UDC: 004.451.9:004.087.5 INFO M: str. 32-37

PREGLED PRIMJENJENIH PRISTUPA ZA SOFTVERSKU ENKRIPCIJU PODATAKA U RAZLIČITIM OPERATIVNIM SISTEMIMA
A SURVEY OF APPLIED APPROACHES FOR SOFTWARE ENCRYPTION OF DATA IN DIFFERENT OPERATING SYSTEMS

Boris Damjanović, Dejan Simić
Univerzitet u Beogradu, Fakultet organizacionih nauka

REZIME: Paralelno sa rastom količine digitalnih informacija koje se smještaju u računarskim sistemima raste i važnost zaštite navedenih podataka. Današnji operativni sistemi nude veliki broj različitih mehanizama za šifrovanje podataka koji često pružaju nejednak nivo zaštite. U ovom tekstu će biti dat pr egled osnovnih kriptografskih rješenja zasnovanih na nivou bloka (block based) i na nivou datoteke (file based) koja su se koristila i koja se još koriste u operativnim sistemima zajedno sa prikazom napora za objedinjavanjem kriptografskih API-ja u operativnim sistemima Linux i Windows.
KLJUČNE REČI: operativni sistemi, bezbjednost i zaštita, enkripcija podataka
ABSTRACT: Along with the increased amount of digital information that are stored in computer systems increases the importance of protecting this information. Today's operating systems offer a number of different mechanisms to encrypt data that often provide unequal levels of protection. In this paper we provide an overview of basic block based and file based cryptographic solutions that are used in operating systems along with the presentation of efforts to unify cryptographic APIs in Linux and Windows OS.
KEY WORDS: operating systems, security and protection, data encryption

I. UVOD

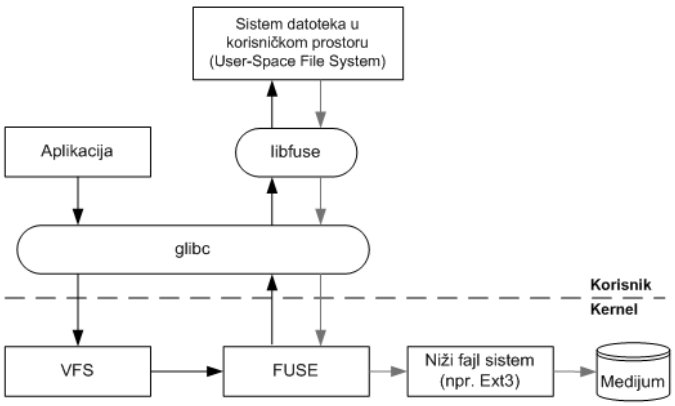
Danas postoje dva osnovna pristupa softverskoj enkripciji podataka na medijumu unutar operativnih sistema: na nivou bloka (block based) i na nivou datoteke (file based). U slučaju enkripcije na nivou bloka, mehanizam za šifrovanje se smješta između sistema datoteka i blok uređaja. U momentu kada fajl sistem treba da upiše blok podataka, taj blok se prije fizičkog zapisivanja na disk šifruje [1]. Fajl sistem je nesvjestan da je došlo do enkripcije podataka. Prednost ovoga pristupa ogleda su u činjenici da je on jednostavan, transparentan i da šifruje sve što se nalazi na uređaju. Njegov osnovni nedostatak je nepostojanje granularne kontrole šifrovanja. Kod šifrovanja na nivou bloka nije moguće tretirati jednu datoteku drugačije od druge. Ovakva enkripcija je jedna od najboljih načina za zaštitu podataka, jer ne samo da je zaštićena svaka datoteka, već je zaštićen i privremeni (temporary) sadržaj koji može sadržavati neke dijelove datoteka [2]. Za razliku od šifrovanja na nivou bloka, kriptografski fajl sistem vrši enkripciju na nivou datoteka. Podaci se ovdje šifruju prije nego što dođu do upravljačkih programa (*device driver*) blok uređaja [1]. Ovakav način omogućava mnogo veću kontrolu načina na koji će datoteke biti šifrovane. Najveći nedostatak ovakvog načina šifrovanja je kompleksnost implementacije i nemogućnost zaštite prostora za straničenje (*swap*).

U ovom tekstu će biti prikazana geneza osnovnih softverskih kriptografskih rješenja koja koriste *block based* i *file based* enkripciju u operativnim sistemima. Navedena rješenja su u jednom istom operativnom sistemu često smještena u različitim modulima i kriptografskim API-ima tako da često pružaju konkurentne kriptografske usluge. Ovako veliki broj opcija stvara konfuziju zbog čega proizvođači softvera neprekidno pokušavaju da objedine, dokumentuju i standardizuju implementirana rješenja. U ovom tekstu će takođe biti prikazani i pokušaji objedinjavanja kriptografskih API-ja u okviru Microsoft Windows te Fedora i Red Hat Linux operativnih sistema.

II. SISTEMI DATOTEKA U KORISNIČKOM PROSTORU (USER SPACE FILE SYSTEMS)

Konvencionalni operativni sistemi obično dijele virtualnu memoriju na prostor koji pripada kernelu i korisnički prostor. Prostor dodjeljen kernelu rezervisan je za izvršavanje kernela, njegovih dodataka i upravljačkih programa. S druge strane, korisnički prostor je dio memorije u kojem se izvršavaju sve korisničke aplikacije i koji se po potrebi može straničiti (*swapping*).

