evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ

Demonstrátor KEYMAKER

Název: **Využití modulu důvěryhodné platformy pro podporu systému TrueCrypt**

Vypracoval: **Ondřej Málek**

Datum: **28. 11. 2009**

evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost
Tato práce vznikla za podpory projektu MŠMT CZ.1.07/2.3.00/09.0067 TeamIT – Budování konkurenceschopných výzkumných týmů pro IT.

Obsah

Kapitola 1

Úvod

V dnešní době neustále vzrůstá využití výpočetní techniky jak v oblasti pracovní nebo akademické, tak v oblasti osobní. S tím souvisí ukládání zvyšujícího se množství různě citlivých dat. Zároveň se zlepšuje interkonektivita a mobilita počítačů, kdy současné trendy směřují k co největšímu používání přenosných počítačů. Společně tento vývoj zvětšuje bezpečnostní rizika spojené s používáním počítačů jako hlavního prostředku pro ukládání a přenos dat.

Ochranou dat v kontextu této práce myslím především jejich důvěrnost, tedy situaci, kdy k nim má přístup pouze oprávněná osoba nebo osoby. S důvěrností souvisí druhý používaný pojem, kterým je důvěryhodné popření existence dat. To je možné ve chvíli, kdy je chráněna nejenom důvěrnost dat, ale zároveň je utajena i jejich existence takovým způsobem, že ji uživatel může dostatečně věrohodně popřít.

Zároveň je nutné dbát na otevřenost a prověřenost technologií použitých v zabezpečování dat. Obecným principem je používání otevřených standardů a technologií, které jsou k dispozici pro obecné bezpečnostní prověření. Zkušenost ukazuje, že proprietární technologie založené na principu „security by obscurity“ nejsou ideálně důvěryhodné.

V této práci představuji možnosti propojení dvou otevřených technologií. První z nich je program TrueCrypt, což je open-source aplikace pro softwarové on-the-fly šifrování dat. TrueCrypt je také celý navržen pro zajištění možnosti věrohodného utajení existence dat. Zároveň v posledních verzích zavádí i systémové šifrování, tedy šifrování systémového disku. Tento systém je velmi používaný a do značné míry bezpečný, co se šifrování dat týká. Na druhou stranu samotné softwarové řešení není již v současné době, kdy neustále narůstá také složitost počítačových systémů, dostatečně důvěryhodné.

Důvěra je základním termínem druhé technologie, se kterou budu pracovat. Tou je důvěryhodné počítání a zejména části založené na modulu důvěryhodné platformy. Jak již napovídá název, důvěryhodné počítání si dává za cíl zavést do výpočetní techniky větší míru důvěry jak v hardware, tak software. Důvěra by měla být jednoduše měřitelná a ověřitelná a také musí vycházet z nějakých neměnitelných kořenů důvěry. K tomu slouží modul důvěryhodné platformy, což je hardwarový čip implementující základ pro techniky důvěryhodného počítání. Standardy důvěryhodného počítání jsou vytvářeny jako otevřené a díky tomu podléhají, stejně jako například zdrojové kódy TrueCryptu, veřejnému zkoumání.

Spojení těchto technologií by mělo vést k vytvoření systému, kdy data budou ukládána nejenom bezpečně, ale také důvěryhodně, a zároveň bude pro uživatele

dostatečně jednoduché a přehledné. Tato práce si dává za cíl přispět k řešení otázek ohledně důvěrnosti uložených dat vypracováním studie, zda je toto propojení možné, případně za jakých podmínek. V rámci těchto úvah bude řešena nejenom důvěrnost dat, ale i případná důvěryhodnost popření jejich existence.

V první části je představen systém TrueCrypt, nejdříve z konceptuálního a poté z implementačního hlediska. Zároveň jsou zde shrnuty jeho bezpečnostní funkčnosti a problémy. Následující kapitola se zabývá důvěryhodným počítáním, kdy jsou představeny principy, na kterých tato technologie stojí, a zároveň detaily směřující k jeho použití na osobních počítačích.

Po rozboru těchto dvou technologií je další kapitola věnována analýze možností jejich propojení, co se týká obecného šifrování dat, kdy je největší důraz kladen na důvěryhodnou ochranu klíčů k šifrovaným datů.

Šifrování systémového disku je v mnoha ohledech odlišné od pouhého šifrování datových svazků. Proto je této problematice věnována samostatná kapitola. Kromě aplikace předchozí kapitoly na systémové šifrování jsou zde rozebírány možnosti ohledně důvěryhodného startu počítače. Na závěr jsou uvedeny dva možné scénáře spojení TrueCryptu a důvěryhodného počítání, jeden s ohledem na jednoduchost použití a zajištění důvěrnosti, druhý s ohledem na maximální možnou popiratelnost dat.

Kapitola 2

TrueCrypt

TrueCrypt [1] je v současné době jedním z nejpopulárnějších řešení pro on-the-fly šifrování disků. Vzhledem k tomu, že funkčnost a možnosti TrueCryptu se v závislosti na použitém operačním systému mění a plná funkčnost je dostupná pouze pro MS Windows, bude se tato kapitola věnovat možnostem, které TrueCrypt poskytuje tomuto systému. První část je věnována vysvětlení některých konceptů, na které byl brán zřetel při vývoji TrueCryptu, druhá se věnuje samotnému programu a implementaci těchto principů a poslední část probírá některé bezpečnostní otázky ohledně návrhu a používání.

TrueCrypt je vyvíjen jako open-source software společností TrueCrypt Foundation a šířen pod vlastní TrueCrypt licencí verze 2.6. Dostupný je pro Microsoft Windows (XP, Vista, Server 2003/2008 a to jak pro 32b, tak i pro 64b verze), Linux a OS X, nicméně plná funkčnost je v současnosti dostupná pouze pro systémy MS Windows. Nejnovější TrueCrypt je verze 6.2 z 11. května 2009, tento přehled se konkrétně věnuje verzi 6.1a z 1. prosince 2008.

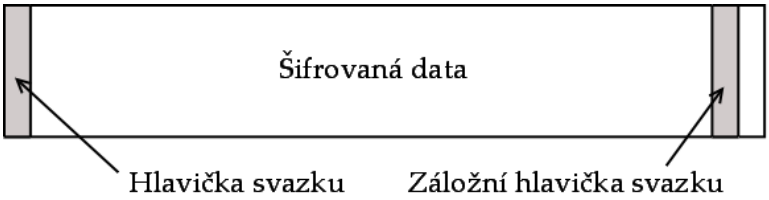
Problémem u práce s TrueCryptem je jeho poměrně restriktivní licence a zároveň uzavřený kolektiv autorů nepodporující odvozené projekty. Kvůli tomu je náročné zařazovat případné odvozené projekty nebo moduly do hlavního projektu.

2.1 Koncepty

V této části jsou uvedeny základní mechanismy a cíle, pro které byl TrueCrypt navržen a které se dalším rozvojem snaží naplňovat. Jeho základním posláním je on-the-fly šifrování dat, přičemž pracuje nad entitou zvanou TrueCrypt svazek.

2.1.1 TrueCrypt svazek

TrueCrypt svazkem může být jak soubor (a to jak pevné, tak variabilní délky), část disku (Partition) nebo celý fyzický disk nezávisle na médiu. Ve všech případech je jeho obsah zvnějšku (tedy nikoliv po namontování, viz 2.2.3) nerozeznatelný od náhodných dat, kterými jej naplní TrueCrypt již při jeho vytváření. To zabraňuje identifikování daného bloku dat jako TrueCrypt svazku, neboť je šifrovaná i jeho hlavička, resp. od verze 6.0 obě hlavičky (druhá je záložní, vytvořená pro případ havárie hlavní). Že je v daném bloku dat skrytý TrueCryptový svazek lze rozpoznat pouze po zadání správného hesla. Nicméně samotná existence souboru se statisticky náhodnými daty bude na většině počítačů podezřelá. Základní struktura TrueCrypt svazku je znázorněna na obrázku 2.1.



Obrázek 2.1: TrueCrypt svazek

On-the-fly šifrování dat pracuje tak, že data jsou šifrována a dešifrována ve chvíli přístupu k nim. Celý TrueCrypt svazek je tedy šifrován, ale dešifrována jsou pouze ta data, která si systém, nebo nějaká aplikace třetích stran, vyžádá. Díky tomu je výkon nutný k dešifrování využit pouze pro opravdu používaná data. Zároveň je zde velká transparentnost vzhledem k systému a uživateli, který nemusí řešit dešifrování dat před jejich použitím (jinak než namontováním TrueCrypt svazku).

2.1.2 Šifrování systému

Šifrování celého systému, resp. systémové části disku, zavádí TrueCrypt až od verze 5.0. Systémový disk je šifrován obdobně jako standardní TrueCrypt svazek. Jsou zde ale rozdíly ve struktuře hlaviček a jiný je i proces vytváření, kdy je navíc na disk nahrán TrueCrypt zavaděč. To ústí v některé problémy ohledně věrohodného popření existence dat. Nicméně pokud nemá uživatel za cíl data popírat a stačí mu jejich přímá ochrana, tak je systémové šifrování velkým krokem vpřed (viz. 2.3.2).

2.1.3 Věrohodné popření

Věrohodné popření existence dat (Plausible Deniability) je jedním ze základních cílů TrueCryptu. Tato problematika se zabývá otázkou, jak utajit nejenom obsah dat, ale jak zároveň důvěryhodně utajit i jejich existenci. V TrueCryptu je mnoho různých částí, které jsou implementovány zvláště s ohledem na možnost popřít existenci dat.

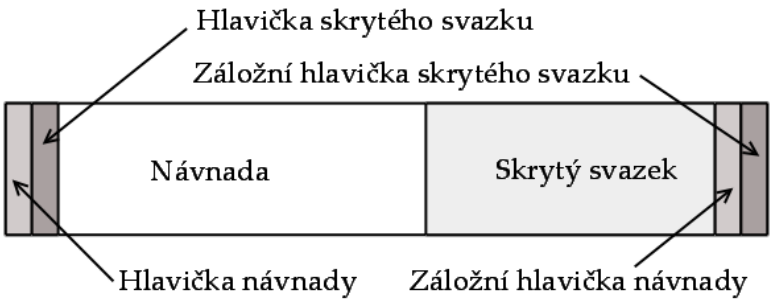
Dobrým příkladem je vnější neodlišitelnost TrueCrypt svazku od náhodných dat. Toto tvrzení neplatí v případě šifrování systémového disku, kdy je v MBR (Master Boot Record) uložen speciální TrueCrypt zavaděč (TrueCrypt Boot Loader), který vyžaduje od uživatele heslo pro dešifrování samotné systémové oblasti. Systémový disk jako takový je sice šifrován naprosto stejně jako jakákoliv jiný TrueCrypt svazek, nicméně sám zavaděč TrueCryptu je dost na prozrazení existence šifrovaného systému. Mnoho na tom nemění ani víceméně kosmetická změna, kdy lze od verze 6.1 měnit výpis zavaděče. Takováto změna může být na první pohled zaměněna za nefunkčnost počítače, samotný zavaděč v MBR může být stále identifikován.

Problémem pro věrohodné popření existence dat je také situace, kdy má uživatel zašifrovanou pouze část disku a zároveň je na nešifrovaném systému nainstalovaný TrueCrypt (nebo v registrech zbylé záznamy po cestovním módu). Tehdy zaniká možnost věrohodně popřít, že části disku, které se tváří jako náhodná data, nejsou ve skutečnosti TrueCrypt svazkem. Potom může být uživatel donucen (například nařize-

ním soudu nebo za pomoci mučení) k odhalení příslušných hesel. Zde pomůže využití skrytých svazků nebo skrytého operačního systému.

2.1.4 Skrytý svazek a operační systém

Skrytý svazek (Hidden Volume) je vlastně svazek uvnitř svazku. I v případě, že je vnější svazek namontován, není možné prokázat, že v něm existuje svazek skrytý. Ve vnějším svazku (označovaném jako návnada) jsou nějaká částečně nebezpečná data, která uspokojí útočníka, a ve vnitřním jsou skutečně důvěrná data. Skrytý svazek je zobrazen na obrázku 2.2.



Obrázek 2.2: Skrytý TrueCrypt svazek

Ještě lepší možností je vytvoření skrytého operačního systému (Hidden Operating System). V takovém případě je použito šifrování systému jak pro skrytý operační systém, tak pro návnadu. Návnada by měla být používána pro běžnou práci, zejména pro jakoukoliv práci se síťovými prostředky. Při používání sítě je velká možnost prozrazení v okamžiku, kdy počítač zjevně komunikoval s okolím a zároveň OS působící jako návnada nebyl spuštěn. Stejně tak není v zájmu bezpečnosti a prosazení možnosti věrohodného popření existence skrytého systému takovémuto běžícímu systému dovoleno zapisovat kamkoliv jinam, než do skrytých TrueCrypt svazků. Díky tomu nemůže ani systém, ani aplikace třetích stran, vynášet informace ze skrytého operačního systému do nižších úrovní zabezpečení.

2.2 Implementace

TrueCrypt pracuje jako diskový ovladač, který mapuje TrueCrypt svazek (TrueCrypt Volume) na fyzický disk počítače. Tím je zajištěna úplná transparentnost pro operační systém a programy počítače. Pro jakýkoliv provoz mimo TrueCrypt se namontovaný (připojený, viz 2.2.3) svazek chová jako jakýkoliv jiný disk v počítači a jako s takovým je s ním pracováno, nepotřebuje žádnou další podporu ze strany ostatních programů. Samozřejmě je tu režie ze strany systému, který musí provádět samotné šifrování, a to jak co se týká systémového času, tak paměti.

2.2.1 Instalace

TrueCrypt se skládá ze dvou částí. Jednou je ovladač, obstarávající samotnou funkčnost, a druhou je grafická aplikace, sloužící k jeho ovládání. Instalace TrueCryptu na počítač vyžaduje administrátorská práva, vzhledem k nutnosti uložit TrueCrypt driver mezi systémové soubory a zapsat příslušné záznamy do registrů. Pro následné používání nejsou administrátorská práva potřebná, neboť základní funkce jako namontování a odmontování svazků jsou dostupné i bez nich.

Kromě klasické instalace je zde ještě druhá možnost. Tou je spouštění v tzv. cestovním módu (Traveller's mode), který umožňuje používat TrueCrypt i bez instalace na daný počítač. Cestovní mód je využíván buď přímým spuštěním binárního souboru, nebo instalováním speciálně upraveného TrueCryptu na přenosné médium (USB disk, CD/DVD apod.). To sice umožňuje pracovat se šifrovanými svazky i na počítači bez nainstalovaného TrueCryptu, jsou zde ale velké nevýhody. Cestovní mód vyžaduje administrátorská práva na daném počítači a zároveň zanechává stopy v registrech počítače. Díky tomu je jediná výhoda, že TrueCrypt bude dostupný kdykoliv je potřeba svazky uložené na stejném médiu namontovat.

2.2.2 Hesla a použité algoritmy

TrueCrypt podporuje různé možnosti zadávání hesla, přesněji podporuje jak heslo jako takové, tzn. řetězec tisknutelných ASCII znaků, tak jeho kombinaci s klíčovým souborem (Keyfile) nebo soubory (může jich být v podstatě neomezené množství). Klíčovým souborem může být libovolný soubor (ať již existující, nebo TrueCryptem speciálně vygenerovaný), přičemž v poslední verzi je podporováno i jeho načtení ze security tokenu. Nezávisle na velikosti klíčového souboru je relevantní a použitý pouze 1MB dat z jeho začátku.

TrueCrypt používá několik různých šifrovacích algoritmů a hašovacích funkcí. V průběhu verzí se samozřejmě vyvíjejí a přehodnocují použité technologie v závislosti na posledních kryptoanalytických objevech, tudíž se mění používané hašovací funkce, šifry i jejich operační módy. Nicméně vývojáři TrueCryptu pečlivě dbají na zachování zpětné kompatibility, kdy nové svazky jsou vytvářeny pouze podle nejnovějších implementovaných funkcí, ale jde korektně pracovat i se svazky verzí předchozích.

Jako kryptografické hašovací funkce jsou v poslední verzi používány algoritmy RIPEMD-160, SHA-512 a Whirlpool. Tyto funkce jsou použité jako kryptografická primitiva pro generování náhodných dat nejen pro použití při generování obsahu svazku během inicializace, ale i pro procedury ohledně vytváření a ověřování klíčů, soli apod.

Z šifrovacích algoritmů to jsou v současnosti AES (Advanced Encryption Standard), Twofish a Serpent. AES je od verze 5.1 implementován také v jazyku symbolických adres, což výrazně zvýšilo jeho rychlost. Všechny používají délku klíčů 256 bitů a vzhledem k použitému módu operací XTS (od verze 5.0 nahradil LRW) mají klíče dva, primární a sekundární. Všechny tyto šifry je možné pro větší bezpečnost (ačkoliv jsou v současné době považovány za bezpečné) použít v předdefinovaných

kaskádách dvou nebo dokonce všech tří šifer (viz tabulka 2.1). Pak každá z nich používá svá vlastní hesla a data jsou dešifrovány postupně dle dané kaskády. Tento přístup samozřejmě znamená podstatné navýšení nejenom bezpečnosti, ale i náročnosti na systémové prostředky.

šifra	**kaskáda**	
AES		
Twofish		
Serpent		
AES	Twofish	
AES	Twofish	Serpent
Serpent	AES	
Serpent	Twofish	AES
Twofish	Serpent	

Tabulka 2.1: Možné šifrovací kaskády

2.2.3 Montování a šifrování svazků

Montování TrueCryptového svazku na systém probíhá tak, že heslo a klíčový soubor (soubory) společně se solí (prvních 64b svazku, jediná nešifrovaná, ale stále náhodná, část svazku) prochází procedurou, kdy je z nich vytvořen tzv. klíč hlavičky (Header key). Pomocí klíče hlavičky je dešifrována hlavička svazku (viz tabulka 2.2), z níž je potom načten hlavní klíč (Master key), který je potom používán pro dešifrování vlastního obsahu svazku.

