Univerzita Hradec Králové
Fakulta informatiky a managementu
Katedra informačních technologií

Zabezpečení certifikátů Eliptických křivek

Bakalářská práce

Autor: Matěj Boura
Studijní obor: Aplikovaná informatika

Vedoucí práce: Ing. Hana Švecová

Hradec Králové Říjen 2021

Prohlášení:

Prohlašuji, že jsem bakalářskou práci zpracoval samostatně a s použitím uvedené literatury.

V Hradci Králové dne 28.4.2022 Matěj Boura

Poděkování:

Děkuji vedoucímu bakalářské práce Ing. Haně Švecové za metodické vedení práce, cenné rady a vstřícnost při konzultacích.

Anotace

Název: Zabezpečení certifikátů Eliptických křivek

Tato bakalářská práce se zabývá analýzou zabezpečení pomocí certifikátů Eliptických křivek a možnostmi jejího využití. Součástí práce je úvod do teorie kryptografie a problematiky Eliptických křivek, kde jsou představeny koncepty nutné pro pochopení algortimů Eliptických křivek. Na závěr první části práce je krátce představena historie kryptografie Eliptických křivek a srovnání jejich efektivity oproti jejich hlavním konkurentům. Dále je rozvedena problematika využití Eliptických křivek v moderních kryptografických systémech, primárně zaměřená na vhodné křivky a algoritmy, které jsou převzaty z aktuálních standardů. V závěru této části práce jsou představeny nejvýznamnější praktické využití kryptografie Eliptických křivek v současnosti. V závěru je provedeno shrnutí výsledků komparace a využití Eliptických křivek.

Annotation

Title: Securing Elliptic Curves certificates

This bachelor thesis deals with the analysis of securing Elliptic Curve certificates and their applications. The thesis introduces the theory of cryptography and the problem of Elliptic Curves, where the concepts necessary to understand Elliptic Curve algorithms are introduced. At the end of the first part of the thesis, the history of Elliptic Curve cryptography is briefly introduced and a comparison of their efficiency against their main competitors. Then the problem of using Elliptic Curves in modern cryptographic systems is elaborated, primarily focusing on suitable curves and algorithms that are taken from current standards. The most important practical applications of Elliptic Curve cryptography in the present day are presented at the end of this section. Finally, a summary of the results of the comparison and use of Elliptic Curves is made.

Obsah

1. Úvod

Tato kvalifikační práce je zaměřena problematiku využití a zabezpečení Eliptických křivek. Primárním důvodem zvolení daného tématu byl obecný zájem o kryptografii a prohloubení znalostí v oblasti šifrování a celkového zabezpečení, jelikož ve všech těchto věcech je ukryt velký potencionál do budoucích let a zároveň jsou problémy se zabezpečením již nyní velmi často diskutovaným tématem (např. bezpečnost kryptoměn).

První části práce je věnována teoretické problematice se zaměřením na Obecnou a Aplikovanou kryptografii. Jsou zde představeny nejpoužívanější druhy zabezpečení. Na tuto část následně navazuje část v praktické části této kvalifikační práce, která je zaměřena na komparaci zabezpečení kryptografie eliptických křivek (ECC, Elliptic-Curve Cryptography). Taktéž je v této části zpracována teorie Eliptických křivek, nutná pro pochopení algoritmů ECC.

V druhé praktické části je proveden rozbor aktuálně doporučovaných a používaných algoritmů ECC, spolu se seznámením s kritérii výběru vhodných křivek pro aplikaci těchto algoritmů. Na závěr této části jsou představeny aktuálně nejvýznamnější aplikace těchto algoritmů v praxi. Jednou z těchto praktických aplikací je možnost digitálních podpisů a jejich případné navázání na elektronické ID karty. Proto je vhodné zmínit také právní stránky tohoto využití, přičemž v České republice se s možností využití kryptografie eliptických křivek pro elektronické podepisování počítalo již dříve. Přesto trvalo delší dobu, než se u nás začaly podpisové certifikáty s využitím eliptických křivek vydávat a využívat v rámci kvalifikovaných certifikátů. Současné nařízení eIDAS (Nařízení Evropského parlamentu a Rady (EU) č. 910/2014 ze dne 23. července 2014 o elektronické identifikaci a službách vytvářejících důvěru pro elektronické transakce na vnitřním trhu) detailně neuvádí, avšak ani nezakazuje využívání ECC pro elektronické podepisování (blíže v dalších kapitolách).

Cíle této práce jsou tedy, představení nutných teoretických základů ohledně kryptografie a matematiky, a následná analýza s komparací algoritmů ECC s návazností na problematiku zabezpečení ECC s využitím v současnosti.

2. Teorie a přínos eliptických křivek k řešení moderních kryptografických systémů

2.1 Teorie eliptických křivek

Nejobecněji lze definovat eliptické křivky jako množinu bodů v rovině splňující rovnici Rovnice 1, na které je zadaný bod O. Eliptické křivky používané v kryptografii jsou algebraické struktury konstruované nad konečným tělesem, které lze algebraicky klasifikovat a každé konečné těleso je pak jednoznačně určeno počtem svých prvků. Proto v tomto textu budu primárně představovat teorii týkající se tohoto druhu eliptických křivek a důsledkem toho nám bude stačit jednodušší přístup převzatý z [1].

Nechť K je těleso, pak tyto eliptické křivky nad tělesem K jsou definované rovnicí

$$y^2 + 2xy = Ax^3 + Bx^2 + Cx + D,$$

Rovnice 1 Eliptická křivka

kde proměnné x, y mohou nabývat hodnot z tělesa K a A, B, C a D jsou konstanty z K. Tuto rovnici lze dále upravit na tzv. Weierstrassův tvar.

$$y^2 = x^3 + \mathrm{Ax} + \mathrm{B},$$

Rovnice 2 Weierstrassův tvar

kde platí

$$4A^3 + 27B^2 \neq 0.$$

Rovnice 3 Podmínka Weierstrassova tvaru

Z těchto rovnic nám vychází definice eliptických křivek, ke které ještě musíme přidat výše zmíněný bod O, který leží v nekonečnu.

$$\mathrm{E} = \{(\mathrm{x},\mathrm{y}):\ y^2 = x^3 + \mathrm{Ax} + \mathrm{B}\} \cup \{\mathrm{O}\}$$

