VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ
ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION
DEPARTMENT OF TELECOMMUNICATIONS

METODY OPTIMALIZACE DIGITÁLNÍCH PODPISŮ

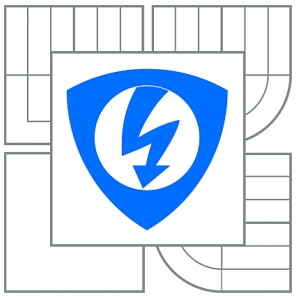
DIPLOMOVÁ PRÁCE
MASTER'S THESIS

AUTOR PRÁCE Bc. ALEŠ ŠPIDLA
AUTHOR

BRNO 2013

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA ELEKTROTECHNIKY A KOMUNIKAČNÍCH TECHNOLOGIÍ

ÚSTAV TELEKOMUNIKACÍ

FACULTY OF ELECTRICAL ENGINEERING AND COMMUNICATION

DEPARTMENT OF TELECOMMUNICATIONS

METODY OPTIMALIZACE DIGITÁLNÍCH PODPISŮ

METHODS FOR OPTIMIZATION OF DIGITAL SIGNATURES

DIPLOMOVÁ PRÁCE

MASTER'S THESIS

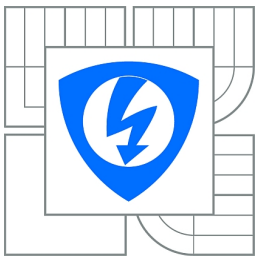
AUTOR PRÁCE — Bc. ALEŠ ŠPIDLA

AUTHOR

VEDOUCÍ PRÁCE — Ing. LUKÁŠ MALINA

SUPERVISOR

BRNO 2013



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

Fakulta elektrotechniky a komunikačních technologií

Ústav telekomunikací

Diplomová práce

magisterský navazující studijní obor
Telekomunikační a informační technika

Student:	Bc. Aleš Špidla	***ID:***	111148
Ročník:	2	***Akademický rok:***	2012/2013

NÁZEV TÉMATU:

Metody optimalizace digitálních podpisů

POKYNY PRO VYPRACOVÁNÍ:

V rámci diplomové práce se zaměřte na druhy digitálních podpisů a možnosti jejich optimalizace v informačních a komunikačních systémech. Zhodnoťte optimalizační techniky, konstrukci daných digitálních podpisů, bezpečnost, efektivitu a jejich praktickou aplikovatelnost do systémů ICT.

Navrhněte kryptografické řešení využívající optimalizačních technik digitálních podpisů pro systémy s centrálním ověřováním (např. sdílené úložiště, sdílené služby) a ověřováním více zpráv v reálném čase (např. Vehicular Ad Hoc Networks). Navržené řešení implementujte a zhodnoťte výkonost a bezpečnost navrženého řešení.

DOPORUČENÁ LITERATURA:

[1] STALLINGS, William. Cryptography and Network Security. 4th edition. [s.l.] : [s.n.], 2006. 592 s. ISBN 0131873164.
[2] MENEZES, Alfred, VAN OORSCHOT, Paul, VANSTONE, Scott. Handbook of applied cryptography. Boca Raton: CRC Press, 1997. 780 s. ISBN 0849385237.

Termín zadání:	11.2.2013	***Termín odevzdání:***	29.5.2013

Vedoucí práce: Ing. Lukáš Malina
Konzultanti diplomové práce:

prof. Ing. Kamil Vrba, CSc.
Předseda oborové rady

UPOZORNĚNÍ:

SIX research centre
sensor, information and communication systems

Faculty of Electrical Engineering
and Communication

Brno University of Technology
Purkynova 118, CZ-61200 Brno, Czechia

http://www.six.feec.vutbr.cz

Výzkum popsaný v této diplomové práci byl realizován v laboratořích podpořených z projektu SIX; registrační číslo CZ.1.05/2.1.00/03.0072, operační program Výzkum a vývoj pro inovace.

MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY

EVROPSKÁ UNIE
EVROPSKÝ FOND PRO REGIONÁLNÍ ROZVOJ
INVESTICE DO VAŠÍ BUDOUCNOSTI

2007–13
OP Výzkum a vývoj pro inovace

Abstrakt

V rámci diplomové práce je stručně popsaná problematika digitálních podpisů a základní metody (RSA, DSA, ECDSA) pro vytvoření a ověření digitálního podpisu. Dále je podrobněji popsána metoda skupinových podpisů navržená autory Boneh, Boyen a Shacham (BBS) a metoda navržená autory Bonen, Shacham (BS). Na to navazuje problematika implementace dávkového podepisování a ověřování pro výše zmíněné metody a jejich praktickém využití, především pro systémy Vehicular ad-hoc network (VANET) a cloudového uložiště v oblasti forenzního IT.

Cílem práce je určit, která z metod BBS a BS je vhodnější pro výše zmíněné systémy z hlediska výpočetní náročnosti. Z tohoto důvodu byl vytvořen program pro porovnání metod BBS a BS, kdy je porovnávána časová náročnost těchto metod. Výsledky měření jsou popsány v závěru práce spolu s odůvodněním, proč jsou jednotlivé metody vhodné pro dané systémy.

Klíčové slova

kryptografie, podpisové schémata, dávkové ověřování, VANET sítě

Abstract

The thesis briefly describes the digital signatures and basic methods (RSA, DSA, ECDSA) for the creation and verification of the digital signature. The method of group signatures designed by Boneh, Boyen and Shacham (BBS) is described in more details as well as the method designed by Bonen and Shacham (BS). The thesis further explores related issue of the implementation of batch signing and verification for the above mentioned methods and their practical application, particularly for systems Vehicular ad-hod network (VANET) and cloud storage in the field of forensic IT.

The purpose of the thesis is to determine which of the methods BBS and BS is more suitable for these systems in terms of computational complexity. For this reason, the author created the program for the comparison of methods BBS and BS. The programme compares the time consumption of the methods for signing and verification of incoming messages. The results of the measurement are summarized in the conclusion along with the justification why the particular methods are suitable for the systems.

Keywords

cryptography, signature schemes, batch verification, VANET network

ŠPIDLA, A. *Metody optimalizace digitálních podpisů*. Brno: Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií, 2013. 54 s. Vedoucí diplomové práce Ing. Lukáš Malina.

Prohlášení

Prohlašuji, že svou diplomovou práci na téma Metody optimalizace digitálních podpisů jsem vypracoval samostatně pod vedením vedoucího diplomové 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 autor uvedené diplomové práce dále prohlašuji, že v souvislosti s vytvořením tohoto projektu jsem neporušil autorská práva třetích osob, zejména jsem nezasáhl nedovoleným způsobem do cizích autorských práv osobnostních a jsem si plně vědom následků porušení ustanovení § 11 a následujících autorského zákona c. 121/2000 Sb., včetně možných trestě právních důsledku vyplývajících z ustanovení části druhé, hlavy VI. díl 4 Trestního zákoníku c.40/2009 Sb.

V Brně dne 28.5.2013

podpis autora

Poděkování

Děkuji vedoucímu mé diplomové práce Ing. Lukáši Malinovi, za poskytování odborných rad a tipů při vypracovávání diplomové práce.

V Brně dne 28.5.2013

podpis autora

Obsah

ÚVOD

Tato práce obsahuje úvod do asymetrické kryptografie včetně popisu jednotlivých podpisových schémat, optimalizačních metod a jejich využití v informačních a komunikačních systémech. V úvodu je stručně vysvětlena problematika digitálních podpisů, včetně popisu certifikační autority, možných útoků na digitální podpis a kvalifikaci podpisových schémat. Druhá kapitola se věnuje nejčastěji využívaným podpisovým schématům RSA (Rivest-Shamir-Adleman), DSA (Digital Signature Algorithm), ECDSA (Ecllipvic Curve Digital Signature Algorithm), dále skupinovým podpisovým schématům BBS (Boneh-Boyen-Shacham), BLS (Boneh-Lynn-Shacham) a BS (Boneh-Shackam). Třetí kapitola přibližuje problematiku dávkového ověřování a agregačních podpisů. V následující kapitole jsou popsány tři případy využití níže popsaných technik v prostředí komunikačních a informačních technologií. Jmenovitě se jedná o problematiku bezdrátových senzorových sítí, cloudové řešení a sítě VANET (Vehicular Ad-Hoc Network). V další části je popis vytvořeného programu pro dokázání výhodnosti využití dávkového podepisování oproti obyčejnému. V této kapitole je kromě praktických ukázek rovněž odůvodnění časové úspory při využití dávkového ověřování. V současné době je totiž při využívání digitálního podepisování kladen důraz na časovou úsporu v tomto směru, jelikož je potřeba zpracovávat v krátké době velké množství přijatých zpráv.

1 Úvod do digitálních podpisů

1.1 Digitální podpis

Digitální podpis má zásadní význam pro autentičnost a zachování integrity dat, jenž byly zaslány příjemci v digitální podobě. Účelem digitálního podpisu je spojit přenášenou zprávu s identitou odesilatele. Digitálně podepsat tedy znamená, pomocí unikátní hodnoty vložit podpis do odesílané zprávy. Tím se zaručí pravost odesilatele a neměnnost zprávy. Samotná struktura podepisovacího procesu vypadá následovně:

- A značí stranu, která zprávu podepisuje.
- M značí soubor zpráv, které se mají podepsat.
- S značí podpis zprávy, případně řetězec znaků pevné délky.
- S_A určuje transformaci M do S, pomocí které se vytváří podpis ze zprávy M na základě strany A.
- V_B je ověřovací proces, který je veřejně znám a jehož cílem je ověřit podpisy vytvořené stranou A.

1.2 Podepisování a ověření

Strana A vytvoří zprávu *m* z množiny M a provede následující:

1. Vypočte $s = S_A(m)$.
2. Odešle dvojici (*m*,*s*), kde *s* je podpis pro zprávu *m*.

Strana B pro ověření provede následující kroky:

1. Získá ověřovací funkci V_A z A.
2. Vypočte hodnotu ověřující pravost podpisu $u = V_A(m, s)$.
3. Jestli-že *u* souhlasí (*u* = true), je podpis přijat. Pokud *u* nesouhlasí (*u* = false), příjemce digitální podpis odmítne.

1.2.1 Požadované vlastnosti pro podepisování a ověřování

Obecně existují dvě základní pravidla, která musí funkce podepisování a ověřování splňovat.

- *s* je platný podpis A od zprávy *m* tehdy a jen tehdy, pokud $V_A(m,s)$ = true.
- Je výpočetně nereálné, aby jiný subjekt než *A* našel pro všechny *m* z *M* takové *s* z *S*, aby se $V_A(m,s)$ = true. Podpis se tedy podle tohoto pravidla vždy jednoznačně váže na zprávu, ke které byl vytvořen.

1.3 Šifrování pomocí veřejného klíče

Asymetrické kryptosystémy využívají dvou klíčů, veřejného a soukromého. Výkon asymetrické kryptografie je oproti symetrické pomalejší. Specifikem je, stejně jako u symetrických systémů, jednocestná funkce $y = f(x)$, kdy je výpočet hodnoty y pomocí argumentu *x* relativně snadný, ovšem opačný výpočet je kvůli výpočtové složitosti prakticky nemožný. Podle toho, jaký je využit typ této funkce, jsou známy kryptosystémy založené na faktorizaci čísla, problému diskrétního logaritmu a na eliptických křivkách. [1]

Aby nedocházelo k nejasnostem, soukromý klíč se používá v kryptosystémech v souvislosti s veřejným klíčem, tedy asymetrických. Tajný klíč se užívá především v symetrické kryptografii. Může to být mírně zavádějící, ale i když dvě nebo více stran sdílejí tajnou zprávu, klíč stále zůstává soukromý, protože jej zná pouze podpisovatel.

Šifrování pomocí veřejného klíče předpokládá, že znalost veřejného klíče *VK* není dostačující k tomu, aby byl zjištěn soukromý klíč *SK*. Jinými slovy je nutné, aby zde byla zastoupena jednosměrná funkce se zadními vrátky (trapdoor function).

1.4 Digitální certifikát

Za digitální certifikát se považuje digitálně podepsaný veřejný klíč, jenž je vydán certifikační autoritou. Jedná se o dokument vydaný třetí důvěryhodnou stranou a slouží k ověření, že majitel digitálního podpisu je důvěryhodná osoba. Certifikát se přiřazuje přímo k digitálnímu podpisu a podle standardu X.509 obsahuje níže uvedené informace. [2]

- Serial Number – tato položka není povinná, její hlavní funkcí je ulehčení orientace.
- Subject – identifikační údaje majitele certifikátu.

- Signature Algorithm – algoritmus použitý k vytvoření podpisu.
- Issuer – identifikační údaje vydavatele certifikátu.
- Valid-From – datum počátku platnosti certifikátu.
- Valid-To – datum konce platnosti certifikátu.
- Key-Usage – účel veřejného klíče (šifrování, ověřování podpisů nebo obojí).
- Public Key – veřejný klíč.
- Thumbprint Algorithm – algoritmus otisku certifikátu.
- Thumbprint – vlastní otisk certifikátu sloužící k ověření neporušenosti certifikátu.