Z pohledu bezpečnosti a důvěrnosti je důležité, že samotné dešifrování hlavičky probíhá formou pokus–omyl, kdy jsou zkoušeny postupně všechny možnosti, jak bylo možno hlavičku zašifrovat. Hledá se ta, při které jsou v hlavičce smysluplné informace, tedy čitelný řetězec „TRUE“ (viz tabulka 2.2).

Druhou důležitou vlastností je nezávislost hlavního klíče na klíči hlavičky, díky čemuž lze vyměnit klíč hlavičky (tedy to, co si pamatuje uživatel, případně obsah klíčového souboru) aniž by bylo potřeba náročně dešifrovat a znovu zašifrovat celý obsah svazku. Také lze tuto proceduru použít pro restartování hesla, kdy může mít administrátor uloženou původní hlavičku s vlastním heslem. U skutečně používaného svazku je hlavička jiná, generovaná jiným heslem. V případě jeho ztráty je možné přistoupit k přehrání hlavičky a disk namontovat pomocí původního hesla.

Při on-the-fly šifrování je hlavní klíč uložený v nestránkované paměti a nikdy (až na výjimku v podobě tzv. memory dump -– kopie operační paměti) se neobjevuje uložený na disku nebo volně přístupné v paměti. Pomocí hlavního klíče jsou pak v operační paměti dešifrována požadovaná data a ta jsou následně předána volajícímu programu. Není tedy dešifrován celý obsah namontovaného svazku, ale pouze konkrétně požadovaná část. Na druhou stranu libovolná aplikace s příslušnými právy ke čtení z daného disku (jak se po namontování hlásí TrueCrypt svazek) si může tyto data odkládat naprosto libovolně, kamkoliv jí to systém povolí. To znamená, že může

offset (v bytech)	**velikost** (v bytech)	**šifrované**	**popis**
0	64	ne	sůl
64	4	ano	řetězec „TRUE“
68	2	ano	verze formátu hlavičky
70	2	ano	minimální verze programu pro otevření
72	4	ano	CRC-32 bytů 256-512
76	16	ano	rezervováno (nastaveno na nulu)
92	8	ano	velikost skrytého svazku (u vnějšího 0)
100	8	ano	velikost svazku
108	8	ano	offset začátku hlavního klíče
116	8	ano	velikost hlavního klíče
124	4	ano	příznaky
128	124	ano	rezervováno
252	4	ano	CRC-32 bytů 64-251
256	různá	ano	hlavní klíče, složené za sebou
512	65024	ano	rezervováno (u systémového vynecháno)

Tabulka 2.2: Hlavička TrueCrypt svazku

bez nejmenších problému vynášet data na nešifrované disky (nebo je dokonce odesílat po síti) bez toho, že by tomuto chování TrueCrypt bránil.

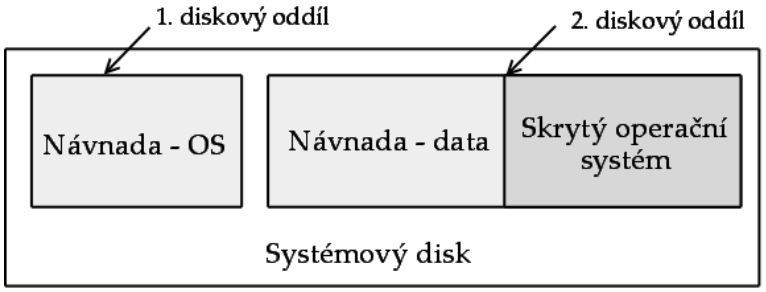
2.2.4 Šifrování systémového disku

Na chování TrueCryptu, kdy různé aplikace, nebo systém samotný, vynášejí data ze šifrovaných disků, bylo několikrát poukázáno jako na závažné ohrožení principu věrohodného popření existence zašifrovaných dat a zároveň nebezpečí pro expozici a narušení důvěrnosti šifrovaných dat [2]. Alespoň částečným řešením nejenom tohoto problému je možnost šifrování systémového disku (nebo systémového svazku), kterou pro systémy MS Windows TrueCrypt zavádí od verze 5.0 a v následujících verzích pravidelně vylepšuje.

Šifrování systémového disku je řešeno pomocí pre-boot autentizace uživatele, tedy zadáním hesla (zde není možné používat klíčové soubory) před vlastním nahráním operačního systému. To je realizováno pomocí TrueCrypt zavaděče (TrueCrypt Boot Loader) přepisujícího původní systémový zavaděč umístěný v prvním sektoru disku, na kterém je nainstalován operační systém. Samotný proces šifrování systémového disku probíhá během normální práce s funkčním systémem, na rozdíl od šifrování logických disků, které je možné šifrovat i s obsahem pouze v systémech Windows Vista/2008. Důležitým (a vyžadovaným) prvkem při šifrování systémového disku je záchranný disk (Rescue Disc), obsahující zálohu TrueCrypt zavaděče a hlavičky svazku pro případy poničení jednoho nebo druhého. Díky záchrannému disku lze v takovýchto případech systém korektně nastartovat. Stejně tak ho lze použít pro restartovaní hesla (viz 2.2.3).

2.2.5 Skrytý svazek a operační systém

Vytváření skrytého svazku je řešeno druhou hlavičkou, která je uložena na speciálním místě svazku (viz 2.1.4), o jejíž dešifrování se TrueCrypt automaticky pokouší nejdříve. Pokud ji úspěšně otevře, pak je namontován vnitřní, skrytý svazek. Znovu se objevuje možnost úniku informací o existenci dat na daném skrytém svazku do nižších úrovní zabezpečení (ať již do normálně šifrovaných, nebo úplně nezabezpečených oblastí počítače). Dalším logickým krokem je tudíž vytvoření skrytého operačního systému.



Obrázek 2.3: Skrytý operační systém

Skrytý operační systém (Hidden Operating System) může být pomocí TrueCryptu vytvořen tak, že v dostatečně velkém oddílu (který musí být první po systémovém oddílu) na systémovém disku je inicializován TrueCrypt svazek. V něm je, zabezpečený nějakými důvěrně vypadajícími daty v „normálně“ šifrovaném svazku, vytvořen skrytý svazek a do něj je překopírován původně nainstalovaný a zašifrovaný systém. Na konci procesu je ještě bezpečně smazán původní systém a na jeho místo je nutno nainstalovat a TrueCryptem zašifrovat systém nový. Ten je označován jako návnada (Decoy) a měl by být používán pro všechny činnosti, pro které není možnost věrohodného popření existence skrytých dat nutná. V pre-boot fázi lze místo hesla k systému působícímu jako návnada použít (odlišné) heslo ke skrytému systému a tím ho spustit. Schéma skrytého operačního systému je naznačeno na obrázku 2.3.

2.3 Bezpečnostní otázky

V následujícím textu se dotkneme některých bezpečnostních otázek a vylepšení v souvislosti s TrueCryptem. Analýza se bude vztahovat k verzi 6.1a z 1. prosince 2008. Referenční verzí, ze které budeme vycházet, bude verze 4.3a z 3. května 2007. Ta byla téměř rok nejnovější stabilní verzí TrueCryptu a tudíž byla velmi rozšířenou, používanou a v neposlední řadě prozkoušenou verzí (např. [3]).

2.3.1 Bezpečnost algoritmů

U šifrovacích algoritmů nastala v posledních verzích změna v podobě preferovaného operačního módu. Předchozí LRW (Liskov, Rivest, Wagner) byl po zjištění některých bezpečnostních problémů vyměněn za XTS (XEX [4] -based Tweaked CodeBook mode

(TCB) with CipherText Stealing (CTS)), který je standardizovaným módem pro AES organizací IEEE a je zvažován organizací NIST. LRW je samozřejmě podporován dále pro zajištění zpětné kompatibility, nové svazky je ale možné vytvářet pouze v módu XTS. Ohledně XTS se během veřejného připomínkování pro NIST objevilo několik problémů [5], a proto by bylo vhodnější použít XEX mód. U něho nejsou problémy s případnými licenčními požadavky třetích stran a efektivně používá jeden klíč tam, kde XTS zavádí dva, aniž by byla zřejmá bezpečnostní výhoda.

Co se týká algoritmů pro šifrování, TrueCrypt stále podporuje šifry AES, Twofish a Serpent. Změna nastala u AES, kdy je nově podporována implementace v jazyce symbolických adres (Assembly language), čímž byla výrazně zlepšena jeho výkonnost. Z hašovacích funkcí již není pro nové svazky podporována SHA-1. Ta byla nahrazena SHA-512. SHA1 je ale stále podporována kvůli zpětné kompatibilitě.

2.3.2 Bezpečnost šifrování systémového disku

Nejvýznamnější změnou, která v novějších verzích nastala, je bezesporu zavedení možnosti šifrování systémového disku nebo oddílu disku. Tuto možnost TrueCrypt zavádí pro následující verze MS Windows: XP, Vista, Server 2003/2008. Pokud je využita, řeší některá možná narušení důvěrnosti dat, případně principu věrohodného popření existence dat.

Prvním problémem je stránkování paměti, kdy si systém odkládá data z operační paměti na disk, přesněji na systémový disk, do speciálního souboru. Zde vzniká bezpečnostní mezera. TrueCrypt se snaží ukládat si svá provozní data (například hlavní klíč) do nestránkované paměti, nicméně nemusí se mu to vždy ze systémových příčin podařit. Tato data by se tedy teoreticky mohla v nešifrované podobě objevit na disku. Stejně tak se ve stránkovacím souboru mohou objevit data ze šifrovaných disků zpřístupněná libovolnou aplikací. Taková data se ukládají do stránkovací paměti a systém je tak může volně odkládat na disk. TrueCrypt při instalaci automaticky stránkování paměti vypíná, nicméně tato operace může způsobovat problémy s během počítače. Šifrování systému je podstatně lepší variantou, protože je stránkovací soubor korektně šifrován.

Druhým problémem je tzv. memory dump, tedy uložení obsahu celé operační paměti do souboru v případě havárie systému. Tento soubor je standardně ukládán na systémový disk a proto pro něj platí stejné možnosti jako pro stránkovací soubor.

Třetí podobnou možností je hibernace počítače, kdy je aktuální nastavení včetně obsahu operační paměti ukládáno pro budoucí použití při buzení počítače. I zde je šifrování systémového disku ideálním protiopatřením.

Šifrování systémového disku je také ideálním opatřením proti únikům šifrovaných dat skrze aplikace systému nebo třetích stran, které by mohly ukládat data čtená z šifrovaných oblastí do vlastních úložišť na nešifrovaném disku (zvláště systémovém). Takovéto úniky byly u TrueCryptu diskutovány jako jedna z nejčastějších možností porušení důvěrnosti a věrohodného popření dat [3].

S ohledem na posílení možnosti věrohodně popřít existenci dat zavádí TrueCrypt zároveň se šifrováním systémového disku možnost vytvořit skrytý operační systém (viz 2.2.5). Tento systém je uložen ve skrytém TrueCrypt svazku na prvním diskovém

oddílu za normálním systémem (také šifrovaným), který funguje jako návnada (Decoy), stejně jako normální data ve vnější oblasti skrytého disku. Takto vytvořený systém může zapisovat pouze na jiné skryté disky (číst může z jakýchkoliv), čímž vytváří velmi bezpečnou a důvěrnou pracovní platformu.

2.3.3 Uživatelská práva

S bezpečností používání TrueCryptu souvisí také jeho zaměření. TrueCrypt je koncipován primárně pro jednouživatelskou stanici. Hlavním výsledkem této koncepce je, že TrueCrypt nepodporuje kontrolu uživatelských práv při provádění kontrolních příkazů. Příkazy jako montovaní a odmontování disku může provádět kterýkoliv uživatel daného systému. Stejně tak všechny namontované disky jsou přístupny všem uživatelům bez rozdílu (pokud to není ošetřeno např. na úrovni NTFS oprávnění). Nasazení TrueCryptu na koncové stanici jednoho uživatele (což je zdaleka nejběžnější a zároveň plánované využití) toto nijak neovlivní, ale nasazení pro multiuživatelský systém je velmi omezeno.

Neexistence kontroly uživatelských práv může přinášet problémy s případným rozšířením uživatelských práv, protože kontrolní příkazy jsou povoleny pro skupinu `Everyone` a vlastní ovladač pracuje pod účtem systému.

2.3.4 Zdrojový kód

Pokud se týká samotného zdrojového kódu, bylo učiněno několik vylepšení jak vzhledem k přehlednosti, tak bezpečnosti celé aplikace. TrueCrypt nyní používá vlastní jmenný prostor a část funkcí a volání byla přejmenována pro jejich lepší identifikaci. Navíc byly dodefinovány konstanty, které se v rámci programu používají.

Z bezpečnostního hlediska vidím jako zajímavé ošetření explicitního přetypování v některých funkcích, kde dříve bylo přetypování pouze automatické. Zároveň novější verze zavádějí ošetření výjimek, které ve verzích předcházejících chyběly. Výjimky jsou nejlépe ošetřovány v novém kódu, zatímco v starších částech jsou zatím upraveny pouze krizové úseky. Nicméně velmi dobře napsané (ohledně výjimek) jsou části spojené se šifrováním systémového disku, kdy by případná neošetřená výjimka mohla způsobovat nefunkčnost celého počítače.

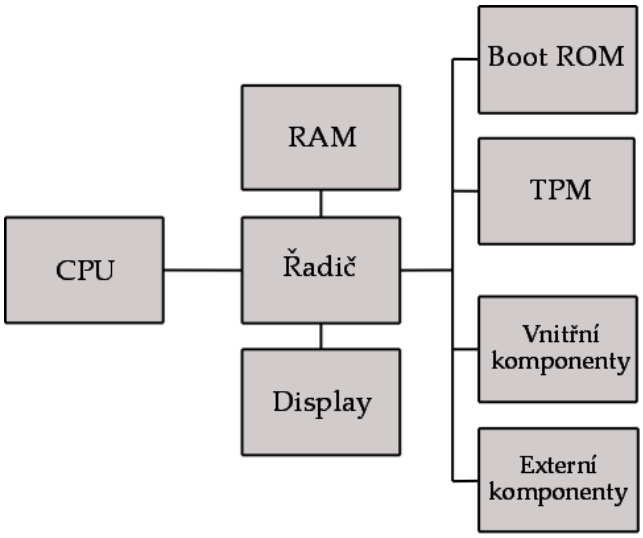
Poněkud nešťastné je používání konstrukce goto, která se obecně nedoporučuje. Zároveň se na některých místech objevuje tzv. polykání výjimek, kdy je odchycována jakákoliv výjimka v dané části kódu a není na ni nijak reagováno. Zde by byl vhodný alespoň nějaký výpis, neboť nezávisle na nebezpečnosti vyvolané výjimky (zvláště pokud jsou odchytávány všechny) je vhodné minimálně upozornit uživatele, že je někde problém.

Kapitola 3

Modul důvěryhodné platformy

V současné době je kladen stále větší důraz na bezpečnost počítačů, přičemž jejich zabezpečení se zvětšováním výkonu, mobility a složitosti stává stále náročnějším úkolem. Existuje několik přístupů k vylepšení softwarových prostředků ochrany počítačů a jedním z nich je technologie důvěryhodného počítání (Trusted Computing). Touto technologií a zejména modulem důvěryhodné platformy (Trusted Platform Module, dále jen TPM) se bude zabývat tato kapitola vycházející zejména z materiálů uvedených ve specifikacích [6, 7, 8, 9, 10].

TPM je implementace funkcí definovaných Trusted Computing Group, zahrnující základní kořeny důvěry, chráněné úložiště a činnosti. Nicméně v rámci osobních počítačů jako TPM rozumíme hardwarovou implementaci, tedy čip integrovaný na základní desce počítače. TPM je hardwarovým základem pro důvěryhodné využití počítačů a spolu s dalšími částmi (hlavně kořenem důvěry pro měření) tvoří základ důvěryhodné platformy. Jeho schématické začlenění do architektury osobního počítače je znázorněno na obr. 3.1.



Obrázek 3.1: Schematické zobrazení PC

3.1 Důvěryhodné počítání

Důvěryhodné počítání (Trusted Computing) je jednou z možností pro zvýšení bezpečnosti počítačových systémů. Jeho základem je hardwarová podpora softwarových prostředků.

Hlavním nositelem myšlenky důvěryhodného použití počítačů je v současné době Trusted Computing Group [11] (dále jen TCG) založená roku 2003, jejímž předchůdcem byla od roku 1999 společnost Trusted Computing Platform Alliance. TCG je nevýdělečnou společností sdružující mnoho velkých i malých firem v oblasti informačních technologií, ať již softwaru nebo hardwaru. Jejím hlavním cílem je vytvářet a prosazovat specifikace pro podporu důvěryhodné výpočetní platformy na základě hardwarových prvků. V současné době je nejnovější specifikace verze 1.2.

Základním pojmem, který je nutno blíže specifikovat, je důvěra (Trust) [12]. Důvěra dle TCG je: „očekávání, že zařízení se bude chovat přesným způsobem vzhledem k danému účelu“, tedy, že se bude chovat, jak vzhledem ke svým vlastnostem a stavu má. Takovéto chování systému (tedy důvěryhodné) je možné zabezpečit pouze v případě, můžeme-li ho nějak důvěryhodně měřit a tato měření důvěryhodně zapisovat, skladovat a ověřovat. Toho je u technologie TPM dosaženo pomocí takzvaných kořenů důvěry (viz 3.5) a na nichž založených řetězech důvěry (viz 3.8.1).