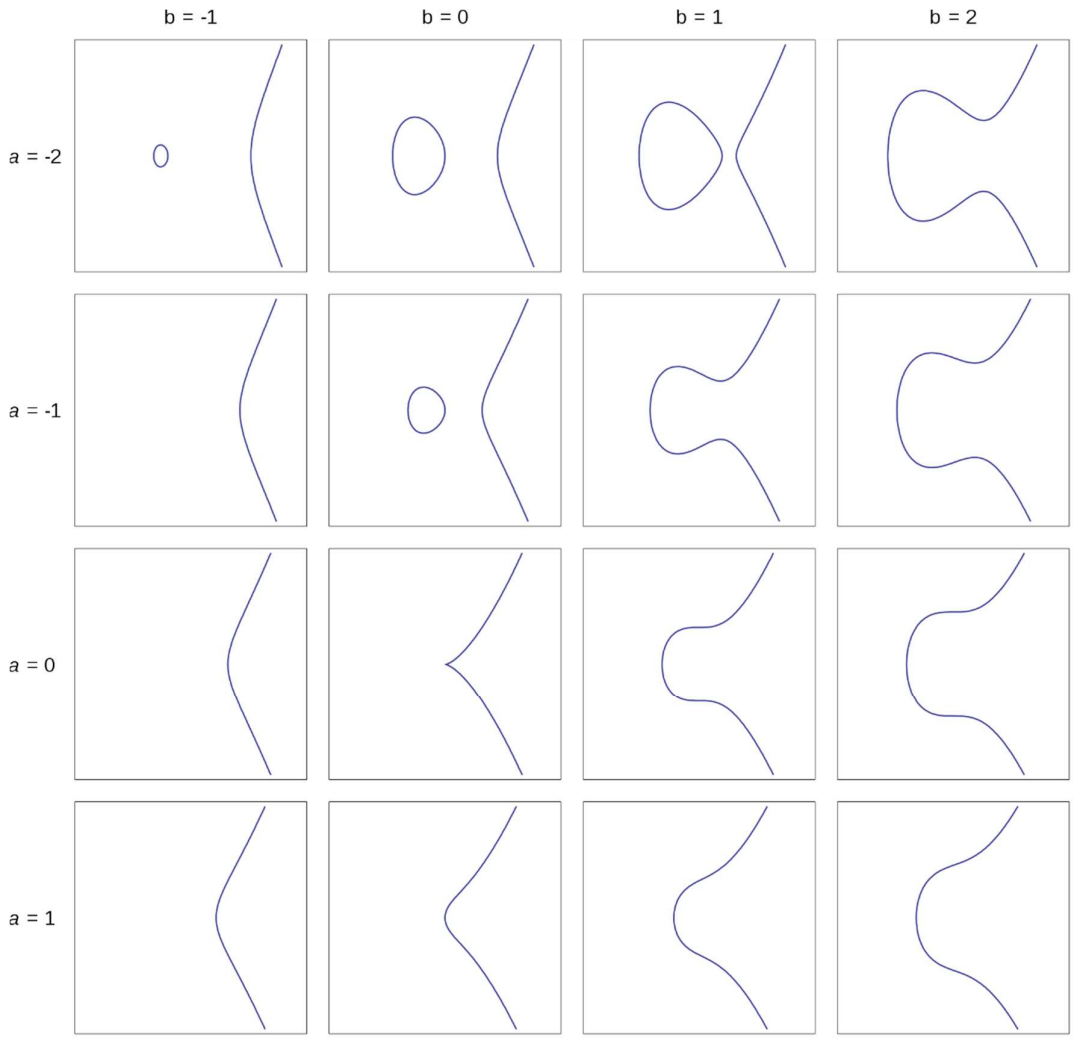
Definice 1 Eliptická křivka

Rovnice 3 nám zajišťuje, aby polynom

$$x^3 + \mathrm{Ax} + \mathrm{B}$$

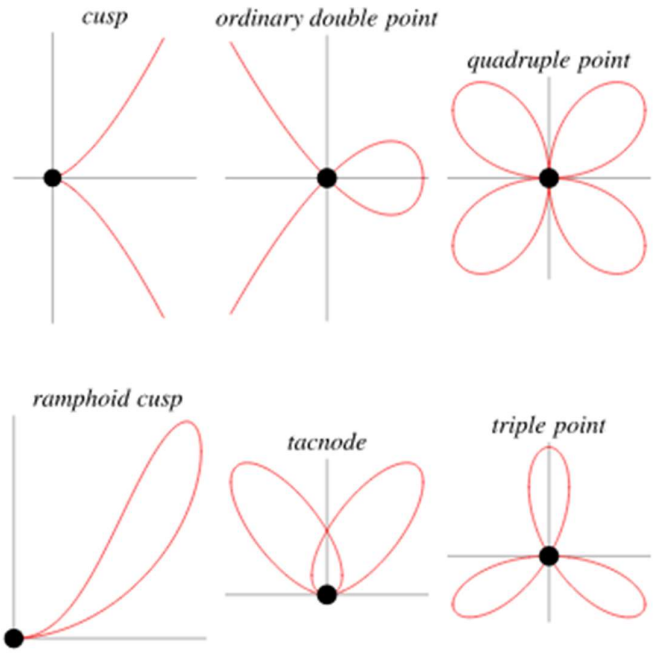
Rovnice 4 Polynom eliptické křivky

neměl vícenásobný kořen, tudíž jeho diskriminant nesmí být roven nule, což je shodné s naší podmínkou, důkaz viz [2]. Důsledkem tohoto nebudou mít eliptické křivky, které uvažujeme ostrý bod tzn. nebudou singulární. Příklady eliptických křivek lze nalézt na Obrázek 1 a příklad singulární křivky pak na Obrázek 2.



Obrázek 1 Příklady eliptických křivek

Zdroj: Stanislav Pal´uch, dostupné na:
https://frcatel.fri.uniza.sk/users/paluch/Kryptografia/ElipticCurve.pdf
strana 10



Obrázek 2 Příklady křivek se singulárními body
Zdroj: Weisstein, Eric W. "Singular Point." From MathWorld--A Wolfram Web Resource. https://mathworld.wolfram.com/SingularPoint.html

2.2 Aritmetika eliptických křivek

2.2.1 Sčítání

Veškeré kryptosystémy založené na bázi eliptických křivek jsou postaveny na základě řešení diskrétního logaritmu, z tohoto důvodu je nutné se seznámit s operací sčítaní bodů na eliptické křivce.

Máme-li zadané dva body

$$P = (x_1, y_1), Q = (x_2, y_2)$$

Rovnice 5 Body na křivce

ležící na eliptické křivce E zadanou dle Rovnice 2, můžeme definovat nový bod R‘ následujícím způsobem. Vedeme přímku p skrz body P, Q. Najdeme průsečík přímky p s eliptickou křivkou E a na tomto místě leží bod R'. Pokud následně uděláme obraz bodu R' přes osu y, tak nalezneme bod R, který je součtem bodů P a Q. Postupem ukázaným v příloze [1] můžeme dojít k vzorcům pro výpočet souřadnic bodu R.

$$P + Q = R = (x_3, y_3)$$

$$x_3 = \mathrm{m}^2 - x_1 - x_2, \qquad y_3 = \mathrm{m} * (x_1 - x_3) - y_1,$$

Rovnice 6 Součet bodů 1

kde *m* je směrnice přímky *p*, kterou lze získat:

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

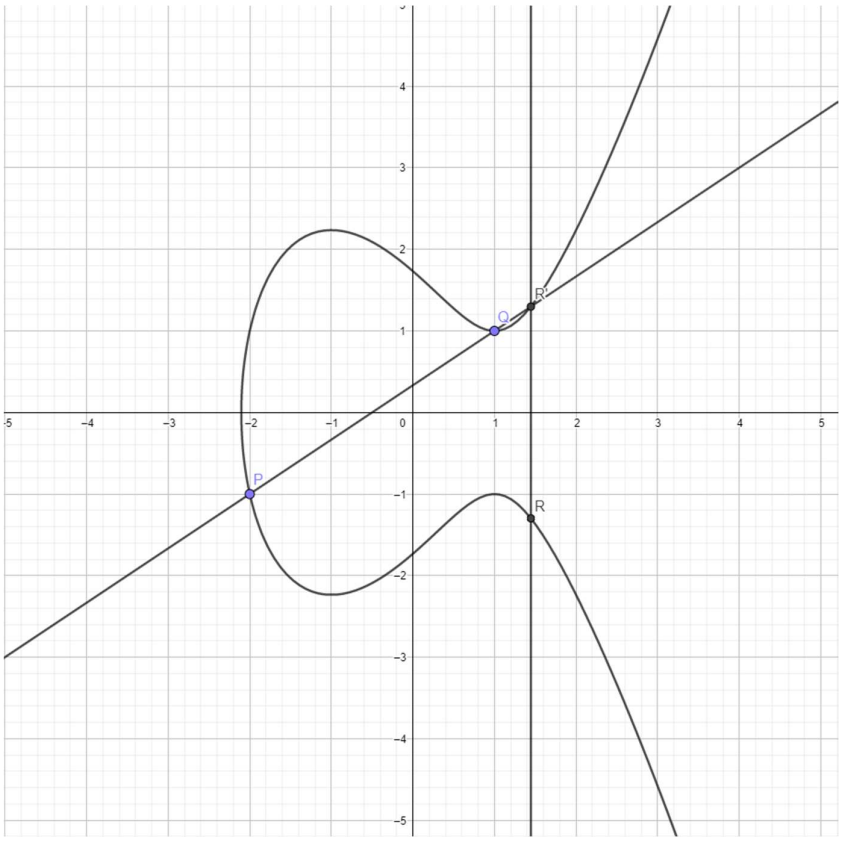
Rovnice 7 Směrnice přímky v součtu 1

Tyto vzorce platí pokud platí:

$$\boldsymbol{x_1 \neq x_2},$$

Rovnice 8 Podmínka součtu 1

příklad takového součtu můžeme vidět níže na Obrázek 3



Obrázek 3 Sčítání bodů první příklad, součet v obrazu průsečíku

Zdroj: autor

Pokud není splněna Rovnice 8, tak nám vznikají tři další varianty výpočtu součtu. První z nich je, pokud mají body sice stejnou hodnotu souřadnice x, ale rozdílnou hodnotu y.

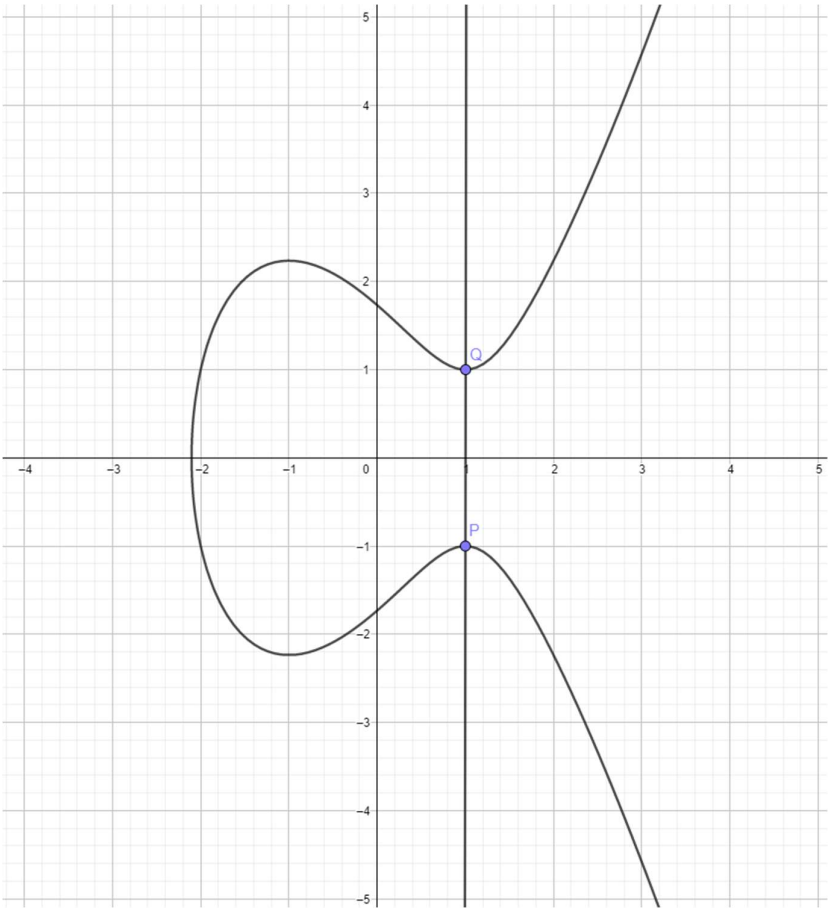
$$\boldsymbol{y_1 \neq y_2}$$

Rovnice 9 Podmínka součtu 2

V tomto případě nám vzniká po propojení vertikální přímka, která protíná eliptickou křivku *E* v nekonečnu neboli v dříve přidaném bodu *O*.

$$P + Q = O = \infty$$

Rovnice 10 Součet bodů 2



Obrázek 4 Sčítání bodů druhý příklad, součet v bodu O

Zdroj: autor

Třetí možností je případ, kdy jsou body natolik blízko u sebe, že můžeme říci:

$$P = Q$$

Rovnice 11 Podmínka součtu 3

V tomto případě přímku vedené skrz tyto body aproximujeme jako tečnu. Najdeme průsečík eliptické křivky s touto tečnou a jeho obraz, který bude součtem těchto bodů. Můžeme dojít k těmto vzorcům, postupem který lze opět najít v literatuře [1].