Mnogi sistemi datoteka u korisničkom okruženju koriste usluge FUSE (*File System in Userspace*) modula [3]. FUSE (*File System in Userspace*) je modul koji se može učitati u jezgro nekog UNIX-u sličnog operativnog sistema. Pomoću ovog modula korisnici mogu da kreiraju svoje vlastite sisteme datoteka bez izmjena izvornog koda jezgra operativnog sistema. Ovaj modul je vrlo koristan za pisanje virtualnih fajl sistema koji, za razliku od tradicionalnih, ne smještaju podatke direktno na medij. FUSE presreće pozive između korisničkog i virtualnog sistema datoteka te kernela dok im dodaje bezbjednosne funkcije [4].



Slika 1: *FUSE modul [4]*

Primjeri sistema datoteka koji koriste FUSE modul u funkciji zaštite podataka su EncFS [5] i CryptoFS [6].

Prema [4], ovi sistemi su slični jer čuvaju datoteke i njihova imena u šifrovanim direktorijumima te zahthjevaju od korisnika da montiraju (mount) šifrovane direktorijume sa korektnom lozinkom da bi pristupili datotekama u njemu. Treba primjetiti da je u okviru CryptoFS projekta najnovija verzija 0.6.0. objavljena još 2007. godine. EncFS je dostupan i na Android platformi u okviru open source aplikacije pod imenom cryptonite, a njegova najnovija verzija 1.7.4. objavljena je krajem 2010. godine.

III. LOKALNI SISTEMI DATOTEKA KOJI KORISTE *NFS* MEHANIZME

Network File System (NFS) [7] je distribuirani fajl sistem protokol koji je originalno razvijen u kompaniji SUN Microsystems 1984. godine, koji je dozvoljavao klijentskim kompjuterima da pristupaju datotekama kroz mrežu na način sličan lokalnom pristupu. Mrežni sistem datoteka (NFS) je dizajniran na *Open Network Computing Remote Procedure Call* (ONC RPC) protokolu, koji je otvoreni standard definisan u IETF RFC publikaciji zbog čega bilo ko može slobodno da ga implementira.

Pošto NFS preusmjerava neke zahtjeve upućene operativnom sistemu prema korisničkom prostoru, neki lokalni sistemi datoteka koriste njegove usluge da olakšaju razvoj. Dva lokalna sistema koji koriste njegove usluge su *Cryptographic File System* (CFS) [8] i *Transparent Cryptographic File System* (TCFS) [9].

CFS (*Cryptographic File System*) [8] predstavlja inetrfejs ka transparentnom UNIX sistemu datoteka koji omogućava automatsku enkripciju sistema direktorijuma pomoću datih lozinki. Korsnici pomoću jednostavnih komandi priključuju kriptografski ključ nekom direktorijumu. Korisnik zatim može da koristi standardne alate i usluge operativnog sistema, a datoteke će automatski biti šifrovane odnosno dešifrovane prilikom pisanja odnosno čitanja. CFS koristi samo DES kao svoj osnovni algoritam.

TCFS (*Transparent Cryptographic File System*) [9] je razvijen na odsjeku *Dipartimento di Informatica ed Applicazione univerziteta di Salerno* u Italiji a trenutno je dostupan za operativni sistem Linux. TCFS predstavlja proširenje NFS sistema koje se ponaša kao NFS ali dozvoljava korisnicima da zaštite svoje datoteke pomoću enkripcije. I ovaj sistem koristi DES algoritam za obezbjeđenje povjerljivosti podataka.

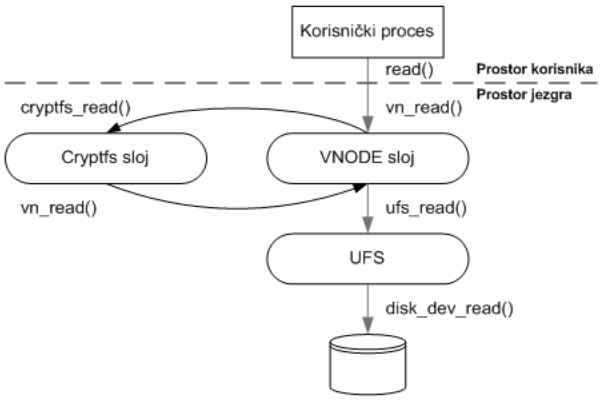
IV. SLOŽENI (STACKABLE) SISTEMI DATOTEKA

Složeni (*stackable*) sistemi datoteka [10] predstavljaju kompromis između sistema datoteka koji funkcionišu na nivou kernela i mrežnih sistema datoteka. Ovakvi sistemi datoteka mogu da funkcionišu iznad bilo kakvog fajl sistema. Oni ne moraju da prenose podatke između prostora jezgra i korisničkog prostora, a portabilni su između većeg broja operativnih sistema. Ovakvi sistemi datoteka se slažu (nadodaju) na postojeću arhitekturu operativnog sistema da bi proširili neku njihovu (obično kriptografsku) funkcionalnost. Oni presreću sistemske pozive i preusmjeravaju ih prema jednom ubačenom sistemskom sloju. Složeni (*stackable*) sistemi datoteka se izvršavaju u jezgru a mogu da operišu iznad bilo kakvog drugog fajl sistema, pri čemu nije potrebno pokretati nikakve dodatne korisničke procese.

