Univerzitet Crne Gore

Prirodno-matematički fakultet

Podgorica

Vladimir Peković

Kodovi za autentifikaciju poruka (MACs)

Specijalistički rad

Podgorica 2013.

Univerzitet Crne Gore

Prirodno-matematički fakultet

Podgorica

Kodovi za autentifikaciju poruka (MACs)

Specijalistički rad

Kriptografija
dr Vladimir Božović

Vladimir Peković

Matematika i računarske nauke

Podgorica, April 2013.

Spisak slika

Slika 1. Princip rada MAC-a
Slika 2. MD5 sažetak poruke
Slika 3. Napad protiv tajnog prefiksa MAC-a
Slika 4. HMAC konstrukcija
Slika 5. Princip enkripcije b bita sa niz i blok šiframa
Slika 6. Enkripcija i dekripcija u CBC režimu
Slika 7. MAC zasnovan na blok šiframa u CBC režimu
Slika 8. Enkripcija i dekripcija u ECB režimu

Sadržaj

1 Uvod **9**

2 Principi MAC-a **11**

3 MAC iz heš funkcija (HMAC) **13**
3.1 Uvod u HMAC 13

3.2 Napad protiv tajnog prefiksa MAC-a 15
3.3 Napad protiv tajnog sufiksa MAC-a 16

3.4 HMAC 16

3.5 HMAC algoritmi 18

4 MAC iz blok kripto-sistema **20**
4.1 Blok kripto-sistemi 20

4.2 CBC 21

4.3 CBC-MAC 22

5 Galois Counter Message Authentication Code (GMAC) **24**
5.1 GCM 24

5.2 GMAC 25

6 Primjer **27**

7 Zaključak **30**

1 Uvod

Ljudi se od davnina bave problematikom slanja poruka nesigurnim kanalima i oduvijek žele sigurno komunicirati. Razvojem tehnologije, a posebno interneta, potreba za sigurnošću prenosa podataka raste. Kako znati da poruka nije putem izmijenjena? Kako znati da je sagovornik onaj za koga se predstavlja? Na papiru je potpis dovoljan dokaz vjerodostojnosti, ali kako provjeriti indentitet pošiljaoca poruke poslate preko interneta?

Integritet podataka garantuje da nije došlo do izmjene sadržaja poruke na njenom putu od izvora do odredišta. Pružanje načina provjere integriteta informacije koja se prenosi ili smješta na nepouzdanim medijima je primarna potreba u svijetu otvorenog računarstva i komunikacije. Mehanizme koji pružaju takvu provjeru integriteta temeljenu na tajnom ključu uobičajeno zovemo "kodovi za autentifikaciju poruke" (MAC, en. Message Authentication Code). Tipično, MAC-ovi se koriste između dvije strane koje dijele tajni ključ kako bi provjerili valjanost informacije izmijenjene između tih strana.

MAC obezbjeđuje i integritet i autentičnost poruke. Autentifikacija je proces u kome se dokazuju identiteti krajnjih elemenata komunikacije. Važnost provjere autentičnosti pošiljaoca poruke se najbolje može primijetiti u finansijskom kontekstu. Na primjer, ukoliko središnja banka pošalje svojoj filijali nalog (digitalni) za uplatu određene svote novca na neki korisnički račun, tada u tom procesu obje komunikacijske strane moraju biti sigurne da su zatražene transakcije te njihove potvrde autentične.

MAC predstavlja niz bita koji se dodaju na originalnu poruku kako bi se obezbijedila autentifikacija pošiljaoca i integritet poruke. MAC vrijednost se dobija tako što se na poruku i tajni ključ, koji posjeduju obje strane komunikacije, primijeni MAC algoritam. Autentifikacija se postiže korišćenjem tajnog dijeljenog ključa, jer samo pošiljalac i primalac imaju tajni ključ. Na prijemu se ponovo računa MAC vrijednost primljene poruke i poredi se sa vrijednošću koja je poslata uz poruku. Ako se te dvije vrijednosti ne poklapaju, došlo je do neovlašćene promjene sadržaja originalne poruke.

