Trusted Platform Module (TPM)

Seminarska naloga iz Kriptografije in teorije kodiranja 2

Ruben Sipoš

Povzetek

Da bi izboljšali varnost računalniških platform, je skupina TCG (Trusted Computing Group) sestavila specifikacijo TP (Trusted Platform). Temelj varnosti, na katerem gradi veriga zaupanja, je TPM (Trusted Platform Module) čip. Z njegovo pomočjo lahko implementiramo preverjanje integritete programskega in strojnega okolja in primernost dokažemo tudi oddaljeni platformi. V tej seminarski nalogi so opisane ponujene zmogljivosti, primeri njihove uporabe in morebitne pomanjkljivosti

Kazalo

1 Uvod

1.1 Motivacija

V zadnjih nekaj letih so računalniki postali eden izmed ključnih elementov našega vsakdanjega življenja. Že nekaj časa nazaj so državni, poslovni in bančni sistemi prenesli večji del obdelave podatkov in njihovo hranjenje v digitalni obliki na računalniške sisteme. Zaradi potrebe po varnosti ti sistemi uporabljajo varne operacijske sisteme in so tudi fizično zaščiteni pred morebitnimi napadi nanje. S pojavom internetnega bančništva, spletnih nakupov in mnogih drugih storitev se je pojavila potreba po večji varnosti tudi na odjemalcih.

Eden izmed razlogov za zelo hitro poslovno rast je odprtost in fleksibilnost PC platforme. Večina uporabnikov postavlja uporabnost pred varnost in se zato upira vsakršnim poskusom povečanja varnosti, ki bi ob tem zmanjšali svobodo uporabnika. Prav tako ne obstaja enoten organ, ki bi diktiral razvoj PC-jev, zato smo priča ogromni raznolikosti. Vedno večji del poslovanja se seli na internet, kar še toliko bolj izpostavlja potrebo po povečanju varnosti na odjemalcih.

Sama programska oprema ne more jamčiti lastne integritete in zadostiti vsem varnostnim potrebam. Da bi dosegli željen nivo varnosti, so potrebne spremembe v strojni opremi. TCG je skupina ustanovljena z namenom, da sestavi specifikacijo, ki bo rešila večino trenutnih problemov varnosti na odjemalcih.

1.2 Snovalci in ideja

Trusted Platform [12] je računalniška platforma, ki vsebuje komponento (predvidoma del strojne opreme), ki ji zaupamo in je temelj zaupanja na katerem gradi programska oprema. En primer tega je Trusted Computing, katerega specifikacijo so spisali v Trusted Computing Group [1], ki je naslednica Trusted Computing Platform Alliance in je bila ustanovljena s strani AMD, HP, IBM, Infineon, Intel, Microsoft in Sun Microsystems. Trenutno ima že več kot 170 članov.

Glavna ideja pristopa, ki ga predlaga TCG, je, da strojna oprema vsebuje funkcije, ki jim moramo zaupati in so temelj zaupanja na katerem gradi programska oprema. Strojna oprema lahko preveri programsko okolje in zagotovi neoporečnost. Vendar pa moramo vedeti, da s pretvorbo platforme v TP ne spremenimo ostalih lastnosti platforme in zato TP najverjetneje ni tako varna kot varni računalnik, ki je bil zasnovan z mislijo na varnost na prvem mestu.

TP nam ponuja izdatno povečanje varnosti za minimalno ceno, kar je ena izmed pomembnejših lastnosti, če hočemo uporabnike spodbuditi k uporabi. TP tudi ni omejena zgolj na PC platformo (kjer se je že precej razširila, predvsem med prenosnimi računalniki), ampak lahko pričakujemo razvoj tudi na področju PDA, mobilnih telefonov itd. Prva verzija specifikacije je bila objavljena že februarja 2001; zadnja verzija je 1.2.

1.3 Sestava TP

Skupina TCG je definirala TP kot skupek zaupanja vredne strojne opreme, programske opreme in zunanjih CA-jev (Certification Authority) potrebnih za izvedbo kriptografskih dokazov. Sama platforma je sestavljena iz treh logičnih blokov.

Core Root of Trust for Measurement (CRTM)

CRTM je ponavadi del BIOS-a. Vsebuje funkcije, ki se izvedejo še preden se naloži operacijski sistem in preverijo integriteto ter tako poskrbijo za varno zagonsko okolje. To se doseže na ta način, da se poračunajo izvlečki vrednosti kritičnih delov zagonskega nalagalnika in operacijskega sistema, ki jih nato preveri TPM.

Trusted Platform Module (TPM)

TPM je dodaten čip na matični plošči [2], ki skrbi za izvedbo vseh zaupanja vrednih operacij in kriptografskih funkcij. Zasnovan je kot pasiven del in ne more neposredno vplivati na izvajanje programov v procesorju in postopek zagona. Znotraj TPM se vrši obdelava podatkov o stanju sistema, ki se lahko shranijo v interno varno shrambo ali pa se iz nje preberejo. Edini direkten vpliv TPM se kaže pri dostopu do ključev shranjenih znotraj interne varne shrambe, ki so dostopni samo v specifičnih okoliščinah (primerna avtentikacija in stanje sistema).

Strojna programska oprema skrbi za komunikacijo z ostalim delom sistema in implementira standardiziran protokol za komunikacijo s programsko opremo (TSS) ter omogoča dostop do strojno podprtih kriptografskih funkcij. Prav tako skrbi za preprečevanje morebitnih fizičnih napadov s spremljanjem senzorjev in po potrebi primerno reagira.

Trusted Platform Support Service (TSS)

Pomemben del platforme je tudi API (Application Programming Interface), ki skrbi za komunikacijo med TPM in operacijskim sistemom. Najbolj nizkonivojski del TSS[1] je gonilnik (na nivoju jedra), ki omogoča komunikacijo in izmenjavo podatkov preko LPC[2] (Low Pin Count) vodila. Naslednje stopnje v programskem skladu so TPM Device Driver Library, TSS Core Services in TSS Service Provider. Njihova naloga je usklajevanje sočasnih dostopov do TPM, pretvorba abstraktnih API klicev v nizkonivojske TPM ukaze, možnost oddaljenega dostopa in varno shranjevanje podatkov na disk v primeru pomanjkanja glavnega pomnilnika.

Da bi bile kriptografske zmožnosti TPM dostopne tudi uporabniku (in ne le operacijskemu sistemu preko TSS), sta na voljo dva standardna API-ja namenjena predvsem za potrebe varne hrambe in možnosti digitalnih podpisov.