Certifikační autoritou se rozumí například server, ale i fyzická osoba, která přiděluje veřejné klíče na základě prokázání identity a přiloženého veřejného klíče žadatele.

1.4.1 Infrastruktura veřejných klíčů PKI

Jádrem této infrastruktury je již zmíněná certifikační autorita (CA), které podléhají další CA, případně registrační autority. Registrační autorita má za úkol ověření údajů žadatele o certifikát. Po ověření uživatele je mu vydán certifikát CRT_{VK}. Součástí je také seznam neplatných certifikátů, které byly znehodnoceny ještě před vypršením životnosti certifikátu. V tomto seznamu si tedy lze ověřit, zda certifikáty protější strany nepozbyly platnosti.

1.5 Kvalifikace digitálních podpisových schémat

1.5.1 Digitální podpisové schéma s dodatkem

V praxi se tato schémata používají nejvíce. Využívají spíše hash funkce a jsou méně náchylné na padělání, jelikož jsou vytvořeny tak, aby bylo nemožné z hashe zprávy vytvořit zpětně zprávu a při jakékoliv změně ve zprávě se hash zprávy změní oproti původnímu. Digitální podpisy s dodatkem využívají především funkce DSA, ElGamal a Schnorr.

Generování klíčů probíhá následovně. Každá entita A si vybere soukromý klíč, který definuje S_A.

$$S_A = \{ S_A, k : k \in R \} \quad (1)$$

S_A poté definuje odpovídající mapování V_A z M_h x S na hodnotu {true, false}. V_A zde zastupuje ověřovací transformaci, kde $m \in M_h$, $s \in S$ a $m = h(m)$ pro $m \in M$. Jedná se o ověřovací transformaci, kterou lze vypočíst bez znalosti soukromého klíče.

$$V_A (m, s) = \text{true, jestliže } S_{A,k} (m) = s \quad (2)$$

Veřejným klíčem entity A je tedy V_A a soukromým je S_A.

Generování podpisu z pohledu entity A probíhá následujícím způsobem. Vybere se element $k \in R$, což je pomocná proměnná pro výpočet podpisu. Následně se vypočte m pomocí jednosměrné funkce $h(m)$ a podpis s.

$$m = h(m) \quad (3)$$

$$s = S_{A,k}(m) \quad (4)$$

Podpis entity A pro m je s. Pro možnost ověření podpisu jinými entitami je m a s. Ověření entitou B by mělo probíhat následovně.
V prvé řádě je třeba obdržet veřejný klíč V_A entity A. Poté se vypočítá $m = h(m)$ a

$$u = V_A(m, s) \quad (5)$$

Jestliže u = true, je podpis ověřen. [3]

1.5.2 Digitální podpisové schéma schopné obnovit zprávu z podpisu

Jak již název napovídá, jedná se o podpisové schéma, které má takovou funkci, kdy lze zprávu obnovit na základě znalosti samotného podpisu. V praxi se tohoto schématu využívá především k podepisování krátkých zpráv. Hlavní algoritmy pro digitální podpisy, které využívají tohoto schématu, jsou RSA, Rabin a Nyberg-Rueppel.

Jakýkoliv takto vytvořený podpis může být převeden do digitálního podpisového schématu s dodatkem vytvořením hash zprávy a podepsáním hodnotou hashe. [2]

1.6 Možné útoky na digitální podpis

Cílem každého útočníka je získat jakoukoliv informaci, která by mu pomohla prolomit zabezpečení, v případě digitálních podpisů získat soukromý klíč, nebo algoritmus generující podpis, s jehož pomocí by byl vytvořen podpis falešný. Jestliže se takovýto útok povede, může se útočník následně vydávat za pravého majitele digitálního podpisu.

- Totální prolomení (Total break)

Útok zvaný total break spočívá v tom, že je útočník schopen vypočítat privátní klíčové informace o autorovi podpisu, případně vytvoří algoritmus, který je schopen nahradit algoritmus původního podpisu.

- Selektivní padělání (Selective foregy)

Protivník je schopen pro určité zprávy nebo její část vytvořit platný podpis.

Útoky na veřejné klíče

- Útok známou zprávou (know-message attack)

Útočník zná podpisy pro soubor zpráv, které jsou mu známy, ale které si sám nevybral.

- Útok zvolenou zprávou (chosen-message attack)

Útočník v prvé řádě získá podpis z vybraného seznamu zpráv, ze kterého se pokusí prolomit podpisové schéma.

- Adaptivní útok zvolenou zprávou (adaptive chosen-message attack)

Útočník využívá podpisů ze zpráv, které jsou spojeny s veřejným klíčem vlastníka. Dále si vyžádá předchozí zprávy obsahující podpis. Kombinací těchto dvou skutečností se snaží prolomit podpisové schéma.

Z pohledu zabránění útoku je tento typ útoku největší hrozbou, protože z dostatečného množství zpráv s odpovídajícími podpisy by útočník mohl vytvořit vzor pro vytvoření svého podpisu. Ačkoliv je tento útok pravděpodobně neuskutečnitelný v praxi, je potřeba, aby podpisové schéma bylo vždy vytvořeno s ohledem na prevenci proti tomuto útoku. [3]

2 Základní metody využívané k podepisování

2.1 RSA (Rivest-Shamir-Adleman)

Kryptosystém RSA využívá problému fakorizace čísla. Jedná se o problém rozkladu daného čísla na součin mocnin prvočísel. RSA tedy využívá trapdoor funkci, o které se zmiňuje kapitola 1.3.

Princip vytvoření RSA parametrů:

V prvé řadě je třeba určit dvě prvočísla p a q. Pomocí těchto čísel se vypočte modulo n a čísla r.

$$n = p.q \tag{6}$$

$$r = (p-1).(q-1) \tag{7}$$

Dále se zvolí veřejný klíč e, který musí být nesoudělný s číslem r. Pokračuje se výpočtem soukromého klíče d.

$$d = e^{-1} \, mod \, r \tag{8}$$

Čísla e a n jsou čísla veřejné. Parametr d je soukromý klíč.

2.1.1 Šifrování a dešifrování RSA

Postup při šifrování začíná rozdělením zprávy na bloky symbolů stejné délky. Každý takový blok se zašifruje pomocí následujícího vztahu a zašifrované bloky spojí do kryptogramu C, který se odešle adresátovi.

$$c_i = z_i^e \, mod \, n \tag{9}$$

Při dešifrování se kryptogram C rozdělí na původní bloky, kdy se každý blok dešifruje pomocí vztahu

$$z_i = c_i^d \bmod n \tag{10}$$

Z bloků z_i se zpětně poskládá zpráva Z.

2.1.2 Podepsání a ověření

Podepsání zprávy *m* provede entita *A* následovně:

1. Vypočte se $m = R(m)$ z rozsahu $[0, n-1]$.

2. Dále se vypočte hodnota *s*.

$$s = m^d \bmod n \tag{11}$$

3. Podpis entity *A* je v tomto případě vypočtené *s*.

Ověření zprávy provádí entita *B*.

1. Entita *B* obdrží veřejný klíč od entity *A* (n, e).
2. Dále se určí hodnota *m* a ověří se, zda $m \in M_R$. Pokud ne, je podpis neplatný.

$$m = s^e \bmod n \tag{12}$$

2.1.3 Bezpečnost a využití

Z praktického hlediska se RSA, a obecně asymetrické kryptosystémy, využívají pro přenos krátkých zpráv, například podepisování, nebo přenos hesla při autentizaci. Bezpečnost RSA je založena především na již zmíněné obtížnosti faktorizovat velká čísla, kdy se za bezpečné pouvažují čísla *n = 2048/4096 bit*. Faktorizace tak velkého čísla je za současných podmínek prakticky nereálná.

Využití RSA při podepisování probíhá pomocí hash funkce, která se vypočítá z podepisované zprávy. Příjemce poté znovu vypočte hash a pokud je stejný, jako hash původní, je zpráva i s podpisem akceptovatelná.

2.2 DSA

Digital Signature Algorithm, tedy DSA, je standard pro digitální podpis, který byl přijat v roce 1993. Poslední úprava byla v roce 2009 ve FIPS 186-3. Vytváření klíčů v tomto algoritmu má dvě fáze. Vybrání parametrů, které jsou sdíleny mezi více uživateli, a následně vytvoření klíčů. Bezpečnost DSA je založena na problému výpočtu diskrétního logaritmu. Určení zmíněných parametrů probíhá následovně.

Nejdříve se zvolí čísla p a q, kde p je 1024 bitové a q 160 bitové. FIPS (Federální standard pro práci s informacemi) doporučuje dvojici p a q mimo zmíněné hodnoty volit také jako (2048,224)bitů, (2048,256)bitů a (3072,256)bitů. Dále se vybere N-bitové číslo g, kde délka N musí být minimálně stejně dlouhá, jako délka vstupu hash funkce. Nakonec se volí číslo h tak, aby platilo:

$$g = h^{(p-1)/q} \bmod p;\ g \neq 1 \tag{13}$$

Výsledkem jsou veřejné parametry p, q a g, se kterými strana A provede následující výpočty pro získání veřejného a soukromého klíče.

- Vybere se náhodné číslo x za podmínky $1 \leq x \leq q - 1$.
- Vypočte se hodnota y ze vztahu

$$y = g^x \bmod p \tag{14}$$

- Hodnota y je veřejný klíč a hodnota x je soukromý klíč strany A.

2.2.1 Podepsání zprávy *m* pomocí DSA

Podepsání samotné zprávy m začíná volbou náhodného čísla k za podmínky $1 \leq k \leq q - 1$ tak, aby platilo $r \neq 1$ u následujícího výpočtu.

$$X = g^k \bmod p \tag{15}$$

$$r = X \ mod \ p \tag{16}$$

Za předpokladu, že hash funkce zprávy *m je* $H(m)$, pak

$$s = k^{-1} \{(H(z) + xr)\} mod \ q \tag{17}$$

Podpis zprávy *m*, který byl vytvořen stranou *A* je tedy (*r*,*s*).

2.2.2 Ověření

V případě, že chce strana B ověřit podpis (*r*,*s*) zprávy *m*, obdrží veřejné parametry *p*,*q* a g a veřejný klíč *y*. Strana B poté začne ověřovat podpis. Nejdříve se ověří, zda jsou *r* a *s* z intervalu celých čísel [1, q - 1]. Poté je třeba vypočítat *w*, u_1 a u^2.

$$w = s^{-1} \ mod \ q \tag{18}$$

$$u_1 = H(m)w \ mod \ q \tag{19}$$

$$u_2 = rw \ mod \ q \tag{20}$$

Pomocí těchto veličin se vypočítá hodnota *X* a následně hodnota *v*, která se porovná s *r*.

$$X = g^{u_1} \ y^{u_2} \ mod \ p \tag{21}$$

$$v = X \ mod \ q \tag{22}$$

Pouze v případě, že se hodnota *v* = *r*, je podpis akceptován. V opačném případě je považován za nedůvěryhodný. [4]

2.2.3 Bezpečnost DSA

Bezpečnost DSA je založena na problému diskrétního logaritmu. Velikost veřejných parametrů je doporučena volit následovně. Zatímco parametr *q* je fixně určen FIPS 186 na 160 bitů, parametr *p* může být libovolným násobkem 64 mezi 512 a 1024 bitů včetně. Doporučená hodnota je alespoň 168 bitů. Standard FIPS 186 však neumožňuje pro *p* zvolit hodnotu větší než 1024 bitů.

Za cenu vyšších implementačních nároků lze u DSA zvýšit bezpečnost přenesením DSA nad algebru bodů rovinné eliptické křivky. Prolomení klíče o odpovídající délce je v tomto případě mnohem náročnější než u DSA. Lze tedy využít eliptických křivek ke zvýšení bezpečnosti s tím, že zůstane zachována délka veřejných parametrů a klíčů, nebo se sníží délka kritických parametrů, aniž by se měnila úroveň bezpečnosti. Použití eliptických křivek vyzývá podpisové schéma ECDSA, které je popsáno v následující podkapitole 2.3. [4]

2.3 ECDSA

Jedná se o variantu DSA, která využívá eliptických křivek pro výpočet doménových parametrů k vytvoření podpisu. Tyto parametry se skládají z vhodně zvolené eliptické křivky *E*, jenž je definována konečným polem F_q, kde můžou nastat dva případy, kdy hodnota *q*, tedy velikost pole, je rovna *p*, nebo v druhém případě 2^m.

V prvním případě nechť F_p je konečné pole, kdy *p* je liché prvočíslo a parametry $a,b \in F_p$ splňují podmínku $4a^3 + 27b^2 \neq 0 \pmod p$. Poté se eliptická křivka $E(F_p)$ přes F_p definována parametry $a,b \in F_p$ skládá ze souboru řešení nebo z bodů $P = (x, y)$ pro $x,z \in F_p$. Rovnice, definující křivku $E(F_p)$, poté vypadá takto:

$$y^2 = x^3 + ax + b \ (mod\ p) \tag{23}$$

Doménové parametry, které je potřeba vypočítat, jsou tyto $T = (p, a, b, G, n, h)$, kde *p* je prvočíslo, *a,b* jsou parametry popsané výše, *G* je základní bod, kdy $G = (x_G, y_G)$ v $E(F_p)$, parametr *n* je prvočíslo, které určuje řád parametru *G*. *H* je celé číslo a jedná se o kofaktor, jenž udává počet prvků na eliptické křivce. Úroveň zabezpečení v bitech je celočíselná hodnota $t \in \{80, 112, 128, 192, 256\}$.