Dalším důležitým termínem je důvěryhodná platforma (Trusted Platform, dále jen TP). Platforma jako taková je definovaná jako „souhrn zdrojů umožňující poskytování služby“ [6], tedy hlavně určitá množina hardwaru s obslužným softwarem. Důvěryhodná platforma je pak taková, u které se můžeme spolehnout na to, že správně a důvěryhodně ohlašuje svoje vlastnosti. Důvěryhodná platforma má jako základní část tzv. důvěryhodný stavební blok (Trusted Building Block, dále jen TBB), který se skládá z kořenu důvěry pro měření a modulu důvěryhodné platformy. V modulu důvěryhodné platformy jsou spojeny kořeny důvěry pro ukládání a kořen důvěry pro hlášení.

3.2 Vlastnosti důvěryhodné platformy

Tato část se zabývá základními vlastnostmi platformy, které musí splňovat, aby byla důvěryhodnou ve smyslu definovaném TCG [6]. TCG udává jako základní a minimální vlastnosti chráněné činnosti, schopnost atestace platformy a jejích částí a měření integrity, jeho skladování a hlášení.

3.2.1 Chráněné činnosti a úložiště

Chráněné činnosti jsou takové příkazy, kterým je dovoleno přistupovat k chráněnému úložišti, tj. do míst, kde je možné bezpečně pracovat s důvěrnými daty. Tyto činnosti se týkají nejenom důvěryhodného měření, ale také jsou to příkazy pro práci s dalšími částmi TPM. Zahrnují například tvorbu kryptografických klíčů, práci s generátorem (pseudo)náhodných dat a další. Jako důvěryhodné úložiště poskytuje TMP (konkrétní implementace důvěryhodného hardwaru) nejenom trvalou paměť (Non-volatile Memory, dále NVM) a nestálou paměť (Volatile Memory, dále jen VM), ale i speciální

chráněné registry zvané registry konfigurace platformy (viz 3.3.8).

3.2.2 Měření integrity

Měření integrity (Integrity Measurement), spolu s jeho následným bezpečným hlášením a případným logováním průběhu, je další z důležitých vlastností důvěryhodné platformy. Měření integrity odpovídá měření důvěryhodnosti daného softwarového nebo hardwarového prvku. Tím, že důvěryhodně můžeme změřit jeho charakteristiky a zároveň je mít někde důvěryhodně uložené (pomocí chráněných činností a úložišť), je možno tuto charakteristiku měřit a porovnávat s hodnotami určenými výrobcem.

Zároveň je nutné uvést, že v případě měření a hlášení integrity nejde o zákaz fungování platformy v nedůvěryhodném stavu. Tato možnost, pokud je žádaná, musí být řešena navázanými prostředky, jako spojení možnosti použití kryptografických klíčů s výsledky měření. Samotné měření integrity má za cíl pouze důvěryhodně změřit tyto hodnoty, aby se jiná část systému mohla rozhodnout, zda jí daný stav platformy vyhovuje a dostačuje.

Základem měření musí být nějaká důvěryhodná část, kterou nazýváme kořen důvěry pro měření (viz 3.5.3). Hlášení měření integrity se v systému provádí pomocí atestace hodnot uložených do registrů konfigurace platformy (viz 3.3.8).

3.2.3 Atestace

Atestací (Attestation) rozumíme zaručení se za správnost a přesnost konkrétních informací. Externí entity se mohou spolehnout na chráněná úložiště a činnosti a na kořeny důvěry (viz 3.5). Kořeny musí být důvěryhodné, aby bylo možné vůbec nějakou důvěru v systému zavést. Platforma musí být schopna zaručit se za svoje charakteristiky ovlivňující její důvěryhodnost.

TCG specifikuje některé základní atestace:

Atestace TPM (Attestation by the TPM) je předložení informace známé pouze TPM, obvykle pomocí klíče atestační identity (viz 3.4.3).

Atestace k platformě (Attestation to platform) je důkaz, že platformě může být důvěřováno při hlášení měření integrity a je to tudíž důvěryhodná platforma.

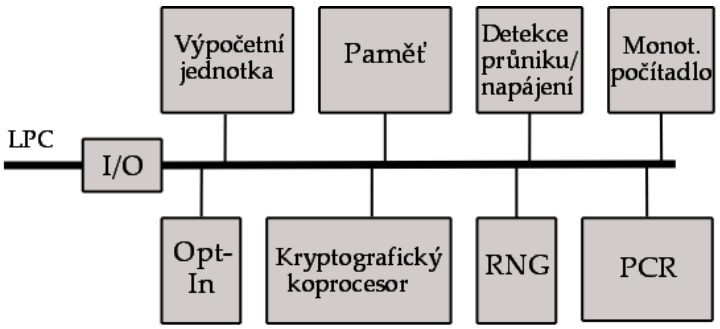
Atestace platformy (Attestation of the platform) je potvrzení, že daná množina hodnot integritních měření pochází z daného TPM. Toto je provedeno podepsáním hodnot pomocí klíče atestační identity (viz 3.4.3).

Autentizace platformy (Authentication of the platform) je potvrzení identifikace platformy pomocí nemigrovatelného klíče (viz 3.6.1).

Všechny formy atestace potřebují jako základ věrohodnou formu identifikace daného TPM nebo jiné entity. Pro TPM tuto identifikaci zajišťuje klíč atestační identity (viz 3.4.3).

3.3 Architektura TPM

Modul důvěryhodné platformy je v podstatě hardwarový čip zajišťující v důvěryhodné platformě kořeny důvěry pro ukládání a hlášení. Skládá se z několika součástí zajišťujících různé služby a funkčnost dalších podporovaných operací. Na obrázků 3.2 je jejich přehled znázorněn graficky.



Obrázek 3.2: Schéma architektury TPM

3.3.1 Vstup a výstup

Jako uzavřený systém musí mít TPM možnost komunikovat se svým okolím. Tato komponenta (I/O) zajišťuje komunikaci skrze protokoly příslušné sběrnice a zároveň uplatňuje přístupové politiky vyžadované Opt-In komponentou (viz 3.3.3) a jakékoliv další vyžadované řízení přístupu. Pro TPM osobního počítače je vybranou sběrnicí pro připojení sběrnice LPC (Low Pin Count, dále jen LPC), což je nízkoúrovňová sběrnice propojující jednotlivé součásti základní desky.

3.3.2 Výpočetní jednotka

TPM musí v průběhu důvěryhodného startu pracovat i v době, kdy není možné spoléhat na dostupnost důvěryhodného výpočetního výkonu. Proto je jednou ze základních komponent výpočetní jednotka (Execution Engine), která zabezpečuje inicializaci TPM a běh základních programů a funkcí.

3.3.3 Opt-In komponenta

Opt-In komponenta (Opt-In) TPM je zodpovědná za změnu stavu TPM a zároveň za dodržování pravidel přístupu nutného ke změně těchto stavů. Zároveň zodpovídá za rozhodnutí o fyzické přítomnosti uživatele, případně zprostředkovává příslušné rozhraní. Opt-In komponenta je také zodpovědná za udržování a prosazování obsahu příznaků souvisejících se stavem TPM.

3.3.4 Paměť

Paměť je v TPM používána jak trvalá (Non-volatile), tak nestálá (Volatile). Nestálá paměť je používána jak pro ukládání dat u kterých je vhodné aby nepřežily restart TPM, tak pro data v průběhu výpočtů. Také do ní mohou být uloženy registry konfigurace platformy (viz 3.3.8).

Trvalá paměť je v TPM využívána pro ukládání povinných dat, například klíče pro potvrzení, a další vnitřní mechanismy, ale je dostupná i pro vnější autorizované procesy. Část trvalé paměti je rezervovaná pro vnitřní použití TPM. Specifikace pro PC určuje, že TPM pro PC by mělo nabízet alespoň 1280B, přičemž alespoň 512B by mělo být uvolněno pro použití operačním systémem a programy.

3.3.5 Detekce napájení a průniku

TPM obsahuje také komponenty odpovědné za detekci změn stavu napájení (Power Detection) a průniku (Intrusion Detection). Detekce napájení je specifikována tak, že TPM musí být upozorněno na jakoukoliv změnu stavu napájení.

TPM musí být připojeno k platformě takovým způsobem, aby se nedalo odstranit nebo přemístit. Detekce průniku musí být na takové úrovni, aby bylo možno fyzickou kontrolou jednoznačně zjistit narušení jeho integrity.

3.3.6 Kryptografický koprocesor

Kryptografický koprocesor (Cryptographic Co-Processor) je zodpovědný za provádění kryptografických operací v rámci TPM. Tyto operace musí zahrnovat generování klíčů pro RSA, stejně jako šifrování a dešifrování pomocí RSA a pomocí symetrického šifrování. Pro hašování musí TPM podporovat funkci SHA-1 a další funkčností je vytváření HMAC. TPM může podporovat další algoritmy (například algoritmy symetrického šifrování pomocí AES), ale pouze výše uvedené jsou povinné.

RSA je algoritmus asymetrické kryptografie, používaný jako základní protokol pro šifrování (a dešifrování) a podepisování dat v TPM. Podporované délky klíčů jsou 512, 1024 a 2048b, přičemž poslední možnost je vyžadovaná u všech povinných klíčů a doporučená u ostatních. Implementace podpisů pomocí RSA musí být dle standardu IEEE P1316 [13].

SHA-1 [14] je standardem pro kryptografické hašovací funkce. V TPM je používaný zejména pro vytváření podpisů.

HMAC [15] se používá hlavně pro autorizační protokoly, kde je využíván pro prokázání znalosti autorizačních dat. HMAC v TPM používá délku bloku 64B a klíč délky 20B.

Symetrické šifrování v rámci TPM se využívá pro šifrování v rámci autorizačních protokolů a pro šifrování shluků dat (Blobs). Pro symetrické šifrování se používá Vernamova šifra s operací XOR, kdy jako klíč se používá nonce (20B) a cílem šifrování jsou zpravidla stejně dlouhá autorizační data. Pro delší data se klíč rozšiřuje pomocí MGF1 [13].

3.3.7 Generátor náhodných dat

Generátor náhodných dat (Random Number Generator, dále jen RNG) pro TPM se skládá ze stavového automatu, přijímajícího a mixujícího nepředvídatelná vstupní data, a jednocestné funkce (SHA-1). Musí zabezpečovat dostatečně kvalitní data pro generování klíčů a další funkce. Generátor musí být v jednom tiku (volání) schopen předat 32B náhodných dat. Pro správné fungování RNG není nutné, aby byl v TPM přítomen generátor skutečně náhodných dat (True RNG, dále jen TRNG).

3.3.8 Registry konfigurace platformy

Registry konfigurace platformy (Platform Configuration Register, dále jen PCR) jsou 160b velká pole pro uchovávání měření integrity. Obvykle jsou uloženy v energeticky závislé paměti v chráněné oblasti. TPM pro PC musí [8] obsahovat 24 PCR číslovaných od 0. Registry 0-15 jsou resetovatelné pouze při restartu hostitelské platformy. Zbylé jsou resetovatelné, pokud jim to přikazuje proces se správnou lokalitou (viz 3.5.4).

Metrika (výsledek měření) je do PCR ukládána jako haš skutečného měření. Vzhledem k tomu, že počet PCR je v TPM omezen, je nutno do nich ukládat změny tak, aby nebylo pro útočníka možné podvrhnout metriku. Nová hodnota daného PCR se tedy skládá jak z nové hodnoty měření, tak z hodnoty předchozí následujícím způsobem: nová hodnota = haš (stará hodnota || výsledek nového měření), kde || udává skládání a jako hašovací funkce se používá SHA-1. Takto je zajištěna jednocestnost, kdy není možné zjistit předchozí měření, a zároveň není tato operace komutativní, záleží tedy na pořadí měření.

PCR jsou používány jednotlivými procesy na základě tzv. lokality (viz 3.5.4). Některé PCR jsou dedikované k použití pro důvěryhodný start systému a jejich použití je rozebráno v části 3.8.2.

3.3.9 Monotónní počítadlo

Monotónní počítadlo (Monotonic Counter) je zařízení poskytující neustále rostoucí a zvyšující se hodnotu. TPM musí podporovat minimálně čtyři počítadla, přičemž jejich implementace může být jak čtyři samostatná počítadla, tak jedno spolu s ukazateli na různé hodnoty. Výstupem z počítadla je 32b hodnota. Všechna počítadla musejí být schopna běžet po dobu minimálně sedmi let za předpokladu přičtení hodnoty každých pět vteřin.

Základní počítadlo se nazývá vnitřní báze (Internal Base) a přístup k němu má pouze samotný TPM. Externí počítadlo (External Counter) je počítadlo, které používají externí procesy. Takové počítadlo může vytvářet a používat proces pouze po autorizaci autorizačními údaji vlastníka TPM.

Monotónní počítadla slouží jako náhrada hodin pro TPM, vzhledem k tomu, že implementace skutečných důvěryhodných hodin by byla příliš nákladná. TPM neudržuje vztah mezi vlastním monotónním počítadlem a skutečným časem. Zodpovědnost za toto svázání je ponechána na entitě, která se na hodnotu počítadla dotazuje.

3.4 Identifikace TPM

Věrohodná a důvěryhodná identifikace je jedním z cílů vývoje důvěryhodné platformy. Je zapotřebí pro atestaci jakýchkoliv dat (například měření integrity) a zároveň i pro většinu ostatních služeb. Tato část ukazuje, jak je identifikace TPM zajištěna, které entity s TPM pracují a také jak se uživatelé autentizují vůči TPM.

3.4.1 Entity ve vztahu k TPM

První důležitou entitou v rámci životního cyklu TPM je výrobce TPM (TPM Manufacturer). Výrobce je zodpovědný za dodržování standardů TCG a jejich splnění potvrzuje vložením certifikátu do TPM. Zároveň zpravidla generuje klíč pro potvrzení (Endorsement Key) jako základní bod identifikace TPM.

Druhou entitou ve vztahu k TPM je vlastník TPM (TPM Owner). Jedno TPM může mít právě jednoho vlastníka, který se prokazuje vůči TPM svými autentizačními daty. Vlastník může svá práva delegovat na další entity, přesněji na uživatele. Vlastník při převzetí vlastnictví generuje hlavní skladovací klíč (Storage Root Key, viz 3.6.2), fungující jako základ hierarchie klíčů a také jako kořen důvěry pro ukládání, a `tpmProof`, což je 20B nonce, tedy náhodná jednorázová hodnota, používaná pro spojení různých dat s daným TPM.

Uživatel TPM je třetí entitou se základním vztahem k TPM. Uživatelem je taková entita, která se prokáže znalostí autorizačních údajů k nějakému objektu chráněnému TPM. Díky tomu neexistuje žádný záznam uživatelů v rámci TPM.

3.4.2 Klíč pro potvrzení

Klíč pro potvrzení (Endorsement Key, dále jen EK) je základním identifikačním prvkem TPM a daný TPM je jednoznačně identifikovatelný skrze svůj klíč pro potvrzení. Zároveň platí, že TPM může mít pouze jeden EK.

EK je pár klíčů RSA délky 2048b. Jeho privátní část se označuje jako `PRIVEK` a veřejná jako `PUBEK`, přičemž přístup ke klíči je možný jen skrze chráněné činnosti a po autentizaci vlastníka TPM, nebo po ověření fyzické přítomnosti uživatele. `PRIVEK` nesmí existovat kdekoliv mimo místa chráněná TPM.

EK je generován uvnitř TPM příkazem `TPM_CreateEndorsementKeyPair` nebo externě a do TPM pouze nahrán. Nezávisle na procesu vytváření musejí být jeho vlastnosti stejné a případný proces externího vytváření musí být bezpečný.

Klíč pro potvrzení musí být potvrzen pomocí osvědčení klíče pro potvrzení (viz 3.4.4), tzn. jeho vlastnosti musí být potvrzeny podepsáním nějakou autoritou. Kdo takové autoritě důvěřuje, může důvěřovat i jí podepsanému osvědčení. Důvěra je problémem při generování EK vlastníkem, a proto je EK zpravidla generován výrobcem, který tak funguje jako příslušná autorita pro osvědčení.

3.4.3 Klíč atestační identity

Klíč atestační identity (Attestation Identity Key, dále jen AIK) se používá pro prokázání identity TPM místo EK, vzhledem k tomu, že větší používání EK jako jediného

identifikačního klíče by vedlo ke svázání veškerých činností TPM dohromady a tím k hrubému porušení soukromí. Jedno TPM může mít neomezený počet AIK a tedy vlastnit pro každé použití jinou identitu, i když v rámci jedné služby stále stejnou a důvěryhodnou.

AIK je stejně jako EK 2048b dlouhý RSA klíč, přičemž jeho privátní část musí být chráněna TPM. Tento klíč je generován uvnitř TPM příkazem `TPM_MakeIdentity`, čímž je zároveň připraven pro ověření certifikační autoritou (Certification Authority, dále jen CA). Ta po potvrzení osvědčení souvisejících s TPM a platformou (viz 3.4.4) vystaví pro daný AIK osvědčení. TPM potom může zpřístupnit AIK k používání pomocí příkazu `TPM_ActivateIdentity`. Oba příkazy potřebují ke spuštění autentizaci vlastníka TPM.

Privátní část AIK lze použít pouze pro generování podpisů. Používá se hlavně k podepisování měření integrity nebo k certifikaci dalších klíčů používaných TPM.

3.4.4 Osvědčení

TCG definuje několik různých osvědčení (Credential). Osvědčení je v podstatě zaručení se za nějakou vlastnost důvěryhodnou autoritou a každé osvědčení by mělo zahrnovat pouze informace relevantní k danému účelu.

Osvědčení klíče pro potvrzení (Endorsement Credential) je vydané entitou generující EK a potvrzuje, že dané TPM vlastní `PRIVEK` patřící k `PUBEK` v daném osvědčení. Dalšími součástmi osvědčení jsou jméno výrobce TPM, číslo modelu TPM a verze TPM.