Microsoft Cryptographic Service Provider (MS-CSP)

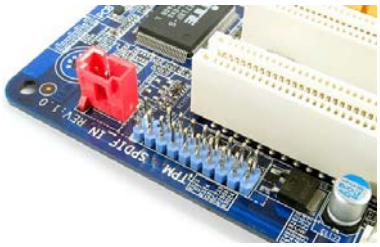
Eden izmed bolj znanih in razširjenih API-jev je MS-CAPI. Precej aplikacij za Windows uporablja funkcije, ki jih ponuja MS-CAPI za izvedbo raznih kriptografskih funkcij kot npr. šifriranje in podpisovanje sporočil v odjemalcu za spletno pošto. Prehod na uporabo TPM in s tem povečanje varnosti je enostaven, potrebno je le izbrati drug CSP.

[1] Odprtokodna implementacija TrouSerS za Linux je dostopna na naslovu http://trousers.sourceforge.net/
[2] Vodilo potrebuje 7 povezav (http://en.wikipedia.org/wiki/Low_Pin_Count).

PKCS#11

Trenutno najbolj uporabljan standard za dostop do kriptografskih funkcij in naprav je PKCS#11 razvit s strani RSA. Prehod na uporabo TPM je enostaven zaradi ponujene pretvorbe med PKCS#11 in TSS API klici. Tako je na primer mogoče preiti na uporabo TPM za varno hranjenje certifikatov uporabljenih v spletnih brskalnikih z minimalno investicijo.

1.4 Sestava TPM



Specifikacija je že nekaj časa objavljena in sedaj se na tržišču že pojavljajo prvi TPM čipi (npr. ST19WP18 [13], Infineon SLB9635 [14]) in tudi matične plošče[3] s podporo za TPM (npr. Gigabyte GA-EP45-DS5, Asus M3A78-VM, Tyan Tempest i5400XT). Prav tako je na tržišču vse več prenosnikov s podporo za TPM (npr. Asus M51Vr AP063E, IBM ThinkPad T60).

Zaupanja vredne kriptografske funkcije so implementirane v TPM čipu, zato strojna in programska oprema zunaj TPM nanje ne more vplivati. Edini možen način komunikacije je preko I/O na LPC vodilu po protokolu specificiranem s strani TCG. Najpogostejše sestavne komponente trenutno prodajanih TPM čipov so:

- 2048-bitni RSA pospeševalnik (podpisovanje, preverjanje, generiranje ključev)

Funkcija	Potreben čas
1024-bitni RSA podpis s KIO[4]	62 ms
1024-bitni RSA podpis brez KIO	206 ms
1024-bitna RSA preverba (e = '$10001')	4 ms
Generiranje 1024-bitnega RSA ključa	1.8 s
2048-bitni RSA podpis s KIO	416 ms
2048-bitna RSA preverba (e = '$10001')	66 ms

- izračun SHA-1 in MD5 izvlečkov
- strojni generator naključnih števil (FIPS 140-2[5] potrjen)
- 8-bitni procesor, 8kB RAM, 16kB EEPROM za hrambo ključev (do 10 let)
- Low Pin Count (LPC) vodilo, ki omogoča hitrosti do 4mB/s
- 1088-bitna MAP (Modular Arthmetic Processor) enota za modularno aritmetiko
- podpora za Intel Trusted Execution Technology (LaGrande) tehnologijo
- varnost pred posegi: zaščita pred pre/podnapetostjo, aktiven ščit, filter visokih frekvenc, senzor nizkih frekvenc, šifriran pomnilnik (MED), zaščita pred SPA in DPA[6] napadi

[3] Trenutno lahko najdemo na tržišču matične plošče različnih proizvajalcev. Prav tako ponudba obsega vse od namiznih do strežniških matičnih plošč.
[4] Ob uporabi kitajskega izreka o ostankih (KIO) se določeni izračuni poenostavijo.
[5] http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf
[6] Določeni napadi temeljijo na merjenju porabe električne energije (simple power analysis, differential power analysis).

- monotono naraščajoč in zaščiten števec (delujoč kot časovnik za preprečevanje replay napadov)
- self-test funkcija
- Platform Configuration Registers (PCR), v katere se shranijo izvlečki ob preverjanju integritete strojne in programske opreme
- certifikati: endorsement, platform, conformance
- endorsement ključ

2 Uporaba TPM

2.1 Začetna točka verige zaupanja

Ker je specifikacija TPM javno dostopna[7] (kar je logično, saj želimo, da bi bila vredna zaupanja), bi se lahko zgodilo npr. da nekdo posnema prisotnost TPM na njihovem CPU. V tem primeru bi lastnik temu računalniku zaupal (saj je seznanjen s tem, kaj se dogaja na njem), vendar pa bi lahko enostavno spreminjal interno stanje simuliranega TPM in tako morebiti dobil prednost ob komunikaciji z drugimi TP. To nikakor ni sprejemljivo, saj je ena izmed pomembnejših nalog TP zagotoviti tudi varno komuniciranje preko spleta. Zaradi tega je bilo uvedenih nekaj dodatnih varnostnih ukrepov.

2.2 Endorsement Key (EK)

Na končni stopnji izdelave TPM čipa izdelovalec zgenerira 2048-bitni par asimetričnih ključev imenovan EK. Ta je nato shranjen v TPM na takšen način, da je nemogoče prebrati privatni del ključa ponovno iz TPM, ampak se ga lahko uporablja zgolj znotraj TPM čipa. Kot dodatno zagotovilo je ključ zaščiten s certifikatom (ki ga ne sme izdati oseba odgovorna za generiranje ključev). Ker velika večina varnostnih funkcij temelji na skrivnosti privatnega dela ključa, moramo zaupati proizvajalcu, da je ključ unikaten in do njega nima dostopa nihče drug. Podprta je tudi generacija EK znotraj TPM z uporabo API funkcije *TPM_CreateEndorsementKeyPair*.

- *Endorsement Certificate*
  Ta certifikat zagotavlja, da je TPM originalen in prihaja od zaupanja vrednega proizvajalca ter da je privatni del ključa res skriven. Vsebuje javni del EK in se ga uporablja za generiranje AIK.

- *Platform Certificate*
  Izda ga izdelovalec matične plošče oz. računalnika. Potrjuje, da je bil vgrajen pravilno delujoč TPM na primerno platformo. Tudi PC se uporablja v procesu generirarnja AIK.