Primjeri ovakvih složenih (*stackable*) sistema datoteka su Cryptfs, NCryptfs i eCryptfs, pri čemu su Cryptfs, NCryptfs nastali na osnovu FIST-a (*File System Translator Language*) [11], koji dozvoljava programerima da opišu sistem na višem nivou i da potom generišu kod pomoću generatora koda [12]. Uslov za interakciju sa bilo kojim fajl sistemom je postojanje dobro definisanog interfejsa sa virtualnim fajl sistemom (VFS). FIST može da generiše sistemske module za različite platforme, poput Solaris, Linux i FreeBSD operativnih sistema.

Cryptfs je složeni kriptografski sistem datoteka koji je nastao na osnovu FiST programskog alata. On je dizajniran više sa namjerom da dokaže FiST koncept nego da bude bezbjedan fajl sistem. Iako koristi samo jedan algoritam (*Blowfish*) i ograničene mogućnosti upravljanja ključevima, poslužio je kao osnova za nekoliko komercijalnih i istraživačkih sistema, kao što je ZIA [13]. Cryptfs koristi Virtualne čvorove (VNode) koji u operativnim sistemima zasnovanim na UNIX-u predstavljaju otvorenu datoteku, drektorijum, uređaj ili drugi entitet za uključenje vlastitog modula u kernel [14], kao što je prikazano na slici 2.

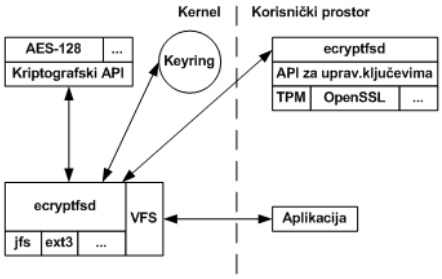
NCryptfs [15] je složeni (*stackable*) sistem datoteka koji je kreiran sa namjerom da obezbjedi bezbjednost, povjerljivost i performanse. NCryptfs je direktni naslijednik Cryptfs sistema, pri čemu je u mnogim oblastima unaprijedio rješenja starijeg sistema. NCryptfs podržava više korisnika, ključeva, načina autentikacije i algoritama šifrovanja. Osnovni cilj NCryptFS sistema bio je da obezbjedi transparentnu i lako portabilnu enkripciju datoteka bez većeg gubitka performansi.



Slika 2: *Cryptfs [14]: složeni kriptografski sistem datoteka*

eCryptfs [16] je složeni kriptografski sistem datoteka koji je kreiran za operativni sistem Linux. Ovakav sistem je složen iznad postojećih montiranih (mounted) sistema datoteka koji se nazivaju niži fajl sistemi. I eCryptfs i niži fajl sistem (ext3, JFS, NFS...) su registrovani u kernelovom virtualnom sistemu datoteka (VFS). eCryptfs presreće pozive nižem fajl sistemu

pa šifruje i dešifruje sadržaje koji se upisuju ili očitavaju iz njega. On dobija ključeve iz keyringa (vlastite baze podataka o ključevima) i koristi kernelov kriptografski API da bi izvršio šifrovanje i dešifrovanje sadržaja datoteka. eCryptfs za sve zahtjeve vezane za upravljanje ključevima koristi ecryptfsd *daemon*, kako je prikazano na slici 3 [16].



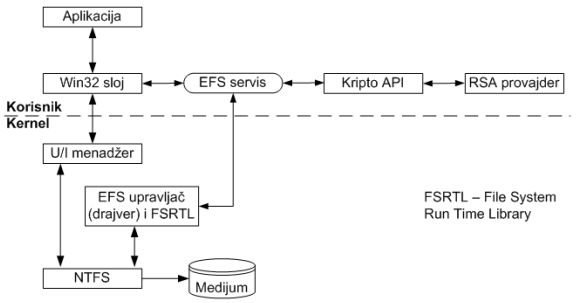
Slika 3: *Način funkcionisanja eCryptfs sistema [16]*

V. *DISK BASED* SISTEMI ZA ENKRIPCIJU DATOTEKA

Sistemi datoteka orjentisani ka disku (*disk based*) [17] obezbjeđuju strukturu nizu bita koji se nalazi na mediju, pružaju usluge imenovanja datoteka i direktorijuma te pristup datotekama i sektorima.

Sistemi datoteka orjentisani ka disku (*disk based*) koji šifruju podatke to čine na višem nivou apstrakcije nego sistemi orjentisani ka blokovima (*block based*) [17]. S druge strane disk based fajl sistemi se izvršavaju na nižem nivou apstrakcije nego složeni (stackable) sistemi datoteka, softverski programi za enkripciju datoteka ili lokalni sistemi koji koriste NFS mehanizme [4]. Ovakvi sistemi imaju pristup svim podacima i atributima karakterističnim za jednu datoteku ili direktorijum tako da mogu da koriste mnogo kompleksniju autorizaciju i autentikaciju nego block-based sistemi datoteka. *Disk based* sistemi datoteka imaju punu kontrolu nad metapodacima i operacijama direktorijuma i datoteka koje obrađuju.