Osvědčení shody (Conformance Credential) je potvrzení o tom, že důvěryhodný stavební blok je důvěryhodný a pro danou platformu jediný. Osvědčení obsahuje jméno entity vydávající osvědčení (například výrobce), výrobce, model a verzi platformy, výrobce, model a verzi TPM. Osvědčení shody není unikátní, ale je navázáno typ a výrobce platformy.

Osvědčení platformy (Platform Credential) má identifikovat výrobce platformy a popisovat její vlastnosti. Osvědčení platformy se odkazuje jak na osvědčení klíče pro potvrzení, tak na osvědčení shody pomocí jejich haše. Toto osvědčení je pevně svázáno s jednou unikátní platformou (skrze haš osvědčení klíče pro potvrzení). Další informace nutné pro vydání jsou jméno výrobce platformy, její model a verze.

Osvědčení platnosti (Validation Credential) je osvědčení měřitelných prvků počítačového systému (ať již softwarových nebo hardwarových) udávající referenční hodnoty pro měření integrity. Předpokládaným vystavovatelem tohoto potvrzení je výrobce dané komponenty ve chvíli, kdy si je jist hodnověrností a spolehlivostí těchto hodnot. Osvědčení platnosti obsahuje jméno osvědčující entity, výrobce, model a verzi měřeného prvku a hodnoty měření. Dále může volitelně obsahovat vlastnosti komponenty.

Osvědčení klíče atestační identity (Attestation Identity Credential) má za cíl potvrdit svázání TPM a AIK a zároveň provést důkaz, že AIK je svázáno s platnými osvědčeními klíče pro potvrzení, shody a platformy. Dále se vystavovatel zaručuje za dodržování očekávání držitele klíče ohledně zachování důvěrnosti. Osvědčení obsahuje odkazy na výrobce a model TPM, výrobce a model platformy a osvědčení shody. Jsou zde pouze odkazy na informace, které nejsou důvěrné.

3.4.5 Autorizace

TPM používá pro autorizaci manipulace s chráněnými objekty a pro přístup k chráněným činnostem několik různých protokolů. Všechny protokoly jsou založeny na použití autorizačních dat (Authorization Data, dále jen AuthData) o velikosti 160b. Tato velikost je zvolena, protože je to velikost výstupních dat z hašovací funkce SHA-1. Předpokladem je, že heslo je hašováno touto funkcí spolu se solí a případnými dalšími hodnotami (například hodnoty z měření integrity) a výsledkem jsou autorizační data.

V rámci TPM existují také autentizační data vlastníka TPM. Nicméně tato data poskytují vlastníkovi přístup jen k datům a činnostem nechráněnými jinými autorizačními daty. Vlastník TPM tudíž nemá přístup k ničemu, co vyžaduje jím nevlastněná autorizační data. Samotná autorizační data jsou zpravidla jediným důkazem o právech nakládat s daným chráněným objektem.

3.4.6 Autorizační protokoly

První dva protokoly jsou použity pro předání důkazu znalosti autorizačních dat bez nutnosti je posílat nebo identifikovat cílové TPM.

Autorizační protokol nezávislý na objektu (Object-Independent Authorization Protocol, dále jen OIAP) je protokol typu výzva-odpověď využívající jako ochranu proti útoku přehráním náhodnou hodnotu a HMAC speciálně konstruovaného řetězce. Relace ustanovená tímto protokolem trvá dokud není jednou ze stran zrušena a příkazy jím přenášené se mohou týkat různých objektů TPM.

Autorizační protokol závislý na objektu (Object Specific Authorization Protocol, dále jen OSAP) je pevně spjatý s objektem, kterého se týká. Stejně jako OIAP používá náhodné hodnoty a HMAC. V rámci ustanovené relace je sice možno zadat více různých příkazů, nicméně musejí pracovat na daném objektu. Tento protokol je vyžadován pro vkládání nebo přepisování autorizačních dat na objektech, neboť obsahuje více náhodných hodnot, které je možno použít pro koncové šifrování následně přenášených dat pomocí symetrického šifrování.

Protokol pro vložení autorizačních dat (AuthData Insertion Protocol, dále jen ADIP) se používá při vytváření entity a vkládání prvotních autorizačních dat. Tento protokol používá relaci navázanou pomocí OSAP a autorizační údaje

předává šifrované pomocí náhodných dat, ustanovených v rámci této relace. Při zřizování nadřazené relace protokolu OSAP je použita autorizace pro nadřazený objekt vůči objektu vytvářenému, například nadřazený klíč v hierarchii klíčů (viz 3.6.3).

Protokol pro změnu autorizačních dat (AuthData Change Protocol, dále jen ADCP) používá dvě relace, jednu ADIP relaci založenou na OSAP pro zašifrování nových autorizačních dat a druhou vnější OSAP nebo OIAP pro autorizaci přístupu k entitě, u které se mají data měnit.

Asymetrický protokol pro změnu autorizačních dat (Asymmetric Authorization Change Protocol, dále jen AACP) je protokol podobný ADCP, ale rodičovská entita nezískává znalost o změněných autorizačních datech. AACP je od verze standardu 1.2 nedoporučen a místo něj je doporučeno používat ADCP.

3.5 Kořeny důvěry

TCG označuje jako kořeny důvěry (Roots of Trust) takové součásti platformy, kterým je nutno důvěřovat, protože jejich dysfunkce (cílená nebo náhodná) není detekovatelná. Standardně se uvádějí tři kořeny důvěry: pro měření, ukládání a hlášení. Kořeny pro ukládání a hlášení obsahuje TPM.

3.5.1 Kořen důvěry pro ukládání

Kořen důvěry pro ukládání (Root of Trust for Storage, dále jen RTS) zabezpečuje ochranu dat fyzicky umístěných mimo samotné TPM a poskytuje pro tato data jak ochranu důvěrnosti, tak ochranu integrity. Také zajišťuje možnost migrace dat mezi různými TPM. Vzhledem k uložení dat mimo TPM na externím paměťovém médiu (například pevném disku) jsou kapacity důvěryhodného ukládání pomocí RTS limitovány pouze velikostí daného paměťového media. RTS také spravuje malou energeticky závislou paměť uvnitř TPM, používanou pro ukládání právě používaných klíčů.

RTS je pevně svázán s vlastnictvím TPM, kdy základem hierarchie asymetrických klíčů použitých k ochraně dat je hlavní skladovací klíč (Storage Root Key, dále jen SRK) generovaný vlastníkem.

3.5.2 Kořen důvěry pro hlášení

Kořen důvěry pro hlášení (Root of Trust for Reporting, dále jen RTR) zodpovídá za vytvoření a udržování identity platformy, zabezpečuje hlášení obsahů PCR a jejich svázání s daným TPM. Také zajišťuje případnou atestaci těchto údajů externím verifikátorem.

Pro TPM v PC tvoří RTR klíč pro potvrzení (viz 3.4.2). Ten je unikátním identifikátorem daného TPM a zároveň je s ním pevně svázán. Nicméně vzhledem k problémům s důvěrností při větším používání EK se používají pro podepisování hodnot z PCR klíče atestační identity (viz 3.4.3).

3.5.3 Kořen důvěry pro měření

Kořenem důvěry pro měření (Root of Trust for Measurement, dále jen RTM) označujeme výpočetní prostředek schopný důvěryhodně měřit integritu. Standardně jsou to normální výpočetní prostředky platformy (CPU) řízené na základě hlavního kořenu důvěry pro měření (Core Root of Trust for Measurement, dále jen CRTM) a jejich bezpečné propojení.

CRTM je kód spuštěný při inicializaci platformy v důvěryhodném stavu a odvozuje se od něho řetězec důvěry. CRTM může být dvojího typu, statický nebo dynamický.

Statický CRTM (Static Core RTM, dále jen S-CRTM) je neměnnou součástí inicializace hostitelské platformy (tj. základní platformy systému) během jejího restartu. S-CRTM je kód, kterým musí začínat inicializace hostitelské platformy. V současnosti u osobních počítačů rozlišujeme dva druhy S-CRTM. V prvním případě je S-CRTM celý BIOS, v druhém je BIOS rozdělen na zaváděcí blok (BIOS boot block) a POST BIOS. Zde již je S-CRTM pouze zaváděcí blok a POST BIOS musí být před použitím změřen a začleněn do řetězce důvěry (viz 3.8).

Dynamický CRTM (Dynamic Core RTM, dále jen D-CRTM), zavedený ve specifikaci verze 1.2, je stejně jako S-CRTM neměnnou součástí hostitelské platformy. Narozdíl od něj nemusí být inicializován po restartu hostitelské platformy. D-CRTM může být inicializován v libovolné fázi života hostitelské platformy, jedinou podmínkou je, že jeho zavolání musí iniciovat důvěryhodný proces.

Přestože je D-CRTM spouštěn až po S-CRTM (v podstatě v jeho rámci), není mezi jejich řetězci důvěry žádná přímá souvislost mimo společné RTS a RTR. S D-CRTM se úzce pojí také pojem lokalita.

3.5.4 Lokalita

Lokalita (Locality) je princip zavedený ve standardu verze 1.2. Umožňuje, aby TPM současně s příkazem přijímal ještě příznak určující důvěryhodnost procesu, který příkaz zaslal. Tato důvěryhodnost se označuje jako lokalita. TPM není schopno ověřit, zda příznak je nastaven správně, ale implementace by měla být taková, aby na provedení útoku byly potřeba netriviální znalosti a specializovaný hardware. V PC je tento příznak realizován vyhrazenými adresami na LPC sběrnici. Definujeme šest možných lokalit, číselně označené 0-4 a Legacy, která jako jediná není povinná.

S principem lokalit je úzce spjat jak D-CRTM, který je ustanoven pod lokalitou 4, tak přiřazení PCR a dalších objektů, například klíčů. PCR jsou rozděleny podle toho, jak s nimi mohou procesy z různých lokalit nakládat. Možnosti jsou: resetovat, číst, používat (například při použití svázaných dat, viz 3.7.1) nebo modifikovat (tedy přidat nové měření).

Lokalita Legacy odpovídá lokalitě 0 při použití vstupních údajů dle specifikace 1.1. Tato lokalita je ponechána z důvodů zpětné kompatibility.

Lokalita 0 je přiřazena použití pro S-CRTM a jeho řetězec důvěry. Tato lokalita může modifikovat PCR 0-15.

Lokalita 1 je prostředí pro důvěryhodný operační systém a může modifikovat PCR 20.

Lokalita 2 je vyhrazena pro běh operačního systému, má možnost resetovat PCR 20-22 a modifikovat PCR 17-22.

Lokalita 3 může být použita pro přídavné komponenty, nicméně její použití je silně implementačně závislé. Lokalita 3 má k dispozici PCR 17-20 a to pouze pro modifikaci.

Lokalita 4 patří k D-CRTM, kdy ji používá důvěryhodný hardware. Přiřazené PCR jsou 17-20 pro reset a 17, 18 pro modifikaci.

PCR které mohou být resetované a modifikované kteroukoliv lokalitou jsou 16 a 23, přičemž lokalita 16 je dedikována pro ladění a 23 pro použití v rámci libovolné aplikace.

lokalita	kontrolující entita	resetovatelná PCR	modifikovatelná PCR
každá	každá	16, 23	16, 23
0	S-CRTM, statický OS	žádná	0-15
1	dynamický OS	žádná	20
2	dynamický OS	20-22	17-22
3	přídavné komponenty	žádná	17-20
4	D-CRTM	17-20	17, 18

Tabulka 3.1: PCR v rámci lokality

3.6 Klíče

Bezpečná a důvěryhodná správa klíčů je jedním z účelů TPM. Díky tomu zabezpečuje jak ochranu dat, tak ochranu identity uživatele ve vztahu k okolí. Klíčem v tomto případě myslíme jmenovitě asymetrický klíč pro algoritmus RSA, což je základní typ klíče pro TPM. Ostatní typy klíčů jsou označeny konkrétně.

3.6.1 Migrovatelnost klíčů

Jednou ze základních charakteristik klíčů je jejich migrovatelnost. Z tohoto pohledu můžeme pohlížet na všechny klíče chráněné TPM. Migrovatelností rozumíme spojení klíče s konkrétním TPM. Podle migrovatelnosti rozdělujeme klíče na nemigrovatelné, migrovatelné a certifikované migrovatelné klíče. Posledně jmenované zavádí až specifikace verze 1.2.

Nemigrovatelné klíče (Non-migratable Keys) jsou takové klíče, které jsou pevně svázané s daným TPM. Jejich privátní část nesmí opustit ochranu TPM založenou na RTS. Některé klíče jsou dle specifikace nemigrovatelné explicitně, například EK nebo AIK. Svázání klíčů s TPM se děje pomocí hodnoty `tpmProof`, která je generována při tvorbě vlastníka.

Migrovatelné klíče (Migratable Keys) jsou klíče schopné migrovat mezi různými TPM. Tak je možné používat data chráněná takovým klíčem na více různých platformách, aniž by byla narušena jejich důvěryhodnost. Důvěryhodnost migrovatelných klíčů je nutně nižší než nemigrovatelných, jak uživatel nemůže plně kontrolovat jejich pohyb, jakmile jednou opustí TPM.

Zda bude klíč migrovatelný určuje buď specifikace, nebo uživatel. Pokud to zadává uživatel, tak při generování klíče rozhodne, zda bude klíč migrovatelný. Pouze u nemigrovatelných klíčů je vyžadováno, aby byly generovány uvnitř TPM. Migrování probíhá pouze se souhlasem majitele TPM. Daný klíč je dešifrován svým rodičovským klíčem, zašifrován veřejným klíčem cíle a poslán.

Certifikované migrovatelné klíče (Certified Migration Keys, dále jen CMK) jsou nově zavedeným druhem ve specifikaci 1.2. Jsou to speciální migrovatelné klíče, které mají ověřitelné vlastnosti. S CMK se vážou pojmy migrační autorita (Migration Authority, dále jen MA) a vybraná migrační autorita (Migration Selection Authority, dále jen MSA). MA je cílová entita migrace, tedy nový vlastník klíče, a MSA je dohlížecí autorita. MSA může být zároveň MA.

Každému CMK je během vytváření přiřazena MSA nebo MA, přičemž takové přiřazení je nevratné a zaznamenané v certifikátu daného klíče. Migrování CMK potom probíhá pod dohledem a za spolupráce MSA. Pokud tedy třetí strana důvěřuje MSA (odkaz na ní je součástí CMK), pak může důvěřovat i samotnému klíči nezávisle na jeho současném umístění.

3.6.2 Typy klíčů

Rozdělení klíčů podle migrovatelnosti je pouze jedním z možných rozdělení klíčů. V této části uvedeme další varianty rozdělení klíčů, včetně pojmenování a určení speciálních klíčů v rámci TPM. Obecně jsou klíče ještě rozděleny na podepisovací a šifrovací klíče. Jak šifrování (a dešifrování), tak podepisování jsou operace prováděné pomocí asymetrických klíčů, nicméně není doporučeno, aby jeden pár klíčů sloužil k oběma operacím.

Podepisovací klíče (Signing Keys) jsou klíče pro podepisování pomocí asymetrické kryptografie. Tyto klíče mohou být migrovatelné i nemigrovatelné.

Skladovací klíče (Storage Keys) slouží pro šifrování a dešifrování dat pomocí asymetrické kryptografie. Jsou použity jak pro ochranu dat ukládaných mimo chráněné úložiště, tak pro ochranu dalších klíčů.

Klíč pro potvrzení (Endorsement Key, viz 3.4.2) je nemigrovatelný šifrovací klíč sloužící jako základní identifikace TPM.

Klíče atestační identity (Attestation Identity Keys, viz 3.4.3) jsou nemigrovatelné podepisovací klíče určené primárně pro podepisování dat pocházejících z TPM.

Klíče pro svazování (Bind Keys) se využívají pro šifrování a dešifrování u malých objemů dat, zejména klíčů symetrické kryptografie.

Zděděné klíče (Legacy Keys) jsou klíče vytvořené mimo TPM. Tyto klíče jsou migrovatelné a mohou být použity jak pro šifrování, tak pro podepisování.

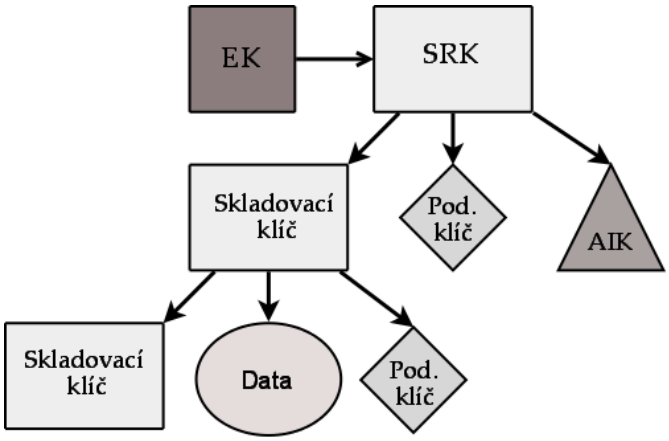
Autentizační klíče (Authentication Keys) jsou symetrické klíče používané na ochranu transportních relací zahrnujících TPM.

Hlavní skladovací klíč (Storage Root Key, dále jen SRK) je nemigrovatelný skladovací klíč generovaný při inicializaci vlastníka TPM jako základ hierarchie klíčů. Je to spolu s klíčem pro potvrzení jediný klíč, který musí být přítomný v TPM neustále. SRK je efektivním kořenem pro ukládání. Stejně jako u EK, SRK může být v TPM pouze jeden.

3.6.3 Hierarchie klíčů

Hierarchie klíčů zabezpečuje rozšíření důvěry z kořene důvěry pro ukládání na ostatní klíče. To zaručuje ochranu samotných klíčů a zároveň ochranu dat, které jsou těmito klíči zašifrované. Kořenem důvěry pro ukládání je hlavní skladovací klíč. Tento klíč je pevně spjat s vlastníkem TPM, při jehož vytváření se generuje. Zároveň je spojen s EK a tím i s daným TPM.

