsem prováděl na svém přenosném počítači s klasickou kancelářskou výbavou (procesor Intel CORE i5, bez přidané grafické karty).

K měření jsem se nejdříve pokoušel využít příkazovou řádku MS Windows (cmd.exe), abych mohl aplikaci pustit ve for cyklu libovolně mnohokrát.Ukázalo se však, že procesor nezvládá obhospodařovat všechny procesy a tak jsem musel měření provádět ručně. Dalším problémem při programovém měření bylo, že přetížení procesoru uměle navyšovalo čas podpisu jedné transakce a tudíž měření nebylo odpovídající.

Naměřená data jsem ukládal do textového souboru prostřednictvím tříd javy BufferedReader a BufferedWriter.

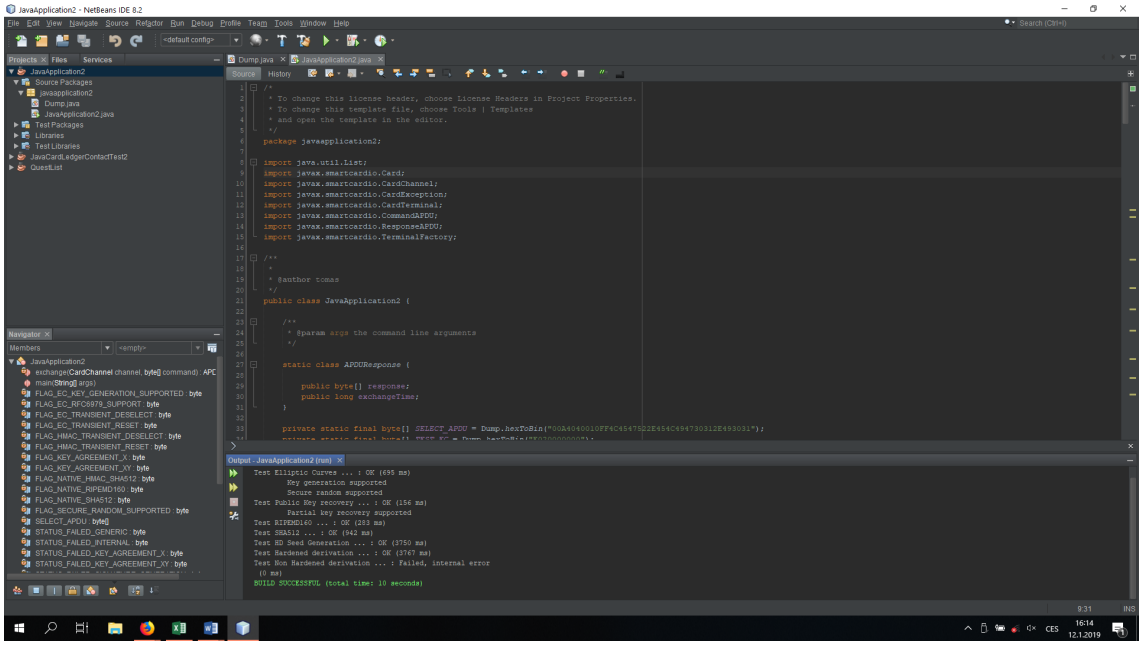
Následně jsem textový soubor přečetl skriptem napsaným v jazyku Python. Tento skript, který jsem vytvořil s použitím knihoven pandas, XlsxWriter, xlrd, matplotlib mi umožnil konstrukci bodového grafu z naměřených hodnot [Obrázek 3]. A následně jsem tímto skriptem vložil naměřená data do souboru .xlsx včetně absolutních odchylek a vypočítal konečný výsledek měření.

Na grafu si můžeme všimnout několika naměřeným hodnot, které se výrazně vymykají průměrné hodnotě. Nicméně vzhledem k jejich nízkému počtu neovlivňují výrazně velikost relativní odchylky výsledné hodnoty. Mírně vzrůstající tendenci hodnot měření v závislosti na čísle měření mohu přisoudit zatížení procesoru, které se zvyšovalo při vzrůstajícím počtu měření. To nám ale měření nekazí, neboť informace o vzrůstajícím počtu milisekund při opakovaném placení bitcoiny po sobě může být pro uživatele také důležitá informace.