Primjer jednog takvog sistema je Microsoft-ov Encryption File System (EFS) [18]. EFS proširuje journaling (dnevnik) a koristi i Windows autentikaciju i liste za kontrolu pristupa (ACL) [19]. EFS je podržan na operativnim sistemima koji su zasnovani na Microsoft NT jezgru, poput Windows 2000, XP, Vista i 7. Slika 4. demonstrira kako EFS proširuje NTFS unutar i van prostora jezgra.



Slika 4: *EFS komponenate u korisničkom prostoru i u prostoru jezgra [18]*

EFS koristi i simetričnu i asimetričnu kriptografiju [20]. Kada se datoteka šifruje po prvi put, kreira se za tu datoteku jedinstven simetrični FEK ključ (*File Encryption Key*), pa se pomoću njega šifruje datoteka. Taj ključ se ugrađuje u datoteku šifrovan korisnikovim javnim ključem. Proces dešifrovanja počinje dešifrovanjem FEK ključa korisnikovim privatnim ključem, koji se nalazi unutar njegovog Windows profila. Do Windows 2000 EFS je koristio DESX algoritam, a kasnije verzije operativnih sistema dodaju i AES i Triple DES.

Z sistem datoteka (*Z File System*) [20] je sistem datoteka kreiran od strane kompanije SUN Microsystems. ZFS je implementiran kao open source softver, a sada je u vlasništvu kompanije Oracle. Oracle Sloaris 11 je Z sistemu datoteka dodao mogućnost transparentnog šifrovanja. Svi podaci i metapodaci fajl sistema (poput vlasništva, lista za kontrolu pristupa, informacija o kvotama itd.) su sakriveni kada se im se zaštiti privatnost u okviru ZFS pool-a. ZFS pool može da sadrži i mješavinu šifrovanih i otvorenih (nešifrovanih) skupova podataka. Pri ovome je šifrovanje aplikacijama i ostalim Oracle Solaris servisima poput NFS i CIFS potpuno transparentno.

VI. *BLOCK BASED* SISTEMI ZA ENKRIPCIJU DATOTEKA

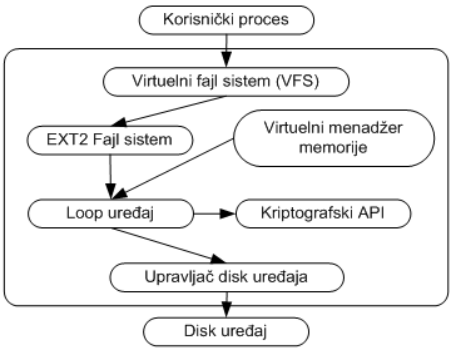
Sistemi datoteka orjentisani ka bloku (*block based*) [4] vrše šifrovanje na nižem nivou apstrakcije od fajl sistema. Ovi sistemi funkcionišu transparentno ispod fajl sistema i šifruju podatke na nivou blokova podataka hard diska ili drugog medijuma.

BITLOCKER dodatak operativnom sistemu je dostupan u *Enterprise* i *Ultimate* izdanjima Windows Vista i Windows 7 operativnih sistema. On je takođe dostupan u *Pro* i *Enterprise* izdanjima Windows 8 operativnog sistema te u Windows Server 2008 R2 i Windows Server 2012 operativnim sistemima. BitlockerTM *Drive Encryption* može da šifruje sve podatke i na sistemskom disku [21]. BitLocker je nametnuo neke nove bezbjednosne zahtjeve koji nisu bili prisutni kod uobičajenih režima rada i algoritama. Zbog toga je kombinovan dobro poznat i istestiran algoritam AES u CBC režimu rada sa jednom novom komponentom koja je dobila ime *Elaphant* raspršivač (*Elaphant diffuser*), zbog čega se navedeni algoritam ponaša kao algoritam modifikovan javnim parametrom (*Tweakable Block Cipher*). On može da u radu koristi tri načina rada: transparentni režim uz upotrebu TPM (*Trusted Platform Module*) 1.2 čipa [22], pomoću korisničke autentikacije kada zahtjeva da korisnik unese PIN ili priključi USB token.

Linux Cryptoloop je modul koji je dodat Linux kernelu da bi se olakšala integracija enkripcije ispod nivoa fajl sistema [23]. Ova vrsta enkripcije funkcioniše bez obzira o kojem se sistemu datoteka radi. Sa *Cryptoloop* modulom, fizički disk ili neka logička particija kao i proizvoljna datoteka se predaje na obradu tzv. *loop* uređaju (*loop device*).

Linux Loopback upravljač (*device driver*) omogućava predstavljanje neke datoteke kao blok uređaja, pri čemu opciono transformiše podatke prije nego što ih očita ili upiše na medijum obično koristeći enkripciju. Trenutno IPSEC i *Cryptoloop* driver koriste ovu mogućnost [10]. Nakon što se

jezgru Linux sistema da ključ pomoću losetup uslužnog programa, *loop device driver* dalje transparentno vrši šifrovanje i dešifrovanje kad god se pristupa loop uređaju. *Loop* uređaj se može inicijalizovati i montirati (*mount*) da koristi bilo koji sistem datoteka. Način funkcionisanja Cryptoloop modula prikazan je na slici 5.