Výsledné parametry umožňují použití několika známých metod pro výběr křivky, jako jsou metody založené na komplexním násobení a metoda založená na obecných algoritmech bodů počítání.

V druhém případě je F_q, kde $q = 2^m$, se jedná o dvě konečná pole, kdy $a, b \in F_q$, za podmínky

$b \neq 0$. Rovnice, určující tuto křivku, vypadá následovně:

$$y^2 + xy = x^3 + ax^2 + b \tag{24}$$

Doménové parametry, které je potřeba vypočítat, jsou tyto, $T = (m, f(x), a, b, G, n, h)$, kde parametry *a,b,G,h* jsou obdobné, jako u F_p , stejně tak parametr *t*. Pro výpočet parametrů je dále potřeba určit hodnotu parametru $m \in \{163, 233, 239, 283, 409, 571\}$ za podmínky *2t* < *m* < *2t´*, kde *t´* je nejmenší možné číslo, větší než *t* z $\{80, 112, 128, 192, 256\}$. Poté se na základě velikosti *m* se určí jeden z níže uvedených polynomů.

$$m = 163;\ f(x) = x163 + x7 + x6 + x3 + 1 \quad (25)$$

$$m = 233;\ f(x) = x233 + x74 + 1 \quad (26)$$

$$m = 239;\ f(x) = x239 + x36 + 1\ nebo\ x239 + x158 + 1 \quad (27)$$

$$m = 283; f(x) = x283 + x12 + x7 + x5 + 1 \quad (28)$$

$$m = 409; f(x) = x409 + x87 + 1 \quad (29)$$

$$m = 571; f(x) = x571 + x10 + x5 + x2 + 1 \quad (30)$$

V následujících kapitolách je popsán postup nastavení ECDSA, generování klíče, vytvoření podpisu a ověřování. [5]

2.3.1 Postup zavedení ECDSA

K tomu, aby bylo možné využít ECDSA, se musí entity A a B řídit určitým postupem. V prvé řadě entita A musí určit, kterou hash funkci bude používat. Poté entita A zavede doménové parametry $T = (m, f(x), a, b, G, n, h)$ nebo $T = (p, a, b, G, n, h)$ na požadované bezpečnostní úrovni. Nakonec entita *B* musí získat hash a parametry domény *t*, které byly určeny entitou A. Poté ověří, že jsou doménové parametry *T* platné. [5]

2.3.2 Nasazení klíče

Entita A stanoví soukromý klíč, který je tvořen dvojicí (d_u, Q_u), která je s *T* využita k podpisovému schématu. Entita *B* poté musí ověřit pravost veřejného klíče Q_u vygenerovaného entitou A. Dvojice (d_u, Q_u) je generována následovně. Vybere se pseudonáhodné číslo *d* z intervalu [*1, n - 1*]. Poté se vypočte hodnota *Q = dG*. [5]

2.3.3 Podepsání zprávy

Samotné podepsání zprávy pomocí ECDSA s využitím klíčů a parametrů, které byly popsány výše, se skládá z následujících kroků.

1. Určí se pár (k, R), kde $R = (x_R, y_R)$.
2. Převede se pole prvku x_R na celočíselnou hodnotu $\overline{x_R}$.
3. Určí se $r = \overline{x_R}$. Jestliže se $r = 0$, je potřeba vrátit se ke kroku 1.
4. Vypočte se hash funkce

$$H = Hash(M) \tag{31}$$

5. Odvodí se celočíselná hodnota e z H.
6. Vypočte se

$$s = k^{-1}(e + rd_U) \bmod n \tag{32}$$

   Pokud $s = 0$, je potřeba vrátit se ke kroku 1.
7. Výstupem je tedy $S = (r, s)$.

2.3.4 Ověření

Podepsanou zprávu je třeba ověřit, což provede entita B pomocí parametrů a klíčů získaných z předchozích procedur. Jako vstup pro ověření bude sloužit zpráva M, údajný podpis $S = (r,s)$ entity A. Postup ověření je poté následující.

1. Pokud r a s nejsou celočíselné hodnoty z intervalu [1, n − 1], ověřovací proces se zastaví a podpis je tím pádem prohlášen za neplatný.
2. Jestliže se proces ověření nezastaví v prvním kroku, vypočte se hash funkce H.

$$H = Hash(M) \tag{33}$$

3. Odvodí se e z H:
   a. Převede se H na bitový řetězec $\overline{H}$.

b. Určí se $\bar{E} = \bar{H}$, jestliže $[log_2 n] \geq 8$

c. $\underline{E}$ se převede na celočíselnou hodnotu *e*.

4. Vypočtou se hodnoty u_1 a u_2, ze kterých se následně vypočte *R*:

$$u_1 = es^{-1} mod\, n \quad (34)$$

$$u_2 = rs^{-1} mod\, n \quad (35)$$

$$R = (x_E, y_R) = u_1 G + u_2 Q_U \quad (36)$$

Jestliže se $R = 0$, je podpis neplatný.

5. Dále se převede x_R na celočíselnou hodnotu $\overline{x_R}$, pomocí které se vypočítá *v*.

$$v = x_R\, mod\, n \quad (37)$$

Pokud $v = r$, je podpis platný. V opačném případě ne. [5]

2.4 Krátké skupinové podpisy navržené trojicí Boneh – Boyen – Shacham (BBS)

Toto podpisové schéma je založeno na metodě Strong Diffie-Hellman (SDH) a bilineárních skupinách. Skupinové podpisy zajišťují anonymitu jednotlivým uživatelům ve skupině. Každý člen této skupiny může zprávu podepsat, ale výsledný podpis udržuje identitu jeho autora v tajnosti. V tomto systému skupinového podepisování existuje třetí strana, která může podpisy sledovat a kontrolovat, případně zakázat členství ve skupině. Jak již bylo zmíněno, jsou krátké skupinové podpisy založeny SDH. Výsledky testování metody dokazují, že SDH je jednoduší a kratší než Strong – RSA. Systém využívá tzv. Zero-Knowledge Proof of Knowledge (ZKPK), což je v podstatě protokol, který jedné straně dokazuje, že matematické tvrzení je pravdivé, aniž by odhalil cokoliv jiného. Případný útočník tedy zjistí pouze informaci o pravdivosti.[6] Skupinové podpisy musí splňovat tyto požadavky [7]:

:

- Spolehlivost a úplnost – platné podpisy musí být vždy správně ověřeny a naopak u neplatných musí dojít k vyhodnocení podpisu jako neplatného.
- Nemožnost podpis padělat – skupinové podpisy mohou vytvářet pouze členové skupiny a skupinový manažer.
- Anonymita – totožnost autora je možné určit pouze pomocí soukromého klíče, který vlastní třetí strana (manažer skupiny).
- Schopnost dohledat autora – manažer skupiny by měl vždycky být schopen vyhledat autora podpisu.
- Nemožnost určit shodu – nelze zjistit, zda podpisy u dvou odlišných zpráv patří jednomu autorovi.
- Falšování podpisu – nelze vytvořit podpis pro uživatele, který není účastníkem skupiny.

2.4.1 Generování, podpis, ověření

Před popisem samotného generování klíče je třeba zmínit notaci a definici skupin. Nechť G_1 a G_2 jsou cyklické skupiny z prvočíselného řádu p s generátorem g_1 a g_2. Dále ať ψ je izomorfizmus z G_1 a G_2 a e je bilineární mapa z G_1 a G_2 promítnutá do G_r.[9]

Generování klíče

- Vybere se náhodné číslo $g_2 \in G_2$ a určí se $g_1 \leftarrow \psi(g_2)$
- Zvolí se hodnoty r_1 *a* r_2 a hodnoty u a v tak, aby platil níže uvedený vztah, kde h je hodnota hash funkce.

$$u^{r_1} = u^{r_1} = h \tag{38}$$

- Zvolí se hodnota γ, pomocí které se vypočte ω

$$\omega = g_2^{\gamma} \tag{39}$$

- Poté se pro všechna i od 1 do n zvolí číslo x_i a vypočte se hodnota f_i.

$$f_i = g_1^{\frac{1}{\gamma + x_i}} \tag{40}$$

Veřejný klíč je poté *gpk* = (g_1; g_2; *h; u; v; w*), klíč vedoucího skupiny je *gmsk* = (r_1, r_2) a soukromý klíč i-tého uživatele je *gks*[i] = (f_i, x_i). [8]

Generování podpisu

Pro podepsání zprávy $M \in \{0,1\}^*$ podpisem σ se postupuje následovně.

- Zvolí se hodnoty $\alpha, \beta, r_\alpha, r_\beta, r_{\gamma_1}, r_{\gamma_2}$ a vypočtou se následující rovnice:

$$T_1 = u^\alpha;\ T_2 = v^\beta;\ T_3 = f * h^{\alpha+\beta} \quad (41), (42), (43)$$

$$\gamma_1 = x * \alpha;\ \gamma_2 = x * \beta \quad (44), (45)$$

$$R_1 = u^{r_\alpha};\ R_2 = v^{r_\beta} \quad (46), (47)$$

$$R_3 = e(T_3, g_2)^{r_x} * e(h, \omega)^{-r_\alpha - r_\beta} * e(h, g_2)^{-r_{\gamma_1} - r_{\gamma_2}} \quad (48)$$

$$R_4 = T_1^{r_x} * u^{-r_{\gamma_1}};\ R_5 = T_2^{r_x} * v^{-r_{\gamma_2}} \quad (49), (50)$$

$$c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \quad (51)$$

$$s_\alpha = r_\alpha + c * \alpha;\ s_\beta = r_\beta + c * \beta \quad (52), (53)$$

$$s_{\gamma_1} = r_{\gamma_1} + c * \gamma_1;\ s_{\gamma_2} = r_{\gamma_2} + c * \gamma_2 \quad (54), (55)$$

- Podpis je $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_{\gamma_1}, s_{\gamma_2})$ (56)

Ověření

Při ověření obdrží entita *B* klíč *gpk* = (g_1; g_2; *h; u; v; w*), podpis $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_{\gamma_1}, s_{\gamma_2})$ a zprávu M. Vypočítají se znovu hodnoty R_1, R_2, R_3, R_4, R_5.

$$R_1 = u^{s_\alpha} * T_1^{-c} \quad (57)$$

$$R_2 = v^{s_\beta} * T_2^{-c} \quad (58)$$

$$R_3 = e(T_3, g_2)^{s_x} * e(h, \omega)^{-s_\alpha - s_\beta} * e(h, g_2)^{-s_{\gamma_1} - s_{\gamma_2}} * (e(T_3, \omega) * e(g_1, g_2)^{-1})^c \quad (59)$$

$$R_4 = T_1^{s_x} * u^{-s_{\gamma_1}};\ R_5 = T_2^{s_x} * v^{-s_{\gamma_2}} \quad (60)$$

Pokud se $c = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5)$, je podpis platný. [8]

2.4.2 Výkonost a délka podpisu BBS

BBS dávkové ověřování dokáže ověřit spoustu podpisů vytvořených mnoha členy skupin na různé zprávy. Základní BBS podpis obsahuje tři prvky z G_1 a šest prvků z $\mathbb{Z}_q$. Naproti tomu modifikovaná verze pro výpočet dávkového ověření obsahuje tři prvky z G_1 a šest prvků z $\mathbb{Z}_q$ a jeden prvek z G_T. Po implementování 170 bitové eliptické křivky MNT je velikost přibližně 2553 bitů, kdy je bezpečnost srovnatelní s 1024 bity metody RSA. Pro aplikace, kde je prioritou šířka pásma je vhodný extrémně krátký podpis BS (Boneh and Shacham), který obsahuje pouze dvě operace pro párování. Tato metoda je popsána v kapitole 2.6.[8]

2.5 Podpisové schéma BLS - Boneh–Lynn–Shacham

Podpisové schéma o délce 160 bitů, které využívá pro svou bezpečnost problém CDH (Computational Diffie-Hellman). Z bezpečnostního hlediska je srovnatelné s metodou DSA, která má délku podpisu 320 bitů. Generování podpisu je násobení na křivce a ověření se provádí pomocí bilineárního párování. [16]

2.5.1 Generování klíče

Vybere se náhodné číslo $x \in Z_p^*$ a vypočte se hodnota $x = g_2^x$. Dále nechť $H : \{0,1\}^* \longrightarrow G_1$ je funkce. Veřejný klíč je poté (g_2, u) a soukromý je x.