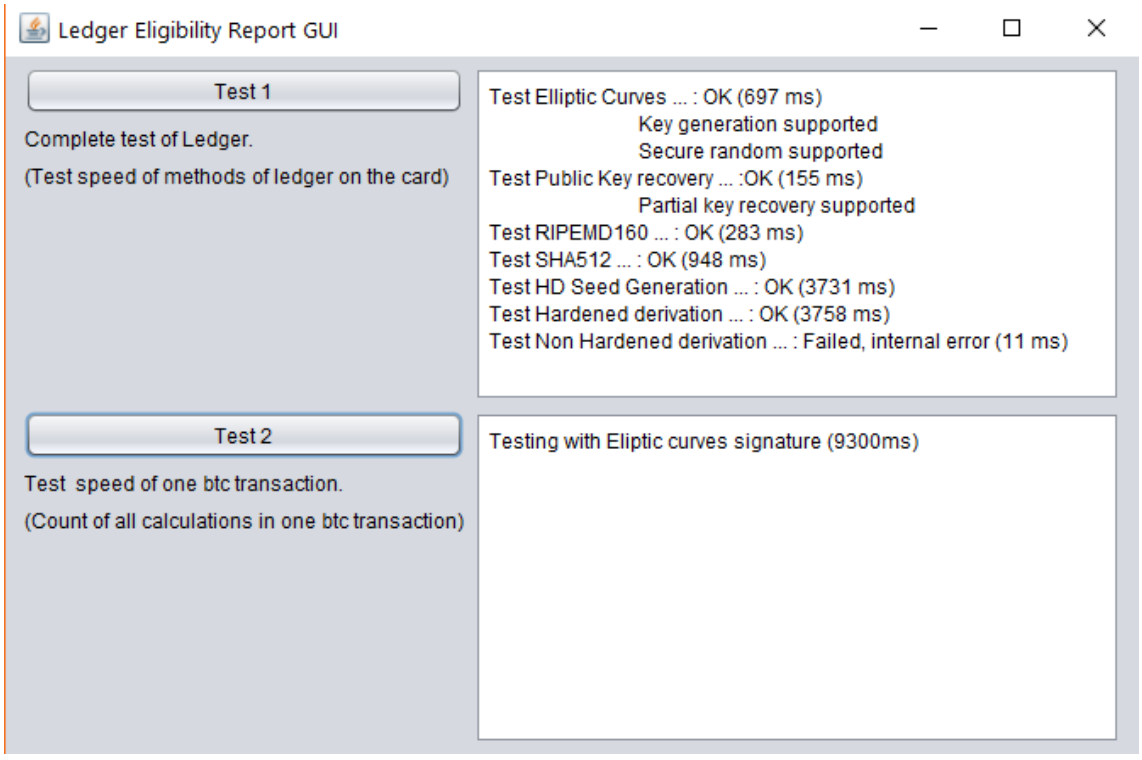
Výsledná hodnota s relativní odchylkou je:

$$(9292.617 \pm 6.941)ms$$

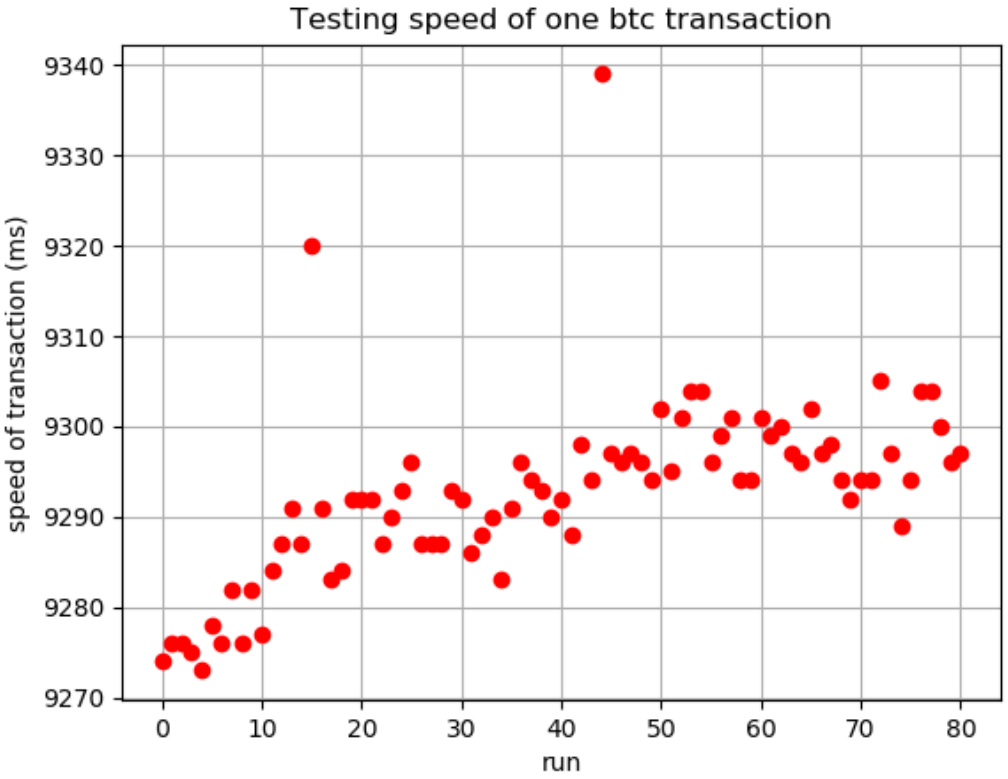
Důvody odchylky jsou zejména jistá míra náhody při generování klíčů a při výpočetních operací použitých v aplikaci. Další důvod může být nedokonalé měření



Obrázek 1: Výpis rychlostí metod do konzole.



