Analiza zaštite podataka u bazama podataka na lokalnom i serverskom nivou

Saša Adamovic, Mladen Veinović, Fakultet za informatiku i menadžment, Univerzitet Singidunum

***Sadržaj* — U ovom radu analizirana je zaštita podataka i informacija u bazama podataka korišćenjem kriptografskih algoritmima. Polazeći od teorijskih osnova baza podataka razmatraju se konkretna rešenja zaštite podataka u bazama podataka. Analiza obuhvata standardna bezbednosna rešenja zasnovana na komercijalno dostupnim kriptografskim algoritmima. Prikazaće se instalacija standardnih kriptografskih sistema nad bazom podataka i bezbednosna rešenja realizovana na aplikativnom serveru. Ceo postupak je zasnovan na AES kriptografskom algoritmu. Obavljena je eksperimentalna analiza uticaja šifrovanja na performanse. Primenjenim (predočenim) rešenjima podaci su zaštićeni u bazi podataka, kao i podaci koji se nalaze u vidu upitnih formi.**

***Ključne reči* — analiza performansi, algoritami za šifrovanje, AES, zaštita podataka, php, mysql, apache, aplikativni server, lokalna zaštita.**

I. Uvod

U PROFESIONALNIM sistemima, zaštita baza podataka je postala visok prioritet u svetu. Cilj svake organizacije je da štiti bezbednosno osetljive podatke u bazama podataka u kojima se nalaze informacije o klijentima i njihovim poslovanjima kao i raznovrsna tehnička dokumentacija. Pri kreiranju strategije za zaštitu baze podataka od značaja je više faktora. Kao dodatna garancija i preduslov za mogućnost realne procene nivoa sigurnosti rešenja jeste i upotreba pouzdanih nosećih softverskih komponenti čiji je izvorni kod dostupan.

Jedan od posrednih uslova za uspešnost rešenja predstavlja i njegova jednostavnost upotrebe. Ideal u konkretnim rešenjima je da krajnji korisnici što manje osećaju prisustvo kriptografskih rešenja, tj. da im se posao ne usložnjava i da se ne menjaju vremenski resursi kod izvršavanja poslova. Realizacija rešenja zaštite nad samom bazom podataka obezbeđuje odličan metod za zaštitu osetljivih podataka, ali sa druge strane dovodi do smanjenja performansi i usložnjava upotrebu. U skladu sa tim, nivo lokalne zastite se postavlja kao moguće rešenje. Sa druge strane, kod rešenja na aplikativnom serveru moguće je vršiti selektivnu primenu šifrovanja. To bi značilo da se mogu šifrovati samo neki, bezbednosno važni podaci kao što su npr. brojevi kreditnih kartica, podaci o klijentima i sl.

Saša Adamović, Fakultet za informatiku i menadžment, Univerzitet Singidunum, Danijelova 32 Beograd; telefon: 381-11-3093248; e-mail: sadamovic@singidunum. ac.yu

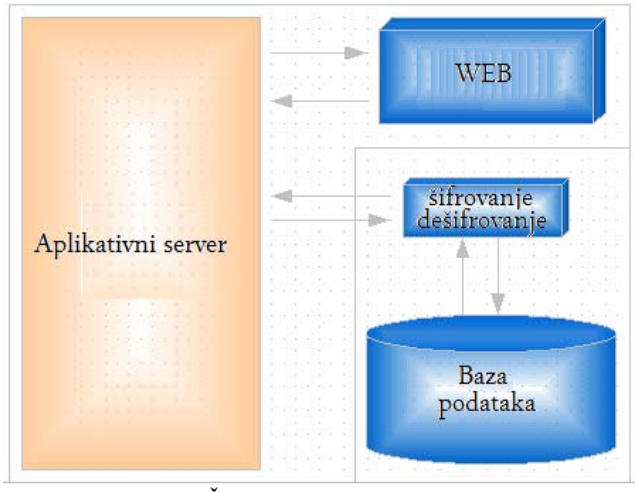
Mladen Veinović, Fakultet za informatiku i menadžment, Univerzitet Singidunum, Danijelova 32 Beograd; telefon: 381-11-3093227; e-mail: mveinovic@singidunum. ac.yu