Hierarchie klíčů se od RTS rozšiřuje pomocí skladovacích klíčů. Rodičovský klíč šifruje svojí veřejnou částí privátní část potomka a tím ho ochraňuje. Základem je samozřejmě SRK a díky tomu se přenáší důvěra v rámci stromu klíčů až k listům. Na obrázku 3.3 je zobrazeno schéma hierarchie klíčů.



Obrázek 3.3: Schéma hierarchie klíčů

Klíče jsou mimo TPM ukládány do struktury `TPM_KEY` nebo `TPM_KEY12` [16]. Ta obsahuje informace o možném využití klíče, jeho typu, hodnotách PCR s kterými je svázán a podobně. Stejně tak obsahuje v otevřeném textu hodnotu veřejného klíče. Poslední část je zašifrovaná pomocí veřejné části rodičovského klíče a obsahuje kromě autorizačních dat ještě haš nešifrované části struktury a privátní klíč. Před použitím je nutno klíč z externě uložené struktury nahrát do TPM pomocí `TPM_LoadKey2` [17].

3.7 Ochrana dat

Ochrana dat, jejich důvěrnosti a integrity, při komunikaci nebo ukládání je důležitou schopností. TPM podporuje v této oblasti čtyři základní operace pro nakládání s daty.

3.7.1 Svazování

Svazování (Binding) je operace šifrování (a dešifrování) pomocí asymetrické kryptografie. Používá se při komunikaci, kdy jsou data zašifrována pomocí veřejného klíče příjemce, který by měl být jediný, kdo vlastní privátní klíč k dešifrování zprávy.

Pokud je jako cílového použito nemigrovatelného klíče TPM, tak jsou data pevně svázána s cílovým TPM a nemohou být (za předpokladu, že cílové TPM pracuje korektně) dešifrována nikde jinde. Jestliže je klíč migrovatelný, tak jsou data svázána pouze s tímto klíčem.

3.7.2 Podepisování

Podepisování (Signing) je standardní operací pro zajištění původu a integrity dat při komunikaci. Pro tuto operaci se používají zpravidla podepisovací klíče, kdy je jimi podepsán haš posílaných dat. Podepisování se provádí privátním klíčem, ověření veřejným klíčem.

3.7.3 Pečetění

Pečetění (Sealing) je operací, při které jsou data svázány se stavem dané platformy (hodnoty určité množiny PCR) a zašifrovány tak, aby dešifrování mohlo být provedeno pouze v daném stavu a příslušnou platformou. Je zde možné použít pouze nemigrovatelný klíč.

3.7.4 Pečetěné podepisování

Pečetěné podepisování (Sealed-Signing) je podepisování využívající možností pečetění k zajištění lepší kontroly ohledně tvůrce podpisu. Při této operaci je do těla zprávy (a tím i do podepsaného haše) zahrnuta i konfigurace platformy (daná hodnotami některých PCR). Díky tomu má příjemce možnost ověřit si, v jakém stavu byla platforma odesílatele v době vytváření podpisu.

3.8 Důvěryhodný start systému

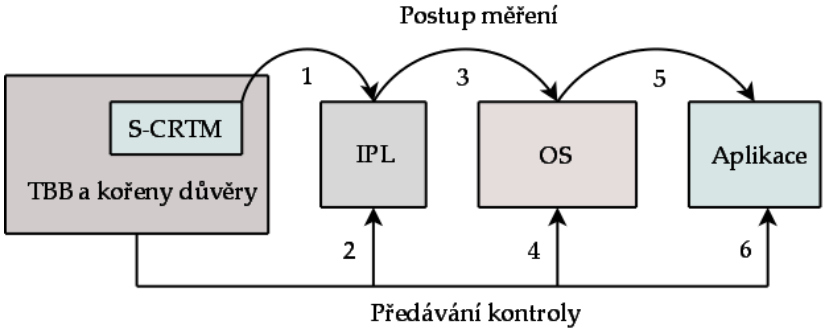
Důvěryhodný start systému (Trusted Boot) je základem pro důvěryhodné použití počítače. Musí zaručit, že danému operačnímu systému je možno důvěřovat. Důvěra se zakládá na řetězci důvěry od statického hlavního kořene důvěry pro měření po operační systém. V této kapitole startem systému myslíme start statického (základního) operačního sytému počítače. Obecně se dá důvěryhodný start systému rozdělit na dvě fáze, přičemž první část ovládá BIOS a druhou startovaný systém.

Důvěryhodný start systému nicméně nevynucuje žádná omezení v případě, že naměřené hodnoty jsou jiné než referenční. Systém je nastartován i takto (tzn. do nedůvěryhodného stavu), ale je zaručeno, že není proveditelné změnit hodnoty měření zpět tak, aby označovaly systém jako důvěryhodný. Každý proces po startu má možnost si tento stav ověřit. Navazování procesů přímo na start systému je samozřejmě možné například pomocí svázání klíčů s hodnotami PCR naměřenými při startu. V tomto případě mohou být klíče uvolněny z TPM pouze v důvěryhodném stavu.

3.8.1 Řetězec důvěry

Důvěryhodný start systému je v podstatě vytváření řetězce důvěry (Chain of Trust) mezi S-CRTM a operačním systémem, resp důvěryhodným operačním systémem. Řetězec důvěry je vyjádření předcházejícího článku řetězce o vlastnostech článku následujícího.

Základem řetězce důvěry při startu systému je S-CRTM, protože řetězec důvěry může začínat pouze v důvěryhodné části a tou je při startu OS S-CRTM. Následné vytváření spočívá v změření další části předtím, než je jí předána kontrola. Hodnota těchto měření se ukládá do PCR 0-7. Na obrázku 3.4 je znázorněno zjednodušené vytváření řetězce důvěry.



Obrázek 3.4: Schéma vytváření řetězce důvěry při startu OS

3.8.2 Použití PCR

Měření provedená v průběhu startu statického operačního systému se ukládají do PCR 0-7. Obecně pro všechny fáze platí, že kód musí být vždy před spuštěním změřen a výsledek důvěryhodně uložen do příslušného PCR. Jakákoliv následná změna v libovolných změřených datech musí vyústit v nové měření nebo v restart platformy.

> Měření musí začínat v S-CRTM. Ten změří sám sebe a výsledek je uložen do **PCR 0**. Do tohoto PCR jsou také uložena měření každého spustitelného kódu patřícího přímo k základní desce (S-CRTM, POST BIOS a Option ROM). Pokud nemůže být měření provedeno, musí CRTM do registrů 0-7 změřit hodnotu `01h` nebo, pokud ani to nelze, upozornit uživatele.

PCR	použití
0	CRTM, BIOS a rozšíření hostitelské platformy
1	konfigurace hostitelské platformy
2	kód přídavných ROM
3	konfigurace přídavných ROM
4	IPL kód
5	IPL konfigurace a data
6	změny stavu napájení
7	vyhrazený výrobci hostitelské platformy
8-15	vyhrazené pro statický OS a hostitelskou platformu

Tabulka 3.2: Použití PCR

K proveditelnému kódu nepatří různá nastavení hardwaru a tudíž nejsou zahrnuta do měření pro PCR 0, ale až do **PCR 1**. Nicméně je možno vynechat veškerá data (například sériová čísla), která by mohla ohrozit soukromí uživatele. Zároveň se musí vynechat všechny automaticky měněné údaje, jako jsou čítače a hodiny. Toto měření provádí BIOS.

Do **PCR 2** se následně měří spustitelný kód obsažený na přídavných ROM (Option ROM) uložených na zařízeních nepatřících do hostitelské platformy (Non-Platform Adapters). Viditelné části těchto ROM měří BIOS, ostatní (pro BIOS dříve nepřístupné) měří před jejich spuštěním již změřený kód obsažený na daném zařízení.

Do **PCR 3** (analogicky k PCR 1) je uloženo měření dat a konfigurací z přídavných ROM. Měření provádí kód na příslušné přídavné ROM.

PCR 4 obsahuje měření IPL kódu (Initial Program Load), tedy kód typicky uložený na začátku MBR tabulky (Master Boot Record). Toto měření a následné předání řízení do IPL představuje posun v řízení od BIOS k operačnímu systému. PCR 4 obsahuje měření jak každého vykonaného IPL, tak dalšího kódu nahraného pomocí IPL. Toto měření se netýká nastavení.

Nastavení a data, která jsou relevantní vzhledem k důvěryhodným vlastnostem platformy (např. výběr bootavacího oddílu), jsou IPL kódem uloženy do **PCR 5**. Zároveň do tohoto PCR mohly být ukládány statické informace z IPL (například geometrie disku), a to ještě před předáním řízení do IPL.

PCR 6 je dedikováno k měření přechodu z různých stavů napájení, přesněji pro měření přechodů při buzení systému z uspání nebo hibernace. Toto PCR může k ukládání měření používat jak fáze před startem operačního systému, tak operační systém po startu. Předělem je měření hodnoty `EV_SEPARATOR`, která je změřena před předáním řízení nastartovanému operačnímu systému.

PCR 7 je určeno pro budoucí využití a prozatím je jeho použití rezervováno

výrobci hostitelské platformy. Nicméně toto PCR nesmí být využito pro pečetění a další operace a obecně je jeho použití nedoporučeno.

3.9 Fungování TPM

3.9.1 Operační stavy

TPM má tři různé operační stavy a každý z nich může nabývat dvou hodnot. Tyto operační stavy, respektive jejich kombinace, dávají osm možností, jak může být limitováno fungování TPM. Od plné funkčnosti po stav, kdy jediné co TPM může provádět je změna stavu.

Může nastat situace, kdy se jednotlivé operační stavy překrývají, tzn. TPM je ve stavu, kdy jedna hodnota příkaz povoluje a druhá zakazuje. V takovémto případě má vždy přednost zákaz provádění příkazu. Jakýkoliv příkaz může být spuštěn pouze tehdy, pokud mu vyhovují všechny operační stavy.

Prvním z operačních stavů má hodnoty **zpřístupněný** (enabled) a **znefunkčněný** (disabled). Pokud je TPM zpřístupněno, pak může provádět standardní příkazy, ale ve znefunkčněném stavu může pouze předávat informace o TPM a měřit nové hodnoty do PCR.

Druhý operační stav udává, zda je TPM **aktivní** (activated) nebo **neaktivní** (deactivated). Neaktivní TPM se chová obdobně jako znefunkčněné, ale může měnit své operační stavy (například změnit vlastníka). Aktivní stav je obdobný zpřístupněnému a TPM může provádět všechny operace.

Posledním operačním stavem je vlastnictví TPM, které může nabývat hodnot **vlastněný** (owned) a **nevlastněný** (un-owned). TPM je ve stavu vlastněný v situaci, kdy TPM má vygenerovaný klíč pro potvrzení (viz 3.4.2) a vlastník se prokázal autentizačními údaji.

3.9.2 Start TPM

Start TPM začíná příkazem `TPM_Init`, který je zavolán při změně napájení TPM. Obvykle to značí start platformy, ale může to být i probouzení z hybernace nebo uspání. Tento příkaz je také možno předat do TPM pomocí fyzického restartu TPM (např. pomocí hardwarového přepínače). Inicializovaný TPM následně čeká na příkaz `TPM_Startup` a odmítá jakékoliv jiné volání.

`TPM_Startup` má parametr se třemi možnostmi, určující v jakém stavu bude TPM po ukončení jeho vykonávání. Možnosti jsou Clear, State a Deactivated. První možnost znamená normální start, obvykle po restartu platformy, druhá určuje obnovení do předem uloženého stavu a poslední uvádí TPM do stavu, kdy nepřijímá jiné příkazy než `TPM_Init`.

`TPM_Startup`, následující po `TPM_Init`, provádí omezené sebetestování TPM. Testování je omezeno hlavně rozsahem funkcí, které je po něm možno důvěryhodně provádět. V jeho rámci se testuje hašování pomocí rodiny příkazů `TPM_SHA1xxx`

a měření a ukládání hodnot do PCR pomocí `TPM_Extend`. Další povolené příkazy v této fázi jsou `TPM_Startup`, `TPM_GetCapability` pro hlášení vlastností TPM a `TPM_ContinueSelfTest` a `TPM_SelfTestFull` pro zahájení plného sebetestování TPM. Po ukončení omezeného testování může TPM buď zahájit další testování, nebo fungovat v limitovaném stavu, co se rozsahu možných příkazů týká.

Plné sebetestování TPM je zahájeno voláním `TPM_ContinueSelfTest` v případě startu TPM, nebo může být také voláno později příkazem `TPM_SelfTestFull`. Plné testování zahrnuje i části již dříve testované v rámci omezeného. Po jeho úspěšném ukončení je TPM v plně funkčním stavu (dle parametru `TPM_Startup`), za předpokladu splnění dalších podmínek (má vlastníka, ustanovený klíč pro potvrzení atp.). Plné testování musí otestovat RNG, PCR (načítání i ukládání hodnot), EK včetně jeho funkčnosti (tzn. fungování RSA), integritu chráněných vlastností (resp. jejich kódu) a jakékoliv známky průniku. Dále by mělo být provedeno testování hašování, šifrovacích algoritmů, zabalení klíčů a všechny další interní mechanismy.

3.9.3 Logování

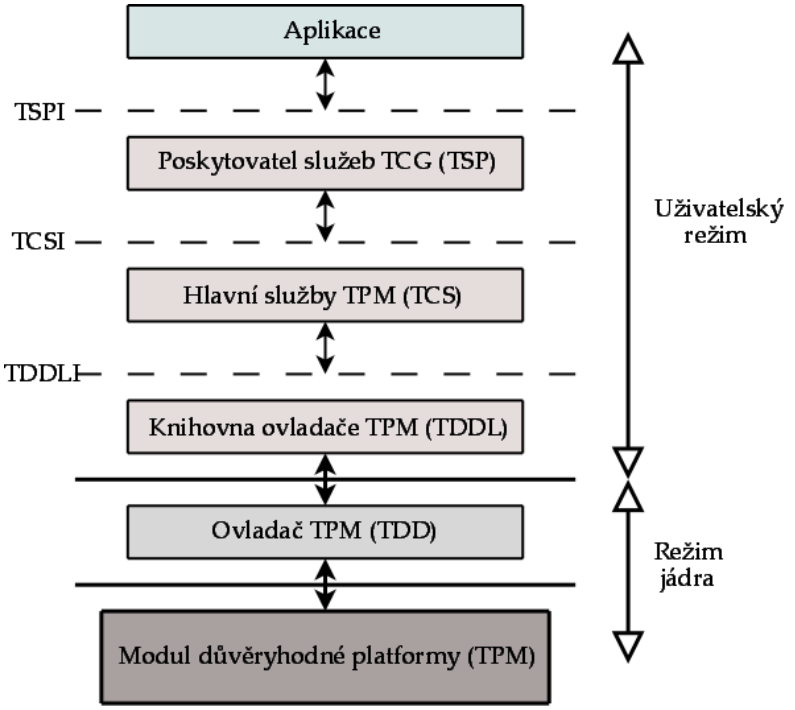
Důležitou funkčností pro kontrolu chování TPM a platformy jako takové je logovaní měření ukládaných do PCR. Samotné měření uložené bezpečně v PCR není průkazné, co se historie operací týká. Proto vytváří TPM log uložených měření (Stored Measurement Log, dále jen SML).

SML se skládá z jednotlivých záznamů, které jsou vytvářeny podobně jako hodnoty pro PCR, tzn. existuje hodnota, která je opakovaně inkrementována měřením a hašována. Takto je i v SML udržována následnost. V záznamu jsou další položky určující typ události a její další presné určení.

SML je ukládán nezávisle na TPM a není ani požadována jeho ochrana. Vzhledem k předpokladu jednocestnosti použité hašovací funkce je možnost útoku na log omezená na útoky typu DoS (Denial of Service). SML je tedy skladován mimo TPM, ale zároveň je nutné ho generovat i v době před startem operačního systému. Tento problém je řešen pomocí vyhrazeného prostoru v operační paměti, do které je ukládán před startem OS, který je poté zodpovědný za překopírování této paměti jako základ samotného SML.

3.10 Softwarový zásobník TCG

Důležitou součástí TPM jsou ovladače, knihovny a rozhraní umožnující s ním komunikovat. Tyto softwarové součástí jsou zahrnuty do softwarového zásobníku TCG (TCG Software Stack, dále jen TSS) [18], jehož součásti a jejich komunikace jsou zobrazeny na obrázku 3.5. Standardy TSS jsou stejně jako celá architektura TPM nezávislé na použitém operačním systému, nicméně některé části se mohou v závislosti na použitém OS měnit. Stejně tak mohou tyto části podporovat nepovinné operace podporované konkrétním TPM, pokud to nenarušuje fungování povinných částí.



Obrázek 3.5: Schéma softwarového zásobníku TCG

3.10.1 Ovladač TPM

Ovladač TPM (TPM Device Driver, dále jen TDD) je základním kamenem komunikace s TPM. Zpravidla je dodáván konkrétním výrobcem TPM, protože jeho základem je kód rozumějící si s daným TPM. Tento ovladač musí být jedinou komponentou, která může komunikovat přímo s TPM, a veškerá komunikace je vedena jeho prostřednictvím. Ovladač TPM běží standardně v režimu jádra.

Pro komunikaci s vyšší vrstvou je výrobcem definované rozhraní TDDI (TCG Device Driver Interface, dále jen TDDI), které zprostředkovává přenos bytových proudů mezi TDD a knihovnou ovladače TCG (TCG Device Driver Library, dále jen TDDL). Tato knihovna pracuje již v uživatelském režimu a poskytuje na systému nezávislý přístupový bod k TPM. Tím stírá rozdíly mezi jednotlivými TPM. Zároveň musí být tato knihovna v systému jako jedinečný jednovláknový proces, aby se zamezilo duplikování příkazů pro TPM. TDDL může v jednu dobu zpracovávat pouze jeden příkaz, výjimkou je přerušení v dané chvíli vykonávaného příkazu.