2.5.2 Generování podpisu

Pro podepsání zprávy $m \in Z_P^*$ je potřeba vypočítat hodnotu σ, která je podpisem zprávy m.

$$\sigma = H(m)^x \tag{61}$$

2.5.3 Ověření

Při ověření entita B obdrží veřejný klíč (g_2, u). Pokud platí níže uvedený vzorec, je podpis platný.

$$e(\sigma, g_2) = e(H(m), u) \tag{62}$$

2.6 Podpisové schéma využívající metodu Verifier-Local Revocation

Implementace metody využívající lokální ověřování je realizována tím, že podpisový ověřovací algoritmus zvaný Revocation List (RL) bude jako dodatečný argument, který obsahuje token pro každého, kdo je ze skupiny odvolán. Ověřovací algoritmus akceptuje všechny podpisy vydány neodvolanými členy skupiny a nevykazuje žádné informace o tom, který odvolaný uživatel vydal podpis. Jestliže byl uživatel odvolán (jeho token byl vložen do RL), podpisy jím vydané již nejsou akceptované. Z toho vyplývá, že podpisy vydané revokovaným uživatelem jsou navzájem korelační. Podpisové schéma využívající VLR metodu obsahuje jako ostatní schémata tři základná algoritmy a to generování klíče, podpis a ověření. [16]

2.6.1 Podpisová schéma BS

V této kapitole je zmíněno využití metody VLR na dané podpisové schéma, navržené dvojící Hovav Shacham a Dan Boneh, kteří stojí také za vznikem zmíněné VLR. Samotná struktura BS metody je následující.

Generování klíče:
Algoritmus pro generování klíče využívá na vstupu hodnotu *n*, což je počet klíčů ke generování.

- Zvolí se hodnota generátoru g_2 v G_2 .A určí se hodnota $g_1 \leftarrow \psi(g_2)$. V případě, že platí rovnice č. 62, musí se postup opakovat.

$$e(\psi(g_2), g_2) = 1 \tag{63}$$

- Dále se určí hodnoty $\gamma \xleftarrow{R} \mathbb{Z}_p^*$ a vypočte ω:

$$\omega = g_2^{\gamma} \tag{64}$$

- S pomocí γ se vygeneruje pro každého uživatele SDH (A_i,x_i) výběrem $x_i \xleftarrow{R} Z_p$ tak, aby byla zachována podmínka z rovnice č. 64 a určí se $A_i \leftarrow g_i^{1/(\gamma+x_i)}$.

$$\gamma + x_i \neq 0 \tag{65}$$

Z výše uvedených výpočtů se určí veřejný klíč $gpk = (g_1, g_2, \omega)$. Pro každého uživatele se vygeneruje soukromý klíč $gsk = (A_i, x_i)$. Odvolávací token *grt* je levá část soukromého klíče, tedy hodnota A_i. Výstupem jsou tedy hodnoty *gpk*, *gsk* a *grt*.

Generování podpisu:

Jako vstup pro generování podpisu se využívá hodnota soukromého a veřejného klíče (*gsk*, *gpk)* a zpráva $M \in \{0,1\}^*$. Postup generování podpisu je následovný:

- Zvolí se náhodná hodnota $r \overset{R}{\leftarrow} \mathbb{Z}_p$ a určí se hodnoty generátorů $(\hat{u}, \hat{v})$ v G_2 z H_0.

$$(\hat{u}, \hat{v}) \leftarrow H_0(gpk, M, r) \in G_2^2 \tag{66}$$

  a vypočítají se jejich obrazy v G_1, tedy $u \leftarrow \psi(\hat{u})$, $v \leftarrow \psi(\hat{v})$.

- Určí se exponent $\alpha \overset{R}{\leftarrow} \mathbb{Z}_p$ a vypočítají hodnoty T_1 a T_2

$$T_1 \leftarrow u^{\alpha}, T_2 \leftarrow A_i v^{\alpha} \tag{67}$$

- Určí se hodnota $\delta \leftarrow x_i \alpha \in Z_p$ a dále hodnoty r_α, r_x a $r_\delta \overset{R}{\leftarrow} \mathbb{Z}_p$
- Vypočítají se pomocné proměnné R_1, R_2 a R_3

$$R_1 \leftarrow u^{r_\alpha} \tag{68}$$

$$R_2 \leftarrow e(T_2, g_2)^{r_x} * e(v, \omega)^{-r_\alpha} * e(v, g_2)^{-r_\delta} \tag{69}$$

$$R_3 \leftarrow T_1^{r_x} * u^{-r_\delta} \tag{70}$$

- Dále je potřeba vypočítat hodnotu $c \in Z_p$ s pomocí H a hodnoty s_α, s_x, s_δ.

$$c \leftarrow H(gpk, M, r, T_1, T_2, R_1, R_2, R_3) \in \mathbb{Z}_p \tag{71}$$

$$s_\alpha = r_\alpha + c\alpha \tag{72}$$

$$s_x = r_x + cx_i \tag{73}$$

$$s_\delta = r_\delta + c \in \mathbb{Z}_p \tag{74}$$

- Výsledný podpis je poté: $\sigma \leftarrow (r, T_1, T_2, c, s_\alpha, s_x, s_\delta)$

Ověření

Ověřovací algoritmus bere na vstup veřejný klíč $gpk = (g_1, g_2, \omega)$, set *rl* nesoucí odvolávací tokeny, podpis σ a zprávu $M \in \{0,1\}^*$. Ověření v prvé řadě kontroluje, zda je podpis pravý, poté zda nebyl vygenerovaný odvolaným uživatelem. Zprávu akceptuje pouze v případě, že projdou ověřením obě dvě pravidla.

- Pro ověření podpisu se nejdříve vypočítá hodnota $\hat{u}, \hat{v}$ stejným způsobem jako při generování podpisu (rovnice č. 66) a jejich obrazy u a *v*.

$$u \leftarrow \psi(\hat{u}), v \leftarrow \psi(\hat{v}) \tag{75}$$

- Poté se znova určí hodnoty R_1, R_2 a R_3

$$\tilde{R}_1 \leftarrow u^{s_\alpha} / T_1^c \tag{76}$$

$$\tilde{R}_2 \leftarrow e(T_2, g_2)^{s_x} * e(v, \omega)^{-s_\alpha} * e(v, g_2)^{-s_\delta} * \left(\frac{e(T_2, \omega)}{e(g_1, g_2)}\right)^c \tag{77}$$

$$\tilde{R}_3 \leftarrow T_1^{s_x} * u^{-s_\delta} \tag{78}$$

- Nakonec se zkontroluje, zda podpis c je korektní

$$c \stackrel{?}{\leftarrow} H(gpk, M, r, T_1, T_2, \tilde{R}_1, \tilde{R}_2, \tilde{R}_3) \tag{79}$$

Jestliže podpis souhlasí, je akceptován. Pokud ne je odmítnut.

- Po kontrole podpisu následuje kontrola, zda nedošlo k podepsáním uživatelem odvolaným ze skupiny. To se provádí pro každý element $A \in rl$. Ověřuje se každé *A* zakódované v (T_1, T_2).

$$e\left(\frac{T_2}{A}, \hat{u}\right) = e(T_1, \hat{v}) \tag{80}$$

Pokud jsou oba algoritmy správně, jak pro podpis, tak pro odvolané uživatele, je vše přijato. V případě, že jeden z algoritmů neprojde kontrolou, je zpráva zamítnuta.

2.6.2 Délka podpisu a výkonost

Skupinový podpis popsaný výše se skládá ze dvou prvků z G_1 a pěti prvků z $\mathbb{Z}_p$. Využitím různých eliptických křivek lze dosáhnout snížení celkové délky podpisu. Například využitím MNT křivek je délka podpisu 1192 bitů a bezpečnost srovnatelná se standardem RSA podpisu o velikosti 1024 bitů. Jestliže se využije křivek navržených Barreto Naehrigem, sníží se délka podpisu na 1122 bitů se zachováním bezpečnosti.
Generování podpisu vyžaduje dvě výpočetní operace pro určení isomorfismu ψ. Výpočet isomorfismu trvá zhruba stejnou dobu jako umocňování v G_1. Takže generování podpisu potřebuje 8 operací pro umocňování a 2 výpočty pro výpočet bilineárního párování. Ověření podpisu potřebuje 6 umocňování a 3+2|rl| výpočty bilineárního párování.[16]

2.7 Silný substituční útok

S pomocí silného substitučního útoku je útočník na základě znalosti podpisu σ schopen vyprodukovat veřejný klíč, který se liší od pravého veřejného klíče a to bez znalosti klíče soukromého. Jestliže nový veřejný klíč vyhovuje všem pravým podpisům, které byly vytvořený ověřenými uživateli, nikoliv útočníky, poté se tento druh útoku nazývá univerzální substituční útok. V ostatních případech se jedná pouze o lokální substituční útok. Tedy v případech, kdy je nový veřejný klíč, vytvořený útočníkem, vyhovující pouze jednomu pravému soukromému podpisu. Tento útok může být proveden na podpisová schémata BBS, BLS. [9]

3 Optimalizační techniky

Optimalizační techniky pro hromadné digitální podepisování vznikají například z důvodu VANET sítí, kde je potřeby urychlit čas podepisování z důvodu přijímání velkého množství dat v krátké době. Tyto techniky jsou popsány v kapitolách 3.1 a 3.2.

3.1 Dávkové ověřování

Algoritmus pro dávkové ověřování digitálních podpisů neověřuje každý podpis zvlášť, ale po skupinách. Tím se docílí ušetření času, při ověřování zpráv, jelikož dávkové ověřování je mnohem rychlejší, než ověřování každého podpisu zvlášť, jak je ověřeno v poslední kapitole na praktických ukázkách.

Dávkového ověřování je vhodné použít například ve VANET sítích. Díky vývoji techniky kupředu se zvyšují také požadavky na komunikaci zařízení, ať už z oblasti automobilového průmyslu, nebo v oblasti výpočetní techniky. Co se týče automobilového průmyslu, čím dál více se zakládá na bezpečnosti a plynulosti provozu. Z toho plynou nároky na zvýšení inteligence vozidel, především komunikace mezi ostatními vozidly. V podstatě se jedná o to, aby vozidla přenášela své zprávy každých 100-300ms ke všem ostatním vozidlům [10]. Aby tyto systémy pracovaly správně, musí se vypořádat s řadou omezujících faktorů. Mezi ně patří omezení přidělení spektra pro různé typy komunikace, které je potřeba přidělit, dále pokud možno co nejkratší a zároveň co nejmenší bezpečnostní riziko. Nemalé omezení rovněž spočívá v tom, že je třeba zaručit důvěryhodnost zpráv a případně mít možnost dohledat subjekty, které šíří zprávy falešné. A třetí omezení se týká toho, že různé zprávy od mnoha různých podpisovatelů je potřeba ověřit a zpracovat rychle.

V následujících řádcích bude popsána vhodnost digitálních podpisů pro tyto systémy. Pokud se jedná o generování jednoho podpisu, není důvod, aby někde vznikl problém. Ten by mohl nastat při podepsání 100 a více podpisů. RSA podepisování je sice rychlé, ale podpis může být o velikosti až 300bitů, v čemž není započítána velikost certifikátu, a pro některé aplikace by tato velikost mohla být nepřípustná. Na druhou stranu podpisová schémata založená na bilineárních mapách využívají sice menších podpisů se stejnou úrovní zabezpečení, ale na druhou stranu potřebují delší dobu pro ověření. Trojice Landsiedel, Wehrle a Götz [10] dokázala, že je někdy vhodnější udělat více drobných výpočtů, než odesílat dlouhé zprávy. Dávkové ověřování tedy spočívá v tom, že ověřuje seznam n (zpráv a podpisů) párovaných jako skupina. Výstup 1, jestliže všechny n podpisy jsou pravé a výstup 0, jestliže jeden nebo více podpisů je nepravých. Algoritmy pro dávkové ověření mohou výrazně zvýšit efektivitu a to tím způsobem, kdy je ověřování n podpisů najednou rychlejší, než ověření všech n podpisů zvlášť. [10]

3.1.1 Definice dávkového ověřování

Algoritmus (generování, podepisování a ověřování) stanovuje podpisové schéma. Generování podpisu bere jako vstup bezpečnostní parametr k a výstupem poté je podepsání a ověření pomocí párů klíčů (sk, pk). Podpis $S(sk, m)$ je poté podpis zprávy m, která je podepsána soukromým klíčem sk a ověření $V(P, \sigma, m) = 1$, pokud σ je platný podpis m podepsán soukromým klíčem sk korespondujícím s veřejným klíčem pk. V ostatních případech je $V = 0$, tedy podpis je neplatný.

Nechť $P_1 ... P_n$ je n podepisovatelů s odpovídajícím párem klíčů K $\{(sk_1, pk_1), \ldots, (sk_n, pk_n)\}$, výstup $Gen(k)$ pro parametr zabezpečení k. Nechť B je seznam obsahující K a *n-tice* prvků (P_{ti}, σ_i, m_i). Skupina(B) je ověřena, pokud pro Skupinu$(B) = 1$ platí, že $(P_{ti}, \sigma_i, m_i) = 1$, a to pro všechna i.