Slika 5: *Cryptoloop modul [23]*

Kao zamjena za Cryptoloop modul od verzije 2.6 operativnog sistema Linux koristi se dm-crypt *plugin* (podsistem). Dm-crypt funkcioniše tako što koristi Device-Mapper [24], infrastrukturu koja je predstavljena u verziji 2.6 operativnog sistema Linux a koja je namjenjena da obezbjedi generički način za kreiranje virtualnih slojeva blok uređaja iznad stvarnih blok uređaja koji komuniciraju sa hardverom. Dm-crypt koristi Linuxov kripto API i omogućava šifrovanje blok uređaja kao što su diskovi, particije ili RAID nizovi. Dm-crypt podržava kriptografske algoritme i režime rada koji su prisutni u Linux-ovom kripto API-ju što uključuje AES [25], DES, Serpent, Twofish, Blowfish, ECB i CBC režim rada, pri čemu je inicijalizacioni vektor baziran na broju sektora [4].

Enkripcija diska na Android operativnom sistemu je zasnovana na Dm-crypt podsistemu, koji je u stvari funkcija jezgra koja funkcioniše nad slojem blok uređaja [26]. Zbog toga ne može da funkcioniše sa YAFFS sistemom datoteka, koji se obraća direktno *raw nand* fleš čipu, ali može da radi sa Emmc i sličnim fleš uređajima koji se kernelu predstavljaju kao blok uređaji.

Iako je Dm-crypt pružio solidan interfejs za enkripciju diska, alati poput dmsetup uslužnog programa koji su služili za konfigurisanje *Device Mapper*-a nisu bili previše pristupačni za rad. Zbog toga je razvijen uslužni program koji se fokusirao na pojednostavljenje konfigurisanja Dm-crypt uređaja, a koji je dobio naziv Cryptsetup [27].

FileValut [28] je metod za enkripciju diska na Mac računarima. Iako je u verziji 1 koja je predstavljena sa operativnim sistemom Mac OS X Panther mogao da šifruje samo home direktorijum, u verziji 2 je pretrpio značajna unapređenja. Mac OS X Lion i noviji koriste FileValut 2 koji šifruje individualne blokove diska, tako da je enkripcija nevidljiva sistemu datoteka.

VII KRIPTOGRAFSKI API-JI U OKVIRIMA JEZGARA OPERATIVNIH SISTEMA

Različite ideje o načinu i mjestu na kojem se vrši šifrovanje podataka u okviru operativnih sistema dovele su do pojave različitih kriptografskih API-ja (*Application Programming Interface*) koji su nerijetko bili smješteni u različite module, a koji često pružaju iste ili slične kriptografske usluge. Zbog toga uporedo sa razvojem kriptografskih API-ja u okvirima operativnih sistema traju napori za objedinjavanjem, dokumentacijom i standardizacijom primjenjenih rješenja.

FIPS (*Federal Information Processing Standard*) 140-2 je globalno prepoznatljiva serija sigurnosnih standarda vlade SAD koji navode zahtjeve za kriptografske module. Do sada su objavljene prva i druga verzija ovoga standarda, a u pripremi je i treća verzija. Objavljene verzije su definisale 4 nivoa sigurnosti, od kojih prvi nivo navodi listu odobrenih kriptografskih algoritama i funkcija, dok se ostali nivoi bave različitim aspektima fizičke bezbjednosti. Prvi standard, FIPS 140-1 nastao je 1994. godine na osnovu prijedloga velikih američkih proizvođača i korisnika kriptografskih usluga. Dokument FIPS 140-2 koji je nastao 2001. godine na osnovu promjena u tehnologiji kao i na osnovu primjedbi proizvođača i korisnika predstavljao je osnovni ulazni dokument za kreiranje ISO/IEC 19790 standarda iz 2006. i 2012. godine. Da bi prošli validaciju, proizvođači softvera su obavezni i da dokumentuju svoje proizvode na uniforman način, a navedeni dokumenti nakon procesa validacije postaju dostupni na NIST (*National Institute of Standards and Technology*) web stranici. Veliki broj današnjih proizvođača kriptografskog softvera i operativnih sistema verifikuje svoje kriptografske module u skladu sa FIPS 140-2 standardom.

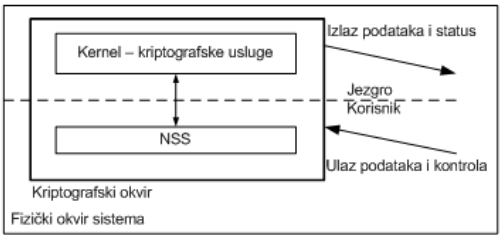
VII-1 *Kriptografski API u okviru Linux operativnog sistema*