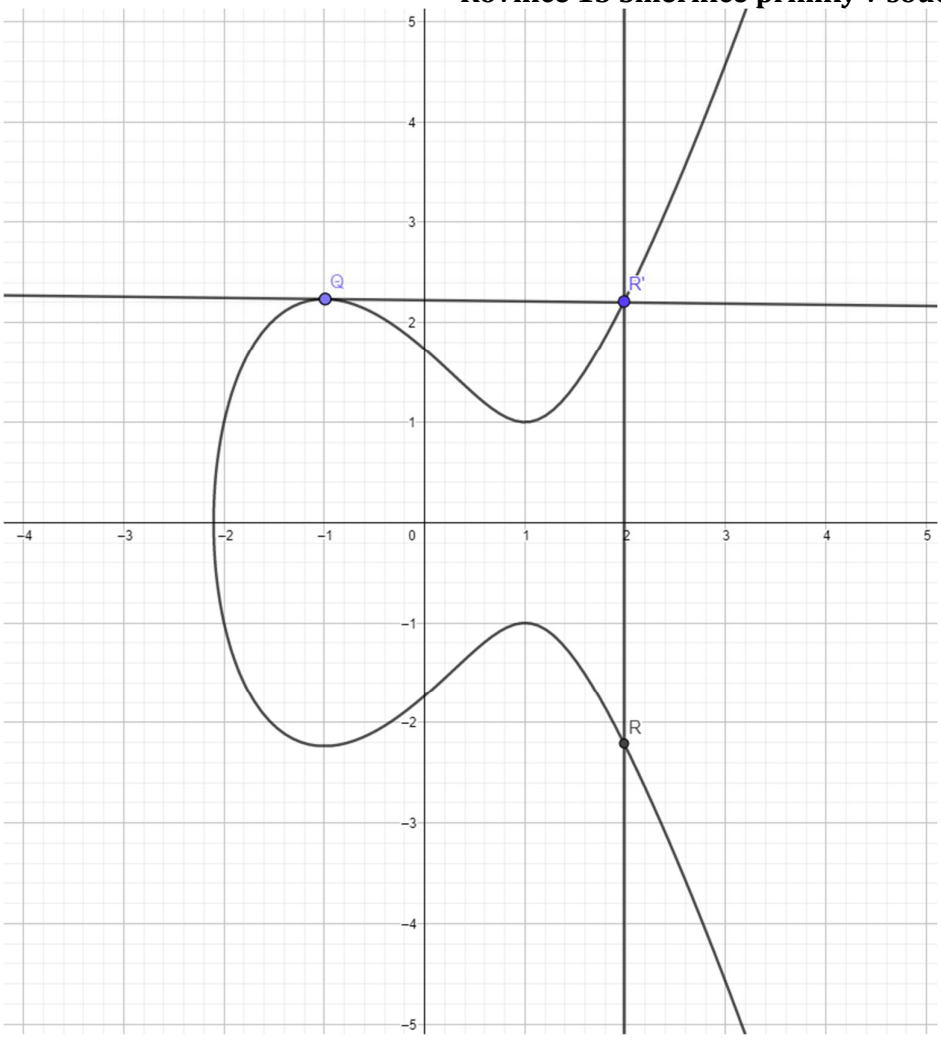
$$x_3 = m^2 - 2x_1, \qquad y_3 = m * (x_1 - x_3) - y_1,$$

Rovnice 12 Součet bodů 3

kde m je směrnice tečny, kterou lze získat:

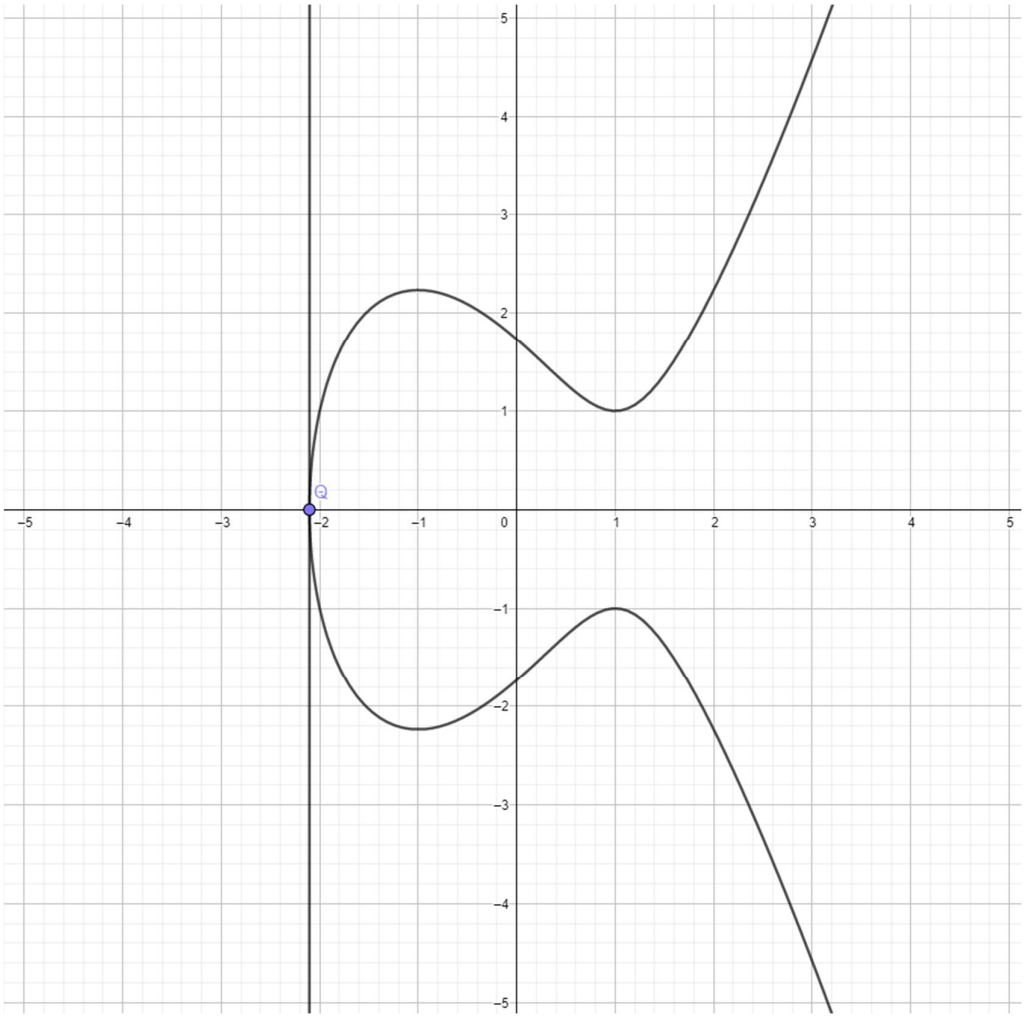
$$m = \frac{3x_1 + A}{2y_1}$$

Rovnice 13 Směrnice přímky v součtu 3



Obrázek 5 Sčítání bodů třetí příklad, totožné body P = Q
Zdroj: autor

V tomto případě se opět setkáváme s problémem, kdy se y-ové souřadnice těchto bodů rovnají nule. V tomto případě nelze vypočítat směrnici tečny a my opět definujeme součet takovýchto bodů jako O, jelikož nám opět vzniká vertikální přímka setkávající se s křivkou E v nekonečnu. Opět tedy platí Rovnice 10**.**



Obrázek 6 Sčítání bodů čtvrtý příklad, totožné body P = Q s y = 0

Zdroj: autor

Nyní se pokusím shrnout výše uvedené poznatky o sčítání bodů na eliptické křivce do algoritmu v pseudokódu.

$if(x_1 \neq x_2)\{$

$m = \frac{y_2 - y_1}{x_2 - x_1}$

$x_3 = \mathrm{m}^2 - x_1 - x_2$

$y_3 = \mathrm{m} * (x_1 - x_3) - y_1$

$P + Q = (x_3, y_3)$

$\}\ else\ \{$

$if(y_1 \neq y_2)\{$

$P + Q = O = \infty$

$\}\ else\ \{$

$if(y_1 = 0)$

$P + Q = O = \infty$

$else\ \{$

$m = \frac{3x_1 + A}{2y_1}$

$x_3 = m^2 - 2x_1$

$y_3 = m * (x_1 - x_3) - y_1$

$P + Q = (x_3, y_3)$

}

}

}

Algoritmus 1 Sčítaní bodů na eliptické křivce

U sčítání bodů na eliptické křivce platí následující pravidla:

1. Komutativita - pro jakékoliv dva body P, Q náležící eliptické křivce E platí:
$$P + Q = Q + P$$
2. Asociativita – pro jakékoliv tři body P, Q, R náležící eliptické křivce E platí:
$$P + (Q + R) = (P + Q) + R$$
3. Existence identity – pro jakýkoliv bod P náležící eliptické křivce E platí:
$$P + \infty = P$$
4. Existence opačného prvku – pro jakýkoliv bod P náležící eliptické křivce E existuje bod P', který také náleží eliptické křivce E a platí:

$$P + P' = \infty$$

Důkazy těchto pravidel lze nalézt v literatuře [1].

2.2.2 Násobení

Dle textu [4] je důležitost eliptických křivek a jejich přínosu spočívá na obtížnosti problému výpočtu diskrétního logaritmu eliptické křivky. Nechť P a Q jsou dva body na eliptické křivce takové, že

$$k * P = Q,$$

Rovnice 14 Násobení bodu skalárem

kde k je skalár. Známe-li P a Q, je výpočetně nesnadné získat k. Je-li k dostatečně velké, je k rovno diskrétnímu logaritmu Q o základu P. Z toho vyplývá, že hlavní operace v algoritmech s eliptickými křivkami souvisí s násobením skaláru *k* s libovolným bodem P na křivce, aby se získal jiný bod Q na křivce. V jednoduchosti nad konečným tělesem E s dostatečně velkým skalárem *k* se jedná o proces sčítání bodu P se sebou samým k-krát. Více o problému diskrétního logaritmu se lze dočíst ze zdroje [6].

2.3 Základy teorie kryptografie

Kryptografie je věda, která zkoumá matematické metody utajování obsahu i prokazování původu přenášených zpráv. Zprávou přitom budeme rozumět číselnou posloupnost, v níž je veřejně známým kódem zakódována informace[8].