Lokalna zaštita se postiže instalacijom kriptografskog modula na aplikativnom serveru ili upotrebom internog kriptografskog modula unutar baze podataka. Kod ovakvog mehanizma šifrovanje se vrši samo na serverskoj strani. Cilj ovakvog mehanizma je da fizički zaštiti podatke na serveru. Podaci se čuvaju u šifrovanoj formi na disku i obezbeđeni su od krađe diska ili neovlašćenog pristupa. Ovakvi kriptografski mehanizmi koriste popularne standardne blokovske kriptografske algoritme (AES, 3DES...).

U ovom radu obrađeni su postupci realizacije šifrovanja unutar baze podataka, kao i realizacija procesa šifrovanja na strani aplikativnog servera. Ova rešenja nude dostupnost izvornog koda, pre svega proširenje raspoloživih algoritama za šifrovanje sopstvenim algoritmom. Kroz analizu performansi navedenih rešenja detaljno se analizira složenost realizacije i opterećenje procesora.

II. Šifrovanje unutar baze podataka

Za realizaciju ovog rešenja lokalne zaštite korišćena je MySql baza podataka. Izvorni kod nekih od realizacija je dostupan i/ili postoji mogućnost proširenja dok su neke realizacije potpuno zatvorene. MySql SUBP ne poseduje sve funkcionalnosti kao sistemi tipa Oracle ali dovoljan skup funkcionalnosti i povoljna cena su pozitivno uticali na prihvatanje ovog softvera od strane tržišta. Danas većina programskih jezika poseduje biblioteke za korišćenje MySQL-a, a postoji i ODBC interfejs za jezike za koje nije ugrađena podrška. Mesto modula za šifrovanje kod lokalne zaštite baze prikazano je na slici 1.



Sl 1. Šifrovanje unutar baze podataka

MySql u svom kriptografskom modulu koristi dva najpoznatija algoritma za šifrovanje AES i DES. U eksperimentalnoj analizi izabran je AES algoritam ("Rijndael") koji koristi funkcije AES_ENCRYPT() i AES_DECRYPT() za šifrovanje i dešifrovanje podataka.

U ovom modulu MySql može da koristi dužinu ključa od 128 ili 256 bita. U radu korišćen je ključ od 256 bita zbog povećanja sigurnosti što se odražava na performanse sistema. Funkcija AES_ENCRYPT() kao rezultat vraća binarni string (šifrat), a funkcija AES_DECRYPT() vraća orginalni string (otvoreni tekst). U slučaju kada se proslede pogrešni argumenti funkcijama, obe funkcije vraćaju kao rezultat nulu.

Za realizaciju funkcije AES_ENCRYPT() kao argumente funkciji prosleđujemo tekst za šifrovanje i ključ algoritma dužine 128 ili 256 bita sa kojim šifrujemo, a za realizaciju funkcije AES_DECRYPT() kao argumente prosleđujemo šifrat i ključ koji smo koristili kod funkcije AES_ENCRYPT() pri šifrovanju.
Navedene su prednosti i mane ovog koncepta zaštite:

Prednosti:
- realizacija AES algoritma u C programskom jeziku,
- podaci se u bazi čuvaju u šifrovanom obliku

Mane :
- nedostupnost izvornog koda
- ključ se nalazi zajedno sa šifrovanim podacima
- promena ključa (proces koji zahteva dešifrovanje i ponovno šifrovanje kompletne baze podataka)
- podaci nisu zaštićeni u toku komumikacije sa bazom

III. Šifrovanje na aplikativnom serveru

Ovaj vid lokalne zaštite se postiže na klijent/server arhitekturi instalacijom kriptografskog modula na aplikativnom serveru. Komponente klijent/server arhitekture moraju se povinovati nekim osnovnim principima kako bi međusobno delovale ispravno. Ovi principi moraju biti jednoznačno upotrebljivi u komponentama klijenta, servera i komunikacionog posrednika. Principi koji moraju biti ispunjeni su:

- hardverska nezavisnost,
- softverska nezavisnost,
- otvoreni pristup za servise,
- distribucija procesa.