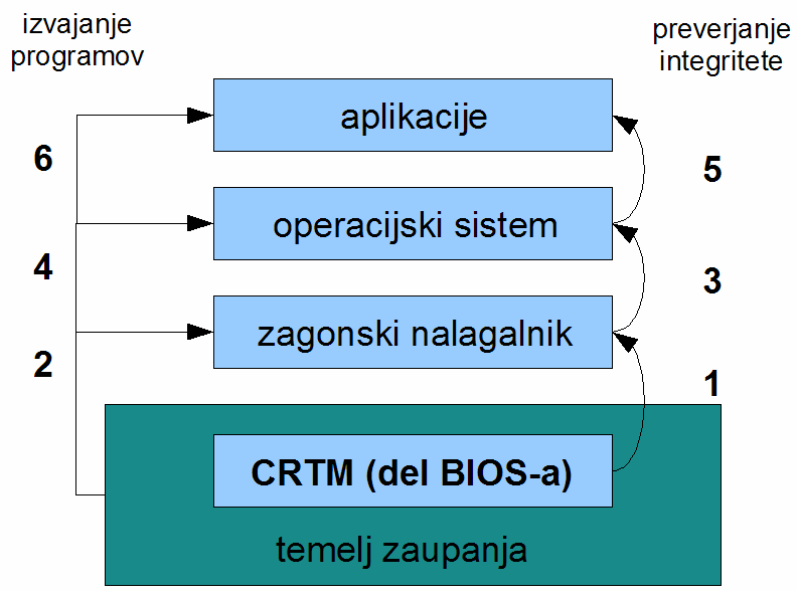
- *Conformance Certificate*
  Ko testni laboratorij opravi predvidene poizkuse in potrdi, da kriptografske funkcije TPM in delovanje matične plošče ustreza specifikacijam TCG, izda certifikat, ki to potrjuje.

[7] https://www.trustedcomputinggroup.org/specs/TPM/

2.3 Preverjanje integritete

Do sedaj smo za varnost na domačih računalnikih skrbeli predvsem z uporabo protivirusnih programov in šifriranja podatkov. TCG predlaga in ponuja povsem drugačen pristop ob uporabi TP. Z varnostjo začnemo že pri najnižjem nivoju (BIOS), kjer za naš temelj postavimo TPM čip, ki mu brezpogojno zaupamo (na podlagi izdanega certifikata o preverjenosti stojne implementacije). Ob izvajanju določenih operacij moramo zaupati tudi nekaterim ostalim komponentam (glavni pomnilnik, sistemski krmilnik, določeni deli procesorja, vodila), saj se temu enostavno ne moremo izogniti ob uporabi odprte platforme kot je npr. PC. Od tukaj naprej gradimo „verigo zaupanja“ vse do končnih aplikacij, ki jih uporabnik uporablja. Vsaka naslednja stopnja gradi na prejšnji, ob tem pa lahko zaupa, da so vse funkcionalnosti nižje stopnje zanesljive, saj so bile že predhodno preverjene.

Na začetku TPM preveri stanje strojnih komponent in zagotovi, da ni prišlo do nedovoljenih sprememb. To dosežemo tako, da poračunamo izvleček stanja strojne opreme (npr. ID-ji naprav in njihovo stanje) in ga shranimo (ali dodamo k obstoječemu) v PCR ter primerjamo s pričakovano vrednostjo. V prihodnosti pa mogoče lahko pričakujemo, da bodo komponente predstavile svoje certifikate o ustreznosti. Ko smo na ta način zagotovili, da je strojna oprema ustrezna (npr. ne vsebuje „zlobnih“ naprav, ki bi poskušale sistemu škoditi), preverimo pravilnost BIOS-a. CRTM nato preveri pravilnost zagonskega sektorja in zagonskega nalagalnika (IPL[8]). Na ta način nadaljujemo gradnjo verige: IPL preveri ključne dele operacijskega sistema (npr. jedro), ti nato preverijo preostale dele, ob zagonu programov le-te preveri operacijski sistem... Tako zagotovimo neoporečnost okolja, ki mu sedaj lahko zaupamo.



[8] Initial Program Loader

2.4 Priklepanje informacij na platformo

Tesno povezano s preverjanjem integritete je tudi priklepanje informacij na platformo. Velikokrat želimo, da so določene informacije (oz. ključi za dostop do njih) na voljo zgolj kadar je okolje neoporečno in je uporabnik uspešno prijavljen. TPM nam omogoča, da na takšen način shranimo ključe znotraj varne hrambe v čipu. Do njih lahko dostopamo zgolj v primeru, da je bilo preverjanje integritete uspešno. Tako zagotovimo, da je dostop mogoč zgolj iz preverjenega programskega okolja, ki pa poskrbi za pravilno upoštevanje pravic za dostop in avtentikacijo uporabnikov.

2.5 Varna hramba

TPM je kriptografska naprava odporna proti posegom (tamper-resistant) in zato primerna za varno hrambo podatkov. Vendar zaradi cenovnih in velikostnih omejitev TPM ponuja zelo omejen prostor za hranjenje podatkov. Da bi kljub temu dosegli možnost zavarovanja potencialno neomejene količine podatkov, je bila uvedena hierarhija zaščitenih objektov. Na vrhu je Storage Root Key (SRK), ki se avtomatsko ustvari ob operaciji prevzema lastništva („Take Ownership“). V primeru, da lastnik TPM-ja prekliče svoje lastništvo, se SRK pobriše in s tem prepreči dostop do vseh varno shranjenih podatkov (kar je dobro iz varnostnega vidika). SRK je asimetrični ključ, ki ščiti ključe, ki ščitijo naslednje ključe ali podatke. CPU sedaj lahko od TPM dobi npr. simetrični ključ (ki je bil varno shranjen znotraj TPM) in ga uporabi za šifriranje večje količine podatkov. Vendar je ključ možno dobiti iz TPM samo, če je stanje sistema takšno kakršno je bilo zahtevano ob shraniti ključa.

Shranimo lahko poljubno količina podatkov, ki so zavarovani prav tako dobro kot SRK in so dostopni samo v predvidenih okoliščinah. Primer uporabe, ki se vse pogosteje pojavlja v vsakdanji rabi, je varovanje podatkov na prenosnem računalniku v primeru kraje. Vsi podatki na disku so zaščiteni in jih zato ne moremo prebrati s fizičnim prenosom diska v drug sistem, prav tako pa ne moremo spremeniti operacijskega sistema in tako zaobiti avtentikacijo, saj v tem primeru sistem več ni v stanju, ki ga TPM zahteva za dostop do ključev.