Nejjasnější způsob, jak si představit potřebu kryptografie je poslání zprávy přes nezabezpečený komunikační kanál. Tímto kanálem může být například počítačová síť. Problém nastane, pokud zpráva obsahuje důvěrné informace. Zprávu by mohl zachytit a přečíst i někdo jiný než ten, komu má být doručena. Nebo, v horším případě, může nějaký útočník upravit zprávu během přenosu takovým způsobem, že to legitimní příjemce nebude moc ani zjistit.

Základním a klasickým úkolem kryptografie je zajistit důvěrnost pomocí šifrovacích metod. Odesílatel zašifruje otevřený text a získá šifrový text. Šifrový text je předán příjemci zprávy, který převede zašifrovaný text zpět na otevřený text dešifrováním. K dešifrování potřebuje příjemce tajný dešifrovací klíč. Útočník tak stále může zachytit zprávu, ale bude se jednat o zašifrovaný text a šifrování by mělo

zabránit tomu, aby se ze zachyceného textu dala získat jakákoliv informace o původním otevřeném textu.

Text nám pomůže nahlédnout do historie a umožní nám zjistit že šifrování je velmi stará metoda, například jedna z nejznámějších šifer Caesarova šifra je stará již více než 2 000 let[10]. Tato šifra je založena na jednoduché symetrické substituční šifře, kdy je šifrovaný text získáván posunem v abecedě o tři znaky.

V roce 1976 publikovali W. Diffie a M. E. Hellman svůj slavný článek s názvem „New Directions in Cryptography“ [11]. V tomto článku autoři představili revoluční koncept kryptografie s veřejným klíčem, který pojmenovali jako asymetrické kryptografické systémy.

Mezi základní cíle kryptografie patří dle textu [12]:

- Autentizace: Proces prokázání totožnosti.
- Soukromí/důvěrnost: Zajištění toho, aby zprávu nemohl číst nikdo jiný než ten, komu je určena.
- Integrita: Zajištění, že přijatá zpráva nebyla žádným způsobem pozměněna.
- Nepopiratelnost: Mechanismus, který prokazuje, že tuto zprávu skutečně odeslal odesílatel.

Jak již bylo dříve zmíněno kryptosystémy se dají rozdělit do dvou základních kategorií.

2.3.1 Symetrické kryptografické systémy

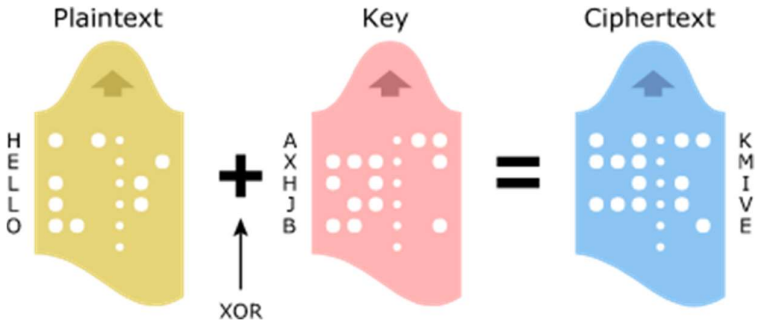
„U symetrických kryptosystémů platí, že zjistit hodnotu dešifrovacího klíče ze znalosti hodnoty klíče šifrovacího je prakticky možné. Z tohoto důvodu se musí utajovat hodnoty obou klíčů, přičemž obvykle platí, že tyto klíče jsou stejné, tj. KD = KE = K, kde K se nazývá tajný klíč. Symetrické kryptosystémy jsou rychlé, a tak se využívají k šifrování velkých objemů dat. Nevýhodou symetrických kryptosystémů je skutečnost, že bezpečné doručení klíče K komunikující protistraně je vzhledem k tajnému charakteru klíče komplikované“[8].

Kryptografická schémata s tajným klíčem se obecně dělí na proudové nebo blokové šifry.

2.3.1.1 Proudové symetrické systémy

Proudové šifry pracují s jedním bitem najednou a implementují určitou formu zpětné vazby, klíč je tedy neustále měněn. Využití proudových symetrických systémů způsobí, že otevřený text je zašifrován do různých šifrovaných textů při použití stejného klíče v proudové šifře.

Nejznámějším příkladem proudové šifry je Vernamova šifra. Princip této šifry je založen na posunutí každého znaku zprávy o náhodný počet pozic v abecedě. Dokonalá náhodnost je velmi důležitý aspekt této šifry, protože klíč musí být stejně dlouhý jako samotný zpráva, tak principiálně není možný útok hrubou silou.



Obrázek 7 Vernamova šifra

Zdroj: https://cryptomuseum.com/crypto/vernam.htm

2.3.1.1.1 Generování náhodných čísel

„Generátory nepředvídatelných čísel klasifikujeme na náhodné a pseudonáhodné. Náhodné generátory generují svá čísla na základě nějakého náhodného fyzikálního děje (např. tepelný šum). Měřením se zjišťuje aktuální hodnota náhodné veličiny a výsledky měření se vhodnou konverzní funkcí g převádějí na čísla pro výstup generátoru. Výstup těchto generátorů je tak skutečně náhodný („true random number generator“). Oproti tomu pseudonáhodné generátory („pseudorandom number generator“) generují svá čísla pomocí vhodného stavového automatu (obr. dole). Stavový automat je hardwarová (např. posuvný registr), nebo datová struktura (např. číslo), která může nabývat více stavů (např. obsah registru, nebo velikost čísla). Stavový automat se nejprve nastaví podle tajné náhodné hodnoty (tzv. seed) do výchozího stavu. Pomocí přechodové funkce f se aktuálnímu stavu pokaždé přiřadí nový následující stav, a tak automat postupně prochází všemi určenými stavy. Speciální konverzní funkce g přitom

každému aktuálnímu stavu automatu přiřazuje číslo, které je výstupem generátoru. Chování pseudonáhodného generátoru je tedy zcela deterministické, tj. oprávněné osoby, které znají funkce *f, g* a seed, dokáží vygenerovat tutéž posloupnost čísel. Požaduje se však, aby neoprávněná osoba došla statistickým testováním vygenerované posloupnosti k závěru, že daná posloupnost je náhodná.“[8]

2.3.1.2 Blokové symetrické systémy

Bloková šifra je takto nazvaná, protože toto schéma šifruje jeden blok dat najednou pomocí stejného klíče pro každý blok. Obecně platí, že stejný blok otevřeného textu bude vždy zašifrován na stejný šifrovaný text, pokud se použije stejný klíč v blokové šifře.

Výběr relevantních blokových šifer je značně rozsáhlejší, proto je částečně převzat z [10]:

2.3.1.2.1 DES – Data Encryption Standard

Dříve nejvíce rozšířený symetrický šifrovací algoritmus, byl vyvinut v 70. letech firmou IBN. Dnes již z důvodu malého klíče (64 bitů z toho 8 kontrolních) není považován za spolehlivý. Princip fungování lze najít na stranách 16-19 ve zdroji [10].

2.3.1.2.2 AES – Advanced Encryption Standard

Nástupce DES, algoritmus Rijndael vybrán v roce 2000 z více kandidátních šifrovacích algoritmů. Tento algoritmus je iterovaná bloková šifra a podporuje různé velikosti bloků a klíčů. Bloky a klíče o velikosti 128, 160, 192, 224 a 256 bitů lze kombinovat nezávisle. Jediný rozdíl mezi šifrou Rijndael a AES je ten, že AES podporuje pouze podmnožinu velikostí bloků a klíčů Rijndaelu. AES stanovuje délku bloku na 128 bitů a používá tři délky klíčů 128, 192 a 256 bitů. Algoritmus AES je v současnosti stále používaný standard například v rámci zabezpečení Wi-Fi sítí dle standardu WPA2.

Šifrování pomocí Rijndael šifry se skládá z počátečního přidání klíče, po kterém následuje použití cyklu *n–1* krát, kde *n* závisí na velikosti bloku a délce klíče. V cyklu probíhá funkce složená z kroků SubBytes, ShiftRows a MixColumns a přidání

klíče. Závěrečné kolo má mírně upravenou funkci, je vynechán krok MixColumns. Vysvětlení jednotlivých funkcí níže. Popis algoritmu Rijndael:

byteString Rijndael(byteString plaintextBlock, key)
{
    InitState(plaintextBlock, state)
    AddKey(state, key_0)
    for (i = 0; i < n − 1; i + +)
    {
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddKey(state, key_i)
    }
    SubBytes(state)
    ShiftRows(state)
    AddKey(state, key_n)
    return state;
}

Algoritmus 2 AES algoritmus

AddKey – přidání klíče do aktuálního stavu (=state) pomocí operace XOR nad aktuálním stavem a klíčem

SubBytes – substituuje všechny bajty stavové matice aplikací funkce S-box, viz [10]

ShiftRows – cyklicky posouvá sloupce stavové matice na základě délky bloku

MixColumns – transformuje každý řádek nezávisle na sobě vynásobením fixním polynomem a zbytku po této operaci modulem [10].

2.3.2 Asymetrické kryptografické systémy

„U asymetrických kryptosystémů naopak platí, že určení hodnoty dešifrovacího klíče ze znalosti hodnoty klíče šifrovacího je prakticky nemožné. Z tohoto důvodu je pak nutné utajovat pouze hodnotu dešifrovacího klíče. Adresát B si nejprve stanoveným postupem vytvoří dvojici šifrovací a dešifrovací klíč. Tato dvojice je odvozena z velkých náhodných čísel, a tak pravděpodobnost, že dva

uživatelé vytvoří stejnou dvojici klíčů, je prakticky rovná nule. Dešifrovací klíč je utajen, a je znám pouze jeho tvůrci B (tzv. soukromý klíč SKB strany B). Hodnotu šifrovacího klíče tvůrce daného kryptosystému je pak zveřejněna. (tzv. veřejný klíč VKB strany B). Platí tak, že KD = SKB a KE = VKB. Adresátovi B potom mohou být zasílány kryptogramy od kohokoliv, bez předchozí znalosti šifrovacího klíče. Velkou nevýhodou je to, že asymetrické kryptosystémy jsou pomalé, a z těchto důvodů používají k šifrování dat o malých objemech. Typicky se jedná o klíče pro symetrické kryptosystémy a o hesla“ [8].