Obrázek 2: Výpis rychlostí metod do grafického prostředí.



Obrázek 3: Graf zachycující naměřené hodnoty v bodovém provedení.

způsobené výpočetním přístrojem, na kterém je aplikace spuštěna. Nepřesnosti v měření mohly být způsobené také nedokonalým přenosem informací.

7.2 Závěr

V závěru práce musím práci zhodnotit. Mohu říci, že práce byla úspěšně dokončena, neboť jsem splnil stanovené cíle na začátku. Nejdříve jsem se seznámil s teoretickým základem problematiky. Prozkoumal jsem dostupnou literaturu a internet a udělal jsem výpis kryptoměn se zvýšenou bezpečností a také výpis jednotlivých kryptopeněženek.

Obtížnější část pak byla pochopení kódu peněženky Bitcoin wallet. Rozepsal jsem třídy a v nich použité metody a pokusil se aspoň základně popsat jejich funkci v programu.

Závěrečným úkolem bylo zprovoznit peněženku Eligibility wallet na přidělené čipové kartě a její propojení s desktopovou aplikací. Úspěšně se mi podařilo nahrát Eligibility wallet na čipovou kartu. Problém nastal ve chvíli, když jsem se pokusil kartu spojit s moji testovací aplikací, neboť zřejmě neseděli bezpečnostní klíče v mém počítači s klíči na kartě. Karta se proto uzamkla a dál se mi s ní nepodařilo komunikovat. Vyměnil jsem tedy kartu za jiný model, opět spojil peněženku a nyní už proběhlo spojení bez problémů. Podařilo se mi získat kontrolní výpis do konzole. Mohl jsem tedy začít pracovat na grafickém prostředí (GUI) v aplikaci, která s kartou komunikovala.

Aplikaci se mi podařilo implementovat a získal jsem interaktivní grafický výstup, ze kterého můžu testovat buď samostatné operace, které kryptopeněženka nabízí, nebo otestovat celou bitcoinovou transakci (tedy sečíst časy výpočtů, které se v transakci využívají).

Vrcholem práce bylo úspěšné měření výkonu peněženky při ověření jedné bitcoinové transakce. Tímto krokem jsem zároveň překročil cíle vymezené na začátku práce.

Práce má i další využití do budoucna. Kryptoměny nejsou mrtvou technologií a mají velký potenciál se vyvíjet do budoucna. Bitcoinovým převodem se dá platit ve více internetových obchodech než dříve a obchodů a banek podporujících platbu bitcoinem neustále přibývá. Čipové karty jako kryptopeněženky by se daly v budoucnosti využívat jako klasické debetní platební karty. To by velmi zjednodušilo uživateli platby s kryptoměnami, neboť nyní je platba Bitcoinem neohrabaná a ve fyzickém světě téměř nemožná.

Má práce může v budoucnu umožnit vývoj kryptopeněženek na čipových kartách a také zpřístupnit informace o těchto technologiích česky mluvícímu obyvatelstvu, které mělo do nynějška k dispozici téměř výhradně anglicky psané texty.

Reference

[1] JEBAVÝ, Josef, Ukázková aplikace šifrovaní s Java Card. Praha: České vysoké učení technické, fakulta elektrotechnická, 7. 3. 2008.

[2] STROUKAL, Dominik, SKALICKÝ, Jan, Bitcoin a jiné kryptopeníze budoucnosti.

[3] en.bitcoin.it/wiki

[4] github.com/bitcoinjs/bitcoinjs-lib

[5] bitcointalk.org

[6] blockgeeks.com/guides/cryptocurrency-wallet-guide

[7] github.com/JavaCardOS/BitcoinWallet

[8] bitcoin.stackexchange.com

[9] kryptomagazin.cz

[10] kingpassive.com/best-privacy-coins-2018/

[11] cryptocompare.com/coins/guides/what-is-zcash/

[12] deeponion.org

[13] github.com/martinpaljak/GlobalPlatformProget-it-now

[14] coinmarketcap.com

[15] h.cash

[16] stealth.com

[17] github.com/LedgerHQ/ledger-javacard-eligibility

[18] D. CHAUM, Blind signatures for untraceable payments

[19] D. CHAUM, A. FIAT, M. NAOR, Untraceable eletronic cash

[20] P. JULIE, Requiem for a Bright Idea

[21] LAURIE, SUSAN, SABETT, SOLINAS, How to Make a Mint: The Cryptography of Anonymous Electronic Cash

[22] cs.wikipedia.org/wiki/Proof-of-Work

[23] Kryptoměny typu Proof of Work a Proof of Stake. www.bitcoin-now.cz

[24] TRADE-ARENA.CZ. Jak fungují Proof of Work a Proof of Stake — Trade-Arena.cz. t.tradearena.cz