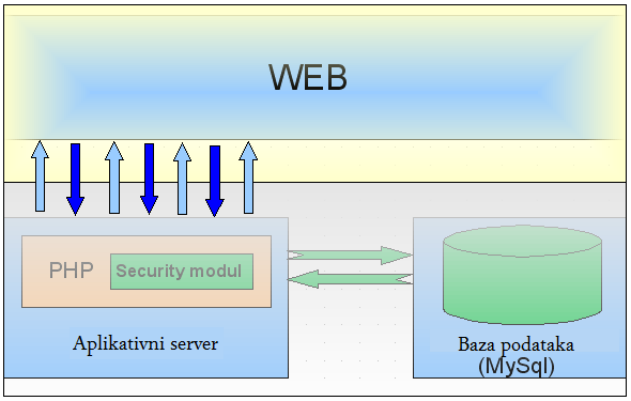
U izradi kriptografskog modula za kompletnu funkcionalnost AES algoritma korišćen je php programski jezik zbog svojih sve naprednijih mogućnosti i zato što pripada grupi open source jezika. Dodatna prednost php-a jeste i mogućnost direktnog korišćenja gotovih komponenti izrađenih u java programskom jeziku.

Algoritam je napisan prema standardima propisanim od strane FIPS-a (*Federal Information Processing Standards*). Navedeni AES algoritam ima podršku za tri dužine ključa (128, 192, 256). Mogućnost korišćenja algoritma u ovom rešenju podrazumeva adekvatan API (tj. mogućnost proširenja rešenja sopstvenim modulima) ili/i dostupnost izvornog koda rešenja.

S obzirom na to da je klijent-server arhitektura iskorišćena u vidu web aplikacija koju može da karakteriše veliki broj korisnika istovremeno, performanse i konkurentni pristup su dva pitanja na koja treba obratiti pažnju pri realizaciji. Različita serverska rešenja imaju različite pristupe za povećanje performansi.

Na slici broj 2 je ilustrovana moguća klijent server arhitektura koju je moguće iskoristiti u procesu zaštite podataka u bazama podataka. Prikazana arhitektura se sastoji od dva servera. Na jedanom serveru se nalazi baza podataka, a na drugom aplikativni server. Aplikativni server implementira php modul sa modulom za šifrovanje i dešifrovanje. Između ova dva servera postoji direktna komunikacija. Podaci koji se razmenjuju su u obliku šifrata.

Imajući u vidu da je php interpreterski jezik, performanse su znatno slabije u odnosu na prethodno analiziranu arhitekturu gde je modul za šifrovanje realizovan u C programskom jeziku i integrisan sa bazom podataka.



Sl 2. Šifrovanje na aplikativnom serveru

Navedene su prednosti i mane ovog koncepta zaštite:

Prednosti :
- podaci u komunikaciji su zaštićeni
- ključ se ne nalazi sa šifrovanim podacima
- dostupnost izvornog koda

Mane :
- realzacija AES algoritma u interpreterskom jeziku (što se odražava na performanse sistema)
- promena ključa (proces koji zahteva dešifrovanje i ponovno šifrovanje kompletne baze podataka)

Ključni su vremenski resursi ili vreme potrebno za jedan "ciklus" algoritma. Algoritmi predstavljaju skup međusobno povezanih elementarnih struktura, za čije izvršavanje se troši značajno procesorsko vreme. Cilj ovog rada je bio da se proveri koliki je uticaj softverskih realizacija algortama za šifrovanje na opterećenost procesora. Upravo iz ovog razloga, analiziran je uticaj vremenske komponente za izvršavanje pojedinih algoritama. U daljem delu ovog rada su prikazani rezultati dobijeni za AES algoritam.