Kriptografija je u određenoj mjeri bila prisutna u većem broju jezgara Linux operativnog sistema, od Theodore Ts'o-ove integracije MD5 i SHA-1 heš funkcija u jezgro [30] do IPSEC implementacije na nivou kernela koju su realizovali Aleksej Kuznjecov (Alexey Kuznetsov) i Dejv Miler (Dave Miller) kada je u jezgro integrisan i CryptoAPI Džejmsa Morisa (James Morris). Linux CryptoAPI podržava tri tipa [31] algoritama: jednosmjerne heš funkcije, algoritme za šifrovanje i za kompresiju podataka. Jezgro operativnog sistema Linux implementira veliki broj simetričnih algoritama i kriptografskih režima rada (modes of operation) sa ograničenom podrškom asimetričnoj kriptografiji. Pojavljuju se sinhroni i asinhroni kriptografski interfejsi, pri čemu su asinhroni korisni za podršku kriptografskom hardveru. Treba reći i da raste podrška hardverski orjentisanim kriptografskim mogućnostima i da je nekoliko algoritama optimizovano prema modernim arhitekturama. Kriptografski API koji se nalazi u jezgru Linux operativnog sistema koriste između ostalih i IPSEC, moduli za enkripciju diska kao što su ecryptfs i dm-crypt kao i rutine za verifikaciju digitalnih potpisa kernelovih modula[32].

Na Linux platformi kriptografske usluge su podijeljene u na mnogo različitih modula koji pokušavaju da pruže iste ili slične usluge, pri čemu različiti moduli nisu međusobno kompatibilni. Konkretno, kriptografske biblioteke koje se isporučuju sa Fedora Linux operativnim sistemom su NSS, OpenSSL, libgcrypt, GnuTLS, Kernel, libcrypt.so (iz paketa glibc), libssh2, nettle, python-crypto i BeeCrypt. Činjenica da veliki broj opcija stvara konfuziju kod razvojnih timova i komplikuje pravne obaveze vezane za izvozne zakone SAD je uočena kao problem u okviru Fedora projekta (sponzorisanog od strane kompanije Red Hat). Zbog toga su u okviru podprojekta Fedora *Crypto Consolidation* [33] preduzeli napore za

dizajnirenje jedinstvenog kriptografskog okruženja koje će biti zasnovano na samo jednom skupu biblioteka koji će biti usklađen sa FIPS 140 standardom. Za ispunjenje ciljeva navedenog projekta odabran je skup NSS kriptografskih biblioteka koje su četiri puta bile verifikovane prema FIPS 140 standardu. *Network Security Services* (NSS) predstavlja skup open source biblioteka koje su namjenjene za razvoj bezbjednosnih klient i server aplikacija na različitim platformama[34]. NSS biblioteka potiče od izvornog koda za SSL biblioteku koji je razvijan u okviru kompanije Netscape, a danas NSS biblioteku razvija veliki broj kompanija uključujući Mozilla fondaciju, Red Hat, Oracle korporaciju, Google i druge.

Softverski kriptografski okvir za kriptografski API u okviru kernela *Red Hat Enterprise Linux* operativnog sistema koji je verifikovan u skladu sa FIPS 140-2 standardom [35] dat je na slici 6:

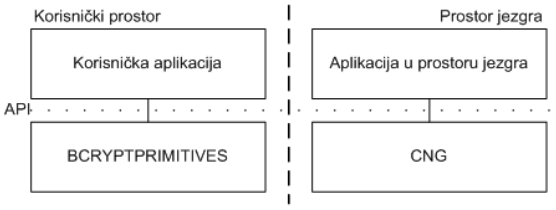


Slika 6: *Red Hat Enterprise Linux 6.2 Kernel Crypto API Cryptographic Module v2.0 - kriptografski okvir sistema*

Treba reći da se u dokumentaciji *Fedora Crypto Consolidation* projekta navodi da je Microsoft vjerovatno otišao najdalje u naporima da objedini kriptografske usluge u okviru jedne dobro integrisane kriptografske biblioteke.

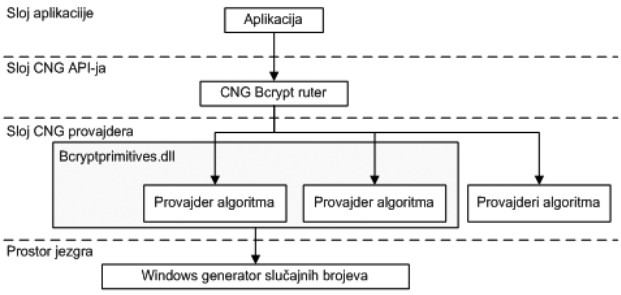
VII-2 *Kriptografski API u okviru Windows operativnog sistema*

U okviru MS Windows operativnog sistema kriptografski API-ji su se razvijali u tri faze [36]. Prva faza počela je sa Windows NT 4.0 operativnim sistemom koji je u korisničkom prostoru implementirao kriptografski API [37] pomoću kombinacije advapi32.dll i crypt32.dll biblioteka koji su dalje pozivale kriptografske provajdere, dok je u prostoru jezgra koristio modul pod nazivom fips.sys. Druga faza počinje sa dolaskom operativnog sistema Windows Vista kada je predstavljen kriptografski API slijedeće generacije [38] (*Cryptographic API: Next Generation* ili CNG). U korisničkom prostoru više nema potrebe da se koriste posredničke usluge modula advapi32.dll i crypt32.dll, već aplikacije u korisničkom prostoru komuniciraju sa bcrypt.dll modulom. Aplikacije u prostoru jezgra pristup kriptografskom API-ju dobijaju pomoću modula pod nazivom ksecdd.sys (*Microsoft Kernel Mode Security Support Provider Interface*). Pri prelasku na novu verziju je radi kompatibilnosti sačuvana mogućnost korišćenja starog kriptografskog API-ja. Treća faza počinje sa dolaskom Windows 7 i Windows Server 2008 R2 operativnog sistema [39]. U korisničkom prostoru kriptografski algoritmi su implementirani u biblioteci bcryptprimitives.dll koja se poziva preko bcrypt.dll biblioteke, dok se u prostoru jezgra koristi drajver pod imenom cng.sys, koji podržava istovremeno i nove CNG pozive ali i pozive upućene ka staroj fips.sys biblioteci.