Tddli (TDDL Interface) je rozhraní pro komunikaci s TDDL sestávající ze tří typů funkcí: servisní (zajišťující komunikaci s TDD), nepřímé (zjišťování a nastavování atributů TPM, TDD nebo TDDL) a přímé (přenášení nebo rušení příkazů pro TPM).

3.10.2 Hlavní služby TCG

Hlavní služby TCG (TCG Core Services, dále jen TCS) jsou modul poskytující nejenom přístup k funkcím TPM, ale zároveň spravující různé prostředky TPM, jako jsou sloty klíčů, vícevláknový přístup nebo správa osvědčení a logů. TCS fungují v uživatelském režimu a jsou zpravidla vykonávány jako systémová služba. TCS musí v rámci platformy existovat pouze jednou a jsou připojeny skrze Tddli k TDDL.

K TCS patří interface pro použití vyšší vrstvou. Toto rozhraní je značeno jako Tcsi (TCS Interface). Tcsi je jednoduché rozhraní, nicméně podporující vícevláknový přístup k TCS (nesmí být výhradní, tzn. entit z vyšší vrstvy může s TCS komunikovat více). Všechny operace pomocí Tcsi (a TCS) musejí mít zaručenou atomicitu.

Existuje také několik dalších vyhraněných komponent zahrnutých v TCG. První z nich je Správce kontextu TCS (TCS Context Manager, dále jen TCSCM) poskytující dynamické ukazatele na množiny souvisejících funkcí. To je důležité pro správné využití prostředků TCS.

Druhou komponentou je Správce klíčů a osvědčení TCS (TCS Key & Credential Manager, dále jen TCSKCM). TCSKCM provádí správu klíčů a osvědčení, aby k daným prostředkům mohli přistupovat různé entity. Zároveň zajišťuje, že k případným soukromým informacím (například v osvědčení klíče pro potvrzení) mají přístup pouze autorizované entity.

Správce událostí TCS (TCS Event Manager, dále jen TCSEM) je TCS součástí zodpovědnou za správu a provázání struktur `TSS_PCR_EVENT` s danými PCR. Zároveň předává kopie těchto záznamů pro použití aplikacemi nebo vyšší vrstvou.

Poslední částí TCS je generátor bloku parametrů TCS (TCS Parameter Block Generator, dále jen TcsipBG), který převádí standardní vstup do TCS (tedy funkce podobné vyšším programovacím jazykům) do bytového proudu očekávaného na straně TPM.

3.10.3 Poskytovatel služeb TCG

Poskytovatel služeb TCG (TCG Service Provider, dále jen TSP) je modul poskytující vlastnosti TPM pro jednotlivé aplikace. Kromě poskytování rozhraní pro aplikace také TSP obsahuje některé funkce, které nepodporuje samotné TPM, např. ověření podpisu. Jako takový je to proces běžící v uživatelském režimu. Zároveň zpravidla existuje jako jeden proces pro každou aplikaci spolupracující s TPM, tj. počet současně běžících TSP není omezen.

Rozhraní k TSP je označováno jako TSPI (TSP Interface). Poskytuje objektově orientovaný vysokoúrovňový přístup k TSP. TSPI by mělo být přítomno v procesu přistupující aplikace, aby mohlo být (v závislosti na implementaci) důvěryhodné například pro vstup autorizačních dat.

Jednou ze dvou hlavních komponent TSP je správce kontextu TSP (TSP Context Manager, dále jen TSPCM), sloužící obdobně jako TCSCM k správě prostředků TSP pomocí dynamických ukazatelů na funkce.

Druhou komponentou TSP jsou kryptografické funkce TSP (TSP Cryptographic Functions, dále jen TSPCF), které implementují vyšší kryptografické funkce a operace, jako například hašovací algoritmy.

Kapitola 4

Využití TPM pro TrueCrypt

Tato kapitola má za cíl představit různé možnosti využití TPM pro šifrování dat TrueCryptem. Šifrování pomocí TrueCryptu je silný nástroj, hardwarová podpora některých částí tohoto řešení je však velmi vhodná pro zvýšení celkové bezpečnosti.

4.1 Model útočníka

Jako první je nutné stanovit alespoň rámcové možnosti útočníka, od nichž odvozuji přínos předložených opatření a použití TPM.

Základem možností útočníka je přístup k cílovému počítači. Zde standardně předpokládám, že má možnost dostat se pouze k vypnutému počítači. Poznámky ke změně ohledně přístupu k zapnutému počítači jsou v části 4.2.

Druhým předpokladem je nemožnost útočníka pozměnit kořeny důvěry TCG (viz 3.5). Z tohoto pohledu by modifikování RTS a RTR znamenalo pozměnit samo chování TPM a změna v RTM by znamenala upravení té části BIOS, která ho tvoří. Možnosti upgradu nebo změny těchto částí samozřejmě musí existovat (co se týká softwaru nebo firmwaru), nicméně tyto možnosti musí být chráněné výrobcem TPM, například pomocí asymetrické kryptografie. Pro účely tohoto textu tudíž předpokládám nemožnost neoprávněné manipulace s kořeny důvěry, neboť takový útok by znemožnil jakékoliv smysluplné využití TPM a je tudíž mimo rámec této práce.

4.2 Cold boot útok

Tento útok, prezentovaný v [19], je založen na fyzikálních vlastnostech současných operačních pamětí, umožnujících přečtení dat v nich uložených i určitou dobu po odpojení přívodu energie.

Cold boot útok je i přes jeho nepopiratelnou proveditelnost někdy podceňován z přesvědčení, že pokud má útočník volný fyzický přístup k běžícímu počítači, tak již jakékoliv následné narušení bezpečnosti nehraje roli, jak již při samotném provádění útoku má přístup k datům v počítači. To nemusí být nutně pravda, například v případě uzamčeného šetřiče obrazovky. Znalost šifrovacích klíčů je navíc důležitější, než znalost dat v daném časovém okamžiku, protože znalost klíčů umožňuje přístup k datům i následně, nezávisle na změnách hlavičky a uživatelského hesla. Jedinou obranou po ztrátě hlavních klíčů je vytvoření nového TrueCrypt svazku a překopírování dat (viz 2.2.3), což může být zejména při systémovém šifrování náročný úkol.

Útok v neinvazivní podobě probíhá tak, že je počítač vypnut a znovu nastartován

z bootovatelného média. Nové prostředí potom vytvoří kopii operační paměti a pokusí se z ní získat data. V našem případě jsou nejohroženější a nejcennější hlavní klíče pro všechny v dané chvíli namontované TrueCrypt svazky. Tyto klíče jsou v paměti přítomné neustále při běhu systému (pokud není daný svazek odmontován) a to nezávisle na stavu systému (například uzamčený šetřič obrazovky).

Na neinvazivní podobu útoku zareagovala TCG novou specifikací [20], v které ukládá S-CRTM funkčnost nutnou pro odstranění této zranitelnosti. Tou je nutnost nulovat, respektive přepisovat, operační paměť již na úrovni S-CRTM, před předáním řízení startu platformy dále. Implementačně je nulování zajištěno pomocí bitu specifikujícího požadavek na přepsání paměti (the Memory Overwrite Request, dále jen MOR), který je součástí S-CRTM. Zároveň je pro urychlení procesu přepsání paměti nabídnuta možnost, kdy by operační systém při správném vypnutí sám vynuloval paměť a dal TPM najevo, že není potřeba při startu paměť čistit znovu.

Invazivní podoba Cold boot útoku předpokládá vyjmutí paměťových modulů z počítače a provedení kopírování jejich obsahu na jiném zařízení. Zároveň mohou být moduly podchlazeny, čímž se řádově snižuje rychlost degradace dat. Útok tohoto typu není řešen ani výše popsanou specifikací, ani jinou možností poskytovanou TPM. Nicméně vyjmutí pamětí je nápadná operace a její provedení není zaměnitelné s normální prací (jako je tomu u neinvazivní formy útoku), díky čemuž je tento scénář méně pravděpodobný.

4.3 Generátor (pseudo)náhodných dat

Další oblastí, ve které může TPM podpořit bezpečnost, je generování náhodných resp. pseudonáhodných dat. TrueCrypt potřebuje ke svému fungování velké množství náhodných dat, ať již pro generování klíčů, nebo pro počáteční plnění svazků (viz 2.1.1). Na náhodnosti do značné míry závisí bezpečnost celého řešení, nejenom co do možnosti věrohodně popřít samotnou existenci dat (viz 2.1.3), ale v případě nekorektního generování klíčů a solí i bezpečnost jako taková .

TrueCrypt ve verzích pro MS Windows generuje psudonáhodná data pomocí získávání dat z mnoha různých zdrojů pseudonáhodných dat, jako jsou různé ukazatele, statistické informace o discích a systému nebo generátor pseudonáhodnych dat obsažený v MS Windows CryptoAPI. Tyto data jsou ukládána do dvou míst (Fast Pool, Slow Pool) a následně filtrována pomocí mixovací funkce. Díky tomu se rozptyluje význam jednotlivých vstupů na výslednou sekvenci.

RNG obsažený v TPM nemusí být nutně generátorem skutečně náhodných dat (viz 3.3.7), nicméně jeho použití je vhodné. Přidáním důvěryhodného zdroje náhodných dat je ještě více rozptýlena významnost ovlivnitelných zdrojů dat vstupujících do mixovací funkce. Vzhledem k množství a rychlosti generování náhodných dat pomocí TPM je tato možnost jediná použitelná. Využívání jenom dat z TPM by extrémně zpomalovalo celé řešení (zvláště v případě generování nových svazků). I pro generování jednotlivých klíčů, kdy by nebyl problém v samotné rychlosti, je lepší využívat více zdrojů dat a jejich filtrování.

4.4 Ochrana TrueCrypt klíčů

Jednou z hlavních schopností TPM je ochrana dat (speciálně klíčů). TPM využívá kořene důvěry pro ukládání (viz 3.5.1) a hierarchie asymetrických klíčů (viz 3.6.3) k vytvoření infrastruktury klíčů, pomocí nichž je možno šifrovat data. Ta jsou šifrována pomocí svazování nebo pečetění. TPM nicméně neochraňuje velká množství dat, takže přímo chránit v rámci jedné operace lze jen data do velikosti 213B při svazování, resp. do 153B při pečetění [18, 21]. Toto úložiště se zpravidla používá pouze pro ochranu klíčů nebo podobně důležitého materiálu.

Pro vyjasnění pojmů budu v této kapitole používat výraz „klíče“ pro klíče TPM, zatímco klíče pro TrueCrypt buď jako „TrueCrypt klíče“, nebo jednotlivě jako „klíč hlavičky“ a „hlavní klíč“ (viz 2.2.3). Pojem „ochrana“ označuje pečetění nebo svazování.

4.4.1 Hierarchie a ukládání klíčů

Důležitou věcí pro zvážení je výběr hierarchie klíčů a jejich ukládání, kde hraje roli hlavně místo, kam klíče odkládat.

Klíče TPM jsou ukládány mimo TPM pomocí hierarchie klíčů (viz 3.6.3). Ta má základ v SRK, což je jediný klíč (spolu s EK), jehož přítomnost v TPM je zaručena. SRK se zároveň nemůže vyskytovat nikde jinde. Ostatní klíče jsou ukládány do struktury `TPM_KEY12`, z které musí být před použitím nahrány do TPM. Toto nahrání je podmíněno znalostí autorizačních údajů pro všechny rodičovské klíče, které ještě nejsou nahrány v TPM. Poté je pomocí rodičovského klíče dešifrována privátní část nahrávaného klíče a po autorizaci (tentokráte pro daný klíč) je klíč připraven k použití.

Díky této architektuře mohou být klíče ukládány v podstatě kdekoliv a přesto jsou chráněny. Pro použití v TrueCryptu je problematické, že struktura `TPM_KEY12` má vnitřní logickou strukturu, podle které je identifikovatelná. To má za následek, že tato struktura by pro zachování principu věrohodného popření existence dat neměla být přímo připojena k samotnému TrueCrypt svazku.

Důležitá je také použitá hierarchie klíčů. Není vhodné používat příliš hlubokou hierarchii, protože je nutné autorizovat použití všech klíčů až ke kořenu. Zároveň nesmí být využit přímo SRK jako klíč. Vzhledem k nutnosti autorizace celého řetězce je nejvhodnější využívání klíčů, kteří jsou přímými potomky SRK. Výjimkou může být situace při šifrování pevného disku.

4.4.2 Migrovatelnost klíčů

Migrovatelnost použitých klíčů, jako obecná vlastnost všech klíčů nastavovaná při vytváření klíče, je důležitou součástí pohledu na ochranu klíčů. Pokud je klíč nemigrovatelný, pak nemůže být použit na jiné platformě, resp. jiným TPM. Nemigrovatelné klíče jsou spojeny s TPM pomocí hodnoty `tpmProof`.

Pro účely podpory TrueCryptu je vhodné zachovat obě varianty, protože propojenost s danou platformou je silně závislá na předpokládaném využití daného svazku. Například pro šifrování mobilního média (například USB úložiště) je zřejmá nutnost

použití migrovatelných klíčů, resp zvážení, zda je podpora pro TPM ochranu klíčů vhodná (nebylo by možné svazek připojit na strojích, pro které nemáme přemigrovaný klíč). Z tohoto důvodu je nutné, nejenom pro zpětnou kompatibilitu, uchovat také stávající možnosti vytváření svazků bez podpory TPM.

Při použití migrovatelných klíčů, kdy je předpokládána kompatibilita s verzí 1.2, je vhodné využít certifikované migrovatelné klíče (viz 3.6.1), které umožňují větší kontrolu nad migračním procesem.

4.4.3 Svazování a pečetění

Svazování (Binding, viz 3.7.1) a pečetění (Sealing, viz 3.7.3) jsou TPM operace pro ochranu dat, připadající do úvahy pro využití u TrueCryptu. V závislosti na použité operaci se mění nejenom okolnosti, za kterých jsou daná data uvolněna (dešifrována), ale také maximální velikost těchto dat (v jedné operaci).

Svazování svazuje data pouze s daným dešifrovacím klíčem. Pokud je klíč migrovatelný, tak jsou svázána pouze s ním, pokud nemigrovatelný, tak jsou pevně spjata i s daným TPM. Použitý klíč musí být typu `TPM_KEY_BIND`. Svazování při použití vytváří strukturu `TPM_BOUND_DATA`, která poskytuje při použití Optimálního asymetrického šifrovacího vyplňování (Optimal Asymmetric Encryption Padding, dále jen OAEP) u RSA klíče velikosti 2048b prostor 213B.

Svazovaní jako takové není TPM operace, je prováděno na úrovni TSP (viz 3.10.3) operací `Tspi_Data_Bind`, nicméně opačná operace `TPM_UnBind` je již operace na úrovni TPM. Vytváření TrueCrypt svazků je možnost prováděná pouze při spuštěném systému a nikoliv v rámci jeho startu (což nemusí být případ otevírání svazku), a tak pro TrueCrypt toto není zásadní omezení.

Pečetění svazuje daná data nejenom s nemigrovatelným klíčem, ale zároveň i se stavem platformy daným stavy určité množiny PCR. Díky tomu jsou zapečetěná data přístupná pouze pro systém na kterém byla zapečetěna a ten musí být ve správném stavu. Svazování probíhá pomocí operace `TPM_Seal` a opačné operace `TPM_Unseal`. Příslušná datová struktura `TPM_SEALED_DATA` poskytuje při použití OAEP a RSA klíče velikosti 2048 prostor pouze 153B. Pokles o 60B proti svazování je dán nutností ukládat data o příslušné množině PCR.

Rozhodnutí, zda použít svazování nebo pečetění, je zásadně spojeno s rozhodnutím o migrovatelnosti klíče a tedy i příslušného TrueCrypt svazku. Při použití migrovatelného klíče je nutné využít svazování. U nemigrovatelných klíčů je silnějším a bezpečnějším nástrojem pečetění, zvláště při správném výběru PCR. Nicméně na obsahu PCR může být závislý již klíč, kdy by pak bylo použití pečetění nadbytečné. Zde záleží na navržené hierarchii klíčů.

PCR vhodných pro pečetění klíčů při použití u TrueCryptu je několik. Rozhodně je vhodné zahrnutou PCR 0, obsahující S-CRTM (například i kvůli částečné obraně před Cold boot útokem, viz 4.2). Ostatní PCR zaznamenávající důvěryhodný start systému (viz 3.8.2) mohou být zahrnuty, pokud je vhodná maximální ochrana.

Velmi vhodnou možností je PCR 23, který je možné používat a resetovat libovolnou aplikací nebo systémem. Tento PCR by bylo možné využít pro měření integrity TrueCrypt ovladače a uživatelského rozhraní, jako ochranu před jejich záměnou.

4.4.4 Ochrana hlavičky

Ochrana celé hlavičky, myšleno její svázání nebo pečetění s pomocí dalšího klíče, je první možností, jak TrueCrypt svazek chránit. Uvažované použití je takové, že místo použití klíče hlavičky by byla celá hlavička chráněná pomocí TPM a tím by byla zajištěna ochrana nejenom hlavního klíče, ale i dalších dat v hlavičce. Nicméně takové řešení naráží na řadu problémů.

Prvním problémem je velikost hlavičky. Standardní TrueCrypt svazek má 4 hlavičky (hlavní a záložní pro standardní a skrytý svazek), každou o velikosti 65536B. To by u každé z hlaviček odpovídalo 308 operacím svazování, resp. 429 pečetění.

Bylo by možné zmenšit hlavičky a vynechat z nich nepotřebné údaje (velká část hlavičky je rezervovaná pro další použití a u systémového šifrování se vůbec nepoužívá). Na druhou stranu by bylo nutné vložit do hlavičky údaje o typu šifrování, protože TrueCrypt při montování svazku tyto data zjišťuje na základě úspěšného dešifrování hlavičky metodou pokus-omyl.