Kodovi za autentifikaciju poruke (MAC), još poznati kao kriptografska kontrolna suma (en. checksum) ili heš funkcija sa ključem, imaju široku primjenu u praksi. Koriste se u finansijskim transakcijama, mobilnim komunikacijama (GSM), autentikaciji internet komunikacija sa SSL/TLS protokolima itd.

U pogledu bezbjednosti i funkcionalnosti, MAC dijeli neke osobine sa digitalnim potpisom, jer on takođe pruža integritet poruke i autentifikaciju poruke. Međutim, za razliku od digitalnih potpisa, MAC je šema sa simetričnim ključem (digitalni potpis poruke se formira korišćenjem tehnike asimetričnih ključeva) i

ne pruža neporicanje. Neporicanje onemogućava da onaj ko je poslao poruku kasnije tvrdi da je nije poslao. Prednost MAC-a je što je mnogo brži od digitalnog potpisa jer je zasnovan ili na blok šiframa ili na heš funkcijama.

Prost primjer MAC-a: Pretpostavimo da banka dobija sledeću poruku od Alise: "Poslati Bobu 1000$". Banka mora biti sigurna da poruka nije izmjenjena, recimo na sledeći nacin: "Poslati Bobu 10$" ili "Poslati Čarliju 1000$". Da bi se napadi ove vrste spriječili koristi se MAC algoritam.

Dvije najpoznatije kategorije MAC-a su: HMAC (Hash-based Message Authentication Code) - koristi se neki od poznatih heš algoritama u realizaciji MAC algoritma (HMAC-MD5, HMAC-SH1), CMAC (Cipher-based Message Authentication Code) – temeljen na algoritmima za kriptovanje blokova.

Rad je organizovan na sljedeći način: poglavlje 2 detaljnije objašnjava šta je MAC funkcija, motivaciju za korišćenje MAC funkcije, kao i njen princip rada. Tu su nevedene i osobine MAC funkcija, kao i spisak sigurnosnih zahtjeva koje one trebaju da ispune. Objašnjeni su pojmovi koji su u vezi sa MAC funkcijom, kao što su integritet, autentičnost, neporicanje.

U poglavlju 3 bavimo se algoritmima koji se koriste u realizaciji MAC algoritma, i ti algoritmi zasnovani su na heš funkcijama, pa se takva realizacija MAC-a naziva HMAC. Heš funkcije moraju da posjeduju određena svojstva da bi se koristile za HMAC, i takve funkcije se nazivaju kriptografske hes funkcije, a u ovom poglavlju navešćemo i osobine idealne kriptogafke heš funkcije. Dalje navodimo neke poznatije algoritme za izračunavanje heš vrijednosti, kao i ciljeve kombinovanja MAC-a sa takvim algoritmima. Navodimo i neke slabosti pojedinih konstrukcija HMAC-a, a zatim i opis algoritma koji se temelji na hešu koji ne pokazuju takve bezbjedonosne slabosti. Na kraju poglavlja navodimo detalje vezane za najpoznatije HMAC algoritme, HMAC-MD5 i SHA-1 algoritam.

Poglavlje 4 opisuje blok šifre i način na koji se MAC realizuje pomoću njih. Predstavljen je način za šifrovanje dugih otvorenih tekstova blok šifrom - CBC.

Poglavlje 5 opisuje GMAC. GMAC je varijanta modaliteta brojača Galois (GCM). GCM je režim šifrovanja koji takođe određuje šifru za autentičnost poruka (MAC). Suprotno GCM režimu, GMAC ne šifruje podatke već samo računa šifru za autentikaciju poruke. Navedena je i UMAC konstrukcija.

U poglavlju 6 je naveden primjer koji predočava nedostatke algoritama koji se ne zasnivaju na MAC-u, tj. vidjećemo da sama enkripcija nije dovoljna da bi se sačuvao integritet podataka.