Slika 7: *CNG arhitektura u Windows 7 i Windows Server 2008 R2 i novijim operativnim sistemima*

Kada su u pitanju korisničke aplikacije, bcryptprimitives.dll modul posjeduje skup funkcija pomoću kojeg se može pristupiti CNG frejmvorku da bi se dobile reference za pristup provajderima pojedinih kriptografskih algoritama [40]:



Slika 8: *Veza modula bcryptprimitives.dll modula sa ostalim kriptografskim komponentama*

Treba reći da su u Microsoftu proveli proces validacije različitih operativnih sistema i odgovarajućih modula iz navedene tri faze u skladu sa FIPS 140-2 standardom, pri čemu su proces validacije proširili i na odgovarajuće kriptografske provajdere koji se isporučuju sa odgovarajućim verzijama operativnog sistema Windows.

VIII. ZAKLJUČAK

U ovom radu prikazana je geneza kriptografskih rješenja koja su se koristila i koja se i dalje koriste u različitim operativnim sistemima. Ako posmatramo *block based* sisteme u okviru operativnog sistema Windows se izdvaja *BitLocker* koji je podržan u novijim edicijama poput *Pro* i *Enterprise* izdanja Windows 8 operativnog sitema. U slučaju Linux operativnog sistema vrlo je aktivan projekat u okviru kojeg se razvija Dm-crypt - najnovija verzija je objavljena 4.avgusta 2013. godine, dok je za Mac OS X aktuelna verzija FileValut 2.

U slučaju disk based sistema u Windows 8 Pro i Enterprise verzijama dostupan je EFS, što potvrđuje da Microsoft ne odustaje od ove tehnologije. U slučaju operativnog sistema Oracle Solaris 11 Z File System prestaje da bude open source, dok od ostalih operativnih sistema treba spomenuti portove za OSX i FreeBsd operative sisteme. Kada se pomenu složeni (*stackable*) sistemi datoteka pod operativnim sistemom Linux aktivno se razvija projekat eCryptfs, koji je zabilježio zadnje izdanje 25.januara 2013. godine.

Kada je u pitanju proces objedinjavanja, dokumentovanja i standardizacije kriptografskih API-ja najdalje je otišao Microsoft sa Windows 7 operativnim sistemom, dok ga u Linux svijetu prati projekat pod nazivom *Fedora Crypto Consolidation* koji je sponzorisan od strane kompanije Red Hat.

ZAHVALNOST

Ovaj rad je dio projekta "Primena multimodalne biometrije u menadžmentu identiteta", finansiranog od strane Ministarstva Prosvete i Nauke Republike Srbije, pod zavodnim brojem TR-32013.