2.6 Attestation (avtentikacija platforme)

Ob poslovanju preko interneta želimo zagotoviti, da je tudi platforma s katero komuniciramo TP in ji zato lahko zaupamo. Preverbo in potrditev, da oddaljena platforma ustreza vsem zahtevam TP, imenujemo (*remote*) *attestation*. Kadar želimo zagotoviti, da je oddaljen sistem TP in na njem teče varen operacijski sistem, moramo preveriti integriteto platforme in veljavnost TPM. Integriteto preverimo tako, da od oddaljenega TPM zahtevamo PCR vrednosti, ki vsebujejo izvleček strojne in programske konfiguracije. Ob tem moramo zagotoviti tudi, da je oddaljen TPM čip veljaven (t.j. da ne komuniciramo s simuliranim TPM ali z neveljavnim ponaredkom). To lahko preverimo npr. s preverbo veljavnosti EK oddaljenega TPM.

3 Primeri uporabe

3.1 BitLocker

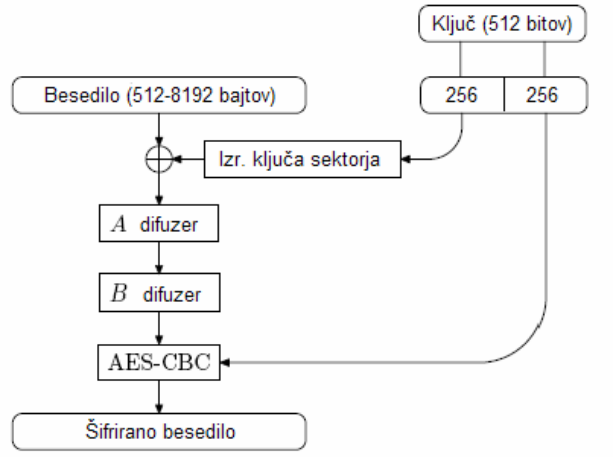
Trenutno najbolj uporabljana rešitev, ki se opira na TPM, je BitLocker [11]. Prvič se je pojavil kot del Windows Vista SP1 in Windows Server 2008. Uporabniku ponuja šifriranje celotnega diska, kar je v zadnjih letih postala ena izmed prioritetnih potreb velikih podjetij. Ocenjuje se, da se letno izgubi oz. je ukradenih 1-2% prenosnih računalnikov v velikih organizacijah. Če si zamislimo veliko podjetje z 100.000 prenosniki, to pomeni nekaj izgubljenih prenosnikov vsak dan. Ob tem se moramo zavedati, da diski v teh računalnikih vsebujejo zaupne podatke, ki lahko negativno vplivajo na poslovanje podjetja, če pridejo v napačne roke, ali celo ogrozijo državno varnost in zasebnost državljanov v primeru kraje prenosnika v lastni državnega uslužbenca.

Da bi se izognili morebitnim težavam ob kraji prenosnika, nam BitLocker ponuja šifriranje celotnega diska, a je hkrati prijaznejši do uporabnika kot ostale rešitve (npr. nizkonivojski gonilnik za šifriranje vseh podatkov na disku), saj ne zahteva dodatne interakcije uporabnika (ni potrebna pametna kartica ali geslo takoj ob zagonu). Varnost tega pristopa temelji na funkcionalnostih, ki jih ponuja TPM. Ključ s katerim so šifrirani vsi podatki na disku je shranjen v TPM in prikljenjen na platformo. Do njega je mogoče dostopati le, kadar izvlečki v PCR-jih ustrezajo želenim. Na ta način se preveri integriteta BIOS-a, MBR-ja, zagonskega nalagalnika in operacijskega sistema. To pomeni, da lahko do ključa (in posledično podatkov na disku) dostopa zgolj preverjen operacijski sistem (za katerega vemo, da bo pravilno upošteval pogoje dostopa do podatkov – npr. s preverbo identitete uporabnika). Kdorkoli lahko na računalniku zažene drug ali modificiran operacijski sistem, saj mu tega nič ne preprečuje, vendar ne bo mogel do ključa. Prav tako do podatkov ne moremo, če fizično prestavimo disk v drug računalnik, saj je ključ shranjen v TPM na prvotni platformi.

Izbira algoritma za šifriranje se je izkazala za dokaj trd oreh. Zagotavljanje zasebnosti podatkov na disku je dovolj enostavno rešiti z izbiro dovolj močnega kriptografskega algoritma, saj BitLocker poskrbi, da so vsi podatki na disku šifrirani z njim. Vendar pa to ne zadošča za nivo varnosti, ki ga želimo. Zagotoviti hočemo, da napadalec ne more z morebitnimi spremembami zapisa na disku zmanjšati varnosti operacijskega sistema in tako priti do podatkov. Splošno sprejeta rešitev zagotavljanja integritete podatkov je uporaba MAC (Message Authentication Code). Ob tem pa se srečamo s težavo: velikost sektorjev je fiksna in zelo tesno povezana z arhitekturo strojne in programske opreme, zato ne moremo dodati MAC v isti sektor ob podatke. Razširitev sektorja na dvojno velikost in uporaba druge polovice za MAC je nesprejemljiva za uporabnike zaradi prevelike cene. Uporaba vsakega *n*-tega sektorja za hrambo MAC vrednosti krši eno izmed osnovnih predpostavk: sprememba *i*-tega sektorja ne vpliva na noben drug sektor. Ta lastnost je pomembna za hitrostne optimizacije kot tudi za integriteto podatkov. Datotečni sistem in mnoge podatkovne baze se zanašajo na to, da ob izpadu elektrike pride do napačnega zapisa zgolj na spremenjenem sektorju (kar pa več ne bi bilo res, saj bi morali popraviti se MAC na ločenem sektorju in bi ob morebitni nekonsistentnosti izgubili več podatkov kot zgolj en sektor). Prav tako te rešitve preprečijo enostavno pretvorbo ob vklopu/izklopu BitLocker zaščite.

Rešitev za katero se je Microsoft odločil ob zasnovi BitLocker je reveževa avtentikacija. Integritete shranjenih podatkov ne preverjamo direktno, ampak se zanašamo, da napadalec z načrtnimi spremembami šifriranih blokov ne bo mogel povzročiti nenaključne spremembe podatkov (v najslabšem primeru se bo sistem zgolj zrušil, ne bo pa se oslabila njegova varnost). V ožjem izboru so bili: tokovne šifre, AES-CBC, Bear, Lion, Beast, VIL, LRW, CMC, EME. Nekatere nimajo

zadovoljive difuzije[9] (tokovne kode, AES-CBC, Beast, VIL, LRW), ostale so prepočasne (Bear, Lion), razbite (Mercy) ali ne dovolj preizkušene (CMC, EME). Na koncu so se odločili za AES-CBC z dodatnim difuzerjem. Na ta način je varnost vsaj tako dobra kot ob uporabi splošno sprejetega AES-CBC, saj tudi ob morebitnem popolnem razbitju zaenkrat še ne dovolj preverjenega difuzerja varnost ostane na nivoju AES-CBC.