Předpokládejme, že $\sigma_1 \ldots \sigma_n$ jsou podpisy ze zpráv $m_1 \ldots m_n$, kterým odpovídají ověřovací klíče $pk_1 \ldots pk_n$. Agregační algoritmus je veřejný algoritmus, který dává P_i, σ_i a m_i $(i = 1,\ldots,n)$ výstupy komprimovaného podpisu σ. Ověřovací algoritmus poté ověřuje, zda je komprimovaný podpis σ platný nebo ne s ohledem na pk_i a m_i, kdy $i = 1, \ldots n$. [11]

3.1.2 Vývoj dávkového ověřování

Skupinovou kryptografii poprvé uvedl Amos Fiat kvůli zvýšení efektivity RSA schématu, kde je mnoho operací prováděno na jednom místě. Během doby bylo navrženo několik metod pro skupinové ověřování, které byly vždy prolomeny, nebo se ukázal jiný druh závady. Jedním z takových bylo podpisové schéma navržené Al- Ibrahimem, které bylo prolomeno D.R. Stinsonem. [11] I přesto vzniklo několik podpisových schémat, které využívají skupinového podepisování. Mezi ně patří podpisové schémata RSA a DSA, které využívá test pomocí malých exponentů, případně podpisová schémata BLS, CL, IBS. [11]

3.2 Agregační podpisy

Agregace podpisů spočívá v tom, aby n podpisů aplikovaných na n odlišných zpráv od n uživatelů bylo možné spojit do jednoho podpisu. Agregace jako taková tedy znamená spojení více prvků do jednoho.

3.2.1 Obecné agregační podpisy

Jedná se o schéma, kde každý uživatel i podepíše svou zprávu $M_{i,,}$ aby obsahovala podpis σ_i. Poté je možné vzít všech n podpisů a aplikovat na ně agregační algoritmus, díky kterému se všechny podpisy $\sigma_1, \ldots, \sigma_n$ zkomprimují do jednoho podpisu σ. Mimo to lze provést také postupnou agregaci, kdy je podpis σ_1 sloučen s podpisem σ_2, čímž vnikne podpis σ_{12}. Ten lze sloučit s podpisem $\sigma_{3,}$ čímž vznikne podpis σ_{123} atd. V tomto schématu si každý uživatel vytváří podpis sám. Podpisy jsou následně shrnuty do jednoho podpisu třetí stranou, která nemusí být nutně ze strany uživatelů. Každé agregační podpisové schéma vychází z klasických podpisových schémat a délka podpisu je stejná jako u klasických. Agregační algoritmus vytváří výsledný podpis z podpisů $\sigma_1, \ldots, \sigma_n$, které patří zprávám $M_1, \ldots, M_n$ a

příslušným veřejným klíčům PK_1,…, PK_n. Výsledkem celého procesu je podpis σ. Ověření probíhá na základě přijatého podpisu σ, u kterého se ověřuje, že je platný vůči zprávám M_1,…, M_n a příslušným veřejným klíčům PK_1,…, PK_n. [12]

Bilineární agregační podpisy

Toto podpisové schéma je založeno na BLS, ale oproti němu vyžaduje skupinu G, jako bilineární skupinu. Stejně jako u schématu BLS, i zde může být každý řetězec podepsán. Schéma umožňuje využít obecné agregační schéma, kde agregační entita, která nesouvisí se skupinou uživatelů, vytvářející podpisy, může do výsledného podpisu zahrnout i již existující podpisy. Schéma obsahuje tři typické algoritmy pro generování a ověření podpisu a dva další pro agregaci.

- **Generování klíče**

Pro konkrétní uživatele se zvolí $x \xleftarrow{R} \mathbb{Z}_p$ a dále se vypočte $v \leftarrow g^x$. Veřejný klíč je zde $v \in G$ a soukromý klíč je $x \in \mathbb{Z}_p$.

- **Generování podpisu**

Každý uživatel pomocí soukromého a veřejného klíče a zprávy $M \in \{0,1\}^*$. Vypočte se hash funkce $h \leftarrow H(v, M)$, kde $h \in G$. Nakonec se vypočte podpis $\sigma \leftarrow h^x$.

- **Ověření**

Ověření probíhá na základě veřejného klíče v, zprávy M a podpisu σ. Vypočte se hodnota $h \leftarrow H(v, M)$, jestliže platí níže uvedený vztah, je podpis platný.

$$e(\sigma, g) = e(h, v) \tag{81}$$

- **Agregace**

Každému uživateli, jehož podpis má být součástí agregace, se přiřadí index i z rozsahu od 1 do n. Každý takový uživatel poskytne svůj podpis $\sigma_i \in G$ a zprávu $M_i \in \{0,1\}^*$. Výsledný podpis se poté vypočte následovně.

$$\sigma \leftarrow \prod_{i=1}^{n} \sigma_i \tag{82}$$

- **Ověření agregace**

Pro ověření je třeba spočítat $h_i \leftarrow H(v_i M_i)$ pro $1 \leq i \leq n$. Pokud platí

$$e(\text{ɞ}, g) = \prod_{i=1}^{n} e(v_i M_i), \tag{83}$$

je agregační podpis platný.

3.2.2 Sekvenční agregační podpisy

Jedná se o variantu agregačních podpisů, ve kterém nejsou podpisy jednotlivě generovány a sloučeny do jednoho, ale podepisující entita převede sekvenční souhrn do jiného, který obsahuje podpis zprávy, dle jeho výběru. Podpisy v tomto schématu jsou ve vrstvách, kdy první podpis je v jádru celého schématu. Stejně jako u obecných agregačních podpisů, i sekvenční mají většinou délku podpisů stejnou, jako je délka obyčejného podpisu. Pro toto schéma probíhá agregace a podepisování v jedné kombinované operaci. Na vstupu je soukromý klíč *SK*, zpráva M_i k podepsání a sekvenční agregační podpis σ' zpráv $M_1,\ldots, M_{i-1}$,kde M_1 je nejvnitřnější zpráva. Výstupem je sekvenční agregační podpis σ pro všechny zprávy. [12]

Agregace s RSA

Provést sekvenční agregaci u metody RSA je problém, protože dva uživatelé nemohou sdílet stejné modulo *N*. Jsou však dvě metody, které toto umožňují, ale vznikají zde určitá omezení. U první metody dochází k omezení při volbě podpisového klíče. Druhá metoda způsobí zvětšení velikosti o jeden bit na každý podpis. [12]

4 Využití digitálních podpisů v komunikačních systémech

4.1 Bezdrátové senzorové sítě a ECDSA

Bezdrátové senzorové sítě jsou používány v mnoha aplikacích náchylných na bezpečnost. Může se jednat například o vojenské, případně lékařské aplikace, jako je sledování stavu pacientů. WSN (Wireless Sensor Networks) lze chápat jako skupinu uživatelů, u kterých se jeví symetrická kryptografie jako dostačující. Nicméně se jedná většinou o systémy, které jsou nasazovány bez obsluhy. V tomto případě může útočník snímače nějakým způsobem napadnout, případně ohrozit komunikaci celé sítě. Tento problém se dá vyřešit využitím

některých z digitálních podpisů. V WSN se využívá primárně metoda ECDSA, protože má malou velikost podpisu na vysoké úrovni zabezpečení. Časová náročnost ECDSA je však relativně velká, protože pro generování podpisu je zapotřebí čas mezi 0,81-2,16s a pro samotné ověření podpisu až 4,32s. Z tohoto důvodu bylo pro WSN vytvořeno nové podpisové schéma zvané Hash-Based. Podpis vzniklý touto metodou má stejnou velikost, jako u metody ECDSA, a je také na stejné bezpečnostní úrovni. Výhodou je hlavně čas potřebný ke generování a ověření podpisu. Například podpis 8bitové zprávy je 7krát rychlejší než u metody ECDSA a ověření až 158krát rychlejší. Tyto časové údaje byly naměřeny na mikrokontroleru Atmel ATMega128, kdy snímač generoval maximálně 2^{10} podpisů s jedním párem klíčů. [13]

4.2 Využití RSA v cloudovém řešení pro forensic IT

Výhoda využití cloudového řešeni s digitálními podpisy pro forensic IT spočívá v tom, že se digitální důkazy mohou proměnit z hmotných důkazů, uskladněných někde v budově, na digitální důkazy v cloudových uložištích. Zde je ovšem potřeba zajistit, aby nebyla data zasílaná pouhým otevřeným textem, který by nebyl šifrován. Hrozilo by odposlouchávání komunikace útočníkem a případné zneužití důkazů. Důkazy by tedy měly být zasílány v zašifrované podobě, což klade zvýšení výpočetních nároků na mobilní zařízení.

Cloudové datové centrum je rozděleno do dvou služeb. Služba pro výpočty a služba pro uložení dat. Předpokládejme, že je potřeba uložit data do digitální evidence forenzního datového centra, které běží v cloudovém řešení. Data jsou zaslána nejdříve do výpočetního centra, kde jsou digitálně podepsána metodou RSA, aby byla zajištěna jejich autentičnost a integrita a až poté do datového uložiště. [14]

4.3 Využití dávkového ověření v síti VANET

Ve VANET sítích, jsou kladeny velké nároky na čas, po který jsou data ověřovány. Jelikož každé vozidlo ve VANET síti musí posílat zprávy každých 100-300ms a je-li takových vozidel v sítí 100, je potřeba ověřovat stovky zpráv za sekundu. Stanice pro DSRC (Dedicated Short Range Communication) obdrží sice jen jeden podpis, ale zbylé podpisy si musí udržet ve vyrovnávací paměti. Ověřování takto příchozích zpráv je značně nepraktické a časově náročné, což by mohlo způsobit problémy v TPD (Tamper Proof Devices), zařízení ve vozidle, které se stará o zpracování, podepisování a ověřování zpráv. Proto byly pro tyto účely navrženy výše popsané metody pro dávkové ověřování a agregační podpisová

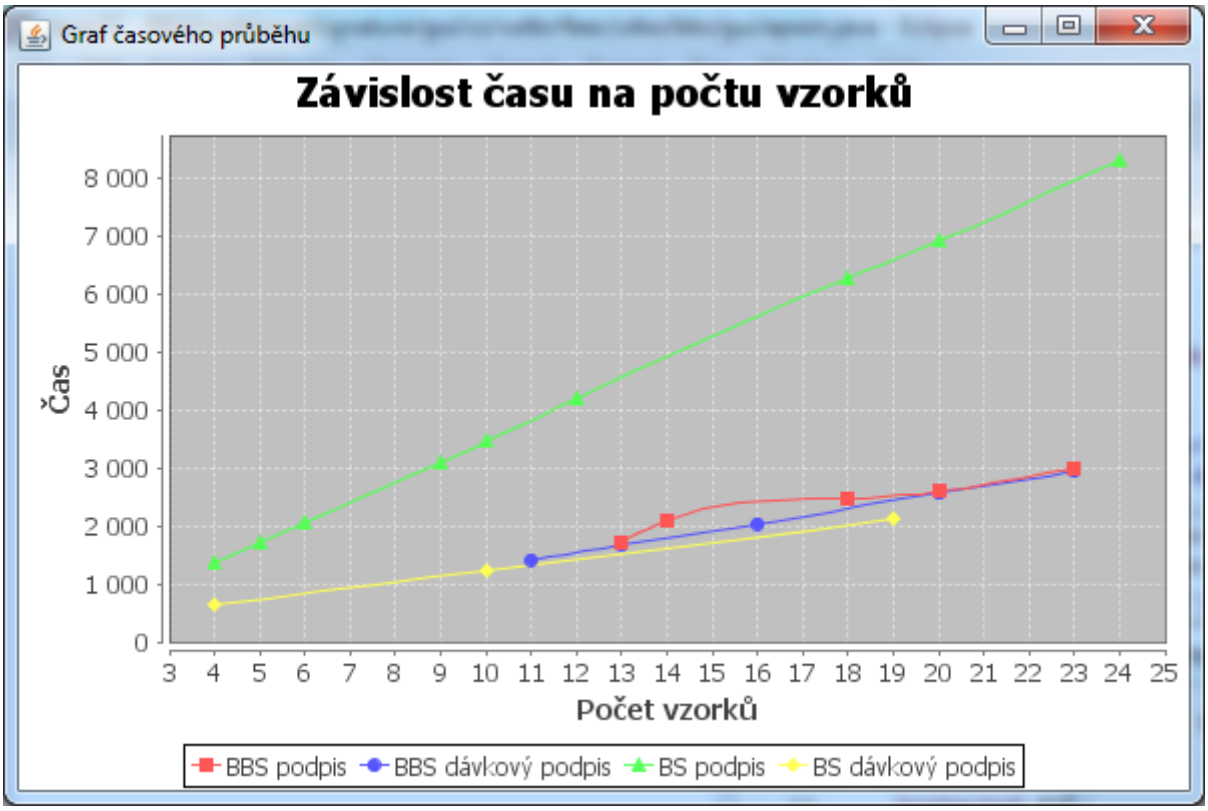
schémata. Mechanizmus skupinového podepisování a ověřování sice řeší problém pro velký počet podpisů, ale bylo potřeba jej ještě dále optimalizovat. Zoptimalizování metody je podpisové schéma IBGS (Identity based group signature) [15] navržené pro prostředí VANET. Jedná se o schéma, které obsahuje skupinu manažerů, členy skupiny, ověřovací autoritu a poskytuje plnou sledovanost a anonymitu, což vyžaduje právě síť VANET. Vlastnosti IBGS umožňují vozidlu přenášet a přijímat zprávy, aniž by byla narušena jejich anonymita. Po splnění všech bezpečnostních požadavků se prokáže, že je skupinový podpis korektní, anonymní a sledovatelný. Jestliže je vše v pořádku, zprávy vygenerované vozidlem budou vždy přijaty jinými vozidly a mohou vést k potenciálnímu zlepšení provozu. [15]