Jak již bylo zmíněno výše, asymetrická kryptografie byla poprvé veřejně popsána na Stanfordově univerzitě profesorem Martinem Hellmanem a postgraduálním studentem Whitfieldem Diffiem v roce 1976. Jejich článek popisuje dvouklíčový šifrovací systém, v němž dvě strany mohou bezpečně komunikovat přes nezabezpečený komunikační kanál, aniž by museli sdílet tajný klíč. Tato metoda závisí na existenci takzvaných jednosměrných funkcí což jsou matematické funkce, které lze snadno spočítat, ale jejich inverzní funkce se počítá poměrně obtížně.

Příkladem může být násobení, nebo umocňování například při výpočtu součinu dvou čísel to nebude náročný problém, ale pokud naopak dostaneme součin dvou čísel a máme z něj získat původní dvě hodnoty bude to úkol nesrovnatelně obtížnější. Nejprve musíme pomocí procesu faktorizace získat všechny možné kombinace, které splňují tento součin a k tomu ještě následně zjistit jaká kombinace z nich je ta správná. Na obdobném principu je založena i početní operace s umocňováním a opačným procesem počítání logaritmů.

2.3.2.1 RSA

Kryptosystém RSA je založen na faktech z elementární teorie čísel, které jsou známy již 250 let. Algoritmus popsali v roce 1977 Ron Rivest, Adi Shamir a Leonard Adleman, dle kterých také nese své označení. Pro vytvoření kryptosystému RSA je třeba vynásobit dvě velmi velká prvočísla a zveřejnit jejich součin *n*, kdy *n* je součástí veřejného klíče, zatímco činitelé *n* jsou utajeni a používají se jako tajný klíč. Základní myšlenka spočívá v tom, že původní činitele *n* nelze získat ze

znalosti součinu. Bezpečnost šifrovací funkce RSA tedy závisí na obrovské obtížnosti faktorizace *n* [13].

RSA je jeden z nejvíce využívaných kryptosystémů, na základě čehož byl vybrán pro komparaci s dalšími kryptografickými systémy za použití eliptických křivek, jelikož jde o jednoho z jejich přímých konkurentů.

Pár veřejného a soukromého klíče RSA lze vygenerovat pomocí algoritmu níže:

1. Zvolte si velká různá prvočísla p a q a vypočítejte n = p - q. Vypočítejte n tak, aby n = p * q.
2. Zvolte si e, které je nesoudělné s ϕ(n), kde ϕ(n) = (p−1) * (q−1). Dvojice (n, e) je zveřejněna jako veřejný klíč.
3. Vypočítejte d, tak že e * d mod ϕ(n) = 1. (n, d) se použije jako tajný klíč.

Algoritmus 3 Generování klíčů RSA

Šifrování je poté definováno funkcí:

$$E: Z_n \rightarrow Z_n, x \rightarrow x^e$$

Rovnice 15 Šifrování RSA

a dešifrování funkcí stejného typu:

$$D: Z_n \rightarrow Z_n, x \rightarrow x^d$$

Rovnice 16 Dešifrování RSA

2.3.2.2 Algoritmus Diffie-Hellman

Po zveřejnění algoritmu RSA Diffie a Hellman navrhli svůj vlastní algoritmus. Algoritmus Diffie-Hellman je využíván pouze pro výměnu tajných klíčů, nikoliv pro autentizaci nebo digitální podpisy.

Tento algoritmus poskytl první praktické řešení problému distribuce klíčů. W. Diffie a M. E. Hellman publikovali svou základní techniku výměny klíčů již spolu s myšlenkou veřejného klíče ve slavném již dříve zmíněném článku "New Directions in Cryptography" v roce 1976 viz [11]. Tento algoritmus také zván Exponenciální výměna klíčů umožňuje dvěma stranám, které spolu nikdy předtím nekomunikovaly, vytvořit vzájemný tajný klíč výměnou zpráv přes veřejný kanál. Toto schéma však odolává pouze pasivní útokům.

Nechť *p* je dostatečně velké prvočíslo, aby bylo obtížné vypočítat diskrétní logaritmy v konečném poli. Nechť *g* je primitivní kořen v tomto konečném poli. Dvojice (p, g) je veřejně známá. Dvojice komunikujících může vytvořit tajný sdílený klíč provedením následujícího protokolu:

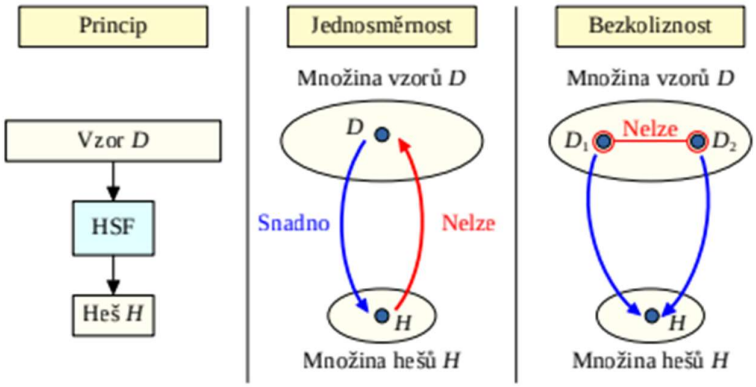
1. První komunikující (dále znám jako Alice) si náhodně zvolí a, 0 ≤ a ≤ p-2, spočte c = g ^ a, poté odešle c druhému komunikujícímu (dále znám jako Bob).
2. Bob náhodně vybere b, 0 ≤ b ≤ p–2, nastaví d = g ^ b a pošle hodnotu d Alici.
3. Alice vypočítá sdílený klíč k = d ^ a = (g ^ b) ^ a
4. Bob vypočítá sdílený klíč k = c ^ b = (g ^ a) ^ b

Algoritmus 4 Výměna klíčů pomocí Diffie-Hellman

2.3.2.3 Hashovací funkce

„Hashovací funkce HSF je kryptografická funkce, která číselnému argumentu D (neboli vzoru) o prakticky libovolné délce (jednotky bitů až triliony triliónů bitů) přiřazuje tzv. hash H, což je číselná hodnota o pevně stanovené délce (typicky o délce 256 až 512 bitů). Formálně budeme tuto funkci zapisovat H = HSF(D). Od Hashovací funkce se vyžadují dvě specifické vlastnosti, které se nazývají jednosměrnost a bezkoliznost. Jednosměrnost (obr. uprostřed) znamená, že určení hodnoty hashe H je pro zadaný vzor D výpočetně snadné, avšak určení hodnoty vzoru D ze znalosti jeho hashe H je prakticky nemožné. Bezkolizností (obr. vpravo) se rozumí, že je prakticky nemožné nalézt nějakou dvojici různých vzorů D1 a D2 takovou, aby jejich hashe byly stejné. V této souvislosti je zapotřebí si uvědomit, že počet číselných posloupností libovolné délky (tj. počet vzorů) je vždy větší, než počet posloupností jediné možné délky (tj. hashů). Z toho pak plyne, že mnoho vzorů musí mít stejný hash (tzv. kolize). Požaduje se však, aby nalezení kolize bylo prakticky nemožné“[8].

Existuje více typů hashovacích funkcí, mezi nejznámější lze zařadit například varianty SHA, nebo MD5.



Obrázek 8 Hashovací funkce
Zdroj: [8]

2.3.2.4 Algoritmy založené na problematice diskrétního logaritmu

V části 2.3.2.1 jsme se zabývali kryptosystémem RSA. Funkce RSA dává prvek m na e-tou mocninu. Je to bijektivní funkce a lze ji efektivně vypočítat. Pokud není známa faktorizace n, neexistuje žádný efektivní algoritmus pro výpočet e. V teorii čísel existují i další funkce které se snadno počítají, ale obtížně invertují. Jednou z nejdůležitějších je umocňování v konečných polích. Pro výpočet inverzní funkce logaritmu exponenciály, tj. pro výpočet x ze znalosti y = g ^ x, není znám žádný efektivní algoritmus a obecně se má za to, že žádný takový algoritmus neexistuje. Tento předpoklad se nazývá předpoklad diskrétního logaritmu.

Mezi tyto algoritmy mimo jiné patří i algoritmy založené na kryptografii eliptických křivek, které rozebereme v následujících částech. Také sem patří jiné algoritmy, které zde stručně popíši.

2.3.2.4.1 ElGamal

Na rozdíl od funkce RSA je ElGamal jednosměrná funkce bez jakýchkoliv dalších informací, které by mohly usnadnit výpočet inverzní funkce.

Generování klíčů, příjemce zpráv postupuje následovně:

1. Zvolí si velké prvočíslo p tak, že p−1 má velký prvočinitel a primitivní kořen g náležící do použitého konečného pole

2. Náhodně zvolí celé číslo x v rozsahu 0 ≤ x ≤ p–2. Tím vznikne trojice (p, g, x) což je tajný klíč.
3. Vypočítá y = g ^ x v konečném poli a tím vytvoří druhou trojici (p, g, y) ze které vzniká veřejný klíč.

Algoritmus 5 Generování klíčů ElGamal

Šifrování odesílatelem probíhá pomocí znalosti veřejného klíče, tak že vynásobí elementy zprávy y ^ k, kde k je náhodně vybrané celé číslo z intervalu <1; p–2>. Dešifrování je následně umožněno znalostí příjemce hodnoty x, kterou má uloženou ve svém tajném klíči.

2.3.2.4.2 DSS – Digital Signature Standard

DSS se měl stát standardní metodou digitálního podpisu pro použití vládními a finančními organizacemi. DSS obsahuje digitální podpisový algoritmus (DSA), který je velmi podobný algoritmu ElGamal. Jde o dalšího konkurenta pro eliptické křivky, zde tedy v oboru digitálních podpisů.