Šifriranje uporabljeno v BitLocker-ju

Ključ, ki se uporablja za šifriranje je velik 512 bitov. Velikost vsakega sektor/blok je potenca števila dve in je med 512 in 8192 bajti. Polovica ključa se uporablja za AES-CBC, druga polovica pa za difuzer, kar zagotavlja neodvisnost v primeru razbitja poljubne polovice.

Prvi korak je XOR sektor ključa s podatki. Glede na velikost sektorja je ključ ponovljen potrebno število krat. Definiran je kot:

$$K_s := E(K_{sec}, e(s)) || E(K_{sec}, e'(s))$$

K_{sec} je polovica ključa namenjena za uporabo v difuzerju
e'() je enaka kot e() le da je zadnji bajt rezultata enak 128 namesto 0

Pred uporabo AES-CBC, gredo podatki skozi dva namenska difuzerja. Po trditvah snovalcev sprememba enega bita povzroči polno difuzijo aktivne 32 bitne besede po približno 1/3 cikla.

Kot zadnji korak se izvede kodiranje z AES-CBC. IV za sektor s se izračuna kot:

$$IV_s := E(K_{AES}, e(s))$$

E() je AES šifrirna funkcija
e() je funkcija, ki pretvori številko sektorja v 16 bajtno vrednost (izbira te funkcije ne vpliva na varnost, če je le injekcija)

Implementacija tega postopka v BitLocker-ju potrebuje ~20ciklov/bajt za AES-CBC in ~10ciklov/bajt za Elephant difuzer na Pentium 4 procesorju, kar je hitreje od maksimalne hitrosti pisanja/branja po disku.

[9] Ena izmed pomembnih lastnosti je, da že zelo majhna sprememba (npr. 1 bit) povzroči velike in težko predvidljive spremembe v nadaljnjih podatkih.

3.2 Možnosti uporabe za namene varnega soprocesorja

Zaenkrat na trgu še ni izdelka, ki bi v celoti izkoristil prednosti TPM in uporabniku ponudil operacijski sistem z vsemi potrebnimi orodji za izvajanje TP filozofije. Microsoft kot največji ponudnik operacijskih sistemov zaenkrat nudi le BitLocker; prav tako kaže, da v prihajajočih Windows 7 še ne bo popolne podpore za vse funkcionalnosti TP.

Eden izmed poskusov neodvisne implementacije varnega operacijskega sistema [3] temelji na razširitvi Linuxa in poskusa doseči TP zgolj z uporabo prosto dostopne dokumentacije. Motivacijski primer bi bila postavitev varnega spletnega strežnika, ki uporabniku zagotavlja, da je v takšnem stanju kot ga je predvidel njegov lastnik (in na njem ni prišlo do kakršnihkoli neželenih sprememb – npr. vdora in spremembe datotek).

Vsebina diska je razdeljena v tri skupine:

- dolgoživi del vsebuje komponente, ki se redko spreminjajo (zagonski nalagalnik in jedro)
- programska oprema (npr. standardna orodja in strežniški programi)
- kratkoživi podatki

Ob vklopu računalnika najprej BIOS preveri MBR, ki nato preveri jedro. Del jedra je Enforcer, ki skrbi za preverjanje programske opreme. Razslojitev vsebine diska v tri dele izdatno olajša posodabljanje programske opreme in spreminjanje podatkov. TPM hrani zgolj izvleček za zagonski nalagalnik in jedro. Modul Enforcer v jedru ima lastno shrambo izvlečkov, ki opisujejo dovoljeno programsko opremo. Na ta način je nadgradnja le-te poenostavljena, saj je potrebno spremeniti zgolj seznam dovoljenih izvlečkov in ne vrhovnega izvlečka v TPM-ju.

Trenutna implementacija je dostopna[10] kot GPL koda, vendar še ima nekaj pomanjkljivosti:

- Uporabnik »root« ima zelo obširne pravice. Brez težav lahko opazuje in spreminja kodo pognanih aplikacij. V primeru, da napadalec dobi dostop do »root« uporabnika, lahko naredi skoraj karkoli.
- Trenutna implementacija zgolj preverja integriteto datotek, preden se le te naložijo. To ni dovolj za zaščito pred morebitnimi neželenimi kasnejšimi spremembami (npr. z direktnim spreminjanjem vrednosti v pomnilniku).
- Odprta ostajajo vprašanja kako zagotavljati svežino delov, ki se pogosto spreminjajo. Kdaj se naj posodobijo izvlečki podatkov, ki jih lahko spreminjajo uporabniki?

3.3 DRM (digital rights management) & možnosti zlorabe

Večina ljudi, ki je že slišala za TPM, ga pozna ali zaradi BitLocker-ja ali pa ga označi[11] kot stvar zasnovano zgolj za namene DRM [5,6]. S tem, ko je človeštvo stopilo v digitalno dobo, je postalo kopiranje podatkov enostavno. Odvija se nenehen boj med avtorji digitalnih izdelkov (in njihovimi zastopniki), ki želijo preprečiti nelegalno kopiranje in ohraniti dobiček, in ljudmi, ki nenehno odkrivajo načine, kako se temu izogniti. Naj naštejem samo nekaj poskusov (neuspešnih in zaenkrat uspešnih, čeprav nihče ne ve kako dolgo še): Macrovision, StarForce, DVD CSS, WMV DRM, HDCP, TPM, ...

TPM predstavlja idealno orodje za vse od uvedbe popolnega DRM do popolnoma diktatorskega

[10] http://enforcer.sourceforge.net/

[11] Po lastnih izkušnjah vem, da ob omembi TPM marsikatera ugledna oseba zgolj zamahne z roko in reče, da je edini namen tega DRM.

sistema brez kakršnekoli svobode. Preverjanje integritete zagotavlja, da so operacijski sistem in programi neoporečni. Na ta način lahko zagotovimo, da je nemogoče obiti preverjanje veljavnosti licence. Prav tako lahko preprečimo vsakršno vohunjenje po pomnilniškem prostoru aplikacij, saj je nemogoče nedovoljeno spreminjati jedro operacijskega sistema. Tako preprečimo morebitno razkritje ključa za dešifriranje DRM vsebin iz naslovnega prostora predvajalnika.