5 Program pro porovnání podpisových schémat BBS a BS

V rámci diplomové práce byl vytvořen program, který slouží k porovnání dvou podpisových schémat a to BBS (Boneh – Boyen – Shacham) a BS (Boneh –Shacham). Program byl navržen tak, aby zobrazoval časové rozdíly podepsání a ověřování zpráv. v závislosti na počtu zpráv, které jsou určeny k podepsání či ověření. Jsou zde zahrnuty následující metody:

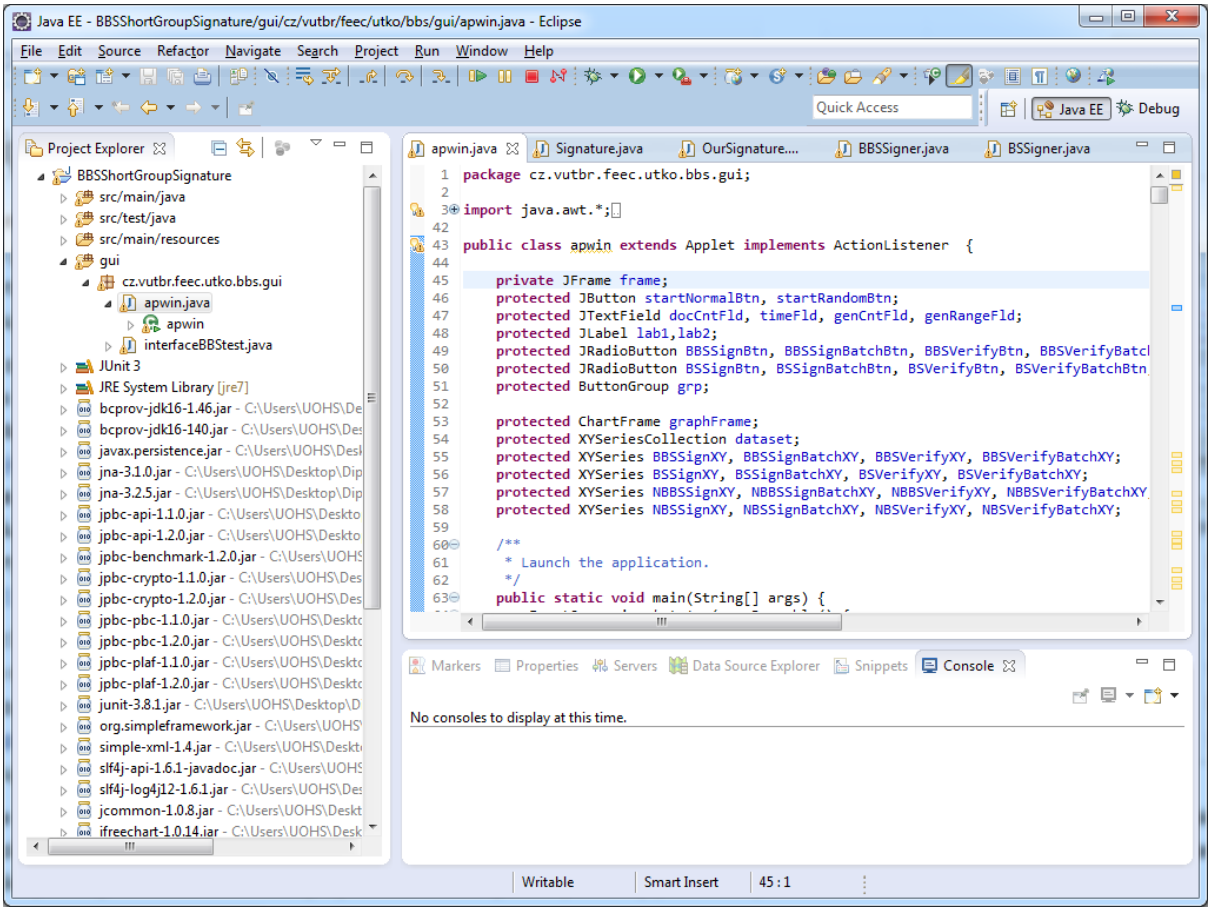
- BBS/BS podpis – podepisují každou zprávu zvlášť
- BBS/BS dávkový podpis – podepisují zprávy po dávkách
- BBS/BS ověření – ověřují každou zprávu zvlášť
- BBS/BS dávkové ověření – ověřují zprávy po dávkách

Výstupem programu je graf, který určuje závislost počtu zpráv na čase a lze z něj tedy porovnat, která z výše uvedených metod je vhodnější. Náhled grafu je zobrazen na následujícím obrázku.



Obr.č.1: Náhled výstupu programu.

Program byl psán v prostředí Eclipse, což je vývojová prostředí určené k programování v programovacím jazyce Java. Pro funkčnost navrženého programu je zapotřebí importovat řadu knihoven, které jsou součástí přílohy na CD nosiči. Mezi ty důležité patří například knihovna iPBC, která zajišťuje subalgoritmy pro výpočet bilineárního párování. Další z použitých knihoven je JFreeChart, sloužící pro generování grafů. Náhled programu Eclipse zobrazuje obr.č.2.



Obr.č.2: Náhled vývojového prostředí Eclipse

5.1 Implementace kryptografických operací

První část kódu zobrazuje některé implementované operace, které program využívá. Jedná se o implementaci části výpočtů pro podpis metody BBS, konkrétně generování parametrů $\alpha, \beta, r_\alpha, r_\beta, r_x, r_{\gamma_1}, r_{\gamma_2}$ a poté rovnic č. 41-47 z kapitoly 2.4.1.

```
public boolean preSign(CipherParameters userPrivateKey){

        this.upk = (BBSUserPrivateKey) userPrivateKey;

        // generování náhodných parametrů
        this.alpha = zp.newRandomElement().getImmutable();
        this.beta = zp.newRandomElement().getImmutable();
        this.ralpha = zp.newRandomElement().getImmutable();
        this.rbeta = zp.newRandomElement().getImmutable();
        this.rx = zp.newRandomElement().getImmutable();
        this.rdelta1 = zp.newRandomElement().getImmutable();
        this.rdelta2 = zp.newRandomElement().getImmutable();
```

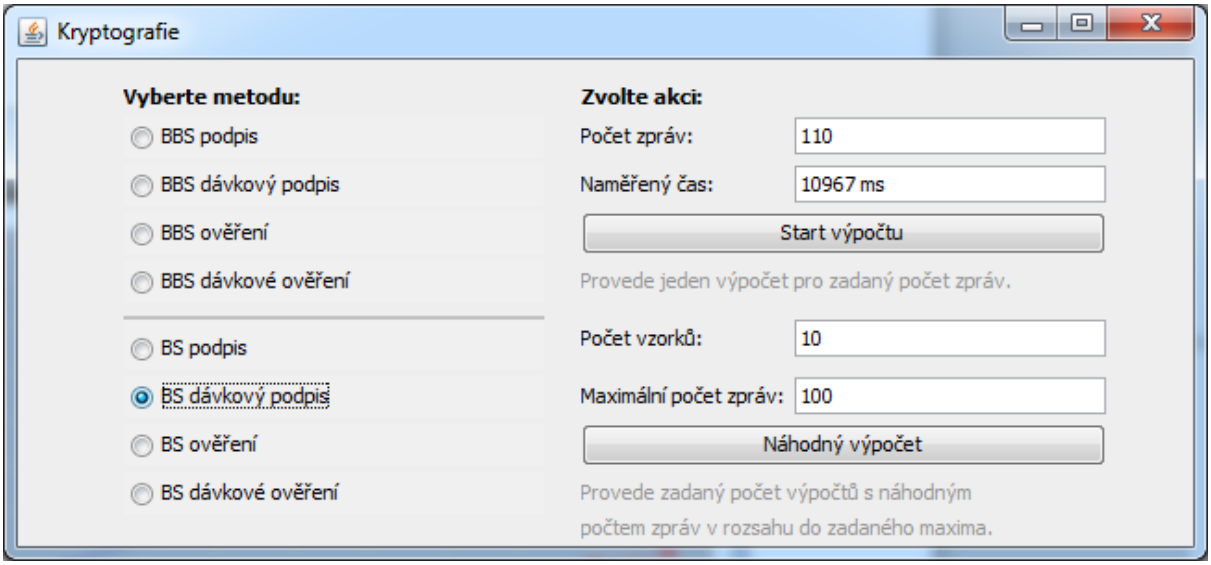
```
        //Výpočet hodnoty T1
        this.t1 = gpk.getU().getElement().powZn(alpha);
        //Výpočet hodnoty T2
        this.t2 = gpk.getV().getElement().powZn(beta);
        //Výpočet hodnoty T3
        this.exp = alpha.getImmutable().add(beta);
              this.t3 =
gpk.getH().getElement().powZn(exp).mul(upk.getA().getElement());

        // //Výpočet hodnoty delta1
        this.delta1 = upk.getX().getElement().mulZn(alpha);
        // Výpočet hodnoty delta2
        this.delta2 = upk.getX().getElement().mulZn(beta);

        //Výpočet hodnoty R1
        this.R1 = gpk.getU().getElement().powZn(ralpha);
        //Výpočet hodnoty R2
        this.R2 = gpk.getV().getElement().powZn(rbeta);
```

Ostatní implementované metody jsou přiloženy v příloze, stejně jako celý program. Zde se jedná pouze o demonstrativní ukázku možného provedení.

5.2 Ovládání programu

Program i grafické prostředí bylo navrženo tak, aby vše bylo uživatelsky snadno pochopitelné a pokud možno co nejjednodušší. Po spuštění programu se zobrazí grafické rozhraní viz obr.č.3, kde je v levé části několik možností pro výběr metody, kterou chceme zobrazit v grafu. Po výběru druhu podpisu případně ověření, jsou dvě možnosti pro vykreslení grafu. V prvním případě se zadává počet zpráv, které chceme podepsat případně ověřit, a poté se spustí výpočet tlačítkem **Start výpočtu** pod touto možností. Následuje vypsání času, který byl potřeba pro podepsání případně ověření zadaného počtu zpráv a vykreslení hodnoty do grafu. V případě zadání dalších vzorků pro stejnou metodu hodnoty v grafu začnou propojovat. Pokud zvolíme jinou metodu podpisu nebo ověření, začnou se hodnoty do grafu vykreslovat odlišnou barvou, což je následně popsáno v legendě pod grafem společně, kde je vypsáno i že se jedná o ruční zadání hodnoty. Druhou možností je využití náhodného generátoru počtu zpráv, kde se zadá, pro kolik vzorků chceme podpis nebo ověření provést a následné maximální počet zpráv. Pokud se zadá počet vzorků např. 10 a maximální počet zpráv 100, program bude podepisovat nebo ověřovat 10 skupin zpráv, které budou obsahovat 0-100 zpráv. Toto se po zadání počtu vzorků a maximálního počtu zpráv spustí tlačítkem **Náhodný výpočet**. Program poté vykreslí graf s dosaženými hodnotami. Obě metody je

možné kombinovat, například v situaci, je nějaká hodnota z náhodně vygenerovaných vzorků zajímavá a je třeba na ni ověřit další metody.



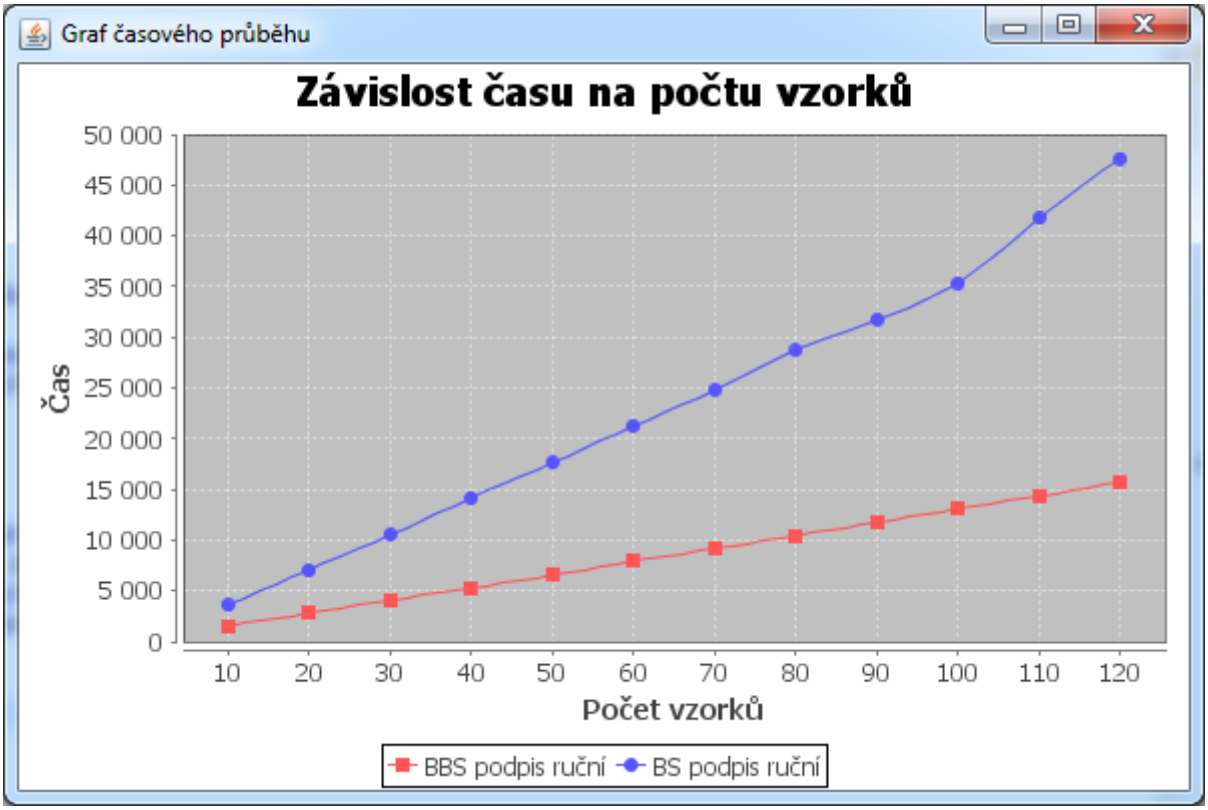
Obr.č.3: Náhled programu

5.2.1 Vykreslování grafu

Graf se začne vždy vykreslovat hned po první provedené akci. Pokud se jedná případ, kdy se ručně zadává počet zpráv, graf vykresluje vzorky pro každou použitou metodu zvlášť vždy po provedeném výpočtu ručně zadaného počtu zpráv. Pokud byla zvolena metoda náhodného počtu zpráv s určitým počtem vzorků, vykreslí se graf najednou. Veškeré výsledky se vypisují do jednoho grafu, kde jsou odlišeny barvami a popsány v legendě, jak již bylo zmíněno.