Rozdíl mezi nimi spočívá v tom, že g není primitivním kořenem v konečném poli, ale prvek řádu q, kde q je prvočíselný dělitel p–1. Navíc se vyžaduje, aby binární velikost q byla 160 bitů.

Podepisování. Zprávy m, které mají být podepsány pomocí DSA, musí být prvky konečného pole. V DSS, se k mapování skutečných zpráv na prvky konečného pole používá hashovací funkce h.

2.3.2.4.3 Kryptografie s eliptickými křivkami

Kryptografie s využitím eliptických křivek (ECC = Elliptic Curve Cryptography) je novějším přístupem a je považována za úžasnou techniku s nízkou velikostí klíče pro šifrování. A zároveň těžkou exponenciálně rostoucí časovou náročností jejich prolomení pro útočníka. V ECC poskytuje 163bitový klíč stejnou bezpečnost ve srovnání s tradičním kryptografickým systémem RSA s 1024bitovým klíčem [4]. Jinak řečeno ECC nabízí při dané velikosti klíče podstatně vyšší bezpečnost. V důsledku toho klíč s menší velikostí umožňuje mnohem kompaktnější implementace pro danou úroveň zabezpečení, což znamená rychlejší kryptografické operace, které mohou běžet na menších počítačích, čipech i kompaktnějším

softwaru. Dále existují extrémně efektivní a kompaktní hardwarové implementace ECC, které nabízejí potenciální snížení nákladů na implementaci ještě nad rámec těch, které jsou způsobeny menšími rozměry samotné délky klíče. Kryptografie eliptických křivek se zdá být nejenom atraktivním kryptografickým systémem s veřejným klíčem pro mobilní/bezdrátové sítě, ale přináší také úsporu šířky pásma (rychlost přenosu). Použití eliptických křivek v kryptografii navrhli V. S. Miller a N. Koblitz již v roce 1985, ale do širšího využití se dostaly až kolem let 2004 a 2005. Kryptografie eliptických křivek není jednoduchá na pochopení pro útočníka, tudíž pro něj není snadné ji prolomit.

3. Zabezpečení a využití certifikátů eliptických křivek

Kryptografie eliptických křivek je přístup k asymetrické kryptografii založený na eliptických křivkách nad konečným polem. ECC je založena na problému diskrétního logaritmu eliptických křivek, což je známý NP-hard problém. Na tomto základě vzniklo mnoho adaptací jiných algoritmů, které přijali užití eliptických křivek. Ovšem na RSA konferenci v roce 2018 bylo deklarováno Commercial National Security Algorithm Suite, což je set kryptografických algoritmů považovaných NSA (National Security Agency) za aktuálně vhodné k užívání. Součástí tohoto setu byly dva algoritmy založené na ECC, a to konkrétně ECDSA a ECDH. Proto si v této části představíme právě tyto algoritmy.

3.1 Vhodné eliptické křivky

"Vhodnou" eliptickou křivkou rozumíme eliptickou křivku E definovanou nad konečným polem F_p splňující následující podmínky:

1. Odolat Pollardovu ρ-útoku, mělo by být #E (F_p), #E(F_p) je označení velikosti množiny možných bodů na dané křivce, dělitelné dostatečně velkým prvočíslem n (například n> 2160).
2. Odolat útoku Semaev-Smart-Satoh-Araki, nemělo by #E (F_p) být rovno p.
3. Odolat MOV redukčnímu útoku, nemělo by n dělit p * k–1 pro všechny 1 ≤ k ≤ C, kde C je dostatečně velké, aby bylo výpočetně nemožné nalézt diskrétní logaritmy v $F_{p^C}^*$. V praxi většinou C> = 20.

V normě FIPS 186-4 doporučuje NIST pět eliptických křivek pro použití v algoritmech ECC s pěti různými úrovněmi zabezpečení. Každá křivka je definována nad prvočíselným polem definovaným zobecněným Mersennovým prvočíslem. Což jsou taková prvočísla, která jsou o jedna menší než celočíselná mocnina čísla 2. Taková prvočísla umožňují rychlou redukci na základě práce Solinase [15]. Všechny křivky mají stejný koeficient a = -3, který byl údajně zvolen z důvodu efektivity, a všechny jejich skupinové řády jsou prvočíselné. Pět doporučených prvočísel je:

$$P_{192} = 2^{192} - 2^{64} - 1$$
$$P_{224} = 2^{224} - 2^{96} + 1$$
$$P_{256} = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$$
$$P_{384} = 2^{384} - 2^{128} - 2^{96} + 2^{32} - 1$$
$$P_{521} = 2^{521} - 1$$

Rovnice 17 Eliptické křivky používané v ECC

Pro 256bitová prvočísla navrhuje SEC2 kromě křivky definované NIST P-256 také křivku secp256k1 definovanou nad F_p, kde p = 2256–232–977. Tato křivka se používá v Bitcoinu. Má 256bitový řád prvočísel. Zajímavé je, že tato volba se odchyluje od těch, které byly provedeny ve FIPS 186-4, v tom, že koeficienty křivky jsou a = 0 a b = 7. To znamená, že secp256k1 má j-invariantu 0 a má tedy velmi zvláštní strukturu, lze si o tom více přečíst v práci [16].

Na výše zmíněné RSA konferenci v roce 2018 se také omezilo použití těchto algoritmů tím, že se zavedla eliptická křivka P-384 jako jediná možná křivka s kterou lze tyto algoritmy použít. Tato křivka je blíže definovaná v dokumentu [17], stejně jako ostatní co jsem zde představil.

3.2 ECDSA – Elliptic Curve Digital Signature Algorithm

ECDSA je algoritmus vycházející z algoritmu DSA ovšem za použití eliptických křivek, definuje pravidla a postup při generování digitálního/elektronického podpisu. ECDSA dělá totéž co jakýkoli jiný algoritmus pro digitální podpis, ale efektivněji, protože ECDSA postačuje použití menších klíčů pro vytvoření stejné úrovně zabezpečení. ECDSA se používá k vytváření certifikátů ECDSA, což je typ elektronického dokumentu, který je využíván k ověření vlastníka certifikátu. Certifikáty obsahují informace o klíči použitém k vytvoření certifikátu,

informace o vlastníkovi certifikátu a podpis vydavatele certifikátu, který je ověřeným důvěryhodným subjektem.

3.2.1 Generování klíče

Nejdříve se musí obě strany shodnout na vhodné eliptické křivce E definované nad konečným polem F_p, s charakteristikou p a základním bodem G ležícím na této křivce. Následně je třeba vypočítat N = #E(F_p) a určit řád n bodu G, pro který platí dle [19]:

1. n je prvočíslo, které dělí N
2. $n > 2\ 160$ a zároveň $n > 4\sqrt{p}$
3. n není rovno p
4. p ^ k–1 pro všechny 1 ≤ k ≤ *20* není dělitelné n

Poté již proběhne generování páru soukromého klíče d a veřejného klíče Q následovně:

1. Vybereme náhodné celé číslo d z intervalu *[1, n–1]*, toto číslo bude naším soukromým klíčem.
2. Vypočítáme bod $Q = d \cdot G$ a tento bod se stane veřejným klíčem.

Algoritmus 6 Generování klíče ECDSA

3.2.2 Generování podpisu

Mějme dokument k podepsání m, výše zmíněné parametry domény a klíčový pár (d, Q) kde d je soukromý klíč a Q je veřejný klíč. Podpis poté probíhá dle [19]:

1. Zvolte náhodné nebo pseudonáhodné celé číslo k, takže $1 \le k \le n-1$
2. Vypočítáme $k * G = (x1; y1)$ a $r = x1 \bmod n$. Pokud je $r = 0$, přejdeme zpět ke kroku 1
3. Vypočítáme $(k \text{ ^ } -1) \bmod n$
4. Vypočítáme $e = SHA\text{-}1(m)$
5. Vypočítáme $s = (k \text{ ^ } -1) * (e + d * r) \bmod n$. Jestliže je s = 0, přejdeme zpět ke kroku 1

6. Podpis pro zprávu *m* je dvojice *(r, s)*

Algoritmus 7 Generování podpisu ECDSA

3.2.3 Ověření podpisu

Pro ověření podpisu (r, s) na m získá ověřovací strana autentickou kopii parametrů domény a související veřejný klíč Q. Ověření poté proběhne následovně [19]:

1. Ověřme, že *r* a *s* jsou celá čísla v intervalu *<1; n–1>*
2. Vypočítáme *e = SHA-1(m)*
3. Vypočítáme *w = (s ^ -1) mod n*
4. Vypočítáme *u1 = e * w mod n* a *u2 = r * w mod n*
5. Vypočítáme *X = u1 * G + u2 * Q*. Jestli je X = O (bod křivky v nekonečnu, viz definice eliptické křivky v kapitole 2.1), tak podpis zamítáme. Pokud ne, tak vypočítáme *v = x1 mod n*, kde *X = (x1; y1)*
6. Pokud je *v = r*, tak podpis přijmeme

Algoritmus 8 Ověření podpisu ECDSA

3.2.4 Srovnání ECDSA a RSA

V následující části srovnám hlavního konkurenta ECDSA a sice algoritmus RSA. Níže v tabulce č. 1 je krátce shrnut přínos ECDSA s následným srovnáním velikost klíče RSA a ECDSA při stejné úrovni zabezpečení.

Symetrické šifry	RSA	ECDSA
80	163	1024
112	233	2240
128	283	3072
192	409	7680
256	571	15360

Tabulka 1 Srovnání velikostí klíčů u různých algoritmů
Zdroj: [14]

Jelikož nezávisí pouze na velikosti klíče, tak jsou níže ještě uvedeny výsledky testů převzaté z díla [13], kde byly porovnávány výkonnostní charakteristiky RSA a

ECDSA, nezávislým otestováním každé ze tří hlavních částí algoritmů, tedy generování klíče, generování podpisu a ověření podpisu.

Testy byly provedeny na počítači s procesorem Intel P4 2,0 GHz a 512 MB paměti RAM. Zpráva použitá pro podepisování byl textový soubor o velikosti 100 KB [13].