Če gremo še en korak dlje, je mogoče z uporabo TPM zagotoviti popoln nadzor nad strojno in programsko opremo uporabnika. V primeru da sistem zazna opremo, ki nima veljavnega podpisa oz. ni v bazi dovoljene opreme, ne dovoli njene uporabe. Kdor uporablja npr. odprtokodno programsko opremo s tem ni omejen, dokler ne dodamo *remote attestation*. Sedaj mora tisti, ki želi komunicirati z oddaljenim sistemom (spletni strežnik, banka, prijateljev računalnik itd.) dokazati, da uporablja TP in s tem odobreno opremo. Še več, z uporabo varne hrambe lahko zagotovimo, da dokumente ustvarjene na TP lahko odpremo samo v neoporečnem okolju (in programom, ki jih je ustvaril), saj so na disk shranjeni šifrirano, do ključa pa lahko dostopa zgolj operacijski sistem, katerega izvlečki v PCR-jih so ustrezni.

Predlogi sprememb specifikacije za zmanjšanje možnosti zlorabe

Da bi zmanjšali možnosti zlorabe TP tehnologije na račun uporabnika [4] in s tem povečali zaupanje uporabnikov ter tako pospešili sprejemanje TP v varnostne namene, je bilo predlaganih nekaj popravkov specifikacije (1.1b), ki pa ne vplivajo na ciljno uporabo.

- CRTM naj omogoča (obvezno) izbiro, katerim konfiguracijam zaupamo. S tem zagotovimo, da uporabnik točno ve, kaj lahko pričakuje. V nasprotnem primeru lokalni uporabnik brez uporabe *remote attestation* ne more preveriti ali je naložena platforma res TP ali pa zgolj njena simulacija.
- Lastnik platforme naj ima možnost menjave vrednosti v PCR-jih. Da lahko še vedno zagotovimo varnost (npr. pri poslovanju z bankami), se naj ob *remote attestation* doda tudi resnična vrednost, vendar zakodirana z EK ključem.
- Dodati je potrebno možnost generiranja novega EK in njegovega certificiranja. S tem preprečimo morebitne zlorabe (npr. s strani proizvajalcev TPM) in preprečimo črnoglede scenarije DRM (ponudniki sedaj več ne morejo vsiliti specifikacije okolja uporabniku).
- SRK mora biti dostopen lastniku platforme, če to želi. S tem rešimo tri pomembne probleme: preprečimo obstoj programske opreme, ki deluje proti uporabnikovi volji (npr. cenzura dokumentov); preprečimo monopole (npr. OpenOffice sicer mogoče ne bi mogel brati Wordovih dokumentov); podatki zaščiteni z neseljivim ključem se lahko rešijo v primeru stojne okvare. V določenih primerih (DRM, avtentikacija) želimo, da SRK ni dostopen nikomur razen platformi. To lahko rešimo tako, da PCR dodatno vsebuje podatek o tem, ali je SRK bil razkrit ali pa ga ne pozna nihče (niti lastnik).
- Programska oprema ne sme biti sposobna ločiti med platformo brez TPM in platformo na kateri je TPM namenoma izključen.

4 Napadi

4.1 Napad na komunikacijski kanal

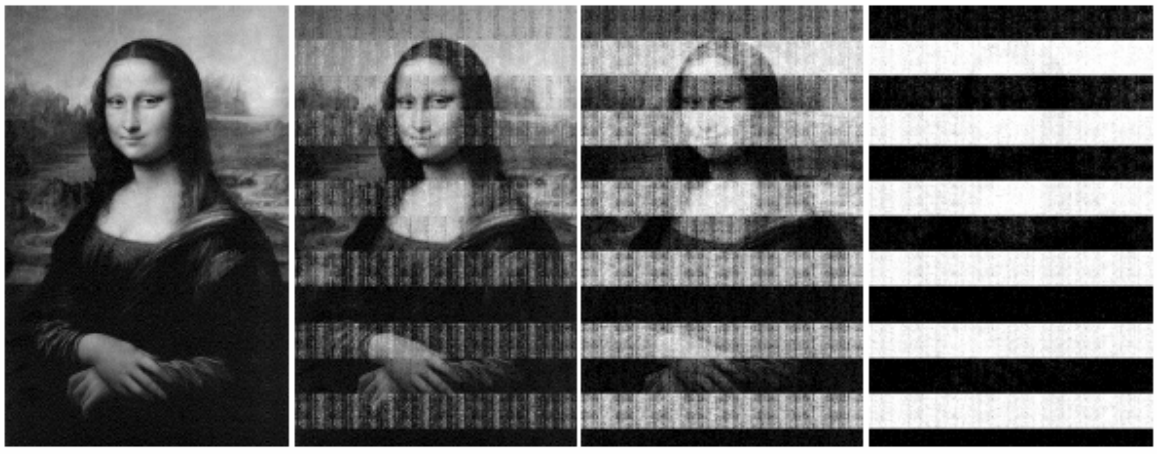
TPM čip je povezan z ostalim delom platforme preko LPC vodila. V večini primerov je TPM dodan na matično ploščo kot razširitvena kartica, kar omogoča enostaven dostop do vodila in možnost spremljanja ter spreminjanja izmenjave. Iz teh razlogov so v nekaterih primerih TPM čipi del matične plošče (in zato težje dostopni) ali celo del veznega nabora (in zato povsem nedostopni).

S pasivnim prisluškovanjem [7] je dokaj enostavno priti do podatkov, ki se prenašajo po vodilu, saj sta specifikacija LPC vodila in protokol za komunikacijo med TSS in strojno programsko opremo v TPM-ju javno dostopna. Na ta način bi npr. lahko prišli do ključa za šifriranje podatkov na disku. Aktiven napad nam ponuja še več možnosti. Najbolj zanimiva je manipulacija vrednosti PCR ob nalaganju platforme, saj bi s tem napadalec lahko zaobšel preverjanje integritete. Prvi poskusi kažejo, da TPM ne zazna aktivne manipulacije LPC vodila in je zato aktiven napad toliko lažji.

4.2 Cold boot napad

4.2.1 Lastnosti DRAM-a

V nasprotju s splošnim prepričanjem moderni DRAM čipi ohranijo večino vsebine kar nekaj časa ob izklopu električnega napajanja [8]. Razni eksperimenti, ki so jih izvedli, to potrjujejo. Kot ekstremen primer je naveden poskus pri katerem so DRAM modul potopili v tekoči dušik za 60 min in po ponovni vstavitvi v računalnik izmerili le 0.17% spremenjenih bitov.



Bitna slika je bila naložena v pomnilnik na sistemu A. Zaporedje prikazuje degradacijo zaradi izklopa napajanja po 5s, 30s, 60s in 5min.