IV. Eksperiment

Za potrebe eksperimenta korišćena su dva međusobno

umrežena računara, aplikativni server i baza podataka. Detaljnije hardverske karakteristike računaraprikazane su u tabeli 1.

Uloga Intel računara je primena kriptografskog rešenja i snimanje rezultata šifrovanja.

TABELA 1: HARDVERSKE KARAKTERISTIKE RAČUNARA KORIŠĆENIH U EKSPERIMENTU

Intel	
Procesor:	Intel Core(TM)2 1.86GHz 4Mb cache
RAM:	2048MB
Mrežna kartica:	Realtek RTL-8169 Gigabit Ethernet

Na oba računara instaliran je Windows Vista operativni sistem.

Instaliran je program za izvršavanje različitih testova vezanih za šifrovanje. Jedna od glavnih osobina ovog programa je to što je realizovan u vidu klijentske i serverske komponente, tako da omogućava praćenje željenih parametara i na klijentskoj i na serverskoj strani, istovremeno, kao i rezultati testiranja performansi AES algoritma.

Nakon postavljanja eksperimentalnog okruženja pristupilo se komparativnom merenju performansi sa sledećim eksperimentima:

- (M1) standardan upis/ispis nešifrovanih podataka u bazi podataka
- (M2) brzina AES algoritma integrisanog u bazi podataka pri šifrovanju/dešifrovanju
- (M3) brzina AES algoritma integrisanog na aplikativnom serveru pri šifrovanju/dešifrovanju
- (M4) brzina AES algoritma integrisanog u bazi podataka pri šifrovanju/dešifrovanju i upisu/ispisu u bazu podataka
- (M5) brzina AES algoritma integrisanog na aplikativnom serveru pri šifrovanju/dešifrovanju i upisu/ispisu u bazu podataka

Pri svakom od merenja beleženi su sledeći parametri:

- ostvarena brzina pri šifrovanju/dešifrovanju izražena u vremenskim jedinicama
- opterećenje procesora

Za sve algoritme su korišćeni isti ključevi dužine 256 bita. Dobijeni rezultati predstavljeni su u delu V Analiza rezultata.

V. ANALIZA REZULTATA

Rezultati merenja su prikazani na slici 3. i u tabeli 2. Na slici 3. se vidi pad brzine šifrovanja podataka pri upisu šifrata u bazu podataka. Imajući u vidu da je za sva merenja korišćena ista hardverska platforma i isti komunikacioni kanal (iste propusne moći) jasno je da razlog usporenja leži u brzini hard diska računara na kom se nalazi baza podataka.

Sl 3. Dijagram sa ostvarenim brzinama AES algoritma

TABELA 2: REZULTATI MERENJA.

Proces	Vreme (s)	CPU (%)
Standardan upis/ispis nešifrovanih podataka u bazi podataka		
Upis	22.30	6
Ispis	1.23	70
Brzina AES algoritma integrisanog u bazi podataka pri šifrovanju/ dešifrovanju		
Sifrovanje	0.18	20
Desifrovanje	0.18	20
Brzina AES algoritma integrisanog na aplikativnom serveru pri šifrovanju/dešifrovanju		
Sifrovanje	5.35	54
Desifrovanje	5.30	54
Brzina AES algoritma integrisanog u bazi podataka pri šifrovanju/ dešifrovanju i upisu/ispisu u bazu podataka		
Sifrovanje	22.66	7
Desifrovanje	0.21	33
Brzina AES algoritma integrisanog na aplikativnom serveru pri šifrovanju/dešifrovanju i upisu/ispisu u bazu podataka		
Sifrovanje	27.34	54
Desifrovanje	5.50	70

U tabeli 2. su brojčano date vrednosti prikazane na dijagramu slike 2. (u sekundama). S obzirom na to da je opterećenje Intel procesora pri korišćenju svakog od algoritama bilo na maksimumu (99,9% + 0,01% za sistemske procese) to jasno ukazuje na usko grlo ovih mehanizama. Takođe, moguće je predvideti 2-3 puta veću brzinu korišćenjem snažnijh procesora i hard diskova veće brzine upisa.