Délka klíče		Doba generování klíče (sekundy)	
ECDSA	RSA	ECDSA	RSA
163	1024	0,08	0,16
233	2240	0,18	7,47
283	3072	0,27	9,80
409	7680	0,64	133,90
571	15360	1,44	679,06

Tabulka 2 Srovnání rychlostí generace klíče ECDSA a RSA

Zdroj: [13]

Z tabulky výše jasně vyplývá další z výhod ECDSA, jelikož podává lepší výkon při generování klíčů o všech velikostech. Primárně se s rostoucí velikostí klíče nezvedá délka generování klíče exponenciálně jako u RSA, ale pouze lineárně.

Délka klíče		Doba generování podpisu (sekundy)	
ECDSA	RSA	ECDSA	RSA
163	1024	0,15	0,01
233	2240	0,34	0,15
283	3072	0,59	0,21
409	7680	1,18	1,53
571	15360	3,07	9,20

Tabulka 3 Srovnání rychlostí generace podpisu ECDSA a RSA

Zdroj: [13]

Výkonnost obou algoritmů se odlišuje minimálně, dokud není větší klíč, kde ECDSA překonává RSA, naopak u kratších klíčů je RSA minimálně před ECDSA. Jedním z důležitých aspektů procesu generování podpisu je, že část času pro každý algoritmus je strávena výpočtem hashe SHA-1 zprávy.

Délka klíče		Doba verifikace podpisu (sekundy)	
ECDSA	RSA	ECDSA	RSA
163	1024	0,23	0,01
233	2240	0,51	0,01
283	3072	0,86	0,01
409	7680	1,80	0,01
571	15360	4,53	0,03

Tabulka 4 Srovnání rychlostí verifikace podpisu ECDSA a RSA
Zdroj: [13]

Z této tabulky již tak pozitivní zprávy pro ECDSA neplynou, ověřování podpisu je oblast, kde RSA výkonnostně předstihne ECDSA. Růst časové náročnosti pro ověření zprávy podepsané pomocí RSA je při použití větších délek klíčů zanedbatelný, rozdíl se projeví až při přechodu z 7 680 na 15 360 bitů. ECC zaostává ve výkonu v každé délce klíče a vykazuje téměř lineární růst pro rostoucí velikost klíčů.

3.3 ECDH – Elliptic Curve Diffie-Hellman

ECDH se od obecného Diffie-Hellmanova algoritmu odlišuje tím, že je založen na problému diskrétního logaritmu eliptických křivek (ECDLP) namísto problému diskrétního logaritmu (DLP). ECDH je anonymní protokol dohody o výměně klíčů, který umožňuje dvěma stranám *A* a *B*, vytvořit sdílené tajemství přes nezabezpečený kanál, kde každá strana má veřejný-soukromý klíč eliptické křivky [20]. Toto sdílené tajemství pak lze využít buď přímo jako klíč nebo pro derivování jiného klíče. Tento vzniklý klíč poté lze využít například jako soukromý klíč k nějaké ze symetrických šifer.

3.3.1 Výměna klíče

Podobně jako u ECDSA se musí obě strany nejdříve shodnout na vhodné eliptické křivce E definované nad konečným polem F_p, s charakteristikou p a základním bodem G ležícím na této křivce. Následně je třeba vypočítat N = #E(F_p) a určit řád n bodu G, pro který platí dle [19]:

1. n je prvočíslo, které dělí N
2. $n > 2\ 160$ a zároveň $n > 4\sqrt{p}$
3. n není rovno p
4. p ^ k–1 pro všechny 1 ≤ k ≤ *20* není dělitelné n

Poté již proběhne hlavní algoritmus výpočtu a výměny klíče následovně [20]:

1. A vypočte $Q = a * G$, kde a je z intervalu $<2, n-1>$, a pošle Q druhému účastníkovi B
2. B obdobně vypočte $R = b * G$, kde b je z intervalu $<2, n-1>$, a pošle R druhému účastníkovi A
3. A i B je přiřazeno R, respektive Q a vypočtou sdílený tajný klíč S
4. S = a * R = b * Q = a * b * G. Obě strany A i B tedy vypočtou stejnou hodnotu a tím je tajný sdílený klíč S ustanoven.

Algoritmus 9 Výměna klíčů ECDH

3.3.2 Srovnání ECDH a DH

Při stejném zabezpečení je všeobecně známo, že velikosti klíčů, které jsou potřeba pro DH jsou asi 6-7krát větší než velikost klíče ECDH. To znamená výhodu 171bitového klíče ECDH namísto 1024bitového klíče DH při přenosu. Také generování parametrů domény těchto dvou přístupů se ve velké míře odlišuje. Test generování doménových parametrů je převzat z [21] a byl prováděn pomocí stacku openssl v procesoru Intel Xeon s frekvencí 3200 MHz s 1024 KB cache a 2048 KB RAM.

ECDH doménové parametry		DH doménové parametry	
Křivka	Čas (sekundy)	Velikost	Čas (sekundy)

prime192v1	0,0475	256 bit	0,0301
prime256v1	0,0676	512 bit	0,5783
sect283k1	0,0931	768 bit	2,4916
sect409k1	0,1784	1024 bit	7,8891
sect571k1	0,3706	2048 bit	14,6603

Tabulka 5 Srovnání rychlostí generace doménových parametrů ECDH a DH
Zdroj: [21]

Z této tabulky je zřejmé, že při stejných úrovních zabezpečení je ECDH skoro vždy rychlejší. Hlavně je růst časů nutných pro generaci lineární oproti exponenciálnímu růstu u DH algoritmu.

	Počet statických instrukcí	Počet dynamických instrukcí	Načtené dynamické instrukce	Uložené dynamické instrukce
ECDH	306/21217	37 766 470	13 506 830	8 688 365
DH	214/20226	57 229 355	21 312 670	12 908 525

Tabulka 6 Srovnání počtu instrukcí nutných u ECDH a DH
Zdroj: [18]

Kromě velikosti klíčů lze konstatovat, že ECDH je rychlejší, než DH jak je vidět ze statistik provádění instrukcí a využití paměti v tabulce 6. Tato tabulka byla sestavena na základě testů na univerzitě v Sieně, kde porovnávaly různé možnosti šifrování a výměny klíčů.

3.4 Příklady užití ECC

3.4.1 Bitcoin Blockchain

První (a největší) využití je v blockchainu Bitcoinu. Bitcoin je elektronická kryptoměna a ústředním prvkem jejího fungování je kryptografie eliptických křivek. Bitcoinové adresy jsou přímo odvozeny z veřejných klíčů eliptických křivek a transakce se ověřují pomocí digitálních podpisů. Veřejné klíče a podpisy jsou zveřejňovány jako součást veřejně dostupného a kontrolovatelného blockchainu, aby se zabránilo dvojímu utrácení. (Veřejný) Blockchain Bitcoinu je deník všech

transakcí, které kdy byly provedeny. Každý blok v tomto deníku obsahuje hash SHA256 předchozího bloku, tímto způsobem se bloky řetězí dohromady počínaje takzvaným genesis blokem. V Bitcoinu se ECDSA soukromý klíč obvykle používá jako účet uživatele. Převod vlastnictví Bitcoinů z uživatele A na uživatele B se realizuje připojením digitálního podpisu (pomocí soukromého klíče uživatele A) hashe předchozí transakce a informace o veřejném klíči uživatele B na konci nové transakce. Podpis lze ověřit pomocí veřejného klíče uživatele A z předchozí transakce. Algoritmus ECDSA se používá i u většiny ostatních kryptoměnových blockchainů jako hlavní algoritmus pro generování klíčů.

3.4.2 SSH – Secure Shell

Kryptografii eliptických křivek lze v protokolu SSH použít na třech místech. V SSH-2, se klíče relace vyjednávají pomocí výměny klíčů Diffie-Hellman. RFC 5656 specifikuje metodu výměny klíčů ECDH používanou v SSH, a to podle standardu SEC1. Každý server má hostitelský klíč, který umožňuje serveru ověřit se vůči klientovi. Server při výměně klíčů zasílá klientovi svůj hostitelský klíč a uživatel si při výměně klíčů ověřuje, zda otisk klíče odpovídá jeho uložené hodnotě. Server pak ověřuje pravost sebe sama podepsáním přepisu výměny klíčů. Tímto hostitelským klíčem může být veřejný klíč ECDSA. Třetí možností je využití veřejného klíče ECDSA pro ověřování klientů.

Dle průzkumu z [16] provedeného v Říjnu 2013 10,3% serverů podporuje sadu šifer ECDSA pro klíč hostitele.

3.4.3 TLS – Transport Layer Security

V protokolu TLS se eliptické křivky mohou vyskytovat na několika místech. RFC 4492 specifikuje sady šifer s eliptickými křivkami pro protokol TLS. Všechny sady šifer specifikované v tomto RFC používají ECDH. Klíče ECDH mohou být buď dlouhodobé (v takovém případě se používají opakovaně pro různé výměny klíčů) nebo krátkodobé (v takovém případě jsou regenerovány pro každou výměnu klíčů). Certifikáty TLS obsahují také veřejný klíč, který server používá k ověření své pravosti, u výměn klíčů pomocí ECDH může být tento veřejný klíč buď ECDSA nebo RSA.

Dle průzkumu z provedeného v Říjnu 2013 10,3% serverů podporuje určitou formu ECDH a poskytl veřejný klíč ECDSA spolu s informací o tom, jakou křivku používá [16].

3.4.4 Elektronické ID

Fyzické čipové karty se stále častěji používají k ověřování uživatelů. Tyto inteligentní karty obsahují kryptografické hardwarové moduly, které provádějí kryptografické výpočty. Nejčastěji tyto karty obsahují soukromé klíče pro šifrování a podpisy. ECC je pro tyto typy nasazení atraktivní volbou, protože snižuje velikosti klíče a výpočetní složitosti ve srovnání s RSA.

Rakouské národní e-ID karty obsahují veřejný klíč RSA nebo ECDSA a lze je použít k poskytování právně závazných digitálních podpisů.