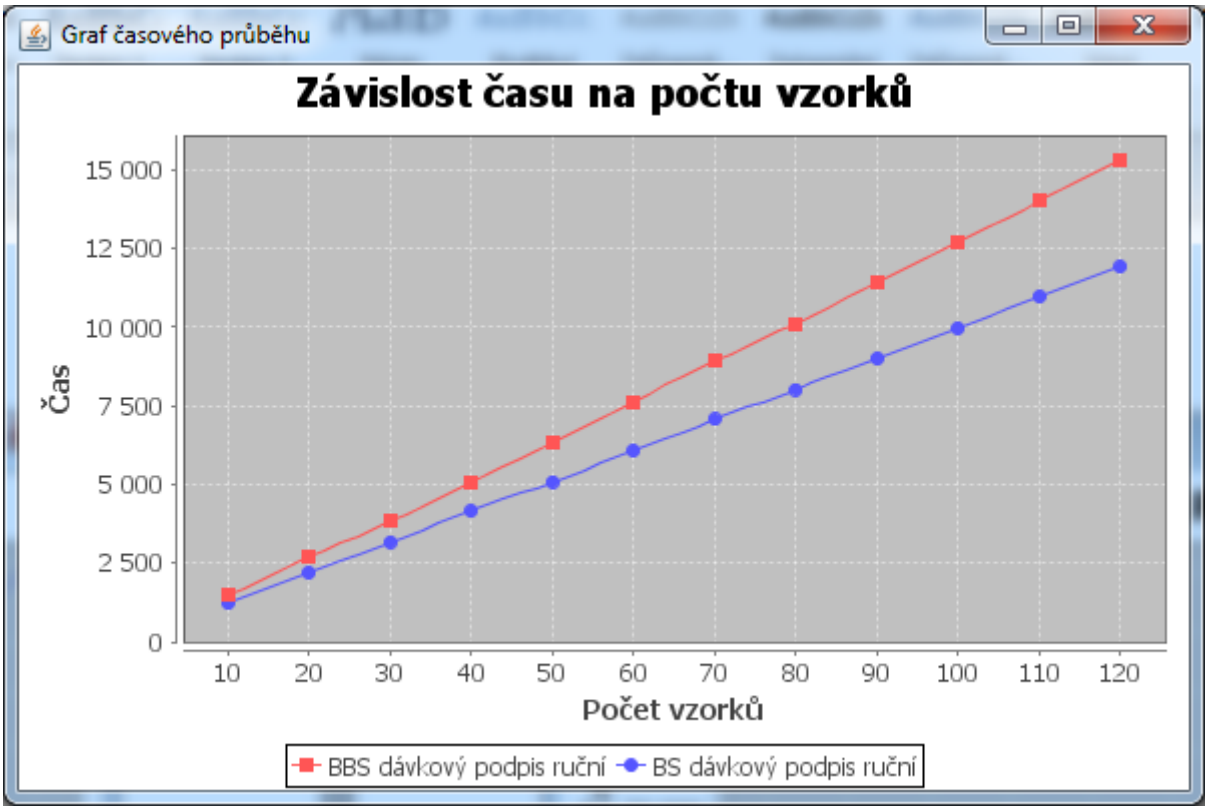
5.3 Porovnání výkonosti implementovaných schémat BBS a BS

Díky programu byly naměřeny níže zobrazené hodnoty. Měřená bylo provedeno na mobilní pracovní stanici Dell Precision M6400 s procesorem IntelCore 2Duo T9600. První měření probíhalo pro podpis BBS s BS s lineárním zvyšováním počtu zpráv od 10 do 120 s krokem 10 zpráv. Jak je vidět z grafu na obrázku č.4, BBS podpis je v tomto případě mnohem rychlejší a s rostoucím se počtem zpráv se i zvyšuje odchylka mezi oběma podpisovými schématy. To je dáno tím, že negeneruje opakovaně hodnoty α,β a bilineární párování provádí pouze jednou a lze tyto hodnoty předem vypočítat offline. V případě BS hodnoty α,β vypočítat předem nelze, jelikož se pro každou zprávu počítají znova. [8]



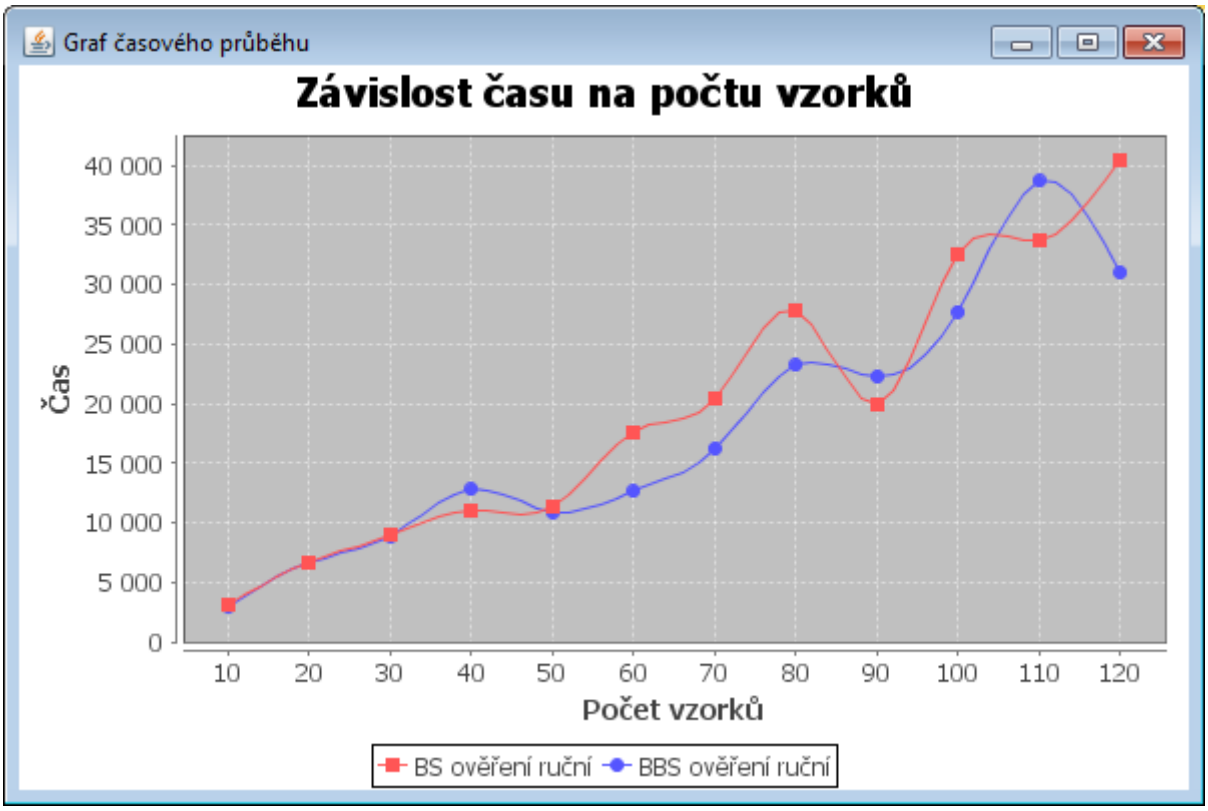
Obr.č.4: Porovnání BBS a BS podpisu aplikovaných na lineárně se zvyšující počet zpráv.

Z grafu na obrázku č.5 jde vidět, že hodnota BS dávkového podepisované je znatelně rychlejší než u podepisování každé zprávy zvlášť, jak bylo popsáno výše. Dávkové podepisování u BS je rychlejší než BBS, jelikož u BS je při výpočtů prováděno 8 operací pro násobení a 8 operací mocniny, oproti tomu u BBS je v obou případech operací 9.

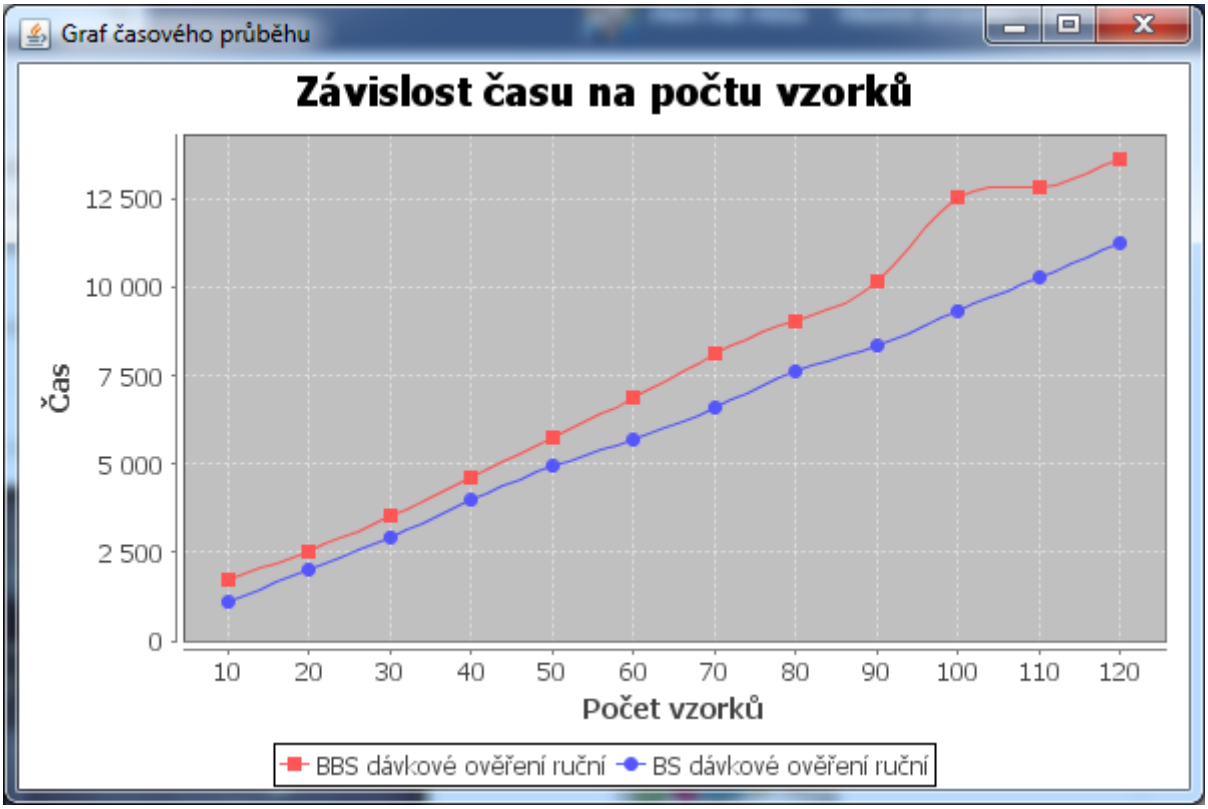


Obr.č.5: Dávkový BBS a BS podpis aplikovaný na lineárně se zvyšující počet zpráv.

Po proměření hodnot u podepisování následovalo měření hodnot obou metod pro ověřování. Jak je vidět na obrázku č. 6, doba ověřovaní se i přes lineární zvyšování hodnot zpráv měnila neúměrně, což se nedá říct o dávkovém ověřování, které dobu ověřování razantně snížilo, což je na obrázku č.7. To je způsobeno tím, že při individuálním ověřování se provádí bilineární párování pro různé dvojice. To může znamenat nelineární dobu výpočtu, což se projevilo i v grafu, jelikož pro každou dvojici trvá výpočet jinou dobu. U dávkového podepisování se výpočet bilineárního párování provádí jen jednou, což zapříčiní lineární růst společně se zvyšujícím se počtem zpráv. Z naměřených hodnot na obrázku č. 4 a č. 5, lze vyčíst, že doba výpočtu normálního podpisu a dávkového se v podstatě neliší. To je dáno možností předem vypočítat bilineární párování u normálního podpisu, a tudíž dávkové ověřování nemá na rychlost podpisu u BBS vliv.