Napomena: U eksperimentu su kao uzorak korišćeni stringovi dužine 300 karaktera i svako merenje je imalo 1000 ciklusa da bi se što bolje procenilo opterećenje šifrovanjem velike količine podataka u što kraćem vremenskom periodu.

VI. ZAKLJUČAK

U radu je razmatrana kriptografska zaštita podataka direktno nad bazom (u okviru izabranog DBMS-a) i na

strani aplikativnog servera. Prikazani su komparativni eksperimentalni rezultati. I jedan i drugi pristup imaju svojih prednosti i mana. Kod lokalne zaštite interveniše se na samoj bazi podataka uključivanjem modula za šifrovanje i dešifrovanje. Podaci u bazi su zapisani u šifrovanom obliku i obezbeđeni su od eventualne krađe hard diska ili neovlašćenog pristupa u bazi. Kod rešenja na aplikativnoj strani u okviru izabranog programskog jezika postiže se fleksibilnost sa mogućnošću selektivnog šifrovanja i dešifrovanja. Mana je što mnogi savremeni jezici (kao što je php) nisu optimizovani za efikasno izvršavanje kompleksnih operacija nad binarnim podacima što može da utiče na performanse ukupnog sistema.

Rezultati ovog rada pokazuju da performanse današnjih PC računara nisu dovoljne za potpuno iskorišćenje mogućnosti kriptografskih mehanizama za šifrovanje na opisanim nivoima. Dobijeni eksperimentalni rezultati ukazuju na potrebu za dodatnim hardverom za šifrovanje umesto softverskog rešenja. Time bi se omogućilo zadržavanje performansi rada sa konkretnom bazom podataka, uz kvalitetno šifrovanje na dodatnom hardverskom modulu. Eventualni pad performansi bi se odnosio samo komunikaciju sa dodatnim hardverom.

Potreba za izučavanjem i razvijanjem metoda kriptografije je sve veća od kako se primenjuju računari, računarske mreže, a sa njima i baze podataka kao jedini vid arhiviranja i kontrolisanja podataka među ljudima. Značaj ove oblasti ima više dimenzija kao što su politička, vojna, ekonomska, socijalna, etička itd.

Ovim radom nije obuhvaćen problem razmene i upravljanja ključevima, što će biti predmet daljih istraživanja.

LITERATURA

[1] William Stallings, "Cryptography and Network Security", Fourth Edition, Prentice Hall, 2006.

[2] A Menezes at all., “Handbook of Applied Cryptography”, CRC Press, 1996.

[3] McDonald D., C. Metz, B. Phan, "PF_KEY Key Management API, Version 2" IETF RFC 2367, 1998.

[4] A. Lee, NIST Special Publication 800-21, Guideline for Implementing Cryptography in the Federal Government, National Institute of Standards and Technology, November 1999.

[5] J. Daemen and V. Rijmen, AES Proposal: Rijndael, AES Algorithm Submission, September *3, 1999.*

[6] Eli Biham and Adi Shamir, Differential Cryptanalysis of the Data Encryption Standard, Springer-Verlag, 1992.

[7] M. Milosavljević, G. Grubor, Osnovi bezbednosti i zaštite informacionih sistema

ABSTRACT

In this paper influence of cryptographic algorithm on CPU load of network computers during data encryption in database. Analyze is focused on changing of time component of the cryptographic algorithm on Windows operating system during implementation of cryptographic module in purpose of data security on local and server level. Results for AES and are given as tables and diagrams. Results have shown that software and hardware obstruct cryptographic algorithms.

TITLE OF THE PAPER IN ENGLISH

Saša Adamović, Mladen Veinović