„Shromáždili jsme 828 911 certifikátů občanských průkazů z databáze LDAP ldap.a-trust.at v lednu 2013. Každý certifikát obsahoval veřejný klíč a podpis RSA od certifikační autority. 477 985 (58 %) certifikátů obsahovalo veřejný klíč eliptické křivky a 477 785 certifikátů bylo správně analyzováno pomocí OpenSSL. Z nich bylo celkem, 253 047 používalo křivku P-192 a 224 738 křivku P-256“ [16].

3.4.5 Elektronický podpis

Od roku 2019 lze používat ECC nejen pro elektronické podpisy, přesněji o kvalifikované certifikáty pro elektronický podpis, ale dle společnosti První certifikační autorita a.s. (I.CA), což je jedna ze tří společností jmenovaná ministerstvem vnitra jako kvalifikovaných poskytovatelů certifikačních služeb pro Českou republiku, lze využít ECC taktéž pro kvalifikované certifikáty pro elektronickou pečeť, komerční certifikáty a komerční technologické serverové certifikáty.

Další dvě certifikační autority, PostSignum v rámci společnosti Česká pošta s. p. a eIdentity a. s., plánují zprovoznění certifikátů založených na ECC taktéž. PostSignum slibuje dokonce zprovoznění již v letošním roce 2022 [9]. Mimo to i Národní certifikační autorita (NCA) zařadila na seznam vydávaných kvalifikovaných prostředků čipovou kartu Starcos (3.5 ID ECC C1R), která eliptické křivky podporuje.

Elektronické podpisy mají již mají praktické využití, jelikož je lze nahrát do elektronické občanky, ta má 8 kontejnerů pro klasické klíče/certifikáty RSA a dalších 8 pro ty založené na ECC. Zatím jsou zde podporované pouze křivky p256, p384 a p521. Tyto podpisy lze využívat jak k podpisování online dokumentů při online podání, např. daňového přiznání, tak k přihlašování do stránek Daňového portálu nebo aplikace Moje VZP. Bohužel jsou tyto funkce závislé taktéž na implementaci appletu na webových stránkách, který musí podporovat ECC abychom byli schopni se přihlásit.

4. Závěr

Tato bakalářská práce měla dva nejdůležitější cíle. Prvním z nich bylo vysvětlení a zanalyzování metod zabezpečení s pomocí eliptických křivek, proto jsem v první teoretické části práce vysvětlil obecné koncepty týkající se eliptických křivek a počítání s nimi. Taktéž jsem představil základy kryptografie a její dělení, aby bylo jasné kam kryptografie eliptických křivek zapadá. V praktické části jsou poté představeny aktuálně významné algoritmy ECC a křivky které lze v těchto algoritmech použít, taktéž kritéria výběru těchto křivek.

Druhým následným cílem bylo srovnání představených algoritmů ECC s jejich přímou konkurencí. Toto srovnání jasně ukazuje výhody ale i nevýhody algoritmů za použití eliptických křivek. Hlavní výhodou se ukazuje být nižší náročnost na velikost klíčů, která zároveň snižuje požadavky na hardware, kde je potřeba tyto algoritmy použít.

Závěr praktické části obsahuje praktické příklady využití ECC. Ze závěru předchozího odstavce lze předpokládat, že ideální možnosti využití jsou u malých zařízení, kde není možné využít velký výkonný hardware. Proto jsou již v současnosti často využívány u elektronických ID a elektronických podpisů, které jsou na tyto karty často nahrávány. Také se dají dobře využít v různých internetových protokolech kde jejich nižší nároky umožní nižší zatížení internetu. Dalším zajímavým příkladem je že se ECC, konkrétně ECDSA, využívá blockchain Bitcoinu a ostatních kryptoměn, kde se tento algoritmus používá pro generování soukromých klíčů, které zde slouží jako adresy uživatelů. Toto je zapříčiněno primárně jeho nižšími nároky na paměť, a tudíž i přenosovou rychlost internetu.

5. Seznam použité literatury

[1] WASHINGTON, Lawrence C. *Elliptic curves: number theory and cryptography*. 2nd ed. Boca Raton, FL: Chapman & Hall/CRC, 2008. Discrete mathematics and its applications. ISBN 978-1-4200-7146-7.

[2] BERAN, Adam. Eliptické křivky nad konečnými tělesy. 2018, 40.

[3] HANKERSON, Darrel R., Scott A. VANSTONE a A. J. MENEZES. *Guide to elliptic curve cryptography*. New York: Springer, 2003. ISBN 978-0-387-95273-4.

[4] SHANKAR, T N a Gadadhar SAHOO. CRYPTOGRAPHY WITH ELLIPTIC CURVES. *International Journal Of Computer Science And Applications*. 2022, **2**.

[5] BLAKE, I., Gerald SEROUSSI, G. SEROUSSI a N. SMART. *Elliptic Curves in Cryptography*. B.m.: Cambridge University Press, 1999. ISBN 978-0-521-65374-9.

[6] POMERANCE, Carl a Shafi GOLDWASSER. *Cryptology and Computational Number Theory*. B.m.: American Mathematical Soc., 1990. ISBN 978-0-8218-0155-0.

[7] MILLER, Victor S. Use of Elliptic Curves in Cryptography. In: Hugh C. WILLIAMS, ed. *Advances in Cryptology — CRYPTO '85 Proceedings* [online]. Berlin, Heidelberg: Springer, 1986, s. 417–426. ISBN 978-3-540-39799-1. Dostupné z: doi:10.1007/3-540-39799-X_31

[8] BURDA, Karel. Kryptografie okolo nás. nedatováno, 132.

[9] *novinky-certifikacni-autority-postsignum-pavel-plachy-cp.pdf* [online]. [vid. 2022-04-12]. Dostupné z: https://www.egovernment.cz/soubor/novinky-certifikacni-autority-postsignum-pavel-plachy-cp/

[10] DELFS, Hans a Helmut KNEBL. *Introduction to cryptography: principles and applications*. 2nd ed. Berlin ; New York: Springer, 2007. Information security and cryptography : texts and monographs. ISBN 978-3-540-49243-6.

[11] DIFFIE, Whitfield a Martin E. HELLMAN. New Directions in Cryptography. In: *Secure Communications and Asymmetric Cryptosystems*. B.m.: Routledge, 1982. ISBN 978-0-429-30563-4.

[12] KESSLER, Gary C. An Overview of Cryptography. nedatováno, 23.

[13] JANSMA, Nicholas a Brandon ARRENDONDO. Performance Comparison of Elliptic Curve and RSA Digital Signatures. nedatováno, 20.

[14] LENSTRA, Arjen K. a Eric R. VERHEUL. Selecting Cryptographic Key Sizes. *Journal of Cryptology* [online]. 2001, **14**(4), 255–293 [vid. 2022-04-03]. ISSN 1432-1378. Dostupné z: doi:10.1007/s00145-001-0009-4

[15] SOLINAS, Jerome A. *Generalized Mersenne Numbers*. 1999.

[16] BOS, Joppe W., J. Alex HALDERMAN, Nadia HENINGER, Jonathan MOORE, Michael NAEHRIG a Eric WUSTROW. *Elliptic Curve Cryptography in Practice* [online]. 734. 2013 [vid. 2022-04-03]. Dostupné z: http://eprint.iacr.org/2013/734

[17] INFORMATION TECHNOLOGY LABORATORY. *Digital Signature Standard (DSS)* [online]. NIST FIPS 186-4. B.m.: National Institute of Standards and Technology. 2013 [vid. 2022-04-03]. Dostupné z: doi: 10.6028/NIST.FIPS.186-4

[18] *A Workload Characterization of Elliptic Curve Cryptography Methods in Embedded Environments*

[19] JOHNSON, Don a Alfred MENEZES. *The Elliptic Curve Digital Signature Algorithm (ECDSA)*. 1999.

[20] HAAKEGAARD, Rakel a Joanna LANG. The Elliptic Curve Diffie-Hellman (ECDH). nedatováno, 4.

[21] DURLANIK, Aytunc a Ibrahim SOGUKPINAR. SIP Authentication Scheme using ECDH. 2005, **8**, 4.

Seznam obrázků

Seznam tabulek

Seznam algoritmů

Seznam rovnic

Seznam definic

6. Přílohy

Oskenované zadání práce

UNIVERZITA HRADEC KRÁLOVÉ
Fakulta informatiky a managementu
Akademický rok: 2020/2021

Studijní program: Aplikovaná informatika
Forma studia: Prezenční
Obor/kombinace: Aplikovaná informatika (ai3-p)

Podklad pro zadání BAKALÁŘSKÉ práce studenta

Jméno a příjmení:	**Matěj Boura**
Osobní číslo:	**I1900153**
Adresa:	Družstevní 803, Nový Bydžov, 50401 Nový Bydžov, Česká republika
Téma práce:	Zabezpečení certifikátů Eliptických křivek
Téma práce anglicky:	Securing Elliptic Curves certificates
Vedoucí práce:	Ing. Hana Švecová Katedra informačních technologií

Zásady pro vypracování:

Cílem práce je analýza a využití Eliptických křivek při zabezpečení certifikátů.

1. Úvod
2. Cíle práce
3. Teorie a přínos eliptických křivek k řešení moderních kryptografických systémů
4. Zabezpečení a využití certifikátů Eliptických křivek
5. Závěr

Seznam doporučené literatury:

[1] K. Burda a Vysoké učení technické v Brně, *Aplikovaná kryptografie*. Brno: VUTIUM, 2013.
[2] M. Oulehla a R. Jašek, *Moderní kryptografie*. 2017.
[3] K. Burda, *Úvod do kryptografie*. 2015.
[4] D. Levický, *Kryptografia o informačnéj a sieťovej bezpečnosti*, roč. 2010. .
[5] *Introduction to cryptography: principles and applications*. New York, NY: Springer Berlin Heidelberg, 2015.
[6] K. Burda, *Kryptografie okolo nás*. 2019.

Podpis studenta: Datum:

Podpis vedoucího práce: Datum: