ENCRYPTING AND SIGNING MESSAGES USING ELLIPTIC CURVE CRYPTOGRAPHY

David Formánek
Gymnázium, Brno, třída Kapitána Jaroše 14
E-mail: davidformanek@seznam.cz

Supervised by: Radek Doležel
E-mail: xdolez35@stud.feec.vutbr.cz

Abstract: This text deals with the use of elliptic curves for cryptographic purposes. It explains the principles of elliptic curve cryptography (ECC) and its advantages over other methods of asymmetric cryptography, it also describes ECDH and ECDSA protocols for secure communications and electronic signature using ECC. An application for ciphering, signing, and verification of messages is a part of the paper.

Keywords: cryptography, elliptic curve, ECC, ECDH, ECDSA

1 ÚVOD

Většina elektronické komunikace není nijak zajištěna proti případnému odposlechnutí, proto je důležité její šifrování, ať už z důvodu, že informace mají v dnešní době často větší hodnotu než hmotný majetek, nebo kvůli prosté ochraně soukromí. Při užití *symetrické kryptografie* se používá pro šifrování i dešifrování stejný tajný klíč, který musí vlastnit obě strany ještě předtím, než začne bezpečná komunikace. Modernější *asymetrická kryptografie* využívá dva různé klíče — veřejný klíč může kdokoliv použít pro zašifrování zprávy, ale pouze její adresát ji může dešifrovat soukromým klíčem (neodvoditelným z veřejného). Z důvodu rychlosti se v praxi používají hybridní kryptosystémy, tzn. kryptografie s veřejným klíčem (např. RSA, Diffie-Hellmann [5]) se využije pouze pro bezpečný přenos klíče a dále se už využívá klasické šifrování. Aplikací asymetrické kryptografie je také *elektronický podpis* zaručující autorství a integritu zprávy vlastníkem privátního klíče.

2 TEORIE KRYPTOGRAFIE ELIPTICKÝCH KŘIVEK

Eliptická křivka [1] je rovinná křivka s obecnou rovnicí

$$y^2 = x^3 + ax + b$$

$a, b \ldots$ parametry křivky, $D = -16(4a^3 + 27b^2)$, $D \neq 0$

Podíváme-li se na křivku jako na grupu, můžeme definovat operaci sčítání bodů. Nejprve stanovíme, že součet trojice bodů křivky ležících v přímce je roven neutrálnímu prvku (0), což je zde tzv. *bod v nekonečnu*, a odtud $P + Q = -R$, kde $-R$ je inverzní prvek k R jako bod s ním souměrný podle osy x. Sečíst lze i dva shodné body (zdvojení bodu) sestrojením tečny. Souřadnice výsledného bodu lze vypočítat analyticky [2]. Protože však pro kryptografické účely potřebujeme přesné a diskrétní hodnoty souřadnic, aplikujeme na výpočty ještě modulární aritmetiku [3], čímž prevedeme křivku nad prvočíselné těleso a dojdeme k rovnicím pro zjištění souřadnic výsledného bodu $-R$ pro zdvojování či obecné sčítání. Každý bod má nyní souřadnice v rozsahu 0 až $(n-1)$ splňující vztah $y^2 \equiv (x^3 + ax + b) \pmod{n}$.

Asymetrická kryptografie je založena na jednosměrných funkcích, u eliptických křivek je tou funkcí *skalární násobení bodu*, které provedeme kombinací operací sčítání a zdvojování, podobně jako u modulárního umocňování [4]. Inverzní operace se nazývá *diskrétní logaritmus v eliptické křivce* a na rozdíl od faktorizace má plně exponenciální časovou složitost. Aplikací ECC na protokoly využívající problém (klasického) diskrétního logaritmu dojdeme k protokolům ECDH (Elliptic curve Diffie–Hellman) a ECDSA (Elliptic Curve Digital Signature Algorithm) tak, že místo modulárního umocňování provádíme násobení na křivkce a jako výsledek bereme x-ovou souřadnici odvozeného bodu. U ECDH tedy dvě strany s privátními klíči k_1, resp. k_2 zvěřejní body k_1P, resp. k_2P a svůj klíč vynásobí se zveřejněným bodem protějšku, čímž oba dospějí k bodu k_1k_2P, ze kterého odvodí klíč pro symetrickou kryptografii.

