VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

HLEDÁNÍ KLÍČŮ ZABEZPEČENÝCH BEZDRÁTOVÝCH SÍTÍ POMOCÍ GPU

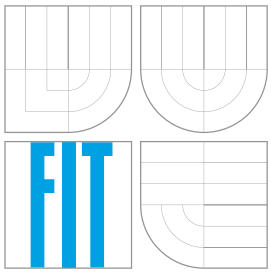
BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE
AUTHOR

RADEK TYRALA

BRNO 2010

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
BRNO UNIVERSITY OF TECHNOLOGY



FAKULTA INFORMAČNÍCH TECHNOLOGIÍ
ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

FACULTY OF INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SYSTEMS

HLEDÁNÍ KLÍČŮ ZABEZPEČENÝCH BEZDRÁTOVÝCH SÍTÍ POMOCÍ GPU

THE USE OF GPU TO SEARCHING FOR THE SECURED WIRELESS NETWORK KEYS

BAKALÁŘSKÁ PRÁCE
BACHELOR'S THESIS

AUTOR PRÁCE — RADEK TYRALA
AUTHOR

VEDOUCÍ PRÁCE — Ing. JIŘÍ JAROŠ
SUPERVISOR

BRNO 2010

Abstrakt

Tato práce se zabývá bezpečností v bezdrátových sítích se zaměřením na sítě typu Wi-Fi. Jsou zde charakterizovány používané bezpečnostní standardy a diskutována jejich slabá místa. Podrobněji se soustředí na bezpečnostní standard WPA2 a na možnosti jeho prolomení. Popisuje princip slovníkových útoků a jejich akceleraci pomocí paralelního zpracování na GPU. Přináší srovnání výkonnosti GPU a CPU pro aplikaci implementující slovníkový útok na bezdrátové sítě zabezpečené standardem 802.11i.

Abstract

This bachelor thesis proposes an analysis of wireless networks security with a particular focus on Wi-Fi type networks. In order to define the central elements of the thesis, let us follow a description of the main steps, namely: The characteristics of the currently applied security standards are provided together with a discussion of their weak points. A somewhat closer insight is offered into the WPA2 security standard as well as into the related breaking possibilities. A description is realized of the principle of dictionary attacks and their acceleration using parallel processing on the GPU. Another important point presented consists in a comparison of the GPU and CPU performance for an application implementing dictionary attack on wireless networks protected with the 802.11i standard.

Klíčová slova

bezdrátové sítě, bezpečnost, Wi-Fi, WPA2, WPA, WEP, GPGPU, OpenCL, CUDA

Keywords

wireless networks, security, Wi-Fi, WPA2, WPA, WEP, GPGPU, OpenCL, CUDA

Citace

Radek Tyrala: Hledání klíčů zabezpečených bezdrátových sítí pomocí GPU, bakalářská práce, Brno, FIT VUT v Brně, 2010

Hledání klíčů zabezpečených bezdrátových sítí pomocí GPU

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Jiřího Jaroše.

.....
Radek Tyrala
14. května 2010

Poděkování

Děkuji vedoucímu práce panu Ing. Jiřímu Jarošovi za projevenou ochotu, poskytnutí odborné pomoci a užitečných rad během tvorby této práce.

Obsah

Seznam zkratek

AES	Advanced Encryption Standard
AMAC	Authenticator's Media Access Control address
ANonce	Authenticator's pseudo-random number (number used once)
AP	Access Point (authenticator)
CCMP	Counter Mode with Cipher Block Chaining Message Authentication Code Protocol
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DNS	Domain Name System
DoS	Denial of Service
ESSID	Extended Service Set Identifier
GPGPU	General-Purpose computation on Graphics Processing Units
GPU	Graphics Processing Unit
HMAC	Hash-based Message Authentication Code
IEEE	Institute of Electrical and Electronics Engineers
IP	Instruction Pointer
IrDA	Infrared Data Association
ISO	International Organization for Standardization
IV	Initialization Vector
KCK	Key Confirmation Key
MAC	Media Access Control address
MD	Message-Digest algorithm
MIC	Message Integrity Check

MITM	Man In The Middle attack
Nonce	pseudo-random number (number used once)
OSI	Open Systems Interconnection
PBKDF2	Password-Based Key Derivation Function 2
PKCS	Public-Key Cryptography Standards
PMK	Pairwise Master Key
PTK	Pairwise Transient Key
RC	Ron's Code
SIMD	Single Instruction, Multiple Data
SHA-1	Secure Hash Algorithm
SMAC	Supplicant's Media Access Control address
SNonce	Supplicant's pseudo-random number (number used once)
STA	STAtion (client, supplicant)
TKIP	Temporal Key Integrity Protocol
UWB	Ultra-WideBand
WEP	Wired Equivalent Privacy
WiMAX	Worldwide Interoperability for Microwave Access
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPA	Wi-Fi Protected Access
WPAN	Wireless Personal Area Network

Kapitola 1

Úvod

S nástupem počítačových sítí se zároveň naskytla otázka bezpečnosti síťové komunikace. Na své významnosti nabrala síťová bezpečnost hlavně s masovým rozšířením Internetu během 90. let 20. století. Bezpečné komunikace bylo potřeba dosáhnout nejen pro vládní organizace, ale také pro společnosti vystupující na Internetu, které získaly povinnost chránit osobní data svých klientů.

Tato problematika vedla k zavedení mnoha bezpečnostních standardů, které poskytují různou úroveň zabezpečení. Spolu s tím se do popředí dostaly také obory jako například kryptografie, která má v souvisloti se zabezpečením komunikace velké uplatnění.

1.1 Bezpečnost v bezdrátových sítích

S příchodem bezdrátových síťových technologií se problematika bezpečnosti komunikace a ochrany citlivých data ještě mnohonásobně zkomplikovala. V tomto případě již utočník nemusí překonat prakticky žádné fyzické překážky a v závislosti na použité technologii může citlivá data získat i na poměrně velkou vzdálenost.

Tato práce se zabývá prověřením zabezpečení komunikace v bezdrátových sítích se zaměřením na možnost prolomení zabezpečení WPA a WPA2 s využitím čipu GPU (Graphics Processing Unit) na grafické kartě. Tohoto cíle se aplikace snaží dosáhnout tzv. *útokem hrubou silou*. Jedná se o slovníkový útok akcelerovaný výpočetním potenciálem čipu GPU.

1.2 Výpočetní potenciál GPU

Od 70. let 20. století, kdy začal vývoj grafických procesorů, bylo jejich využití zamýšleno výhradně pro grafické operace. S využitím GPU pro obecné typy výpočtů tzv. GPGPU (General-Purpose computation on Graphics Processing Units) však přišli největší výrobci grafických karet jako je nVidia a ATI teprve s počátkem 21. století. V GPU byl objeven vysoký potenciál pro určité typy výpočtů a tato technologie se rychle rozšířila. Dnes se přední výrobci grafických karet předhánějí, který produkt bude lepší a výkonější nejen pro grafické aplikace, ale také pro obecné výpočty na GPU.

1.3 Struktura práce

V úvodních kapitolách se práce zabývá rozdělením bezdrátových technologií a jejich popisem z hlediska vývoje a využití. Jsou zde uvedeny vlastnosti jednotlivých síťových technologií

a jejich vzájemné srovnání. Větší pozornost je věnována popisu bezpečnostních standardů využívaných v sítích typu Wi-Fi. Jsou zde diskutovány jejich slabá místa a zároveň jsou jednotlivé bezpečnostní standardy porovnány z hlediska kvality zabezpečení, kterou poskytují. Okrajově jsou shrnuty šifrovací algoritmy využívané v zabezpečení bezdrátové komunikace. Práce přináší bližší pohled na klasifikaci a popis útoků typických pro bezdrátové sítě.

Cílovou architekturou vyvíjené aplikace je zejména grafická karta a čip GPU. Z tohoto důvodu je jedna kapitola věnována popisu architektury čipu GPU. Je zde uvedeno porovnání jednotek GPU a CPU a dále jsou popsány technologie umožňující vývoj aplikací pro GPU a provádění obecných výpočtů na GPU.

Další část práce se již soustředí na vyvíjenou aplikaci. Je zde uvedena analýza problému, popsán princip na jakém je aplikace vystavěna a vysvětlen způsob získání údajů potřebných pro fungování aplikace. Dále se v práci nachází popis implementace, jsou zde shrnuty využité algoritmy a popsána struktura aplikace. Podrobněji se práce věnuje popisu principu převodu aplikace z implementace pro CPU na GPU.

Závěrečná část se věnuje testování vytvořené aplikace a analýze získaných výsledků. Pomocí grafů jsou zde znázorněny výsledky testování na několika grafických kartách v porovnání s výsledky implementace na CPU. V závěru je shrnuta funkčnost aplikace, její smysl a využitelnost v praxi.

Kapitola 2

Typy bezdrátových technologií

První pokusy o uskutečnění bezdrátové komunikace se objevily již v roce 1887, kdy Heinrich Hertz ve své laboratoři demonstroval vytvoření rádiových vln. Než se však první významný typ bezdrátové komunikace (IrDA) dostal do počítačů uplynulo více než 100 let.

V počítačové komunikaci se uplatnilo mnoho různých bezdrátových technologií, které se od sebe v mnoha ohledech liší. Specifická bývá pro každou technologii především frekvence, na které přenos probíhá. S tím přímo souvisí například maximální vzdálenost, na kterou lze komunikaci uskutečnit. V této kapitole je charakterizováno několik bezdrátových technologií, které tvoří hlavní linii v průběhu vývoje bezdrátové komunikace. Uvedené technologie jsou rozděleny do skupin WPAN, WLAN a WMAN, které se od sebe liší zejména maximálním dosahem dané technologie.

2.1 WPAN

Jedná se o osobní bezdrátové sítě (Wireless Personal Area Network), pro které je typický dosah v řádu několika metrů. Sítě WPAN jsou často využívány pro komunikaci s bezdrátovými periferními zařízeními jako je například myš, klávesnice nebo tiskárna. Patří sem například technologie IrDA, Bluetooth, UWB[1], Z-Wave[2] či Zigbee[3].

2.1.1 IrDA

Asociace IrDA vznikla v roce 1993 za účelem vytvořit specifikaci pro nízkonákladovou a jednoduchou bezdrátovou komunikační technologii. Technologie IrDA umožňuje uskutečnit přenos na velmi malé vzdálenosti v řádech několika metrů. Původní specifikace uvádí přenosovou rychlost maximálně 115 kb/s. Pozdější úpravy a rozšíření však uvádí rychlost až do 16 Mb/s. Mezi komunikujícími zařízeními musí být přímá viditelnost a maximální úhel odklonu 15 stupňů. Tato omezení byla odstraněna až s nástupem modernějších technologií jako je Bluetooth či Wi-Fi. Na poli personálních počítačů a mobilních telefonů je technologie IrDA sice na ústupu, v některých typech vestavěných zařízení, jako jsou například dálkové ovladače, však stále zůstává značně rozšířená. Během jednoho roku se na celém světě vyprodukuje více než 40 miliónů zařízení obsahujících tuto technologii [20].

[1] http://www.intel.com/go/uwb/
[2] http://www.z-wave.com/
[3] http://www.zigbee.org/

2.1.2 Bluetooth

Oproti IrDA spočívá hlavní přednost technologie Bluetooth ve zvýšení maximálního dosahu, nárůstu propustnosti a také odstranění podmínky přímé viditelnosti komunikujících zařízení. První specifikace verze 0.7 vytvořená skupinou SIG (Bluetooth Special Interest Group) vznikla v roce 1998. Nejnovější specifikace je Bluetooth verze 3.0 standardizovaná v roce 2009. Bluetooth pracuje ve frekvenčním pásmu 2.4 GHz s maximálním dosahem v řádu desítek metrů a při teoretické maximální propustnosti do 24 Mb/s [1]. Tato technologie se uplatnila zejména v oblasti mobilních telefonů a personálních počítačů. Z hlediska bezpečnosti umožňuje Bluetooth provést autentizaci komunikujících zařízení pomocí klíčů a šifrovat přenášená data. Skutečnost, že provedení autentizace a šifrování závisí pouze na nastavení komunikujících zařízení, otevírá cestu mnoha typům útoků jako je například Bluejacking [15].

2.2 WLAN

V současné době jsou nejrozšířenějším způsobem pro bezdrátovou komunikaci v oblasti počítačů bezesporu sítě typu WLAN (Wireless Local Area Network). Tato technologie se rozšířila mezi poskytovatele Internetu a také do domácností mezi běžné uživatele počítačů. Jedná se o lokální bezdrátové sítě standardizované souborem standardů 802.11 společností IEEE v roce 1997 [5]. Sítě WLAN se stále vyvíjejí a s tím také souvisí doplňování nových standardů do souboru 802.11.

Tyto sítě se často označují souhrným názvem Wi-Fi[4]. Vlastníkem této obchodní značky je společnost Wi-Fi Alliance, která tímto termínem označuje nejprve jen standard 802.11b. Postupně se však tento název rozšiřuje na všechny standardy z rodiny 802.11 definující bezdrátové sítě.

Nejvýznamější standardy a jejich hlavní charakteristiky jsou uvedeny tabulce 2.1. [11] V této tabulce je za účelem srovnání uvedena také charakteristika technologie WiMAX, která však patří do kategorie sítí WMAN a je specifikovaná souborem standardů 802.16. Podrobnosti o technologii WiMAX jsou uvedeny v části 2.3.1.

Tabulka 2.1: Porovnání bezdrátových sítí WLAN a WMAN

Standard	Rok	Frekvence	Maximální rychlost	Dosah (mimo budovy)
802.11a	1999	5 GHz	54 Mbit/s	120 m
802.11b	1999	2.4 GHz	11 Mbit/s	140 m
802.11g	2003	2.4 GHz	54 Mbit/s	140 m
802.11n	2009	2.4/5 GHz	600 Mbit/s	250 m
802.16e WiMax	2005	2–6 GHz	30 Mbit/s	desítky km

Vytvoření vlastní jednoduché sítě WLAN je v dnešní době dostupné, a díky snaze výrobců bezdrátových zařízení také celkem triviální téměř pro každého. To s sebou však přináší značná rizika v oblasti zabezpečení. Existuje sice několik bezpečnostních standardů, které nám umožňují dosáhnout relativně spolehlivého zabezpečení bezdrátového přenosu a ochranu proti potenciálním útočníkům, avšak jejich využití není při konfiguraci bezdrátové sítě striktně vyžadováno. Domácí a nezkušení uživatelé si často nepřipouští skutečnost,

[4] http://www.webopedia.com/TERM/W/Wi_Fi.html

že by se jejich síť mohla stát terčem nějakého útočníka nebo si bezpečnostní rizika v nezabezpečené síti dostatečně neuvědomují. Právě v těchto případech bývá bezpečnost velmi často zcela opomíjena. Tento problém se však zdaleka netýká pouze domácích uživatelů, kde bývá citlivost přenášených dat v porovnání se společnostmi uchovávajícími rozsáhlé databáze s údaji o svých klientech, nesrovnatelně nižší. V minulosti došlo k několika alarmujícím případům, kdy i velké společnosti, včetně bank, používaly zastaralé a naprosto nevyhovující standardy nebo dokonce otevřené spojení bez jakéhokoliv šifrování.

V dnešní době se již situace alespoň na poli velkých společností zlepšila. Avšak značné procento domacích uživatelů, menší firmy, a dokonce i mnoho poskytovatelů internetového připojení zůstává o krok pozadu. Pro získání přístupu k nedostatečně zabezpečené bezdrátové síťi nebo k odchycení komunikace probíhající na takové síti člověk zdaleka nemusí být sofistikovaným útočníkem a dokonce k tomu nepotřebuje ani hluboké znalosti principu zabezpečení a komunuikace v bezdrátových sítích. Díky nepřebernému množství nejrůznějších návodů a nástrojů volně dostupných na Internetu se potenciálním útočníkem může stát za velmi krátkou dobu téměř kdokoliv se základními dovednostmi práce na počítači.

Obvzláště neuspokojivá situace stále panuje u mnohých poskytovatelů bezdrátového připojení. Jedná se typicky o poskytovatele zaměřující se na pokrytí menších obcí, výjimkou však často nejsou ani poskyvatelé ve velkých městech. Nejčastěji problém spočívá v nedostatečně zabezpečeném nebo dokonce vůbec nezabezpečeném přenosu dat mezi koncovým bodem uživatele a přístupovým bodem poskytovatele. V tomto případě by uživateli Internetu od takového poskytovatele k ochraně dat nepomohlo ani sebelepší zabezpečení jeho domácí sítě. Útočník by citlivá data mohl jednoduše získat při jejich přenosu mezi uživatelem a poskytovatelem.

Z tohoho důvodu by jedním z kritérií při výběru poskytovatele bezdrátového připojení k Internetu měl být způsob zabezpečení přenosu dat. Možnost výběru mezi více poskyvateli Internetu je však v mnohých lokalitách stále nedosažitelným luxusem. Poskytovatele s monopolním postavením v daných lokalitách pak nic nenutí zlepšovat úroveň nabízených služeb včetně zabezpečení.

2.3 WMAN

Sítě WMAN (Wireless Metropolitan Area Network) jsou charakteristické maximálním dosahem v řádu desítek kilometrů. Tento typ sítí má představovat alternativu k sítím WLAN a zaměřit se především na k pokrytí odlehlých oblastí. Nejvýznamějším představitelem tohoto typu sítí je technologie WiMAX.

2.3.1 WiMAX

WiMAX (Worldwide Interoperability for Microwave Access) je na trhu bezdrátové komunikace poměrně mladou technologií. Je tvořena souborem standardů IEEE 802.16 z roku 2005. Tato technologie se pomalu rozšiřuje a jsou do ní vkládána vysoká očekávání. Předpokládá se velká konkurenceschopnost k Wi-Fi, DSL a kabelovým sítím. Velkou předností této technologie je schopnost přenosu dat v porovnání s Wi-Fi na mnohonásobně vyšší vzdálenosti při zachování propustnosti v řádu desítek megabitů za sekundu [6].

Vzhledem ke značnému zájmu většiny mobilních operátorů a poskytovatelů Internetu lze očekávat velké rozšíření a další rozvoj této technologie. Sítě WiMAX by se v blízké budoucnosti mohly ukázat jako vhodné řešení problému se zajištěním konektivity k síti Internet pro velmi špatně dostupné a odlehlé oblasti.

Kapitola 3

Bezpečnost bezdrátové komunikace

S rozvojem bezdrátových technologií bylo zapotřebí zajistit adekvátní zabezpečení komunikace a důvěryhodnost přenášených dat. Za tímto účelem bylo vytvořeno několik bezpečnostních standardů jako je WEP, WPA či WPA2[1]. Tyto standardy se od sebe v mnoha ohledech značně liší a s tím také souvisí kvalita zabezpečení, kterou nabízí. Bezpečnost je zpravidla zajištěna pomocí dvou mechanismů, které podle referenčního modelu ISO/OSI probíhají na spojové vrstvě. Jedná se o mechanismus autentizace komunikujících stran a šifrování přenášených dat.

Následující bezpečnostní standardy našly své největší uplatnění v sítích WLAN, které jsou vzhledem ke svému rozšíření a dosahu vystaveny největšímu počtu nejrůznějších typů útoků. Podrobné rozdělení a charakteristika útoků na sítě WLAN se nachází v kapitole 5.

3.1 WEP

Jedná se o zkratku z anglických slov *Wired Equivalent Privacy*, což ve volném překladu znamená bezpečnost odpovídající komunikaci po metalických spojích. Za tímto účelem byl protokol WEP navržen, ovšem významu svého názvu tento protokol zdaleka nedostál.

Všechny sítě 802.11 mají zabudovaný protokol WEP, který používá symetrický postup šifrování, kdy se pro šifrování a dešifrování používá stejný algoritmus i stejný klíč. Protokol WEP nabízí pouze jednostrannou autentizaci a to ve směru klienta vůči přístupovému bodu. Autentizace může být otevřená, což je implicitní volba a představuje v postatě nulovou autentizaci. Druhou možností je autentizace sdíleným klíčem, která probíhá prostřednictvím výzvy a odpovědi, kdy na základě požadavku vyslaného klientem na autentizaci sdíleným klíčem (athentication request) mu přístupový bod zašle text (challenge), který klient musí zašifrovat svým klíčem a odeslat zpět přistupovému bodu. Ve skutečnosti se ověřuje totožnost síťové karty, nikoliv samotné osoby uživatele, což je jedna z hlavních slabin autentizace v rámci WEP. [11]

Pro šifrování dat je využit šifrovací algoritmus RC4 (viz 4.1). WEP umožňuje pro šifrování použít klíč s délkou 64 nebo 128 bitů. Součástí klíče je však také dynamicky se měnící vektor IV (Initialization Vector), který má vždy délku 24 bitů. Délka uživatelského klíče potom činí jen 40 resp. 104 bitů.

Protokol WEP byl do praxe zaveden v roce 1999, ale velmi brzy se ukázalo, že kvůli svým nedostatkům neposkytuje dostatečnou úroveň zabezpečení ani pro domácí uživatele.

[1] Označení pro bezpečnostní standard 802.11i (http://standards.ieee.org/getieee802/download/802.11i-2004.pdf)

Největším nedostatkem je kromě minimální autentizace statický klíč a pouze 24 bitový inicializační vektor. Tento vektor se sice s každým paketem mění, ale vzhledem k jeho délce existuje jen 2^{24} různých kombinací. V reálném čase pak dochází k opakování stejných inicializačních vektorů [11]. Pokud utočník odchytne dostatečné množství zašifrovaných paketů, může z nich získat klíč pro přístup do bezdrátové sítě zabezpečené protokolem WEP i bez použití slovníkového útoku (Brute Force Attack) [14]. V současné době existuje mnoho nástrojů, které umožňují kromě monitorování a odposlechu také prolomení šifrovacího klíče pomocí odchycených paketů. Jedná se například o aplikace Aircrack-ng[2], WepCrack[3] či AirSnort.

Vzhledem k uvedeným nedostatkům protokolu WEP jej není vhodné používat pro zabezpečování jakékoliv komunikace. Doporučuje se využívat modernější zabezpečovací standardy jako je WPA a WPA2. WPA s protokolem TKIP (Temporal Key Integrity Protocol) je možné po aktualizaci firmwaru použít i na zařízeních, které původně podporovaly pouze protokol WEP.

3.2 WPA

Bezpečnostní standard WPA (Wi-Fi Protected Access) vznikl v roce 2002 jako reakce na nedostatečné zabezpečení, které poskytoval protokol WEP. WPA zahrnuje funkce standardu 802.11i, který se měl objevit až v roce 2004. Jedná se zejména o funkce, které mohou být implementovány bez hardwarových úprav existujících zařízení. Oproti protokolu WEP přináší WPA lepší zabezpečení dat a také autentizaci komunikujících zařízení pomocí 802.1x. WPA je zpětně kompatibilní s protokolem WEP a dopředně kompatibilní s WPA2. [19]

Šifrovací algoritmus zůstává z důvodu zajištění kompatibility stejný jako u protokolu WEP, data jsou tedy šifrována pomocí proudové šifry RC4.

WPA implementuje protokol TKIP, který je určen k řešení hlavních nedostatků WEP (často označován jako WEP-fix). Obsahuje funkce pro dynamické regenerování klíčů, kontroly integrity zpráv a číslování paketů na ochranu proti útokům typu replay. [11]

Avšak ani WPA při použití protokolu TKIP neposkytuje absolutní zabezpečení. V roce 2009 publikovali japonští vědci Toshihiro Ohigashi a Masakatu Morii způsob, jak lze při použití útoku *Man in the middle* falzifikovat zasílanou zprávu během jedné minuty [10].

3.3 WPA2

WPA2 vzniklo v roce 2004 jako standard 802.11i a v současné době představuje nejmodernější způsob zapezpečení bezdrátových sítí. Kromě slovníkových útoků, které jsou účinné pouze při použití slabého hesla nebyl zatím zaznamenán žádný připad prolomení tohoto zabezpečení.

WPA2 nabízí stejný způsob autentizace jako WPA, tedy pomocí 802.1x. Místo protokolu TKIP je implementován protokol CCMP[4], který pro šifrování dat používá standard AES (Advanced Encryption Standard).

Tento standard je implementován ve všech moderních zařízeních sloužících pro komunikaci v sítích WLAN, vzhledem k bezpečnosti, kterou poskytuje, je doporučeno jej přednostně využít.

[2] http://www.aircrack-ng.org/
[3] http://wepcrack.sourceforge.net/
[4] protokol CCMP je definován ve standardu 802.11i

Kapitola 4

Šifrování dat

Již v dobách starého Říma se lídé snažili přijít na způsoby, jak zašifrovat důležité informace tak, aby je nepřítel nemohl získat. Mezi starodávné šifry se řadí například známá Caesarova šifra, která spočívá v posunutí každého znaku zprávy o stejný počet pozic. Podobné šifry jsou sice zajimavé, avšak vzheledem k jejich jednoduchosti má většina z nich v dnešní době využití pouze pro skautské hry a nemůže být řeč o jejich nasazení pro šifrování komunikace v bezdrátových sítích.

Šifrováním obecně se zabývá obor kryptografie. V současnosti využívané šifry můžeme rozdělit na symetrické, které pro zašifrování i dešifrování zprávy používají identický klíč, a šifry asymetrické, které využívají principu veřejného a soukromého klíče. Pomocí veřejného klíče je zpráva zašifrována, pomocí soukromého klíče pak dešifrována. Výhodou asymetrické kryptografie je fakt, že komunikujícím stranám odpadá nutnost předání tajného soukromého klíče.

V oblasti zabezpečení bezdrátové komunikace našly uplatnění dvě symetrické šifry. Jedná se o proudovou šifru RC4 a blokovou šifru Rijndael, kterou využívá standard AES (4.2).

4.1 Proudová šifra RC4

Proudové šifry jsou často využívány v aplikacích, kde je kladen velký důraz na vysokou rychlost a malé zpoždění. Šifrovací algoritmus RC4 navrhl Američan Ronald Rivest v roce 1987 pro společnost RSA Data Security. Toto šifrování však obsahuje několik slabin, které jej umožnily prolomit. Existuje několik typů útoků, které se na tyto slabiny zaměřují a dokáží tuto šifru překonat, jedná se například o tzv. Tracking Attack. [8]

Tento způsob šifrování je implementován bezpečnostními standardy WEP a WPA.

4.2 AES

Jedná se o standard, který byl schválen americkou vládou jako naprosto bezpečný způsob šifrování dat. Pro vlastní šifrování byla ve veřejné soutěži vybrána v roce 2000 symetrická šifra Rijndael, kterou vynalezli Belgičané Joan Daemen a Vincent Rijmen. Standard AES je nástupcem staršího standardu DES (Data Encryption Standard), který byl v roce 1999 prolomen. [3]

Tento šifrovací standard je využit pro šifrování dat v bezpečnostním standardu WPA2.

Kapitola 5

Klasifikace útoků na bezdrátové sítě

Na bezdrátové sítě existuje již od počátku jejich vzniku velké množství nejrůznějších typů útoků. Podle způsobu provedení rozdělujeme útoky nejčastěji do dvou velkých skupin, a sice na pasivní a aktivní útoky.

5.1 Pasivní útoky

Při provádění pasivních útoků nedochází k žádné modifikaci provozu probíhajícího na cílové síti. Tyto útoky nejsou vedeny za účelem poškození či falšování přenášených dat. Jejich cílem může být odposlech citlivých dat, získání informací o síti, provozu či komunikujících klientech.

V případě nezabezpečené bezdrátové sítě má útočník absolutní přehled nad topologií sítě, připojenými klienty a všemi daty, které se v této síti přenáší. Situace je však komplikovanější pokud je pasivní útok prováděn na zabezpečenou bezdrátovou síť. Stále je sice možné monitorovat komunikaci mezi klienty avšak bez znalosti šifrovacího klíče nelze komunikaci dešifrovat a získat tak případná citlivá data.

Šifrovací klíč je však také možné získat cestou pasivních útoků a to u všech třech používaných zebezpečení.

V případě protokolu WEP stačí k prolomení a získání šifrovacího klíče odchytit dostatečné množství paketů obsahujících inicializační vektor a použít některý z volně šiřitelných programů, jako je například aplikace Aircrack-ng, podrobněji zmíněná v 3.1.

V případě použití zabecpečení WPA či WPA2 je nutné odchytit pakety přenášené během zahájení komunikace klienta a přístupového bodu. Jedná se zejména o čtyři konkrétní pakety tvořící takzvaný *čtyřcestný handshake*, během něhož je provedena autentizace obou zařízení a ustanovení šifrovacího klíče. Tyto pakety sice nejsou šifrované,sdílené heslo z nich však nelze získat, protože se mezi komunikujícími stranami nikdy nepřenáší. Pomocí informací z těchto paketů je ale možné provést slovníkový útok, který může být v případě slabého hesla úspěšný.

Pokud se útočníkovi touto cestou podaří získat šifrovací klíč, může monitorovat a odposlouchávat provoz i na zabezpečené síti, aniž by provedl jediný aktivní útok.

Pasivní útoky jsou obecně velmi nebezpečné. Je poměrně jednoduché je provést a získat pomocí nich veškerou komunikaci, která na síti probíhá. Zároveň je však téměř nemožné je odhalit.

5.1.1 Monitorování provozu

Monitorování provozu na sítí patří mezi typické pasivní útoky. Tento útok je možné provést v podstatě s libovolnou bezrátovou síťovou kartou. Síťovou kartu stačí programově přepnout do tzv. *promiskuitního* (monitorovacího) módu. V tomto režimu karta přijímá nejen jí určené pakety, ale i všechny ostatní pakety, které zachytí.

5.1.2 Analýza provozu a dat

Analýzu provozu je možné provádět v reálném čase nebo zpětně na zakladě odchycených paketů. Analýza provozu na síti slouží útočníkovi k získání obecnějších informací o síti, jako je například časová výtíženost sítě. Na základě těchto informací je útočník schopen vyhodnotit například to, v jaké době je nejvyšší pravděpodobnost odchycení citlivých dat.

Analýzou konkrétních datových paketů zachycených na síti se zabývá velké množství programů. Jedním z nejznámějších je například volně šiřitelný, multiplatformní program Wireshrak[1], dříve známý pod jménem Ethereal.

5.2 Aktivní útoky

Existuje velké množství aktivních útoků, vždy však spočívají v konkrétním zásahu do provozu na síti. Jejich provedení je opět snažší na nezabezpečených či slabě zabezpečených sítích. Sítě chráněné nejnovějšími bezpečnostními standardy jako je WPA a WPA2 jsou proti většině těchto útoků odolnější.

5.2.1 Odmítnutí služby

Tento útok se často označuje zkratkou DoS, která vychází z anglického názvu *Denial of Service* [9]. Při tomto útoku se útočník snaží zabránit klientům dané sítě v komunikaci. Tohoto cíle lze dosáhnout dvěma způsoby.

Jednak se může útočník pokusit zahltit přístupový bod generovanými pakety (útok typu replay) nebo jednodušše všem klientům na síti opakovaně rozesílá podvržené deautentizační pakety. Tyto pakety jsou v nezašifrované podobě, z tohoto důvodu lze tento typ útoku využít také na sítích se zabezpečením WPA či WPA2 i bez znalosti šifrovacího klíče.

5.2.2 Modifikace zpráv

V případě nezabezpečených bezdrátových sítí, kdy je komunikace nešifrovaná a autentizace otevřená, je pro útočníka velmi snadné modifikovat pakety a podvrhovat jejich odesílatele.

Modifikace zpráv se často uplatňuje v rámci útoku MITM (Man In The Middle attack), kdy je komunikace mezi klienty na síti v podstatě řízena útočníkem. Tohoto lze docílit takzvaným *otrávením ARP tabulky*, ve které je uchován překlad IP adres na MAC adresy zařízení v místní síti. Útočník rozešle podvržené ARP pakety klientovi a přístupovému bodu, pomocí kterých přesvědčí klienta, že IP adresa přístupového bodu odpovídá MAC adrese útočníka, a přístupový bod o tom, že IP adresa klienta odpovídá MAC adrese útočníka. Všechna komunikace tak proudí přes útočníka, ten může pakety přeposílat nepozměněné, modifikované, zahazovat je, či vytvářet a posílat vlastní podvržené pakety.

Mezi často podvrhované pakety patří například odpovědi DNS serverů. Tím je možné nic netušícího klienta přesměrovat v podstatě na libovolnou webovou adresu.

[1] http://www.wireshark.org/

Tomuto útoku lze předejít použitím některého standardu pro zabezpečení bezdrátové komunikace. Protokol WEP však používá velmi slabé zabezpečení integrity zpráv pomocí kontrolního součtu *CRC-32* (Cyclic Redundancy Check). Útočník je i bez znalosti šifrovacího klíče schopen změnit obsah paketu a vypočítat validní kontrolní součet. Bezpečnostní standardy WPA i WPA2 obsahují sofistikovanější mechanismy proti modifikaci zpráv. U WPA je integrita zpráv zajištěna kontrolním součtem, který je počítán algoritmem *Michael* nad celou nezašifrovanou zprávou, poté vložen do paketu a spolu se zprávou zašifrován. V případě WPA2 je kontrolní součet vypočítán algoritmem *CCM* [18].

5.2.3 Rogue AP

Mezi typické útoky na bezdrátové sítě patří vytvoření falešného přístupového bodu, tzv. *rogue AP*. Utočníkovi k tomu postačuje obyčejná bezdrátová síťová karta a vhodné programové vybavení. Bezdrátovou kartu je pak možné nastavit tak, aby se chovala jako přístupový bod na zvoleném kanálu a s nastaveným ESSIDem. Zároveň je možné přístupovému bodu nastavit libovolný typ zabezpečení.

Útočník tak může například vytvořit nezabezpečenou bezdrátovou síť s názvem "*free wifi hotspot*"a připojeným klientům přidělit IP adresu i nasdílet internetové připojení. Všechna data klientů však budou proudit přes útočníka, který je může libovolně měnit.

Další možností je využití rogue AP k prolomení šifrovacího klíče u protokolu WEP. Útočníkovi stačí vytvořit kopii zvoleného bezdrátového bodu. Ve skutečnosti stačí nastavit stejný ESSID a příznak zabezpečení protokolem WEP. Klienti se totiž při připojování k bezdrátové síti řídí hodnotou ESSIDu a ne MAC adresou přístupového bodu. Pokud se klient místo ke správné síti připojí k utočníkem vytvořené síti, je na něj zahájen útok. Tyto útoky jsou vystavěny na principu, který využívá slabinu v autentizaci v rámci protokolu WEP. Autentizace v tomto případě není žadná nebo se autentizuje pouze klient vůči přístupovému bodu. Mezi tyto útoky patří například *Hirte attack*[2] nebo *Caffe-Latte attack*[3]. Pro získání šifrovacího klíče k bezdrátové síti v tomto případě utočník vůbec nemusí být v dosahu přistupového bodu. Pro uspěšné provedení těchto útoku stačí být v dosahu klienta, který zná šifrovací klíč. Tato skutečnost je hlavním rozdílem od ostatních běžných útoků na protokol WEP.

V případě zabezpečení WPA/WPA2 může utočník pomocí rogue AP získat handshake klienta, který se k útočníkem vytvořené síti připojí. Informace z tohoho handshaku mohou být využity při slovníkovém útoku (viz 7.2).

[2] http://www.aircrack-ng.org/doku.php?id=hirte
[3] http://www.airtightnetworks.com/home/resources/knowledge-center/caffe-latte.html

Kapitola 6

Využití GPU

GPU, čip na grafické kartě, sloužil od počátku pouze k urychlení vykreslování obrazu na výstupním zobrazovacím zařízením počítače. Jeho smyslem bylo jen odlehčení zátěže procesoru. Tato skutečnost však přestává platit počátkem 21. století, kdy je v GPU objeven obrovský výpočetní potenciál také pro obecné typy výpočtů. V současné době se tato oblast stále rozvíjí a pro GPU se nachází různá uplatnění, například na poli fyzikálních simulací, bioinformatiky nebo zpracování obrazových a zvukových signálů.

Tato kapitola se zabývá architekturou grafického čipu, porovnáním jeho výpočetního výkonu s klasickým procesorem a technologiemi, které umožňují provádět obecné typy výpočtů na grafických kartách.

6.1 Architektura GPU

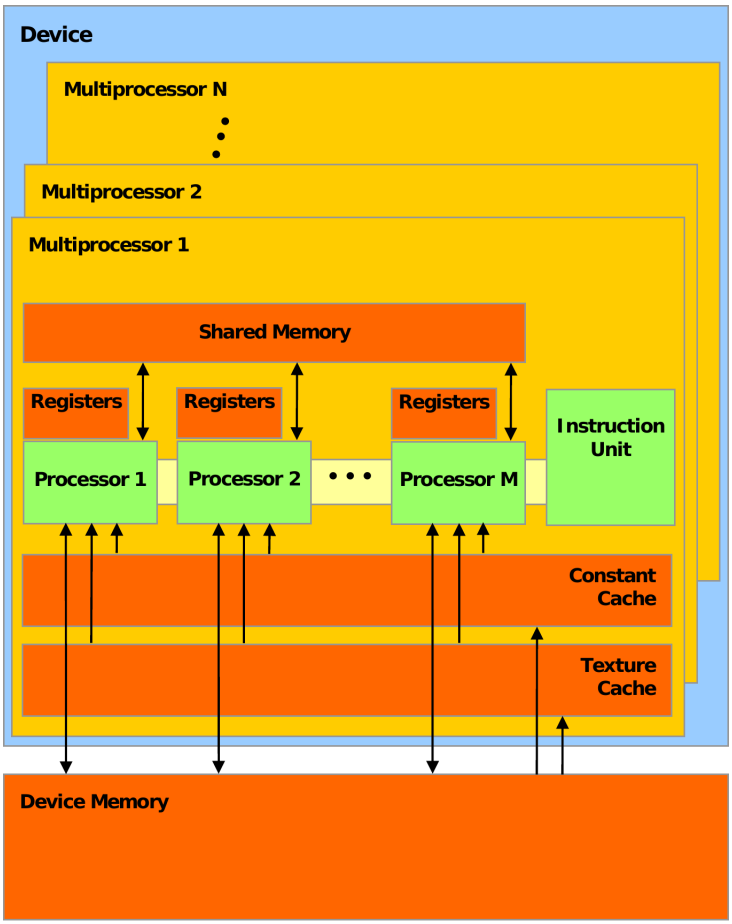
Grafický čip GPU se skládá obecně z velkého počtu výpočetních jader, které jsou po osmi uspořádány do multiprocesorů. Každý multiprocesor má k dispozici vlastní registry, lokální vyrovnávací paměť, paměť konstant a paměť pro textury. Grafická karta obsahuje ještě globální paměť, která je sdílená pro všechny multiprocesory. Schéma uspořádání jednotlivých částí GPU je znázorněno na obrázku 6.1.

Grafická karta nVidia GeForce GTX 285, která byla využita při vyvojí a testování aplikace v rámci této práce, obsahuje 240 výpočetních jader, které jsou rozděleny do 30 multiprocesorů. V každém multiprocesoru se nachází 16 384 registrů, vyrovnávací paměť o velikosti 16 kB a paměť pro konstanty o veliksti 64 kB. Grafická karta je vybavena 1 GB globální paměti typu GDDR3 pracující na frekvenci 1242 MHz.

Rychlost vypočetních jader u GPU je sice nižší než u CPU, jejich síla však spočívá v jejich počtu. Každý problém, který má být na GPU efektivně řešen, musí být nejprve dekomponován a upraven tak, aby jej bylo možné dobře paralelizovat. Pro dosažení maximálního zrychlení je totiž nutné, aby byly pokud možno všechny výpočetní jednotky neustále vytíženy. To znamená rozdělit běh programu na několik tisíc nezavislých vláken, které mohou běžet paralelně.

Jednotlivé multiprocesory jsou označovány jako jednotky SIMD (Single Instruction, Multiple Data). To znamená, že každý multiprocesor je schopen v jednom okamžiku, obsluhovat více vláken, aniž by mezi nimi musel přepínat. Nevýhodou tohoto hromadného zpracování je přítomnost pouze jednoho IP (Instruction Pointer) registru na každém multiprocesoru. To s sebou přináší omezení, že všechna vlákna zpracovávaná na jednom multiprocesoru musí vykonávat vždy stejnou instrukci.

V případě nVidia CUDA jsou jednotky SIMD nazývány pojmem *warp*. Maximální velikost warpu je na současných CUDA GPU 32 vláken. Všechna vlákna zpracovávaná na GPU jsou tedy rozdělena do skupin po 32 vláknech na jednotlivé warpy, které jsou vykonávány na multiprocesorech.



Obrázek 6.1: Schéma GPU [2]

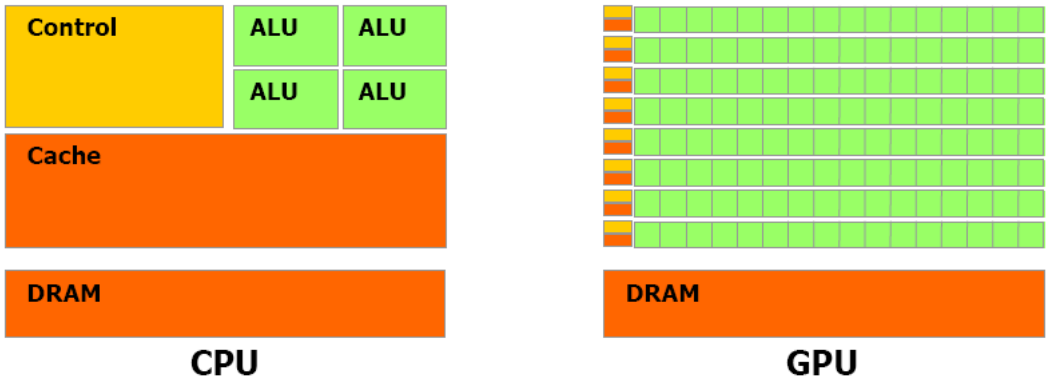
6.2 Porovnání s CPU

Na obrázku 6.2 je zjednodušené schéma demonstrující hlavní rozdíly mezi procesorem a grafickým čipem. Hlavní předností CPU je velmi rychlá a velká vyrovnávací paměť, vysoká výkonnost při vykonávání jednoho vlákna, podpora vstupně vystupních operací a také kvalitní predikce skoků. Naproti tomu spočívají hlavní přednosti GPU v obrovském počtu výpočetních jader a rychlých lokálních pamětech.

Současná GPU nabízejí obrovský výpočetní potenciál. Nejnovější grafický čip od společnosti nVidia označovaný kódovým názvem *fermi* obsahuje 480 vypočetních jader[1], což je oproti současným CPU až 100 násobně více. Pokud však implementací na GPU chceme dosáhnout znatelného zrychlení, musíme nejprve provést důkladnou analýzu implemento-

[1] http://www.nvidia.com/object/product_geforce_gtx_480_us.html

vaného problému a zvážit, jestli je daný problém dobře paralelizovatelný a tedy vhodný pro zpracování na GPU.



Obrázek 6.2: Porovnání CPU a GPU [2]

6.3 CUDA

S touto technologií přišel jeden z největších výrobců grafických čipů, společnost nVidia, v roce 2007. Technologie CUDA umožňuje provádění obecných výpočtů na grafické kartě. Její součástí je programovací jazyk označovaný jako *C for CUDA*. Jedná se o standardní jazyk C s rozšířeními od spolenosti nVidia umožňujícími efektivní a snadný vývoj aplikací pro GPU. Nevýhodou technologie CUDA je podpora pouze GPU od společnosti nVidia[2].

6.4 OpenCL

Standard OpenCL (Open Computing Language) byl vytvořen společností Apple pro multiplatformní paralelní programování na moderních procesorech. Jeho první stabilní verze byla vydána v roce 2008. V současné době tento standard spravuje a rozvíjí organizace Khronos[3], jeho nejnovější vydaná verze je 1.0.48 z října 2009. OpenCL obsahuje programovací jazyk založený na jazyce C (standard C99) s rozšířeními jako je možnost zápisu kernelu (funkce vykonávaná na GPU).

Při výběru technologie pro tvorbu aplikace v rámci této práce byla právě podpora více platforem rozhodující vlastností pro volbu OpenCL.

[2] http://www.nvidia.com/object/cuda_gpus.html
[3] http://www.khronos.org/opencl/

Kapitola 7

Zaměření a logika vyvíjené aplikace

Bezpečnostní protokol WEP již řadu let není považován za bezpečný. Existuje mnoho aplikací, které zneužívají bezpečnostních slabin protokolu WEP a umožňují i poměrně nezkušeným útočníkům získat šífrovací klíč k takto zabezpečené síti během několika minut.

Zabezpečení WPA s protokolem TKIP bylo provizorním řešením až do roku 2004, kdy jej nahradil standard 802.11i známý jako WPA2. Jak bylo zmíněno v 3.2, tak i případě WPA s protokolem TKIP již byly provedeny uspěšné útoky, které umožnily modifikaci přenášených zpráv bez znalosti šifrovacího klíče. Tyto útoky dokazují přítomnost bezpečnostních slabin v protokolu TKIP.

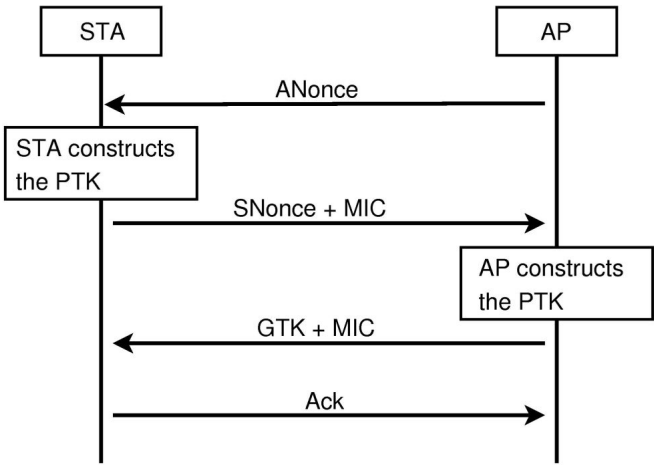
Z těchto důvodů je aplikace vyvíjená v rámci této práce zaměřena na bezpečnostní standard WPA2. Toto zabezpečení využívá protokol CCMP, který pro šifrování dat implementuje standard AES. Tento standard je uznán také americkou vládou jako bezpečný.

Cílem aplikace tedy není nalezení slabin a prolomení tohoto standardu, nýbrž maximální zefektivnění slovníkových útoků. Tohoto cíle se aplikace snaží dosáhnout využitím výpočetního potenciálu GPU.

7.1 Návrh aplikace

Celá aplikace by mohla být tvořena triviálním skriptem, který by každé heslo ze slovníku nastavil do parametrů připojení některého síťového klienta a ten by se postupně s každým heslem k síti zkusil připojit. Toto řešení by sice bylo velmi jednoduché avšak naprosto neefektivní. Testování by bylo jednak velmi pomalé a navíc by se útočník musel po celou dobu trvání útoku nacházet v dosahu dané sítě, což by značně zvyšovalo pravděpodobnost jeho odhalení. Přistupový bod by také mohl zaznamenávat neuspěšné pokusy o připojení a takového klienta po určitém počtu neuspěšných pokusů odmítat.

Mnohem efektivnějším řešením je získání údajů pro slovníkový útok z čtyřcestného ustanovení spojení (Four-Way Handshake viz obrázek 7.1) mezi klientem a přístupovým bodem. Jedná se o čtyři nezašifrované pakety, pomocí kterých proběhne autentizace klienta a přístupového bodu a zavedení šifrovacího klíče pro další komunikaci. Sdílené heslo v těchto paketech přeneseno není a to ani v zahashované podobě.



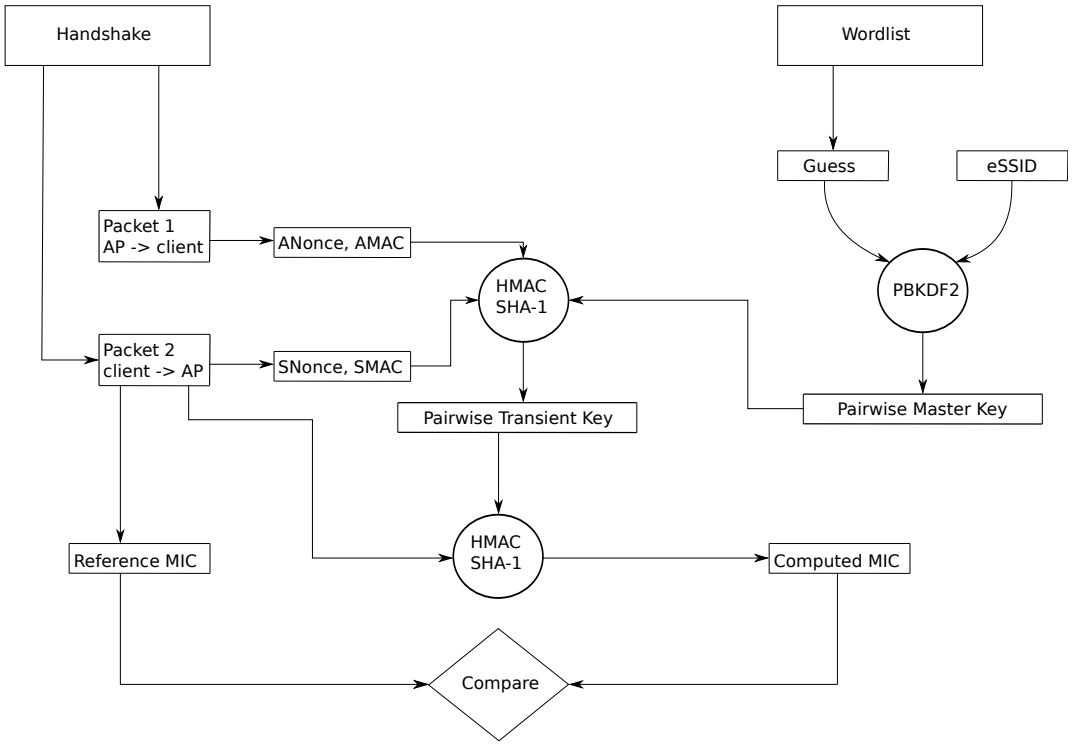
Obrázek 7.1: Four-Way Handshake

Klient (STA) nejprve na základě znalosti sdíleného hesla a identifikátoru sítě ESSID (Extended Service Set Identifier) vytvoří pomocí funkce PBKDF2 256 bitů dlouhý klíč PMK (Pairwise Master Key). Shodný klíč PMK vygeneruje také přístupový bod. Pomocí pseudonáhodné funkce, jejímž vstupem je aktuální čas a klíč PMK, vygeneruje přístupový bod 256 bitové pseudonáhodné číslo, tzv. *ANonce* (Authenticator's pseudo-random number), které zašle v prvním paketu čtyřcestného handshaku klientovi.

Po přijetí prvního paketu vygeneruje klient stejným algoritmem vlastní 256 bitové pseudonáhodné číslo, tzv. *SNonce* (Supplicant's pseudo-random number). Klient pomocí algoritmu HMAC-SHA-1, jehož vstupem je AMAC (MAC adresa přístupového bodu), SMAC (MAC adresa klienta), ANonce, SNonce a klíč PMK, vygeneruje 512 bitový klíč PTK (Pairwise Transient Key). Nakonec použije prvních 128 bitů tohoto klíče, označovaných jako KCK (Key Confirmation Key), pro vygenerování 128 bitového klíče pro kontrolu integrity zprávy, tzv. MIC (Message Integrity Check). Z pohledu vyvíjené aplikace jsou nejdůležitějšími informacemi ve druhém paketu čtyřcestného handshaku, který odesílá klient přístupovému bodu, právě hodnoty SNonce a MIC.

Všechny klíče jsou generovány pseudonáhodnými funkcemi, ze shodných vstupních dat tedy dostaneme vždy stejný klíč. Díky tomu je přístupový bod schopen vygenerovat stejný PTK a z něj následně MIC. Takto získaný MIC porovná s hodnoutou v poli MIC v paketu, který přijal od klienta. Pokud se hodnoty shodují, znamená to, že klient zná správné sdílené heslo. Tímto proběhla úspěšná autentizace klienta vůči přístupovému bodu.

V dalších dvou paketech čtyřcestného handshaku proběhne ještě autentizace přístupového bodu vůči klientovi a potvrzení zavedení šifrovacího klíče. Z pohledu vyvíjené aplikace však pro získání potřebných informací k provedení slovníkového útoku plně postačují první dva pakety čtyřcestného handshaku. Princip fungování aplikace je znázorněn na obrázku 7.2.



Obrázek 7.2: Princip fungování aplikace

7.2 Získání potřebných informací

K provedení slovníkového útoku potřebuje útočník znát ESSID dané sítě, mít k dispozici obsáhlý slovník s potenciálními hesly a odchytit alespoň první dva pakety čtyřcestného handshaku. První paket obsahuje jedinou důležitou informaci a tou je ANonce. Z druhého paketu pak útočník získá SNonce a referenční hodnotu MIC. MAC adresy klienta i přístupového bodu jsou obsaženy v obou paketech.

Pro získání handshaku musí být útočník vybaven bezdrátovou síťovou kartou přepnutou do monitorovacího režimu a softwarovým nástrojem pro zpracování zachycených dat. Mezi tyto nástroje patří například analyzátor síťového provozu Wireshark nebo program Airodump-ng. Tyto aplikace používají pro uložení odchycených paketů standardizovaný formát *pcap*, který je mezi síťovými aplikacemi velmi rozšířený.

Handshake může být útočníkem odchycen jen v době, kdy se klient připojuje k přístupovému bodu. Útočník má pak dvě varinanty, jak handshake získat. Pokud je klient k přístupovému bodu již připojen, může počkat dokud se klient neodpojí a znovu nepřipojí nebo dokud se neobjeví jiný klient. Tato varianta je pro útočníka z hlediska jeho utajení výhodnější, protože na sít neprovádí žádný aktivní útok. Na druhou stranu může být tato varianta velmi náročná z časového hlediska. Jako druhou alternativou se pro útočníka jeví donucení klienta k opětovné autentizaci využitím aktivního útoku, konkrétně deautentizací připojeného klienta.

7.3 Vytvoření kvalitního slovníku

Jelikož se jedná o slovníkový útok, odvíjí se od kvality slovníku úspěšnost celého útoku. V ideálním případě by slovník obsahoval všechny existující kombinace znaků, které lze při použití zabezpečení WPA nebo WPA2 v hesle použít. Heslo může mít v tomto případě délku 8 až 63 znaků. Znaky musí mít v tabulce ASCII hodnotu 32 až 126. Existuje tedy 95 různých znaků, které mohou být v hesle použity.

Pomocí variací s opakováním lze spočítat, že všech možných hesel o délce pouze 8 znaků, ve kterých by se vyskytovala jen malá písmena anglické abecedy spolu s číslicemi 0 až 9 by existovalo 36^8, což je zhruba $2.82 \cdot 10^{12}$ možností. Vytvoření programu pro vygenerování tohoto slovníku by nepředstavovalo žádné komplikace. První problém by se objevil při hledání místa pro jeho uložení. Takto rozsáhlý slovník by pro uložení na disk potřeboval 23 TB volného místa. Současné technologie by již umožnily uložení takto velkého slovníku v případě jeho rozdělení na několik částí a použití více externích disků. Otestování všech slov, které tento slovník obsahuje, by však v současné době trvalo v řádu několika let.

To jsme však brali v úvahu pouze hesla o délce 8, sestávající z 36 různých znaků. Celkový počet existujících hesel, tedy hesel délky 8 až 63 složených z 95 možných znaků, získáme součtem následujících variací s opakováním:

$$95^8 + 95^9 + 95^{10} + \cdots + 95^{60} + 95^{61} + 95^{62} + 95^{63} = 3.99 \cdot 10^{124}$$

Takto obrovský počet hesel nejsme a v dohledné budoucnosti nebudeme schopni uložit na žádné dostupné medium. I kdybychom tuto překážku odstranili tím, že bychom vygenerovaná hesla neukládali a rovnou testovali, zabere nám jejich vyzkoušení při rychlosti 1400[1] hesel za vteřinu zhruba $9.04 \cdot 10^{113}$ let.

Vzhledem k uvedeným problémům nelze slovník výtvářet na základě generování hesel ze všech platných znaků. Na druhou stranu však můžeme předpokládat, že většina bezdrátových sítí, pokud má být jejich heslo pro člověka zapamatovatelné, nebude chráněna hesly delšími než 15 znaků. Znaky v těchto heslech navíc s velkou pravděpodobností nebudou náhodné a budou představovat existující slova. Především méně zkušení uživatelé často používají hesla jako jsou například, rodná čísla, telefonní čísla, jména vlastní, partnerů, dětí nebo domacích zvířat. Pokud je jméno kratší než minimální přípustná délka hesla, doplní ji takový uživatel nejčastěji inkrementujícími se číslicemi nebo rokem narození.

Kvalitní slovník by měl obsahovat řádově desítky až stovky milionů smysluplných hesel. Takto velký slovník pro útočníka představuje ideální poměr mezi testovacím časem, potřebnou pamětí a pravděpodobností nalezení správného hesla.

[1] Rychlost testování programem Aircrack-ng na počítači s procesorem Intel Core 2 Duo 2.8 GHz.

Kapitola 8

Popis implementace

Implementace vyvíjené aplikace byla provedena v jazyce C pod operačním systémem Linux. První implementace byla za účelem ladění a pozdější možnosti porovnání výkonnosti, vytvořena pro CPU. Po dokončení fází optimalizace a testování byla část aplikace za účelem dosažení vyšší rychlosti přepsána do OpenCL tak, aby mohl výpočet probíhat na GPU.

8.1 Použité algoritmy

Aplikace pro svoji činnost využívá několik funkcí, které implementují standardizované kryptografické a hashovací algoritmy.

8.1.1 PBKDF2

Tato kryptografická funkce pro vytváření klíčů je specifikovaná ve standardu PKCS #5 [7]. Základ této funkce vytváří pseudonáhodná hashovací funkce a její mnohonásobná iterace. Standard doporučuje 1000 iterací jako minimální počet. V případě WPA při generování klíče PMK je jako pseudonáhodná hashovací funkce použita HMAC-SHA-1 s počtem iterací 4096.

Takto vysoký počet iterací je použit z důvodu zesílení klíče právě proti slovníkovým útokům. Generování klíčů tak útočníkovi zabere více času a celý útok je výrazně pomalejší.

8.1.2 HMAC

Jedná se o kryptografickou funkci pro generování klíčů, která pro svou činnost potřebuje funkci pro generování pseudonáhodných čísel. K tomuto účelu využívá hashovací funkci SHA-1 nebo MD5. Při generování klíče PMK je využita funkce HMAC-SHA-1. Tato funkce je použita také při počítání hodnoty MIC v čtyřcestném handshaku u WPA2-CCMP. Pokud se jedná o WPA-TKIP je hodnota MIC počítána pomocí HMAC-MD5.

8.1.3 SHA-1

Hashovací funkce SHA-1 vznikla v roce 1995 jako náhrada hashovací funkce SHA-0, u které byly objeveny kolize[1]. I přesto, že již existuje novější funkce SHA-2 a pracuje se na funkci SHA-3, je tato funkce velmi rozšířena v mnoha aplikacích různých typů. Kromě zabezpečení bezdrátových sítí našla funkce SHA-1 uplatnění také v protokolech a aplikacích jako jsou

[1] vyprodukování stejných hashů pro různá data

SSH, TLS, SSL, IPsec nebo v systému digitálních podpisů. Tato funkce pracuje nad libovolnými daty menšími než 2^{64} bitů. Jejím výsledkem je vždy 160 bitový hash.[4] Popisem útoků na tuto funkci a hledáním možných kolízí se podrobně zabývá publikace [17].

8.1.4 MD5

Hashovací algoritmus MD5 publikoval v roce 1992 Američan Ronald Rivest. Jedná se o nástupce hashovací funkce MD4 z roku 1991. Funkce MD5 se uplatnila v mnoha bezpečnostních protokolech. Velmi často se využívá k uložení hesel (například v systémech Unix) nebo k vytvoření kontrolního součtu souborů. [12]

Již v roce 1996 demonstroval německý kryptograf Hans Dobbertin, že je algoritmus MD5 schopen vyprodukovat identické hashe pro dvě různé zprávy v případě, že inicializační vektory mohou být libovolně zvoleny. V roce 2004 byl publikován způsob, jak během přibližně jedné hodiny vygenerovat dva různé řetězce se stejným výsledným hashem i při použití standardních inicializačních vektorů. [16] Z tohoto důvodu se v současné době od použití hashovací funkce MD5 v bezpečnostních aplikacích upouští.

8.2 Struktura aplikace

Celá aplikace sestává ze pěti hlavních částí:

1. zpracování vstupních dat
2. vytvoření PMK z hesla ve slovníku a ESSIDu sítě funkcí `pbkdf2()`
3. získání PTK z PMK a z hodnot ANonce, SNonce, AMAC a SMAC funkcí `hmac()`
4. výpočet MIC z PTK a z hodnot všech bajtů 2. paketu handshaku funkcí `hmac()`
5. porovnání vypočítané hodnoty MIC s hodnoutou MIC ve 2. paketu handshaku

Část programu reprezentovaná body 2 až 5 se v cyklu opakuje pokaždé s novým heslem ze slovníku a to až do doby, dokud není v bodě 5 nalezena shoda mezi referenční a vypočítanou hodnotou MIC nebo do vyčerpání slovníku.

8.3 Profilace aplikace

Největší výhodou využití GPU oproti CPU je nepochybně možnost paralelního zpracování řádově tisíců vláken. Rychlost zpracování na GPU může být vyšší než na CPU pouze tehdy, obsahuje-li program dostatečné množství paralelizovatelných výpočtů. V takovém případě můžeme podle [13] dosáhnout až několikanásobného zrychlení i oproti plně optimalizované implementaci pro CPU. Přesná hodnota zrychlení závisí na povaze řešeného problému a hardwarovém vybavení.

Pomocí GPU se tedy vyplatí počítat pouze paralelizovatelné části programu, jejichž v podstatě sériové zpracování zabere implementaci pro CPU nejvíce času. Z tohoto důvodu byla vyvíjená aplikace nejprve implementována pro CPU v jazyce C a analyzována pomocí profilovacího programu *gprof*. Analýza proběhla na počítači s procesorem Intel Core 2 Duo s frekvencí jádra 2.8 GHz pod operačním systémem Linux. Testovací slovník obsahoval 10 000 hesel, přičemž až poslední heslo bylo korektní.

Výsledky analýzy znázorněné v tabulce 8.1 ukázaly, že program tráví přibližně 99,06 % času v jediné funkci. Jedná se o funkci `pbkdf2()`, která pro každé heslo ze slovníku vypočte klíč PMK. Pro výpočet tohoto klíče využívá kryptografickou funkci HMAC a hashovací funkci SHA-1.

Funkce `pbkdf2()` pro svou činnost sice potřebuje funkci HMAC-SHA-1 avšak jejich obecnou implementaci ve funkcích `hmac()` a `getSHA1()` nevolá. Za účelem maximálního urychlení aplikace, byl kód těchto funkcí upraven a optimalizován. Negativním následkem těchto úprav byla ztráta obecnosti funkcí `hmac()` a `getSHA1()`. Tyto funkce jsou v aplikaci potřebné nejen pro výpočet klíče PMK funkcí `pbkdf2()`, ale také pro výpočet klíče PTK a hodnoty MIC. Z tohoto důvodu nemohla optimalizovaná avšak méně obecná implementace nahradit původní implementaci těchto funkcí. Optimalizovaný kód byl proto vložen přímo do funkce `pbkdf2()`. Touto úpravou bylo dosaženo téměř trojnásobného zrychlení.

Tabulka 8.1: Výsledek profilace aplikace programem gprof

čas v %	kumulativní čas v *s*	vlastní čas v *s*	počet volání	název funkce
99.06	46.32	46.32	10 000	`pbkdf2()`
0.75	46.67	0.35	40 000	`getSHA1()`
0.19	46.76	0.09	20 000	`hmac()`
0.00	46.76	0.00	10 005	`str2bin()`
0.00	46.76	0.00	3	`hexToByte()`
0.00	46.76	0.00	1	`bin2str()`

8.4 Transformace do OpenCL

Výsledky profilace programem *gprof* tedy jednoznačně určují, že jediným místem v programu, ve kterém je možno docílit markantního zrychlení je právě funkce `pbkdf2()`.

Při každém volání této funkce proběhne vypočet hashe SHA-1 16 386krát. Přesněji 8 192krát proběhne hashování ESSIDu pro získání prvních 160 bitů klíče PMK, 8 192krát proběhne hashování ESSIDu pro získání zbývajících 96 bitů klíče PMK a 2krát hashování hesla při inicializaci výpočtu.

Počet iterací hashovací funkce při každém volání funkce `pbkdf2()` se nabízí jako ideální místo pro paralelizaci a zpracování na GPU. Jednotlivé iterace cyklu, během kterého hashování probíhá však nejsou nezávislé. Každý další hash je počítán z hashe získaného v předchozí iteraci. Vzhledem k této skutečnosti není vícevláknové paralelní zpracování v tomto případě možné.

Jako efektivní řešení tohoto problému se ukázalo paralelní zpracování celé funkce `pbkdf2()` pro určitý blok hesel ze slovníku. To znamená, že nejsou paralelizované jednotlivé operace při vypočtu každého klíče PMK, nýbrž je souběžně počítán větší počet klíčů PMK. Tento počet je dán konstantou `PASSWORD_COUNT`, která tak určuje počet paralelně zpracovávaných vláken programu a přímo tedy ovlivňuje celkovou rychlost aplikace. S rostoucí hodnotou této konstanty roste také rychlost celé aplikace. Zároveň se však zvyšuje paměťová náročnost aplikace na grafickou kartu. Kromě velikosti paměti má grafický čip omezení také v maximálním počtu aktivních warpů na multiprocesor a tedy i v maximálním počtu aktivních vláken.

Z tohoto důvodu je nutné hodnotu konstanty `PASSWORD_COUNT` optimálně zvolit podle parametrů grafické karty, na které aplikace poběží. Pokud je konstanta příliš vysoká, skončí běh aplikace s chybou uvedenou ve výpisu 8.1.

Výpis 8.1: Chybové hlášení

```
$ ./guwap input wordlist
CL_OUT_OF_RESOURCES
Error in clEnqueueNDRangeKernel, Line 576 in file oclGUWAP.cpp
```

Konstanta `PASSWORD_COUNT` je implicitně nastavena na hodnotu 15 360, což je ideální hodnota pro grafickou kartu nVidia GeForce GTX 285, na které probíhalo testování. Tato hodnota byla z velké části zjištěna experimentálně na základě testování (viz 9.1). Teoreticky by se měla rovnat maximálnímu počtu aktivních warpů na multiprocesor vynásobeného počtem vláken v jednom warpu a celkovým počtem multiprocesorů. V případě testované grafické karty by to bylo $32 \cdot 32 \cdot 30$, což odpovídá hodnotě 30 720. Avšak již okolo hodnoty 17 000 dochází k chybě uvedené ve výpisu 8.1. Hodnota získaná experimentálně byla zaokrouhlena na 15 360 proto, aby mohla být vlákna beze zbytku rozdělena po 32 vláknech na 16 aktivních warpů na každý multiprocesor.

8.5 MultiGPU režim

Za účelem dosažení ještě většího zrychlení byla finální verze aplikace upravena tak, aby mohla souběžně pracovat na všech dostupných GPU v systému. Počet dostupných GPU je zjištěn funkcí `clGetDeviceIDs()`. Následně jsou na všech GPU naalokovány potřebné zdroje a konstanta `PASSWORD_COUNT` je vynásobena počtem dostupných GPU. Na každém GPU je tedy zpracováván blok hesel, který odpovídá nastavené hodnotě konstanty `PASSWORD_COUNT`. Maximální počet využitých GPU může být upraven nastavením příslušné hodnoty konstantě `MAX_GPU_COUNT`.

Výsledky testování na dvou GPU jsou uvedeny v části 9.2.

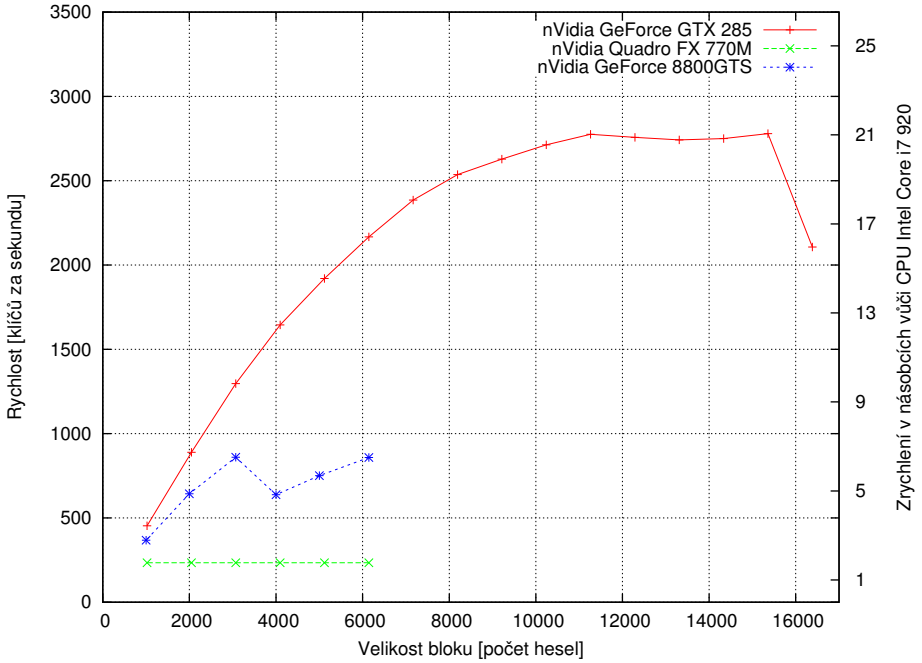
Kapitola 9

Interpretace výsledků

Obě implementace aplikace, tedy verze pro CPU a verze pro GPU, byly testovány na počítači s procesorem Intel Core i7 920 2.67 GHz a grafickou kartou nVidia GeForce GTX 285.

9.1 Porovnání výsledků s výsledky na CPU

Z grafu 9.1 je patrný rozdíl ve výpočetním výkonu mezi CPU[1] a GPU. Takto vysokého zrychlení bylo dosaženo díky paralelizaci výpočtu na grafické kartě. Pro porovnání obsahuje graf také výsledky testování na grafických kartách nVidia GeForce 8800GTS a nVidia Quadro FX 770M.



Obrázek 9.1: Porovnání výkonu CPU a GPU v závislosti na velikosti bloku hesel

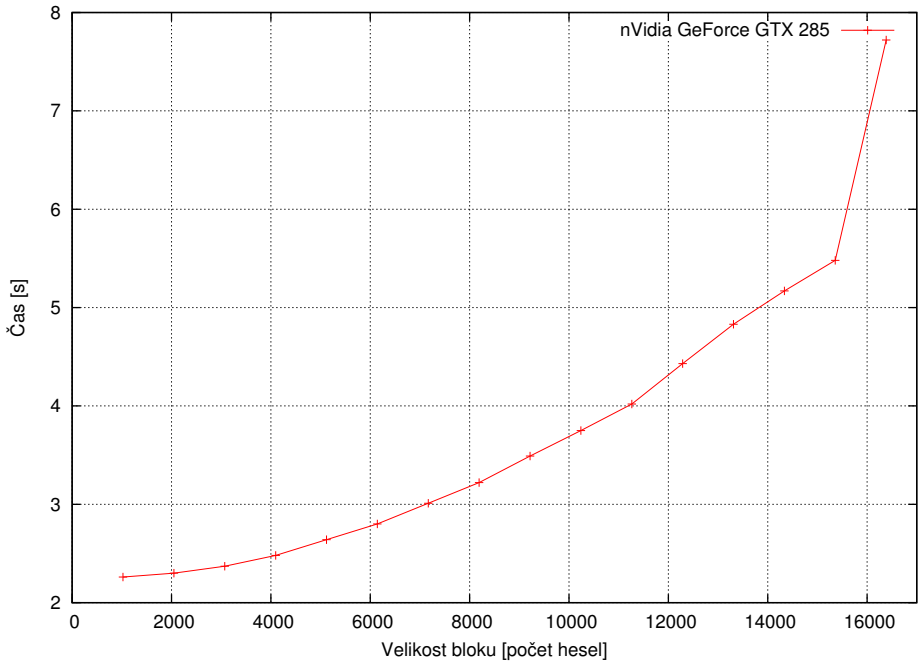
[1] jedná se o jednovláknovou implemntaci pro CPU

Naprosto odlišný průběh karty nVidia Quadro FX 770M je způsoben jejími rozdílnými parametry. Oproti grafické kartě nVidia GeForce GTX 285 vybavené 240 výpočetními jádry, které pracují na frekvenci 1476 MHz, obsahuje grafická karta nVidia Quadro FX 770M pouze 32 výpočetních jader pracujících na frekvenci 1250 MHz. Grafická karta nVidia GeForce 8800GTS obsahuje 96 výpočetních jader na frekvenci 1200 MHz.

Aby byl výkonnostní průběh u grafické karty nVidia Quadro FX 770M lépe čitelný, museli bychom vzhledem k nízkému počtu výpočetních jader snížit rozmezí bloku načítaných hesel. (viz příloha A)

Důkladným testováním na velkém vzorku dat[2] bylo zjištěno, že při nastavení optimální velikosti bloku načítaných hesel[3] trvá jedna iterace zhruba 5.53 vteřin. Přičemž výpočet na GPU zaujímá 98.5 % tohoto času. Úloha procesoru činí přibližně jen 80 ms během každé iterace, zbytek času tráví CPU čekáním na dokončení výpočtu na grafické kartě. Tento čas by sice bylo možné využít například pro načtení dalšího bloku hesel ze slovníku avšak vzhledem ke skutečnosti, že činnost procesoru během jedné iterace trvá včetně načtení bloku hesel pouhých 80 ms, je tato časová úspora bezvýznamná.

Časová náročnost jednotlivých iterací narůstá pouze se zvyšováním hodnoty konstanty `PASSWORD_COUNT`, která určuje velikost bloku načítaných hesel. Tato závislost je znázorněna v grafu 9.2.



Obrázek 9.2: Délka iterace v závislosti na velikosti bloku hesel

Z hlediska paměťové náročnosti se aplikace chová stejně jako v případě časové náročnosti. Odvíjí se tedy opět od velikosti bloku načítaných hesel. Při ponechání implicitního nastavení velikosti bloku načítaných hesel, je ze souboru se slovníkem, popřípadě ze standardního vstupu, načteno 15 360 hesel, reprezentovaných textovými řětězci délky 8 až 63 znaků. Hesla jsou následně převedena do binární podoby, kdy je každé heslo reprezentova-

[2] slovníky od 10 000 až po 1 000 000 hesel

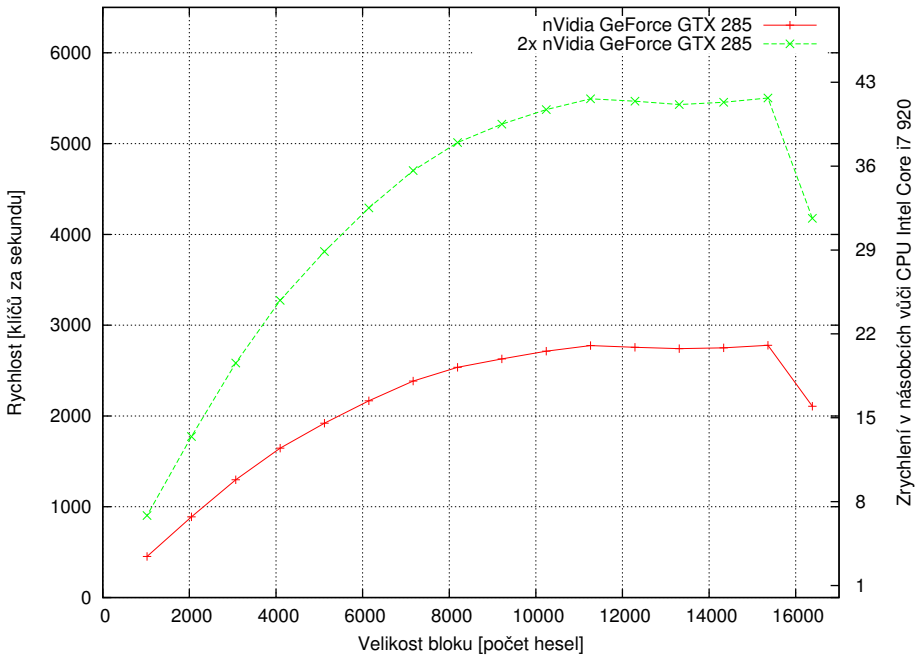
[3] pro kartu nVidia GeForce GTX 285 je to přibližně 15 360

náno polem 16 čtyřbajtových integerů. Pro uložení binárních hesel v rámci jedné iterace potřebujeme v tomto případě $15\,360 * 16 * 4$ bajtů, což je 960 kB.

Kapacita vyrovnávací paměti na současných CPU se pohybuje v řádu několika MB. CPU má tedy k těmto datům přístup během několika nanosekund. Vyrovnávací paměť na grafickém čipu má však kapacitu pouze 16 kB. Z tohoto důvodu musí být data uložena v globální paměti. Přístup do globální paměti zabere 400 až 600 taktů jádra[4], což při frekvenci jádra okolo 1.4 GHz představuje 300 až 400 ns. V porovnání s přístupovou dobou do vyrovnávací pamětí procesoru trvá přístup do globální paměťi grafické karty až stokrát déle.

9.2 Testování multiGPU režimu

Tento režim byl otestován na dvou grafických kartách nVidia GeForce GTX 285. Porovnání výsledků s verzí pro jedno GPU se nachází v grafu 9.3. Bylo dosaženo rychlosti okolo 5500 hesel za vteřinu, což představuje 1.98 násobné zrychlení oproti verzi s jedním GPU. Zrychlení není přesně dvojnásobné z důvodu zvýšené režie při použití více GPU.



Obrázek 9.3: Srovnání výkonu při multiGPU režimu

Při čtení dat z grafických karet bylo nejdříve využito asynchronní čtení. Synchronizace byla prováděna na základě údálostí (*events*) pomocí funkce `clWaitForEvents()`. Tato knihovní funkce však během čekání na jednotlivé události neuspává vlákno procesoru, který je tak plně vytížen aktivním čekáním. Z tohoto důvodu je ve finální verzi implementováno synchronní čtení a vlákno aplikace běžící na CPU je během provádění výpočtu na GPU uspáno funkcí `usleep()`. Vyhneme se tak aktivnímu čekání a získáme procesorový čas pro jiné aplikace. Tento čas může být například využit pro generování hesel jinou aplikací v režimu načítání hesel ze standardního vstupu.

[4] http://www.eecg.toronto.edu/~moshovos/CUDA08/arx/microbenchmark_report.pdf

Kapitola 10

Závěr

Aplikace vyvinutá v rámci této práce prezentuje jednoduchou implementaci slovníkového útoku na bezdrátové sítě zabezpečené standardy WPA a WPA2. Její implementací pro GPU[1] bylo oproti verzi pro CPU[2] dosaženo 21 násobného zrychlení, což představuje velmi výraznou akceleraci slovníkového útoku.

Vzhledem k počtu hesel, které útočník potřebuje otestovat, se jedná o dobře paralelizovatelný problém. Implementace pro čip GPU, který se vyznačuje právě schopností paralelního zpracování velkého množství vláken, je v tomto případě tedy jednoznačně efektivnější než zpracování na CPU.

Při rychlosti testování okolo 5500 hesel za vteřinu, což je rychlost dosažená s konfigurací dvakrát nVidia GeForce GTX 285, lze i velmi rozsáhlé slovníky otestovat v rozumném čase. S použitím tohoto systému jsme schopni otestovat až 475 milionů hesel za den, což mnohonásobně převyšuje počet slov českého jazyka[3] a naprosto deklasuje použití slabých hesel. Získání přístupu do sítí chráněných bežnými hesly vyskytujícími se ve slovnících je pak již téměř jisté.

Cílem vývoje této aplikace bylo zejména upozornit na skutečnost, že prolomení slabého hesla nezabrání ani použití nejnovějších bezpečnostních standardů.

Tato aplikace spolu se všemi ostatními implementacemi slovníkových útoků na zabezpečení WPA a WPA2 bude mít uplatnění dokud všechny bezdrátové sítě nebudou zabezpečeny opravdu komplexními a netriviálními hesly nebo dokud nebudou tyto bezpečnostní standardy nahrazeny novými, z důvodu jejich prolomení neslovníkovým útokem. Druhá varianta je jako závěr vzhledem k nepoučitelnosti většiny uživatelů mnohem pravděpodobnější.

[1] nVidia GeForce GTX 285
[2] Intel Core i7 920 2.67 GHz
[3] http://www.ujc.cas.cz/poradna/porfaq.htm

Literatura

[1] Cordeiro, C.; Abhyankar, S.; Agrawal, D. P.: Scalable and QoS-Aware Dynamic Slot Assignment and Piconet Partitioning to Enhance the Performance of Bluetooth Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, ročník 5, č. 10, 2006: s. 1313–1330, ISSN 1536-1233, http://dx.doi.org/10.1109/TMC.2006.156.

[2] Corporation, N.: OpenCL Programming Guide for the CUDA Architecture. 2010, http://developer.download.nvidia.com/compute/cuda/3_0/toolkit/docs/NVIDIA_OpenCL_ProgrammingGuide.pdf.

[3] Daemen, J.; Rijmen, V.: *The Design of Rijndael*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2002, ISBN 3540425802.

[4] Eastlake, D., 3rd; Jones, P.: US Secure Hash Algorithm 1 (SHA1). 2001.

[5] Gast, M. S.: *802.11 Wireless Networks: The Definitive Guide*. Sebastopol, CA, USA: O'Reilly & Associates, Inc., 2002, ISBN 0596001835.

[6] Ghosh, A.; Wolter, D.; Andrews, J.; aj.: Broadband wireless access with WiMax/802.16: current performance benchmarks and future potential. *Communications Magazine, IEEE*, ročník 43, č. 2, Feb. 2005: s. 129–136, ISSN 0163-6804, http://dx.doi.org/10.1109/MCOM.2005.1391513.

[7] Kaliski, B.: PKCS #5: Password-Based Cryptography Specification Version 2.0. 2000.

[8] Mister, S.; Tavares, S. E.: Cryptanalysis of RC4-like Ciphers. In *SAC '98: Proceedings of the Selected Areas in Cryptography*, London, UK: Springer-Verlag, 1999, ISBN 3-540-65894-7, s. 131–143.

[9] Needham, R. M.: Denial of service. In *CCS '93: Proceedings of the 1st ACM conference on Computer and communications security*, New York, NY, USA: ACM, 1993, ISBN 0-89791-629-8, s. 151–153, http://doi.acm.org/10.1145/168588.168607.

[10] Ohigashi, T.; Morii, M.: An Implementation and Evaluation for a Message Falsification Attack on WPA. 2009, http://jwis2009.nsysu.edu.tw/location/paper/A%20Practical%20Message%20Falsification%20Attack%20on%20WPA.pdf.

[11] Pužmanová, R.: *Bezpečnost bezdrátové komunikace: Jak zabezpečit Wi-Fi, Bluetooth, GPRS či 3G*. Brno: Computer Press, 2005, ISBN 80-251-0791-4.

[12] Rivest, R.: The MD5 Message-Digest Algorithm. 1992.

[13] Ryoo, S.; Rodrigues, C. I.; Baghsorkhi, S. S.; aj.: Optimization principles and application performance evaluation of a multithreaded GPU using CUDA. In *PPoPP '08: Proceedings of the 13th ACM SIGPLAN Symposium on Principles and practice of parallel programming*, New York, NY, USA: ACM, 2008, ISBN 978-1-59593-795-7, s. 73–82, http://doi.acm.org/10.1145/1345206.1345220.

[14] Tews, E.; Beck, M.: Practical attacks against WEP and WPA. In *WiSec '09: Proceedings of the second ACM conference on Wireless network security*, New York, NY, USA: ACM, 2009, ISBN 978-1-60558-460-7, s. 79–86, http://doi.acm.org/10.1145/1514274.1514286.

[15] Thom-Santelli, J.; Ainslie, A.; Gay, G.: Location, location, location: a study of bluejacking practices. In *CHI '07: CHI '07 extended abstracts on Human factors in computing systems*, New York, NY, USA: ACM, 2007, ISBN 978-1-59593-642-4, s. 2693–2698, http://doi.acm.org/10.1145/1240866.1241064.

[16] Thompson, E.: MD5 collisions and the impact on computer forensics. *Digital Investigation*, ročník 2, č. 1, 2005: s. 36 – 40, ISSN 1742-2876, doi:DOI:10.1016/j.diin.2005.01.004, http://www.sciencedirect.com/science?_ob=MImg&_imagekey=B7CW4-4FM544N-1-3&_cdi=18096&_user=640830&_pii=S1742287605000058&_orig=na&_coverDate=02%2F28%2F2005&_sk=999979998&view=c&wchp=dGLbVzW-zSkWA&md5=5c1788b3d0e6c9dc154a39ee259afd46&ie=/sdarticle.pdf.

[17] Wang, X.; Yin, Y.; Yu, H.: Finding Collisions in the Full SHA-1. In *Advances in Cryptology – CRYPTO 2005*, Springer Berlin / Heidelberg, 2005, ISBN 978-3-540-28114-6, s. 17–36.

[18] Whiting, D.; Housley, R.; Ferguson, N.: Counter with CBC-MAC (CCM). 2003.

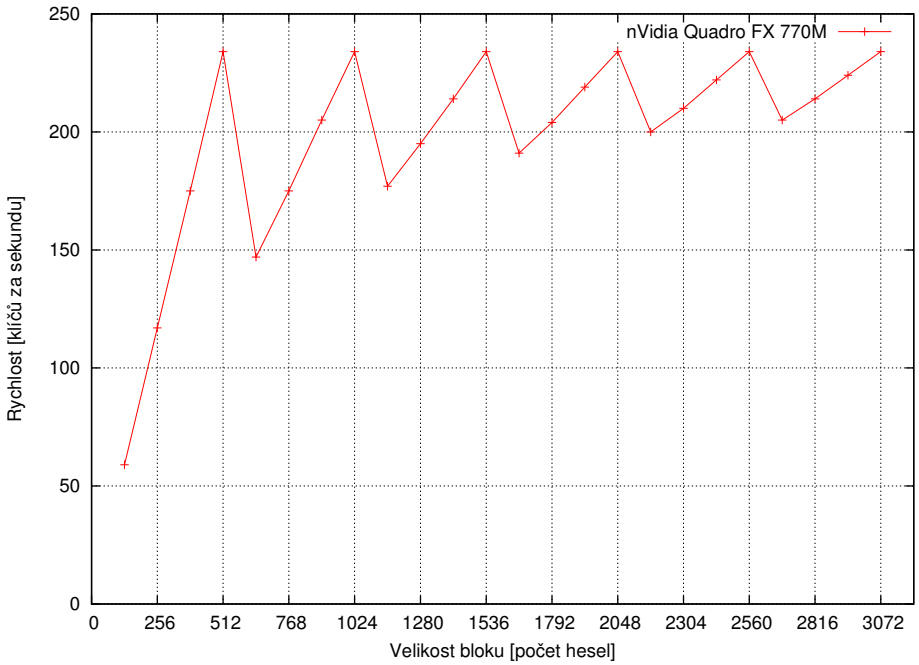
[19] Wi-Fi Alliance: Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks. 2003, http://www.ans-vb.com/Docs/Whitepaper_Wi-Fi_Security4-29-03.pdf.

[20] Williams, S.: IrDA: past, present and future. *Personal Communications, IEEE*, ročník 7, č. 1, Feb 2000: s. 11–19, ISSN 1070-9916, http://dx.doi.org/10.1109/98.824566.

Příloha A

Grafy výsledků pro GK nVidia Quadro FX 770M

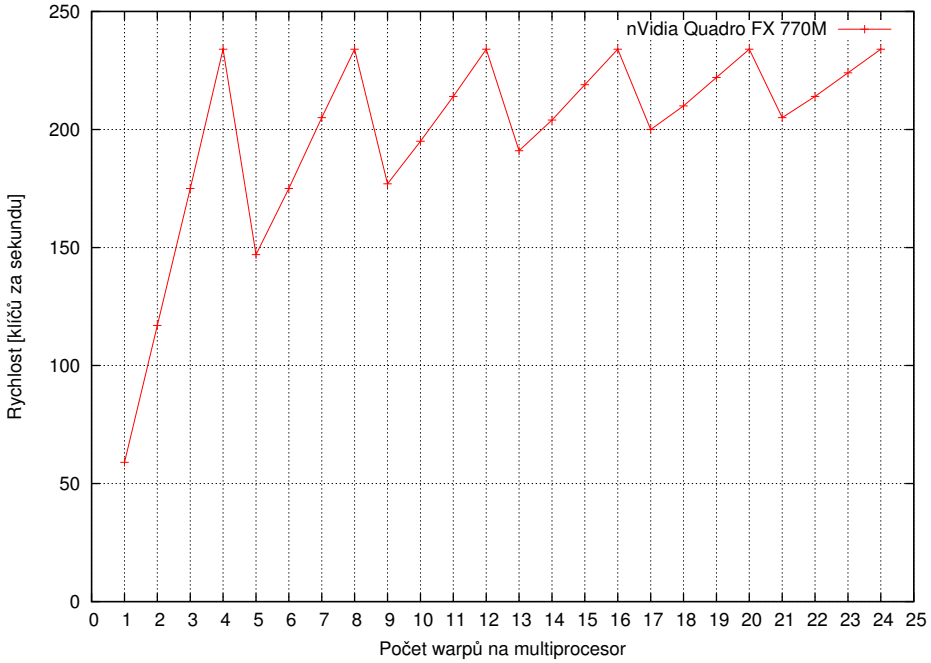
Podle specifikace společnosti nVidia může být na této kartě maximálně 24 aktivních warpů na multiprocesor. To znaméná celkem 3072[1] souběžně zpracovávaných vláken. Graf A.1 znázorňuje závislost mezi počtem vláken a rychlostí výpočtu. V grafu je možné nalézt určitou periodu, se kterou se opakují minima a maxima.



Obrázek A.1: Výkonnostní analýza GPU v závislosti na počtu vláken

[1] 32 vláken ve warpu a celkem 4 multiprocesory

Pokud přepočítáme počet vláken na počet warpů (graf A.2), zjistíme že maxima jsou jen v bodech, kdy je počet warpů na jeden multiprocesor dělitelný čtyřmi nebo osmi beze zbytku. Toto číslo souvisí s počtem výpočetních jader v jednom multiprocesoru, kterých se zde nachází právě osm.



Obrázek A.2: Výkonnostní analýza GPU v závislosti na počtu warpů

Příloha B

Obsah přiloženého CD

`/bin/CPU/`	zkompilovaná verze aplikace pro CPU
`/GPU/`	zkompilovaná verze aplikace pro GPU
`/doc/tex/`	zdrojové soubory textu bakalářské práce
`/manual.txt`	popis zprovoznění a ovládání aplikace
`/projekt.pdf`	text bakalářské práce
`/input/input-TKIP`	ukázkový vstupní soubor 1
`/input-CCMP`	ukázkový vstupní soubor 2
`/template`	předloha vstupního souboru
`/src/CPU/`	zdrojové soubory verze pro CPU
`/GPU/`	zdrojové soubory verze pro GPU
`/wordlists/`	testovací slovníky

Příloha C

Zprovoznění a ovládání aplikace

Pro zprovoznění aplikace je nutné mít nainstalovaný ovladač grafické karty, umožňující provádění GPGPU výpočtů včetně příslušného kompilátoru. Aplikace byla testována s ovladači nVidia 190.29 a 195.36.15 a s CUDA toolkitem obsahujícím OpenCL verze 1.0.

Zdrojové kódy jsou vybaveny Makefilem, pro jejich překlad tedy stačí zadat příkaz `make`. Aplikace očekává dva parametry, prvním je textový soubor se vstupními daty a druhým je slovník obsahující hesla k testování. Formát souboru se vstupními daty je dán předlohou v souboru *template*. Konkrétní podobu tohoto souboru prezentují soubory *input-TKIP* a *input-CCMP*. Pokud druhý parametr chybí, jsou hesla načítína ze standardního vstupu. Aplikace vypíše nápovědu a ukončí se, pokud je spuštěna se špatným počtem parametrů nebo s parametrem `-h`, popřípadě `--help`.

Chování aplikace lze upravit změnou hodnot některých konstant v souboru *guwap.c*. Tyto konstanty a jejich implicitní hodnoty nastavené pro grafickou kartu nVidia GeForce GTX 285 jsou uvedeny v tabulce C.1.

Tabulka C.1: Konstanty ovliviňující chování aplikace

název konstanty	implicitní hodnota	popis konstanty
`PASSWORD_COUNT`	15 360	velikost bloku načítaných hesel
`SLEEP_TIME_USEC`	5 300 000	doba uspání vlákna CPU při čekání na GPU (v μs)
`MAX_GPU_COUNT`	0	počet GPU, které smí aplikace využít (0 = všechny)
`VERBOSE`	`false`	výpisy mezivýsledků (PMK, PTK, MIC)