Testni sistem	Št. sekund brez elektrike	Količina napak (%) na delavni temp.	Količina napak (%) pri -50ºC
A	60 300	41 50	brez 0,000095
B	360 600	50 50	brez 0,000036
C	120 360	41 42	0,00105 0,00144
D	40 80	50 50	0,025 0,18

4.2.2 Shranjevanje vsebine DRAM modulov

Da bi shranili vsebino DRAM modula ne potrebujemo posebne opreme. Takoj, ko sistem s ciljnim modulov vklopimo, se ponovno prične osveževanje celic in s tem preneha izguba podatkov. Nato lahko na različne načine shranimo oz. prenesemo vsebino. Da bi uničili čim manj podatkov ne smemo naložiti operacijskega sistema ampak zgolj manjši program (s pomočjo PXE mrežnega zagona, USB ključka ali česa podobnega), ki ve storiti želeno operacijo in uniči minimalen kos podatkov v pomnilniku za lastne potrebe.

4.2.3 Rekonstrukcija ključev

Čeprav nam ta pristop omogoči, da dobimo kopijo vsebine pomnilnika, s tem še vedno nismo pridobili celotnega ključa (za katerega smo vedeli da se nahaja v pomnilniku pred napadom), saj se je zaradi izklopa elektrike določeno število bitov pokvarilo. Naiven pristop k rekonstrukciji bi bilo poizkušanje vseh ključev z nizko Hammingovo razdaljo. Vendar to v primeru 256 bitnega AES ključa pri 10% izgubi še vedno pomeni $>2^{56}$ možnosti.

Alternativen pristop je uporaba ostalih namigov, ki jih lahko najdemo v pomnilniku. Ker algoritmi ob šifriranju v pomnilnik shranijo še veliko ostalih stvari in ne zgolj ključa, si lahko pomagamo s temi informacijami ob rekonstrukciji. Če najdemo v pomnilniku v naprej izračunane strukture (npr. zaporedje ključev bločne šifre, praštevili p in q ipd.), lahko te podatke uporabimo kot neke vrste kodo za odpravljanje napak pri rekonstrukciji ključa. Tako lahko ob razumnih predpostavkah rekonstruiramo DES ključ v 98% primerih ob 50% napaki, če je v pomnilniku tudi 16 podključev. Podobno lahko rekonstruiramo AES ključe ob 15% napaki v delčku sekunde ob prisotnosti zaporedja podključev in RSA ključe ob 1024 bitnih praštevilih in 6% napaki v manj kot treh minutah.

4.2.4 Iskanje ključev v pomnilniku

Preden pričnemo z rekonstrukcijo ključev želimo avtomatsko poiskati morebitne lokacije le-teh v pomnilniku. Naiven pristop bi bil, da bi poskusili vsak košček pomnilnika kot morebiten ključ ter preverili ali dešifrira nase podatke. Ob prisotnosti napak v ključih ta pristop postane neizvedljiv. Z uporabo podobne ideje o podključih kot v prejšnjem delu lahko postopek pohitrimo.

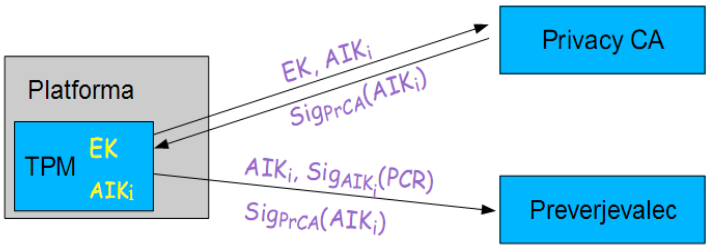
6.2.5 Napad na šifrirane diske

S prihodom TPM in uporabo šifriranja celotnega diska (npr. BitLocker, FileVault, TrueCrypt[12], dm-crypt, Loop-AES) se je povečala varnost podatkov v primerih kraje prenosnikov. Vendar pa s cold boot napadom lahko v veliki večini primerov pridemo do ključa uporabljenega za šifriranje diska. V primeru uporabe BitLocker-ja v »basic mode« (zgolj uporaba TPM brez npr. dodatne pametne kartice) lahko v manj kot pol ure pridemo do ključa za šifriranje diska. Napad uporablja USB ključek, ki vsebuje prirejeno verzijo Linuxa. Računalnik najprej vklopimo in naložimo Windows, nato pa ponastavimo in zaženemo iz USB ključka.

4.3 Direct Anonymous Attestation

Ena izmed funkcionalnosti, ki nam jih TPM ponuja je *remote attestation*. Na ta način lahko oddaljeni platformi dokažemo veljavnost našega TPM in prikažemo PCR vrednosti. Osnoven protokol [10], s katerim preverjevalcu dokažemo, da na naši platformi teče varen OS je, da mu pošljemo $EK, auth_{EK}(PCR)$. Problem tega pristopa je, da lahko dva različna preverjevalca ugotovita, da se pogovarjata z isto platformo (kar seveda ni zaželeno).

TP specifikacija v1.1 (uporaba AIK)



Attestation Identity Key (AIK)
Za potrebe atestacije lahko na TP ustvarimo (poljubno mnogo) AIK. Ta vsebuje psevdonim (izbran s strani lastnika) za našo platformo in dokazuje veljavnost našega TPM.

Tudi ta pristop ima nekaj slabosti. Pogosta potreba po izdajanju novih AIK naredi *Privacy CA* za ozko grlo. Če *Privacy CA* in preverjevalec sodelujeta, lahko odkrijeta identiteto uporabnika. *Privacy CA* moramo zaupati, da bo izdala certifikate zgolj veljavnim TPM.

TP specifikacija v1.2 (uporaba DAA [10])

- generiramo DAA ključ
- *DAA issuer* preveri nas EK in izda certifikat $Sig_{IS}(DAA)$ za nas DAA ključ
- Namesto da bi preverjevalcu poslali certifikat, mu dokažemo, da ga imamo.
    - Pošljemo AIK_i, $Sig_{AIKi}(PCR)$

[12] Brezplacna in odprtokodna resitev dostopna na naslovu http://www.truecrypt.org/

- Dokažemo da smo podpisali z DAA ključem AIK_i, ImePreverjevalca, Čas
- Dokažemo da imamo v lasti Sig_{IS}(DAA)

Ob uporabi te rešitve se *DAA issuer* in preverjevalec ne moreta povezati in tako razkriti identitete uporabnika. Odpravimo tudi ozko grlo, saj mora biti DAA certificiran le enkrat. Protokol temelji na Camenisch-Lysyanskaya shemi [9] in dokazu poznavanja diskretnega logaritma. Da lahko odkrijemo pobegle TPM (npr. če je nekdo ukradel DAA in ga sedaj uporablja zase), moramo sporočilu dodati še

$$Nym = t^{DAA\_secret} \bmod p.$$

S tem nekoliko zmanjšamo zasebnost (odvisno od načina izbire *t*).