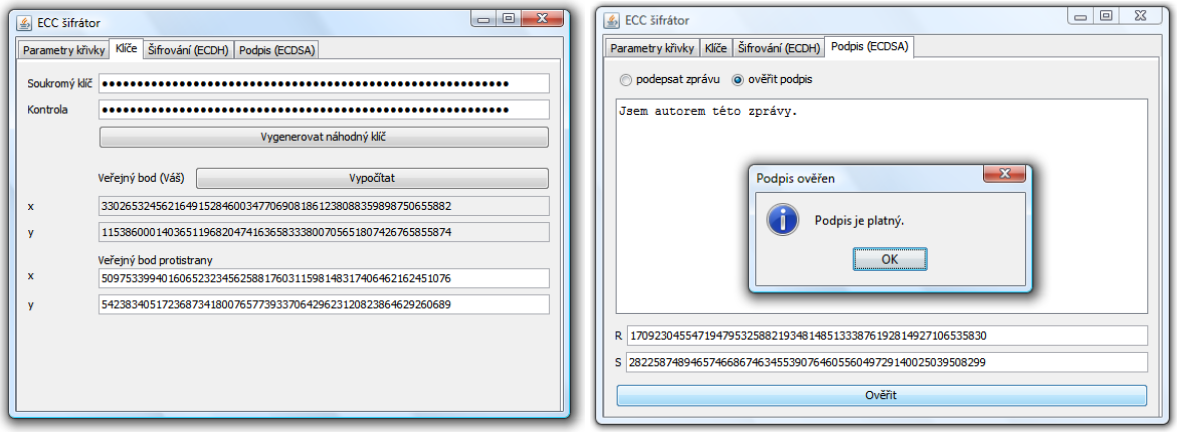
3 APLIKACE PRO ŠIFROVANÍ A PODEPISOVÁNÍ ZPRÁV

Součástí práce je multiplatformní aplikace (obr. 1 na str. 3) využívající kryptografie eliptických křivek a popsaných algoritmů k realizaci protokolů ECDH a ECDSA. Program umožňuje vybrat jednu ze standardizovaných křivek [6], testovací křivku s malými hodnotami, nebo zvolit křivku s vlastními parametry. Po zadání soukromého klíče (nebo jeho vyplnění pseudonáhodným generátorem, který aplikace obsahuje) a ověření validity zadaných parametrů odvodí souřadnice veřejného bodu. S veřejným bodem protistrany poté již umožňuje šifrování i vytváření a ověřování elektronického podpisu s využitím dané eliptické křivky. Aplikace byla navržena pro v praxi použitelnou bezpečnou komunikaci, jejím cílem ale není zajištění samotného přenosu zprávy, pro tyto účely lze však použít zaběhnutých komunikačních prostředků.

3.1 VNITŘNÍ POPIS PROGRAMU

Program je vytvořen v jazyce Java s využitím grafické knihovny `Swing`. Kromě třídy **eccgui.Main**, která pouze volá metodu `main` třídy `Okno`, se aplikace skládá ještě z šesti tříd:

- Třída **ecc.Curve** reprezentující eliptickou křivku nad prvočíselným tělesem v sobě uchovává parametry křivky (a,b,n), kontroluje, zda n je opravdu prvočíslo, a že diskriminant křivky je nenulový.
- **ecc.Point** reprezentuje bod na eliptické křivce nad prvočíselným tělesem. V konstruktoru kontroluje, zda bod leží na zadané křivce. Třída `Point` zajišťuje zdvojování a sčítání bodů, při sčítání ověřuje, zda se jedná o body téže křivky, a v případě součtu dvou shodných bodu automaticky zavolá metodu na zdvojování, aby nedošlo k výpočtu převrácené hodnoty čísla 0. Pomocí kombinace metod pro sčítání a zdvojování počítá daný násobek bodu. Všechny výpočty se provádějí pomocí metod třídy `java.math.BigInteger`, neboť datový typ `int` má příliš malou velikost, a i většina parametrů je typu `BigInteger`.
- **ecc.Params** v sobě uchovává veřejné kryptografické parametry (křivku, výchozí bod, řád bodu a veřejné body) a sama slouží jako vstupní parametr pro třídy `Ecdh` a `Ecdsa`. Kontroluje, zda jsou všechny údaje vyplněny a řád bodu je prvočíselný.
- Třída **ecc.Ecdh** zajišťuje bezpečnou dohodu na společném šifrovacím klíči pomocí ECC. Využívá třídy `Point` pro odvození společného bodu a šifrovací klíč stanoví jako prvních 128 bitů výstupu hašovací funkce SHA-256. Šifruje a dešifruje zprávy pomocí blokové šifry AES-128, k čemuž používá třídu `javax.crypto.Cipher`. Třída `Ecdh` má privátní konstruktor a odkaz na objekt se získává pomocí statické metody `getInstance`, která vygeneruje nový klíč v případě, že šifruje poprvé nebo se změnily parametry, v opačném případě vrátí předchozí instanci. Součástí třídy jsou také privátní metody pro problematický převod mezi polem bytů a řetězcem (`String`) v hexadecimální soustavě.



Obrázek 1: Grafické uživatelské rozhraní vytvořené aplikace

- Účel třídy **ecc.Ecdsa** je generování a ověřování elektronického podpisu. Hašovací funkcí je tu opět SHA-256, tentokrát oříznutá na bitovou délku řádu výchozího bodu křivky. Kromě podpisů zajišťuje třída `Ecdsa` také generování pseudonáhodného čísla menšího než zadaný parametr.
- **eccgui.Okno** je největší třída aplikace. Zajišťuje generování grafického rozhraní, zachytávání uživatelských akcí, vyplňování formulářových polí a volání ostatních tříd programu. Ošetřuje také všechny výjimky zobrazením vyskakovacího okna (podle typu výjimky) s popisem chyby.

4 ZÁVĚR

Praktickou částí práce byla tvorba kryptografické aplikace a její popis. Vytvořený program je spustitelný na všech operačních systémech s podporou Java SE, nevyžaduje instalaci a jeho použití k bezpečné komunikaci je velmi jednoduché. Odvození šifrovacího klíče zabere v řádu desetiny sekund (závisí na výkonu počítače a nastavení) a samotné šifrování již probíhá téměř okamžitě, přitom při správném použití zajišťuje utajení zprávy na mnoho let. Aplikace také umožňuje elektronické podpisování, které s dostatečnou ochranou soukromého klíče garantuje autorství zprávy lépe než klasický podpis.

REFERENCE

[1] *Wikipedia, the free encyclopedia* [online]. 2001 [cit. 2010]. Elliptic curve. Dostupné z WWW: <http://en.wikipedia.org/wiki/Elliptic_curve>.

[2] *Certicom* [online]. 2000 [cit. 2010]. 2.2 Elliptic Curve Addition: An Algebraic Approach . Dostupné z WWW: <http://www.certicom.com/index.php/22-elliptic-curve-addition-an-algebraic-approach>.

[3] *Wikipedie* [online]. 2006 [cit. 2010]. Modulární aritmetika. Dostupné z WWW: <http://cs.wikipedia.org/wiki/Modul%C3%A1rn%C3%AD_aritmetika>.

[4] *Wikipedia, the free encyclopedia* [online]. 2001 [cit. 2010]. Exponentiation by squaring. Dostupné z WWW: <http://en.wikipedia.org/wiki/Exponentiation_by_squaring>.

[5] *Wikipedia, the free encyclopedia* [online]. 2001 [cit. 2010]. Diffie—Hellman key exchange. Dostupné z WWW: <http://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>.

[6] Digital Signature Standard (DSS). In *FIPS PUB 186–3* [online]. USA : NIST, 2009 [cit. 2010]. Dostupné z WWW: <http://csrc.nist.gov/publications/fips/fips186-3/fips_186-3.pdf>.