REFERENCE

[1] D.Defreez, Android Privacy Through Encryption, Master Thesis of Science in Mathematics and Computer Science, Southern Oregon University, 2012
[2] J.Fuller, J.Ha, D.O'Brien, S.Radvan, E.Christensen, A.Ligas, Fedora Security Guide, http://docs.fedoraproject.org
[3] M. Szeredi, Filesystem in USEr space. http://sourceforge.net/projects/avf, 2008.
[4] S.M. Diesburg and A.A.Wang, A Survey of Confidential Data Storage and Deletion Methods, ACM Computing Surveys (CSUR), Volume 43 Issue 1, novembar 2010, Article No. 2
[5] V. Gough, EncFS: Encrypted file system. http://arg0.net/wiki/encfs, 2008.
[6] C. Hohmann, CryptoFS. http://reboot.animeirc.de/cryptofs/, 2008.
[7] R.Sandberg, D. Goldberg, S. Kleiman, D. Walsh, B. Lyon, "Design and Implementation of the Sun Network Filesystem," USENIX Conference Proceedings, USENIX Association, Berkeley, CA, Summer 1985.
[8] M.Blaze, A Cryptographic File System for Unix, ACM Conference on Computer and Communications Security 1993
[9] M.Ermelindo: Transparent Cryptographic File System, Linux Journal, Volume 1997 Issue 40es, Aug. 1997, Article No. 3
[10] C.P. WRIGHT, Operating System Support for Extensible Secure File Systems, Technical Report FSL-04-02, maj 2004
[11] E.Zadok and J. Nieh, FiST: A language for stackable file systems. In Proceedings of the Annual USENIX Technical Conference. USENIX Association, Berkeley, CA, 55–70, 2000.
[12] E.Zadok, FiST: A System for Stackable File - System Code Generation, PhD Thesis, Columbia University, 2001
[13] M. Corner and B. D. Noble. Zero-interaction authentication. In The Eigth ACM Conference on Mobile Computing and Networking, septembar 2002
[14] E.Zadok, I.Badulescu, and A.Shender, Cryptfs: A Stackable Vnode Level Encryption File System, Technical Report CUCS-021-98. Computer Science Department, Columbia University, jul 1998. Dostupno na: http://www.cs.columbia.edu/~library/.
[15] E.Zadok, I.Badulescu, and A.Shender, Extending File Systems Using Stackable Templates, Proceedings of the USENIX Annual Technical Conference, Monterey, California, USA, jun 1999
[16] M.Halcrow, eCryptfs: a Stacked Cryptographic Filesystem, Linux Journal, Volume 2007 Issue 156, april 2007
[17] C.P. Wright, Operating System Support for Extensible Secure File Systems, Technical Report FSL-04-02, maj 2004
[18] Microsoft Corp., Using Encrypting File System, Tech. rep. 2005. Dostupno na: http://technet.microsoft.com/en-us/library/bb457116.aspx.
[19] Microsoft Corp., How EFS works. Tech. Rep. 2008. Dostupno na: http://technet.microsoft.com/en-us/library/cc962103.aspx.
[20] Darren Moffat, How to Manage ZFS Data Encryption, Oracle technetwork, 2012
[21] N.Ferguson, AES-CBC + Elephant difuser A Disk Encryption Algorithm for Windows Vista, Microsoft White Paper, 2006
[22] IBM Blueprints, Securing Sensitive Files With TPM Keys, IBM Corporation, 2009
[23] M.J.O. Saarinen, Encrypted watermarks and linux laptop security, Proceeding WISA'04 Proceedings of the 5th international conference on Information Security Applications, Springer-Verlag Berlin, Heidelberg, 2005
[24] Red Hat Inc. Device-mapper Resource Page. Website. http://sources.redhat.com/dm/
[25] B.Damjanović and D.Simić, Performance evaluation of AES algorithm under Linux operating system, Procedings Of The Romanian Academy, Series A, Mathematics, Physics, Technical Sciences, Information Science, Volume 14, Number 2, pp. 177–183, april - jun 2013,
[26] Notes on the implementation of encryption in Android 3.0, Android open source project Teach Info, http://source.android.com/tech/encryption/android_crypto_implementation.html
[27] W. M.Petullo, Disk encryption in Fedora: Past, present and future, RedHat Magazine, jan. 2007, http://magazine.redhat.com/2007/01/18/
[28] Apple Technical White Paper, Best Practices for Deploying FileVault 2, Deploying OS X Full Disk Encryption Technology, avgust 2012, http://training.apple.com/pdf/WP_FileVault2.pdf
[29] Cryptographic Module Validation Program, National Institute Of Standards And Technology, Dostupno na: http://csrc.nist.gov/groups/STM/cmvp/index.html, Pristupano: avgust 2013
[30] J.L.Cooke and D.Bryson. Strong Cryptography in the Linux Kernel. In Proceedings of the Linux Symposium, pages 139-144, Ottawa, Canada, jul 2003.
[31] J.Morris, The Linux Kernel Cryptographic API, Linux Journal, Volume 2003 Issue 108, april 2003
[32] J.Morris, Overview of Linux Kernel Security Features, Linux.com, jul 2011, Dostupno na: http://www.linux.com/learn/docs/727873-overview-of-linux-kernel-security-features/, Pristupano: avgust 2013
[33] Fedora Crypto Consolidation, Fedora Project, Dostupno na: https://fedoraproject.org/wiki/FedoraCryptoConsolidation, Pristupano: avgust 2013
[34] Network Security Services, Mozilla Developer Network, Dostupno na: https://developer.mozilla.org/en/docs/NSS, Pristupano: avgust 2013
[35] FIPS 140-2 Security Policy Document, Red Hat Enterprise Linux 6.2 Kernel Crypto API Cryptographic Module v2.0, februar 2013
[36] Microsoft Corp., FIPS 140 Evaluation, februar 2012, Dostupno na: http://technet.microsoft.com/en-us/library/cc750357.aspx,
[37] R. Coleridge, The Cryptography API, or How to Keep a Secret, avgust 1996, Dostupno na: http://msdn.microsoft.com/en-us/library/ms867086.aspx
[38] Microsoft Corp., MSDN, Cryptography API: Next Generation, Dostupno na: http://msdn.microsoft.com/en-us/library/windows/desktop/aa376210(v=vs.85).aspx, Pristupano: avgust 2013
[39] Microsoft Corp., TechNet, Cryptography Next Generation, Dostupno na: http://technet.microsoft.com/en-us/library/cc730763(v=ws.10).aspx, Pristupano: avgust 2013
[40] FIPS 140-2 Security Policy Document , Microsoft Windows Cryptographic Primitives Library (bcryptprimitives.dll) Security Policy Document, maj 2011

M. Sc. Boris Damjanović,
Kontakt: damjanovic@koledzprijedor.org
Oblasti interesovanja: zaštita računarskih sistema, kriptografija, informacioni sistemi, interakcija čovek-računar, programiranje i jezici, paralelizacija koda

Dr Dejan Simić, redovni prof., FON
Kontakt: dsimic@fon.bg.ac.rs
Oblasti interesovanja: elektronski sistemi plaćanja, primena kriptografije, zaštita podataka i računarskih sistema, PCI standardi, informacione tehnologije i njihova primena