Obr.č.6: Ověřování metodou BS a BSS



Obr.č.7: Dávkové ověření metodou BBS a BS

Z výsledků je patrné, že podpisové schéma BS je vhodné pro dávkové podepisování i ověření, jelikož je vždy rychlejší než BBS schéma. Výsledek rovněž ukazuje, že dávkové ověření je zpravidla rychlejší, než podepisování každé zprávy zvlášť. BS schéma je rychlejší, neboť využívá tzv. VLR (Verifier local revocation) a je také výpočetně méně náročné než BBS. VLR využívá toho, že zprávy informující o zrušení některých uživatelů skupiny zpracovává pouze ověřující strana.

Z výše uvedených grafů lze vyčíst, že v případě omezení 10 sekundovým intervalem na podepsání a ověření zpráv je vhodnější využít metodu BS s dávkovým podepisováním a ověřováním, jelikož je v obou případech rychlejší než metoda BBS. Dá se říct, že metoda BS je vhodnější v každém případě pro systémy vyžadující vlastnosti dávkového podepisování a ověřování. Metoda BBS je poté vhodná pro situace, kdy není potřeba využívat dávkového podepisování, jelikož pří podepisování každé zprávy zvlášť je rychlejší, což je vysvětleno v úvodu této kapitoly.

5.4 Konkrétní měření provedené na implementované metody BBS a BS

Pro porovnání konkrétních hodnot byly zvoleny následující postupy měření. V prvním případě šlo o porovnání normálního podepsání a ověření jedné zprávy pomocí obou metod. Výsledné hodnoty zobrazuje tabulka č.1.

Metoda	Naměřený čas pro 1 zprávu
BBS podpis	205ms
BBS ověření	463ms
BS podpis	220ms
BS ověření	362ms

Tab.č.1 Podepsání a ověření jedné zprávy.

Poté bylo provedeno měření pro různé skupiny zpráv. Měření probíhalo opět pro obě metody BBS a BS. Ověřovaly se nejprve každá zpráva zvlášť a poté se využilo dávkového ověřování. Počty zpráv ve skupinách byly 10, 50 a 100 zpráv. Výsledky měření jsou zobrazeny v tabulce č.2.

Metoda	10 zpráv	50 zpráv	100 zpráv
BBS	2345ms	10790ms	21502ms
BS	2284ms	12292ms	20745ms
BBS dávkové	727ms	2912ms	5021ms
BS dávkové	590ms	2592ms	3885ms

Tab.č.2 Ověření různého počtu zpráv.

Jak je vidět v tabulce č. 1, hodnoty pro podepsání a ověření jedné zprávy se v obou metodách moc neliší, což lze vyčíst také z grafu č.1, kde se hodnoty značně lišily až při vyšších počtech zpráv. Naproti tomu měření pro dávkové ověřování dokazuje, že BS metoda je v tomto případě lepší, jelikož dokáže skupinu 100 zpráv ověřit o necelých 23% rychleji než metoda BBS. Při porovnání individuálního ověřování a dávkového je vidět, že je časová úspora značná pro obě metody značná. Pro 50 zpráv se jednalo u metody BBS o 73% a u metody BS o necelých 80%.

6 Zhodnocení výkonosti, bezpečnosti a délky podpisu BBS a BS

Jak bylo ověřeno v praktické části, BS metoda je v případě dávkového ověřování výkonnější než BBS, což je způsobeno nižší výpočetní náročností. Jak bylo popsáno v kapitole 2.6.2, skupinový podpis BS se skládá ze dvou prvků z G_1 a pěti prvků z $\mathbb{Z}_p$. BBS jak základní tak modifikovaný je složitější, jelikož základní BBS podpis obsahuje tři prvky z G_1 a šest prvků z $\mathbb{Z}_q$. Naproti tomu modifikovaná verze pro výpočet dávkového ověření obsahuje tři prvky z G_1 a šest prvků z $\mathbb{Z}_q$ a jeden prvek z G_T, což je popsáno v kapitole 2.4.2.

Délka podpisu BS, s bezpečností srovnatelnou se standardem RSA podpisu o délce 1024bitů, může být až 1192 bitů. Metoda BBS se stejnou úrovní bezpečnosti má délku podpisu 2553 bitů.

7 Praktické využití

Cloudové uložiště a VANET sítě jsou prostředí, kde je zapotřebí využít dávkového ověřování. Co se týče cloudového řešení, například v oblasti forensic IT, jak je popsáno v kapitole 4.2, kde jsou zasílány data z určité společnosti, která potřebuje, aby byla zajištěna autentičnost a nezpochybnitelnost dat, stejně tak jako informace o odesilateli, je vhodné využít metodu BBS, kdy lze prvky T_1, T_2, T_3 předem vypočítat offline, stejně tak lze předem vypočítat bilineární párování. Jelikož u cloudového řešení předpokládáme jednu ověřující a jedu podepisující stranu, s vyšším výpočetním výkonem, je lepší využít právě metodu BBS, která je rychlejší pro výpočet jedné zprávy.

V případě VANET sítí, kde si informace mezi sebou vyměňují komunikační jednotky ve vozidlech navzájem je zapotřebí rovněž rychlého ověření a zachování autentičnosti a identity odesilatele, aby nedocházelo k podvržení zpráv, případně aby bylo možno odhalit útočníka,

který šíří škodlivé zprávy, které jsou založeny na fiktivních informacích. V případě, že by byl útočník schopný zasílat do VANET sítě milné informace, mohlo by to vést k záměrnému vzniku dopravních komplikací. Proto je zapotřebí i zde využívat digitálních podpisů. Zde je výhodnější použít metodu BS, jelikož se jedná o velké množství zpráv, a metoda BS, jak bylo zjištěno měřením, je v dávkovém ověřování rychlejší než metoda BBS.

Co se týče bezpečnosti obou metod, jsou skupinové podpisy bezpečnější, jelikož lze tímto způsobem poskytnout ochranu soukromí jednotlivým uživatelů.

8 Závěr

V práci byly popsány základy digitálních podpisů, včetně možných útoků na ně. Dále byly vysvětleny nejpoužívanější podpisová schémata, včetně popisu rovnic pro generování veřejných a soukromých klíčů, generování podpisu a samotného ověření. Mezi schémata patří RSA, které využívá problematiky faktorizace čísel, DSA vycházející z problému výpočtu diskrétního logaritmu a podpisové schéma ECDSA, které využívá eliptických křivek pro výpočet parametrů k vytvoření podpisu. Další řešenou problematikou byly skupinové podpisové metody BBS a BS. Ty využívají problémy Strong Diffie-Hellman (SDH) a bilineárních skupin. Na závěr druhé kapitoly je popsán substituční útok, který je možné provést na schémata BBS a BLS.

Dále byly popsány výhody optimalizačních technik, jako dávkové ověřování a agregování podpisů. Tyto techniky vznikly především kvůli potřebě podepisování a ověřování více zpráv najednou, což umožňuje urychlit komunikaci mezi danými zařízeními, neboť není potřeba ověřovat každou zprávu zvlášť. Agregování podpisů zkomprimuje více podpisů do jednoho a sníží tím celkovou délku zprávy, čímž sníží čas potřebný pro přenos. Dávkové ověřování výrazně sníží dobu potřebnou pro ověření většího množství zpráv, což bylo také ověřeno na vytvořeném programu.

Program, který vycházel z již předem naimplementované metody podepisování BBS (Boneh, Boyen and Shacham), podle které byly následně vytvořeny metody pro podepisování a ověřování BS (Boneh, Shacham). Program byl vyvinut ve vývojovém prostředí Eclipse. V 5. kapitole je popsána struktura programu, jeho ovládání a výstupy. Program by měl být přínosem pro názornou ukázku výhody dávkového podepisování a ověřování a případnou aplikaci metod v praxi. V další kapitole byly ukázány procentuální rozdíly mezi implementovanými metodami, kdy bylo zjištěno, dávkové ověřování BBS nebo BS je až o 80% rychlejší než podepisování každé zprávy zvlášť. V poslední kapitole je popsáno možné využití v cloudovém uložišti, případně ve VANET sítích. Pro cloudové uložiště je vhodnější metoda BS, která je výpočetně méně náročná a podepisovatel nemusí být stále online. Z tohoto důvodu musí pro VANET sítě být využita metoda BBS, jelikož komunikační jednotky ve vozidlech mezi sebou komunikují neustále.

Literatura

[1] BURDA, Karel. BEZPEČNOST INFORMAČNÍCH SYSTÉMŮ. Brno, 2005

[2] Public key certificate. [online]. Dostupné z: http://en.wikipedia.org/wiki/Public_key_certificate

[3] MENEZES, Alfred J., Paul C. VAN OORSCHOT a Scott A. VANSTONE. *HANDBOOK of APPLIED CRYPTOGRAPHY*. 1996.

[4] *Digital Signature Algorithm* [online]. Dostupné z: http://en.wikipedia.org/wiki/Digital_Signature_Algorithm

[5] JOHNSON, Don, Alfred MENEZES a Scott VANSTONE. The Elliptic Curve Digital Signature Algorithm (ECDSA). 2001, s. 56. Dostupné z: http://cs.ucsb.edu/~koc/ccs130h/notes/ecdsa-cert.pdf

[6] Zero-knowledge proof [online]. Dostupné z: http://en.wikipedia.org/wiki/Zero-knowledge_proof

[7] BONEH, Dan, Xavier BOYEN a Hovay SHACHAM. Short Group Signatures. 2004, s. 19. Dostupné z: http://www.iacr.org/archive/crypto2004/31520040/groupsigs.pdf

[8] FERRARA, Anna Lisa, Matthew GREEN, Susan HOHENBERGER a Michael Ostergaard PEDERSEN. Practical Short Signature Batch Veri. 2009, s. 24. Dostupné z: http://eprint.iacr.org/2008/015.pdf

[9] TAN, Chik-How. Key Substitution Attacks on Provably Secure Short Signature Schemes. 2005, s. 2. Dostupné z: http://dl.jbnu.ac.kr/file/IEICE/IEICE-TranFundElecCommandCompSci/ieice_ietfec_2005_e88-a-02/0611.pdf

[10] CAMENISCH, Jan, Susan HOHENBERGER a Michael Ostergaard PEDERSEN. Batch Verification of Short Signatures. 2007, s. 18. Dostupné z: http://eprint.iacr.org/2008/015.pdf

[11] ZAVERUCHA, Gregory M. a Douglas R. STINSON. Group Testing and Batch Verification. 2009, s. 18. Dostupné z: http://eprint.iacr.org/2009/240.pdf

[12] BONEH, Dan, Craig GENTRY, Ben LYNN a Hovav SHACHAM. A Survey of Two Signature Aggregation Techniques. s. 10. Dostupné z: http://crypto.stanford.edu/~dabo/papers/aggsurvey.pdf

[13] DAHMEN, Erik, Christoph KRAUB a Tang-Qei WU. Short hash-based signatures for wireless sensor networks. 2012, s. 14. Dostupné z: http://www.cdc.informatik.tu-darmstadt.de/~dahmen/papers/DK09.pdf

[14] LIN, Chu-Hsing, Chen-Yu LEE a Tang-Qei WU. A Cloud-aided RSA Signature Scheme for Sealing and Storing the Digital Evidences in Computer Forensics. 2012, s. 4. Dostupné z: http://www.sersc.org/journals/IJSIA/vol6_no2_2012/31.pdf

[15] SAIFUL ISLAM MAMN, Mohammad a Atsuko MIYAJI. An Optimized Signature Verification System For Vehicle Ad hoc NETwork. 2012, s. 8. Dostupné z: http://arxiv.org/pdf/1208.5096.pdf

[16] SHACHAM, Hovav. Collected Papers where Every Theorem Is Filled with Grief. 2005, s. 97. Dostupné z: http://cseweb.ucsd.edu/~hovav/dist/thesis-hyperref.pdf

Seznam použitých zkratek:

BBS - Boneh-Boyen-Shacham

BLS - Boneh-Lynn-Shacham

BS - Boneh-Shackam

DSA - Digital Signature Algorithm

DSRC - Dedicated Short Range Communication

ECDSA - Ecllipvic Curve Digital Signature Algorithm

IBGS - Identity based group signature

RL - Revocation List

RSA - Rivest-Shamir-Adleman

SDH - Strong Diffie-Hellman

TPD - Tamper Proof Devices

VANET - Vehicular Ad-Hoc Network

VLR - Verifier-Local Revocation

Obsah elektronické přílohy:

Elektronická příloha obsahuje následující soubory:

- DiplomováPráce.pdf – elektronická verze diplomové práce
- bbszip.rar – složka obsahující program spustitelný ve vývojovém prostředí Eclipse a potřebné knihovny pro jeho zprovoznění.