Sedaj lahko ločimo detekcijo pobeglih TPM od preverbe platforme.

- najprej TPM pošlje dokaz o DAA in *Nym*
  - uporablja se dolgoročna izbira *t*
  - preveri se črna lista, naredi se frekvenčna analiza
  - izda se enkraten certifikat vezan na TPC preko DAA
- nato TPM pošlje dokaz o DAA in dokaz o enkratnem certifikatu
  - uporabi se naključen *t*

5 Zaključek

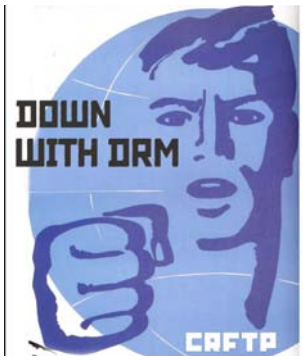
Zaradi vse večje odvisnosti od računalnikov na vseh področjih (poslovna raba, bančništvo, javna uprava, zasebna raba) se tudi računalniški varnosti posveča vse več pozornosti. Da bi zagotovili primeren nivo varnosti, do sedaj sprejete metode (npr. protivirusni programi) pogosto ne zadostujejo več. Potrebna je celovitejša rešitev primerna za domače in poslovne uporabnike osebnih računalnikov.

Prvi poskus, ki je dosegel prodor na množični trg je Trusted Platform (TP) zasnovan s strani Trusted Computing Group (TCG), ki jo podpirajo skoraj vsi večji proizvajalci računalniške opreme. Osnovna ideja tega pristopa je vzpostavitev temelja zaupanja osnovanega na strojni opremi, na katerem nato gradi veriga zaupanja. S pomočjo kriptografskih pristopov vsaka stopnja najprej preveri naslednjo in zagotovi njeno integriteto preden ji pred nadzor nad izvajanjem. Na ta način lahko zagotovimo, da se platforma na koncu naloži v predvideno in neoporečno stanje. Pri komunikaciji z oddaljenimi sistemi lahko nato dokažemo, da je naša platforma zaupanja vredna in tudi sami preverimo oddaljeno platformo.

Strojni del TP predstavlja TPM (Trusted Platform Module) čip, ki opravlja vse osnovne zaupanja vredne operacije. Ponuja nam še nekaj dodatnih funkcionalnosti kot npr. varno hrambo podatkov, kriptografske funkcije, strojni generator naključnih števil itd. Na tržišču je vse več matičnih plošč, ki ali že vsebujejo TPM čip ali pa imajo dodaten priključek za njegovo priključitev. Še največjo zastopanost pa opazimo pri prenosnih računalnikih, pri katerih je trenutno najbolj iskana lastnost (kar se varnosti tiče) šifriranje celotnega diska, saj to ob morebitni kraji zagotavlja zasebnost shranjenih podatkov.

Ob pregledu specifikacije in primerov uporabe v praksi, lahko odkrijemo nekatere pomanjkljivosti. Nekaj kriptografskih napak je bilo odkritih, predlagane so bile rešitve in tudi implementirane, saj se po potrebi specifikacija posodablja (tako je trenutna verzija 1.2). Nekaj

problemov ostaja še nerešenih (npr. cood boot napad), vendar obstajajo začasne rešitve ali pa je varnost še vedno zadostna za zastavljene cilje.



Marsikdo opozarja na spornost uporabe TPM za DRM (Digital Rights Management). Ob morebitni zlorabi TPM s strani podjetij in/ali države je mogoče z njegovo pomočjo doseči popoln DRM, cenzuro, nadzor nad sistemom, monopole itd. Nekaj rešitev, ki bi preprečile nekaj skrajnih scenarijev je že bilo predlaganih, vendar še niso bile sprejete v specifikacijo. Kljub temu, da ogromno ljudi TPM označuje zgolj kot orodje za DRM, zaenkrat njegove uporabe v ta namen še ni zaslediti.

V bližnji prihodnosti bodo verjetno skoraj vsi računalniki opremljeni s TPM. V kolikšni meri se ga bo uporabljalo in v kakšne namene (povečanje varnosti uporabnika ali DRM) pa še ni mogoče reči. Vendar so strahovi glede DRM verjetno prenapihnjeni (in bo v najslabšem primeru TPM zgolj nekaterim povzročil nekaj sivih las); splošna uporaba za namene povečanja varnosti pa je verjetno tudi še daleč (npr. Windows 7 verjetno še ne bo uporabljal TPM za preverjanje postopka nalaganja).

Literatura

[1] https://www.trustedcomputinggroup.org/groups/tpm/
[2] http://en.wikipedia.org/wiki/Trusted_Platform_Module
[3] John Marchesini, Sean Smith, Omen Wild, Rich MacDonald: Experimenting with TCPA/TCG Hardware, Or: How I Learned to Stop Worrying and Love The Bear
[4] Klaus Kursawe, Christian Stuble: Improving End-user Security and Trustworthiness of TCG-Platforms
[5] Lucky Green: Trusted Computing Platform Alliance: The Mother(board) of all Big Brothers
[6] http://www.cl.cam.ac.uk/~rja14/tcpa-faq.html
[7] Klaus Kursawe, Dries Schellekens, and Bart Preneel: Analyzing trusted platform communication
[8] J. Alex Halderman et al.: Lest We Remember: Cold Boot Attacks on Encryption Keys
[9] J. Camenisch and A. Lysyanskaya. A signature scheme with effcient protocols.
[10] Jan Camenisch: Direct Anonymous Attestation: Achieving Privacy in Remote Authentication
[11] Niels Ferguson: AES-CBC + Elephant difuser, A Disk Encryption Algorithm for Windows Vista
[12] Siani Pearson: Trusted Computing Platforms, the Next Security Solution
[13] ST19WP18 TPM Datasheet
[14] Infineon TPM 1.2 SLB 9635 TT 1.2 Product Brief

Obstaja tudi nekaj knjig, ki se ukvarjajo s TP in TPM. V njih je pregledno opisana celotna tematika in tudi vsi morebitni splošni (kriptografski) pojmi, s katerimi se ob tem lahko srečamo.

- D. Challenger et al., A Practical Guide to Trusted Computing, IBM Press
- Sean W. Smith, Trusted Computing Platform: Design and Applications, Springer
- Steven Kinney, Trusted Platform Module Basics, Newnes
- Boris Balacheff et al., Trusted Computing Platforms: TCPA Technology in Context, Prentice Hall