I pokud by každá hlavička zabírala místo pouze v jedné datové struktuře (TPM neumí struktury nijak řetězit, každý příkaz `TPM_Unbind` nebo `TPM_Unseal` zpracovává pouze jednu strukturu), pak stále bude každý svazek potřebovat čtyři TPM struktury obsahující hlavičky. Zároveň musí existovat k tomuto svazku dvě struktury `TPM_KEY12`, jedna pro standardní svazek, druhá pro skrytý. Tyto objekty by bylo možné uložit do samostatné „klíčové struktury“, která by ale musela být oddělena od samotného svazku.

Problémem, vycházejícím z rozdílných návrhových principů u TrueCryptu a TPM, je identifikovatelnost datových struktur TPM. Všechny uvažované datové struktury mají zřejmou vnitřní strukturu, odlišující je od náhodných dat. Proto, kdykoliv je ve hře nutnost věrohodně popřít existenci dat, resp. TrueCrypt svazku, není tato možnost výhodná.

4.4.5 Ochrana klíče hlavičky

Další možností, jak chránit TrueCrypt svazky pomocí TPM, je ochrana klíče hlavičky. Obdobně jako u ochrany celé hlavičky je zde možnost chránit pomocí TPM všechna data, obsažená v hlavičce svazku. Použily by se dvě struktury `TPM_KEY12` pro klíče a čtyři struktury typu `TPM_BOUND_DATA` nebo `TPM_SEALED_DATA` pro uložení klíčů hlavičkek. Tyto struktury by bylo třeba oddělit od TrueCrypt svazku, vzhledem k jejich vnitřní logické struktuře. Bylo by je ale možné spojit do jedné „klíčové struktury“, skrze kterou by se daný svazek dal pomocí TPM namontovat.

Ochrana klíče hlavičky je velmi podobná ochraně hlavičky v případě, že by byla hlavička zmenšena na velikost ukladatelnou v jedné struktuře. Pouze je zde přidán jeden krok navíc při montování svazku, kterým je dešifrování klíče hlavičky.

Toto řešení má výhodu v jednoduchosti implementace, kdy by nebylo nutné měnit prakticky nic, kromě způsobu, jakým je derivován klíč hlavičky z uživatelského hesla nebo klíčového souboru. Místo toho by se těmito údaji autorizoval TPM klíč pro dešifrování hesla hlavičky a dál by montování svazku probíhalo standardním způsobem.

Při ochraně klíče hlavičky by byla potřeba pouze malá změna v samotných prin-

cipech TrueCryptu, ale na druhou stranu vzniká nutnost existence externí struktury pro uložení šifrovaných hlaviček a klíče, která by v případě připojení k nenamontovanému TrueCrypt svazku narušovala princip věrohodného popření existence dat.

4.4.6 Ochrana hlavního klíče

Třetí možností, jak chránit TrueCrypt svazky skrze ochranu klíčů, je přímo ochrana hlavních klíčů v rámci každé z hlaviček. Tato možnost se od předchozích liší tím, že neumožňuje pomocí TPM chránit data v hlavičkách svazku, ale chrání pouze hlavní klíč a skrze něj samotný obsah TrueCrypt svazku.

Samotné otevírání hlavičky TrueCrypt svazku by v tomto případě zůstalo zachováno. Nicméně hlavička nebude obsahovat hlavní klíč v otevřené podobě. Místo toho zde bude uložena struktura `TPM_KEY12` a `TPM_DATA_BIND`, resp `TPM_DATA_SEAL`. Pomocí TPM by pak byl nahrán klíč z hlavičky a jeho prostřednictvím získáno heslo z druhé struktury.

Hlavička obsahuje dostatek prostoru pro tyto struktury, vzhledem k množství místa rezervovaného pro budoucí využití. Jediný problém s místem by byl u šifrování systémového disku, kde není hlavička dostatečně prostorná, resp. neobsahuje rezervované místo. Tuto hlavičku by tedy bylo nutné rozšířit.

Ochrana TrueCrypt svazku pomocí ochrany hlavičky je varianta chránící přímo nejdůležitější část hlavičky, při současném zachování výhod architektury TrueCryptu. Vzhledem k tomu, že není nutné, aby existovala externí datová struktura pro uložení klíče a dalších TPM struktur, zůstává zachována náhodná struktura dat a tudíž možnost věrohodně popřít existenci TrueCrypt svazku. Zároveň tento přístup nemění další mechanismy, jako je například možnost vytváření nové hlavičky.

Co se týká náročnosti na implementaci, vzniká zde problém pouze u použití pro šifrování systémového disku, vzhledem k odlišné architektuře hlaviček a nedostatku místa. Pro standardní svazky nevyžaduje velké změny ani ve formátu hlavičky, ani ve vytváření a montování svazků.

4.5 Využití čipových karet

V současné době (verze 6.1a) podporuje TrueCrypt kryptografické tokeny komunikující přes PKCS #11 [22]. Kryptografickým tokenem se zde myslí jakékoliv zařízení splňující danou normu, primárně jsou to však kryptografické čipové karty.

TrueCrypt umožňuje ukládání klíčových souborů na čipové karty, díky čemuž je klíčový soubor lépe chráněn. Kromě toho je možné generování nových klíčových souborů přímo na tokenu, kdy odpadá možnost diskreditace při uložení na disk (předpokládám bezpečnou komunikaci mezi čtečkou a počítačem).

Tyto a další možnosti, hlavně ukládání celé TrueCrypt hlavičky na čipové karty, jsou podrobněji rozebrány v [23, 24]. Nicméně tyto práce operují ještě se starou verzí hlavičky (nová přišla ve verzi 6.0), tedy s hlavičkou o velikosti pouze 512B, a bez existence záložních hlaviček. Samotná velikost hlaviček je problém řešený i v části 4.4, kdy je pro použití s TPM nutný návrat k této nebo obdobné velikosti. Záložní hlavičky nijak dále neovlivňují možné použití čipových karet, protože by bylo možné záložní

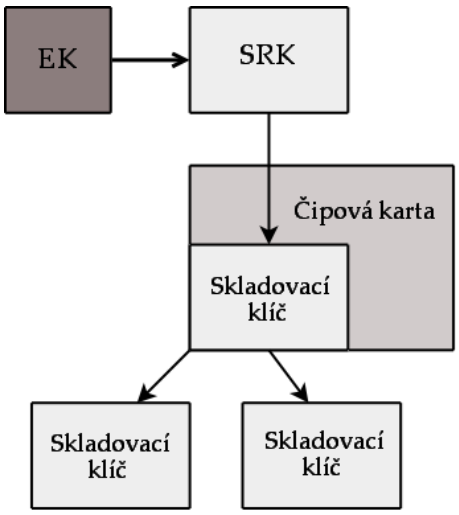
hlavičku dle rozhodnutí uživatele buď uložit, na úkor místa, nebo vynechat.

Zřejmým využitím čipových karet v případě použití TPM je ukládání TPM klíčů (a jiných dat v TPM strukturách), tedy použití čipové karty jako bezpečného úložiště. Možnosti ochrany klíčového materiálu byly probrány v předchozí části. Z tohoto pohledu jde čipové karty použít u každého z těchto přístupů.

U ochrany klíče hlavičky nebo celé hlavičky by se na čipovou kartu ukládaly „klíčové struktury" sestávající z dvou struktur `TPM_KEY12` a čtyř `TPM_BOUND_DATA`, nebo `TPM_SEALED_DATA` v případě pečetění. Při montování svazku by se pak uživatel autorizoval vůči svojí čipové kartě (připojené k počítači) a z ní by byl do TPM nahrán příslušný klíč (po autorizaci vůči klíči). Tím by byla (s platformou v důvěryhodném stavu) nahrána data, ať již klíč hlavičky nebo celá hlavička, a montování svazku by dále probíhalo standardním způsobem. U ochrany hlavního klíče by bylo nutné uložit obě hlavičky (normálního a skrytého svazku).

Při všech těchto možnostech je nutné zmenšit hlavičku a zároveň se rozhodnout, zda ukládat záložní hlavičky nebo ne. Toto rozhodnutí by mělo být na uživateli. Ideální by bylo mít záložní hlavičku uloženou na jiné čipové kartě, která by byla uložena na bezpečném místě pro případ ztráty.

Čipovou kartu je také možné použít pro uložení TPM klíče, následně použitého pro otevírání všech ostatních klíčů. Díky tomu by se na kartu posunul efektivní kořen hierarchie klíčů, jak je naznačeno na obrázku 4.1.



Obrázek 4.1: Schéma posunutí základu hierarchie klíčů na čipovou kartu

Další možnost, která se v těchto souvislostech nabízí, je ukončení montování svazku v případě odejmutí karty. Nepřítomnost karty ve čtečce indikuje nepřítomnost obsluhy a tedy v nejhorším případě útok na daný počítač. Proto by bylo vhodné pro takto namontované TrueCrypt svazky zavést opakované dotazování na přítomnost čipové karty se správnými údaji. K zajištění důvěryhodnosti takového dotazování je vhodné ho implementovat pomocí monotónního počítadla (viz 3.3.9), které zajišťuje důvěryhodný zdroj časového signálu. Pokud bude inkrementování daného počítadla

zajišťovat TrueCrypt ovladač a zároveň klíč k TrueCrypt svazku bude vázán na důvěryhodné měření tohoto ovladače, je důvěryhodost zajištěna pomocí TPM.

Ověřování by mělo být založeno jak na čase (inkrementování vzhledem k systémovému času), tak na počtu přístupu k danému svazku. Tím by byla zajištěna ochrana (alespoň částečná) proti útokům skrze systémový čas.

Kapitola 5

TPM a šifrování systémového disku

Šifrování systémového disku je velmi silný prostředek pro zajištění ochrany dat, řešící mnoho problémů vznikajících s pouhým šifrováním datových svazků nebo souborů, jako ukládání dočasných souborů (viz 2.3.2). Nicméně šifrování systémového disku je stále náchylné k některým problémům, přičemž je možné je pomocí TPM částečně nebo úplně eliminovat. Zároveň je možné využít TPM pro posílení stávající ochrany.

5.1 Věrohodné popření systémového šifrování

Systémový disk šifrovaný pomocí TrueCryptu je z velké části stejný jako jiné TrueCrypt svazky. To znamená, že je z vnějšku nerozlišitelný od náhodných dat. Nicméně existuje jedna část tohoto řešení, která musí být již z principu v otevřené (tedy nešifrované) podobě. Touto částí je zavaděč TrueCryptu (TrueCrypt Boot Loader).

Zavaděč systému, nebo také IPL kód, je kód zodpovědný za nahrání operačního systému. Je uložený (na PC) v oblasti MBR tabulky (Master Boot Record), která odpovídá prvním 512B na systémovém disku. Této části je při startu předána kontrola poté, co je kompletně nastartován BIOS. Zavaděč je první část disku, resp. operačního systému, která dostává kontrolu a je překlenovacím prvkem mezi pre-OS prostředím a operačním systémem.

V případě TrueCryptem zašifrovaného systémového disku je standardní IPL kód MS Windows (resp. jakýkoliv zavaděč, uložený v MBR tabulce) přepsán zavaděčem TrueCryptu, který vzhledem ke své velikosti zabírá i část prostoru za MBR tabulkou na prvním cylindru disku. Zavaděč TrueCryptu je zodpovědný za přijetí hesla od uživatele, dešifrování hlavičky systémového disku a nahrání hlavního klíče z hlavičky do paměti. Po vykonání těchto částí již startuje z přístupného disku ovladače a jádro operačního systému.

Z uživatelského hlediska start šifrovaného systému probíhá tak, že po předání řízení zavaděči je uživateli vypsán dotaz na heslo k šifrovanému systému. Po jeho zadání start systému probíhá standardním způsobem, jako kdyby systém nebyl šifrován.

Problém narušení principu věrohodnosti dat skrze existenci zavaděče TrueCryptu má několik možných řešení. Některé z nich jsou implementovány v rámci verze 6.1a, nicméně jsou pouze kosmetické a neobstojí před skutečnou analýzou. Změny jsou hlavně v oblasti zobrazení informací při dotazu na heslo, kdy nejnovější zavaděč umožňuje editaci celého výpisu nebo jeho úplnou eliminaci. Po startu pak uživatele vítá prázdná obrazovka s blikajícím kurzorem, čekající na zadání správného hesla.

Nicméně toto ukrytí obstojí pouze na první pohled. V samotném zavaděči je stále možné jednoduše přečíst v otevřeném textu řetězec „TrueCrypt Boot Loader“. Ani změna tohoto řetězce, ačkoliv je bez újmy na funkčnosti možná, nemění zavaděč natolik, aby nebyl rozpoznatelný podle svého chování a kódu.

První řešení nevyužívá funkčnosti TPM a spočívá v umístění zavaděče na externí médium, například záchranný disk (vytvářený povinně jako součást šifrování systémového disku) nebo USB token. Díky tomu je pak počítač bez hesla a externího média současně nerozlišitelný od nenainstalovaného a nepoužívaného počítače.

Druhým řešením je využití skrytého operačního systému zároveň s pomocí ochrany skrze TPM klíče, jak bylo nastíněno v části 4.4. Zde by systém použitý jako návnada startoval bez použití pre-boot hesla. K jeho nastartování by byl použit TPM klíč bez autorizačních údajů, pouze s přihlédnutím k důvěryhodnosti startu (viz 5.2). Díky tomu by v normálním běhu návnady nebylo na první pohled poznat, že je disk šifrován. Při odhalení šifrování by bylo jasné vysvětlení šifrování disku vzhledem k zaručení důvěryhodnosti systému. Ve skutečnosti by tedy nešlo o věrohodné popření systémového šifrování jako takového, ale o vysvětlení jeho existence.

Pro skutečně důvěrné fungování by v tomto případě bylo během pre-boot fáze startu platformy snímáno heslo, resp. stisk klávesy inicializující zadávání hesla. Po jeho zadání by platforma nastartovala do skrytého operačního systému, umožňujícího bezpečnou práci. Případně by bylo možné využít bootovatelného média s jiným TrueCrypt zavaděčem a s heslem k skrytému operačnímu systému.

5.2 Důvěryhodný start

Důvěryhodný start je funkčnost TPM vhodná pro posílení (lépe zavedení) důvěryhodnosti systémového šifrování. Jeho využití je vhodné hlavně pro zabezpečení zavaděče TrueCryptu, který, jako jediná část systémového šifrování uložená v otevřeném textu, je možným bodem průniku do řešení.

Zavaděč TrueCryptu je na disku uložen v otevřené podobě. Jeho vliv na věrohodné popření existence dat, resp. TrueCrypt svazku, je popsán v části 5.1. Nezávisle na problému věrohodného popření existence dat je zavaděč TrueCryptu možným slabým místem. Je to jediná nešifrovaná část, a proto je pro útočníka přístupná i při vypnutém systému (útočník může nastartovat systém z přenosného média). Zároveň je to část zodpovědná za přijímání hesla a montování systémového disku. Proto je její ohrožení, například programy pro ukládání hesel (Keylogger), kritické.

V standardním TrueCryptu není integrita zavaděče nijak testována, pokud vliv zavaděče nezapříčiní pád systému nebo nefunkčnost montování systémového disku. Proto je vhodné kontrolu integrity ošetřit. K tomu je ideální využít důvěryhodného startu. Jinou z možností je samozřejmě start systému pomocí zavaděče na nepřepisovatelném médiu, například pomocí záchranného disku, kdy není možné do samotného zavaděče zasáhnout zvenčí.

Měření TrueCrypt zavaděče je jako hodnota IPL ukládáno do PCR 4, jeho případné nastavení do PCR 5. Tato hodnota by měla být rozhodně zahrnuta do množiny PCR umožnující použití klíče pro šifrování systémového disku. Vzhledem k možnosti

upravení předchozí části řetězce důvěry je zároveň nutné kontrolovat i předchozí PCR. Měření PCR 0 také obsahuje obranu proti útoku studeným startem (viz 4.2).

5.3 Použití klíčů

Použití klíčů pro ochranu TrueCrypt svazků bylo z obecného hlediska rozebráno v části 4.4. Nicméně u šifrování systémového disku nastává několik odchylek od normálního běhu, které je nutno speciálně ošetřit.

Jednou z odchylek u systémového šifrování je jiné rozmístění a velikost hlaviček. Systémový svazek má hlavičku umístěnou na konci a pouze o velikosti 512B. Není zde ani hlavička pro skrytý svazek, ani záložní hlavičky. Zároveň je zde zavaděč, který, pokud není uložen na nepřepisovatelném mediu (viz 5.1 a 5.2), je možné použít pro externí uložení dat.

Díky této konstelaci je nejvhodnější ochranou systémového svazku pomocí TPM ochrana klíče hlavičky. Klíč TPM použitý pro jeho zašifrování, stejně TPM struktura obsahující klíč hlavičky a stejná dvojice pro případný skrytý operační systém, mohou být uloženy v rámci TrueCrypt zavaděče. V případě problémů s jeho integritou pak nebude uvolněno ani dešifrování hlavičky. Ideálním případem je uložení takto upraveného zavaděče například na bootovatelné USB.

Zároveň je také systémové šifrování, resp. TPM klíč pro přístup k systémovému svazku, vhodným místem pro zahájení hierarchie klíčů místo SRK. Tento TPM klíč by měl za rodiče přímo SRK a zároveň by sám byl rodičem následně vytvářených klíčů pro další TrueCrypt svazky. Stejně by bylo možné navázat druhý hierarchický strom na TPM klíč k skrytému operačnímu systému. Díky tomu by vznikly dvě oddělené hierarchie klíčů k TrueCrypt svazkům, oddělené od sebe podle úrovní soukromí a možného popření dat.

Samotné použití by záleželo také na použité ochraně klíčů, neboť u ochrany hlavičky (nebo jejího klíče) a při použití TPM klíčů bez autorizace by uživatel nemusel po bezpečném nastartování systémového šifrování zadávat žádná další autorizační data.

5.4 Čipové karty

Použití čipových karet pro podporu systémového šifrování naráží na obtížnost přístupu k čtečce v pre-boot prostředí. Vzhledem k různým ovladačům různých čteček je integrace těchto ovladačů do pre-boot prostředí nereálná (připojují se samozřejmě i problémy s místem). Z tohoto důvodu není možné využít čipové karty stejným způsobem, jako pro běžné svazky.

Nicméně je zde možnost pro použití při montování dalších svazků na základě hierarchie klíčů. Pokud bude základem hierarchie klíč k systémovému disku (potomek SRK), jak bylo představeno v 5.3, je vhodné vložit do hierarchie klíč uložený na čipové kartě. Tento klíč by byl potomkem hlavního klíče hierarchie a zároveň rodičem všech ostatních klíčů. Díky tomu by při diskreditaci klíče k šifrovanému operačnímu systému bylo stále nutné vlastnit čipovou kartu (a její autorizační údaje) pro namontování dalších (datových) svazků.

Při použití jednoho klíče z kořene hierarchie uloženého na čipové kartě sice vzniká potřeba autentizace uživatele vůči kartě (navíc k autorizacím použití klíčů), nicméně přidaná hodnota, co se týká bezpečnosti celého řešení, je velká.

5.5 Návrh použití

Zde jsou představeny dvě možná použití TPM pro podporu TrueCryptu vycházející z této a předchozí kapitoly. V prvním případě je kladen důraz na jednoduchost použití a důvěrnost dat, v druhém na možnost důvěryhodně popřít existenci určitých dat na počítači.

5.5.1 Důvěrnost dat

Cílem prvního scénáře je představit lehce spravovatelné zabezpečení počítače pomocí TrueCryptu a TPM, kdy není předpokládána nutnost důvěryhodného popření existence všech dat v počítači. TrueCrypt je zde tedy primárně použit pro zajištění důvěrnosti dat.

Základem je TPM podpora pro šifrování systémového disku, kdy je chráněn klíč hlavičky TrueCrypt svazku. „Klíčová struktura“ je uložena v rámci TrueCrypt zavaděče. Tento TPM klíč (nutný k získání klíče hlavičky) je potomkem SRK a je použit jako základ hierarchie klíčů. Od něj je následně odvozen další klíč, který je uložen na kryptografické čipové kartě. Klíče dalších svazků jsou vytvářeny jako potomci tohoto klíče.

Klíč hlavičky systémového TrueCrypt svazku je svázán s důvěryhodným stavem celé platformy (PCR 0-6), stejně jako klíč uložený na kartě. Oba klíče jsou chráněny zadáním autorizačních údajů.

Dalšími TrueCrypt svazky jsou datové disky, případně další TrueCrypt svazky, včetně možných skrytých svazků. Jejich ochrana je provedena pomocí ochrany hlavního klíče, kdy je dle rozhodnutí uživatele TPM klíč buď uložen v rámci hlavičky, nebo na čipové kartě. Existence autorizačních dat pro TPM klíče k ostatním svazkům je na rozhodnutí uživatele, nicméně tyto klíče jsou svázány s měřením TrueCrypt driveru ukládaného do PCR 23.

Ukládání na čipovou kartu je vhodné pouze pro extrémně důvěrná data, pro která je nutné odmontování při nepřítomnosti karty. Proto je zde aktivní dotazování vůči čipové kartě s případným odmontováním disku při neúspěchu. Pro ostatní svazky platí nutnost jednorázového připojení karty, kvůli nahrání jejich rodičovského klíče.

Z ostatních navržených možností využívá tento scénář náhodných dat generovaných pomocí RNG na TPM, kdy je zahrnuje jako jeden ze vstupů do TrueCrypt RNG. Pro ochranu klíčů je použito pečetění přes PCR 0-6 a PCR 23 s uloženým měřením TrueCrypt driveru. Základem hierarchie klíčů jsou dva nemigrovatelné klíče. Klíče od jednotlivých datových svazků mohou být v případě potřeby migrovatelné, ale preferované je certifikované migrování.

5.5.2 Popiratelnost dat

Druhým možným scénářem je použití pro zabezpečení možnosti důvěryhodně popřít data na počítači. Tento scénář je schválně doveden až do maximální možné ochrany bez ohledu na pohodlí uživatele, proto není pro praktické použití příliš vhodný.

Základem je vytvořená návnada v podobě standardně šifrovaného operačního systému. Návnada by měla být vytvořena přesně stejným způsobem jako v části 5.5.1. Díky tomu je dostatečně důvěryhodná. Pouze v hierarchii klíčů pro návnadu nejsou žádné klíče pro skryté svazky. Vhodné je, aby alespoň jeden ze svazků měl uložený klíč na čipové kartě, pro věrohodné vysvětlení její existence. Návnada také musí obsahovat dostatečně důvěrná data ve standardním TrueCrypt svazku na diskovém oddílu, kde je uložen skrytý operační systém.

Skrytý operační systém je startován pomocí bootovatelného média, které místo klíče k návnadě obsahuje klíče k skrytému operačnímu systému. Další části řešení jsou u skrytého operačního systému podobné jako u návnady, pouze je vyžadováno nejvyšší zabezpečení. Proto by všechny následné svazky měly být chráněny pomocí ukládání hlavičky na čipovou kartu a TPM ochrany hlavního klíče. Díky tomu zde není vyžadován klíč, tvořící u návnady přechod mezi klíčem k systému a klíči k datovým TrueCrypt svazkům. Všechny TPM klíče budou svázány s důvěryhodným stavem platformy a TrueCrypt ovladače a mají nastavená autorizační data.

Části, které jsou volitelné u návnady, jsou u skrytého operačního systému nastaveny na maximální bezpečnost. Proto by všechny použité klíče měli být nemigrovatelné. Stejně tak je samozřejmostí použití TPM RNG a odmontování svazků v případě vyjmutí karty.

Kapitola 6

Závěr

Tento demonstrátor se skládá zečtyř věcných kapitol, první se zabývá systémem TrueCrypt, druhá modulem důvěryhodné platformy a důvěryhodným počítáním a poslední dvě tvoří analýza možných propojení těchto dvou technologií.

V první části jsem popsal systém TrueCrypt, což je open-source program pro on-the-fly šifrování dat pro operační systémy MS Windows, Linux a OS X. Vzhledem k částečnému zacílení práce na šifrování systému, kdy tuto funkčnost TrueCrypt podporuje pouze u MS Windows, je práce zaměřená na fungování pod tímto systémem.

TrueCrypt je založen na principu svazků, datových oddílů odpovídajících souboru nebo diskovému oddílu. Tyto svazky jsou montovány k systému pomocí uživatelského hesla nebo klíčových souborů a následně používány jako standardní pevné disky. Díky tomu je z hlediska systému a aplikací třetích stran používání TrueCryptu jednoduché a bez další režie (kromě strojového času a paměti).

Jedním z důležitých principů, které se TrueCrypt snaží podporovat, je možnost věrohodně popřít existenci dat. S tímto cílem je navržena samotná struktura TrueCrypt svazku, která je nerozeznatelná od náhodných dat. Další funkčností pro důvěryhodné utajení existence dat je skrytý oddíl, kdy je v rámci jednoho TrueCrypt svazku vytvořen ještě druhý s jiným heslem. Obdobná situace je možná u šifrování systému při vytváření skrytého operačního systému.

V rámci první části jsem také provedl zhodnocení bezpečnosti TrueCryptu a to jak z hlediska principů na kterých je založen, tak vlastní implementace. Pokud je použito všech možností, pak je výsledné řešení bezpečné s ohledem na důvěrnost i věrohodné popření existence dat. Nicméně existuje několik problémů, kdy zvláště zavaděč TrueCryptu je úzké místo celého systémového šifrování.

Druhá část je věnována rozboru modulu důvěryhodné platformy (TPM) a možnostem, které v rámci konceptu důvěryhodného počítání nabízí. Důvěryhodné počítání si bere za úkol zavést do výpočetní techniky jednoduše měřitelnou a ověřitelnou důvěru. Standardy důvěryhodného počítání jsou publikovány jako otevřené, takže podléhají veřejnému zkoumání a dohledu. Základem praktické implementace těchto principů je právě modul důvěryhodné platformy. Ten poskytuje počítači hardwarové prostředky nutné k zavedení tří tzv. kořenů důvěry. Kořeny důvěry tvoří naprosté minimum nutné pro zajištění důvěry v systému. Kořeny důvěry pro ukládání a hlášení tvoří TPM a jako kořen důvěry pro měření funguje BIOS systému.

Ve třetí části této práce jsem rozpracoval možnosti propojení těchto dvou systémů v rámci podpory šifrování standardních TrueCrypt svazků. TPM nabízí TrueCryptu nejenom navýšení bezpečnosti u již použitých technologií, jako například další zdroj

náhodných dat, ale i zcela nové možnosti. Jednou z nich je ochrana proti tzv. cold boot útoku, kterým je možné získat šifrovací klíče přímo z operační paměti.

Základem této kapitoly je ale ochrana svazků pomocí větší bezpečnosti šifrovacích klíčů. Navrhl jsem tři možné využití důvěryhodného počítání pro jejich ochranu. Je možno chránit celou hlavičku TrueCrypt svazku, klíč k této hlavičce, nebo pouze hlavní šifrovací klíč k datům. Zároveň jsem uvedl několik možných využití čipových karet k dalšímu posílení bezpečnosti. Jejich použití je bráno v kontextu s důvěryhodným počítáním.

Následující kapitola rozebírá podporu TPM pro šifrování systémového disku, které je v určitých ohledech odlišné od šifrování normálního datového svazku. Vznikají tak problémy, jaké není nutné řešit u datového svazku. Největším je zavaděč TrueCryptu, který je jedinou nešifrovanou a nechráněnou součástí řešení. V souvislosti s ním navrhuji možnosti, jak zajistit jeho bezpečnost a důvěryhodnost. Mimo nových problému zmiňuji i změny ohledně ochrany klíčů a použití čipových karet. Poslední částí jsou dva možné scénáře, kde představuji spojení předchozích návrhů. Jeden je vypracován s ohledem na jednoduchost použití a důvěrnost dat, druhý s ohledem na maximální popiratelnost exitence dat.

V této práci jsem zpracoval možnosti využití technologie důvěryhodného počítání pro posílení důvěry v rámci šifrovacího systému TrueCrypt, s ohledem na proveditelnost a výhody pro bezpečnost celého řešení. Tyto výhody byly řešeny jak z hlediska důvěrnosti dat, tak jejich věrohodného popření.

Literatura

[1] Truecrypt – free open-source disk encryption software for windows vista/xp, mac os x, and linux, 2008. Dokument dostupný na <http://www.truecrypt.org/> (duben 2009).

[2] Alexei CZESKIS et al. Defeating encrypted and deniable file systems: Truecrypt v5.1a and the case of the tattling os and applications. Technical report, Dept. of Computer Science and Engineering, Univ. of Washington, 2008. Dokument dostupný na <http://www.schneier.com/paper-truecrypt-dfs.pdf> (duben 2009).

[3] Vlastimil KLÍMA. Truecrypt : profesionální ochrana dat zdarma, 2007. Dokument dostupný na <http://www.root.cz/clanky/truecrypt-profesionalni-ochrana-dat-zdarma> (duben 2009).

[4] Phillip ROGAWAY. Efficient instantiations of tweakable blockciphers and refinements to modes ocb and pmac. Technical report, University of California, 2004. Dokument dostupný na <http://www.cs.ucdavis.edu/~rogaway/papers/offsets.pdf> (duben 2009).

[5] Moses LISKOV and Kazuhiko MINEMATSU. Comments on xts-aes. Technical report, The College of William and Mary and NEC Corporation, 2008. Dokument dostupný na <http://csrc.nist.gov/groups/ST/toolkit/BCM/documents/comments/XTS/XTS_comments-Liskov_Minematsu.pdf> (duben 2009).

[6] Trusted Computing Group. *TCG Specification Architecture Overview*, 1.4 edition, 2007. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/tcg_architecture_overview_version_14> (duben 2009).

[7] Trusted Computing Group. *TPM Main : Part 1 Design Principles*, 2007. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/tpm_specification_version_12_revision_103_part_1__3> (duben 2009).

[8] Trusted Computing Group. *TCG PC Specific Implementation Specification*, 2003. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/pc_client_work_group_pc_specific_implementation_specification_version_11> (duben 2009).

[9] Trusted Computing Group. *TCG PC Client Specific TPM Interface Specification*, 2005. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/pc_client_work_group_pc_client_specific_tpm_interface_specification_tis_version_12> (duben 2009).

[10] Trusted Computing Group. *TCG PC Client Specific Implementation Specification For Conventional BIOS*. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/pc_client_work_group_specific_implementation_specification_for_conventional_bios_specification_version_12> (duben 2009).

[11] Trusted Computing Group, 2009. Dokument dostupný na <https://www.trustedcomputinggroup.org> (duben 2009).

[12] Eimear GALLERY. An overview of trusted computing technology. In Chris Mitchel, editor, *Trusted Computing*. The Institution of Electrical Engineers, 2005. ISBN 0863415253.

[13] Microprocessor Standards Committee of the IEEE Computer Society. *IEEE Standard Specifications for Public-Key Cryptography (P1363)*, 2000. ISBN 0738119563.

[14] National Institute of Standards and Technology. *FIPS 180-1 - Secure Hash Standard*, 1995. Dokument dostupný na <http://www.itl.nist.gov/fipspubs/fip180-1.htm> (duben 2009).

[15] H. KRAWCZYK, M. BELLARE, and R. CANETTI. Hmac: Keyed-hashing for message authentication, 1997. Dokument dostupný na <http://www.ietf.org/rfc/rfc2104.txt> (duben 2009).

[16] Trusted Computing Group. *TPM Main Part 2 TPM Structures*, 2006. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/tpm_specification_version_12_revision_103_part_1__3> (duben 2009).

[17] Trusted Computing Group. *TPM Main Part 3 Commands*, 2006. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/tpm_specification_version_12_revision_103_part_1__3> (duben 2009).

[18] Trusted Computing Group. *TCG Software Stack (TSS) Specification Version 1.2*, 2007. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/tcg_software_stack_tss_specification__version_12_errata_a> (duben 2009).

[19] J. Alex HALDERMAN et al. Lest we remember: Cold boot attacks on encryption keys. In *USENIX Security Symposium*, 2008. Dokument dostupný na <http://citp.princeton.edu/pub/coldboot.pdf> (duben 2009).

[20] Trusted Computing. *TCG Platform Reset Attack Mitigation Specification*, 2008. Dokument dostupný na <http://www.trustedcomputinggroup.org/resources/pc_client_work_group_platform_reset_attack_mitigation_specification_version_10> (duben 2009).

[21] David Challener et al. *A practical Guide to Trusted Computing*. IBM Press, 2008.

[22] RSA Laboratories. *PKCS #11 v2.20: Password-Based Cryptography Standard*, 2004. Dokument dostupný na <ftp://ftp.rsasecurity.com/pub/pkcs/pkcs-11/v2-20/pkcs-11v2-20.pdf> (duben 2009).

[23] Tobiáš Smolka. *Nástroj pro automatizaci obsluhy dat na kryptografické čipové kartě*. Bakalářská práce, Masarykova Univerzita, Fakulta Informatiky, 2008. Dokument dostupný na <https://is.muni.cz/auth/th/172671/fi_b/text_prace.pdf> (duben 2009).

[24] Jiří Kamínek. *Doplnění podpory pro čipové karty do šifrovacího programu TrueCrypt*. Bakalářská práce, Masarykova Univerzita, Fakulta Informatiky, 2005. Dokument dostupný na <https://is.muni.cz/auth/th/98985/fi_b/bc_10052006.pdf> (duben 2009).

Seznam zkratek

AACP	Asymmetric Authorization Change Protocol
ADCP	AuthData Change Protocol
ADIP	AuthData Insertion Protocol
AES	Advanced Encryption Standard
AIK	Attestation Identity Key
AuthData	Authorization Data
CA	Certification Authority
CMK	Certified Migration Keys
CRTM	Core Root of Trust for Measurement
D-CRTM	Dynamic Core Root of Trust for Measurement
DoS	Denial of Service
EK	Endorsement Key
LPC	Low Pin Count
MA	Migration Authority
MBR	Master Boot Record
MOR	the Memory Overwrite Request
MSA	Migration Selection Authority
NVM	Non-volatile Memory
OIAP	Object-Independent Authorization Protocol
OSAP	Object Specific Authorization Protocol
PC	Personal Computer
PCR	Platform Configuration Register
RNG	Random Number Generator

RTM	Root of Trust for Measurement
RTR	Root of Trust for Reporting
RTS	Root of Trust for Storage
S-CRTM	Static Core Root of Trust for Measurement
SML	Stored Measurement Log
SRK	Storage Root Key
TBB	Trusted Building Block
TCG	Trusted Computing Group
TCS	TCG Core Services
TCSCM	TCS Context Manager
TCSEM	TCS Event Manager
Tcsi	TCG Core Services Interface
TcsipBG	TCS Parameter Block Generator
TCSKCM	TCS Key & Credential Manager
TDD	TPM Device Driver
TDDI	TPM Device Driver Interface
TDDL	TCG Device Driver Library
Tddli	TCG Device Driver Library Interface
TP	Trusted Platform
TPM	Trusted Platform Module
TRNG	True Random Number Generator
TSP	TCG Service Provider
TSPCF	TSP Cryptographic Functions
TSPCM	TSP Context Manager
TSPI	TCG Service Provider Interface
TSS	TCG Software Stack
VM	Volatile Memory