2 Principi MAC-a

MAC je funkcija koja zavisi od simetričnog ključa k i poruke x. Koristićemo sledeću notaciju:

$$m = \mathrm{MAC}_k(x).$$

Princip rada MAC-a i verifikacija predstavljeni su na Slici 1.

Slika 1. Princip rada MAC-a

Motivacija za korišćenje MAC-a je da Alisa i Bob žele da detektuju bilo koju manipulaciju poruke x prilikom prenosa. Za ovo, Bob izračunava MAC kao funkciju od poruke i dijeljenog tajnog ključa k. On šalje Alisi i poruku x i autentifikovanu oznaku m. Pošto je ovo simetričan proces, Alisa, po prijemu poruke x i autentifikovane oznake m, jednastavno ponavlja korake koje je Bob sprovodio prilikom slanja poruke: ponovo proračunava autentifikovanu oznaku pomoću poruke x i dijeljenog tajnog ključa k i provjerava da li se slaže sa Bobovim m.

Osnovna pretpostavka ovog sistema je da će MAC-ovo računanje donijeti netačan rezultat ako je poruka x promijenjena u toku prenosa. Dakle, integritet

poruke je obezbijeđen. Osim toga, Alisa se sada uvjerila da je Bob bio tvorac poruke s obzirom da samo dvije stranke sa istim tajnim ključem k imaju mogućnost da izračunaju MAC, tj. autentifikacija poruke je obezbijeđena.

MAC algoritmi su familija funkcija f_k parametrizovanih tajnim ključem k, sa slijedećim svojstvima:

1. lakoća izračunavanja - za poznatu funkciju f_k, sa zadatim vrijednostima k i ulaza x, $f_k(x)$ je lako izračunati. Taj rezultat se zove MAC vrijednost ili MAC.

2. kompresija - f_k preslikava ulaz proizvoljne konačne dužine na izlaz $f_k(x)$ određene dužine n.

3. funkcija preslikavanja je sirjektivna (potencijalno mnoge poruke imaju isti MAC) - nalaženje takvih poruka sa identičnim MAC-om mora da bude veoma teško.

U praksi, poruka x je često mnogo duža od odgovarajućeg MAC-a. Dakle, izlaz MAC računanja je autentifikovana oznaka fiksne dužine koja je nezavisna od dužine ulaza. Sve važne osobine MAC-a su:

- **Kriptografski kontrolni zbi**r: MAC generiše kriptografski sigurnu autentifikovanu oznaku za datu poruku.
- **Simetričnost**: MAC se zasniva na tajnom simetričnom ključu. Stranke moraju da dijele tajni ključ.
- **Proizvoljna veličina poruke**: MAC prihvata poruke proizvoljne dužine.
- **Fiksna dužina izlaza**: MAC generiše fiksne veličine autentifikacionih oznaka.
- **Integritet poruke**: MAC obezbjeđuje integritet poruke, bilo kakve manipulacije poruke tokom prenosa će biti detektovane od strane primaoca.
- **Autentičnost poruke**: primaoc je siguran u porijeklo poruke.
- **Nema neporicanja**: kako je MAC zasnovan na simetričnim principima, oni ne daju neporicanje.

Poslednju tačku je važno imati na umu. Pošto dvije strane koje komuniciraju dijele ključ, ne postoji mogućnost da se dokaže od strane trećeg neutralnog lica, recimo sudije, da li poruka i njen MAC potiče od Alise ili Boba. Dakle MAC ne nudi zaštitu u slučajevima kada ili Alisa ili Bob nijesu iskreni. Simetrični tajni ključ nije vezan za određenu osobu, već za dvije strane, a samim tim sudija ne može napraviti razliku između Alise i Boba u slučaju spora.

3 MAC iz heš funkcija (HMAC)

3.1 Uvod u HMAC

Ukoliko su poruke dugačke, korišćenje kriptovanja sa javnim ključem za potpisivanje cijele poruke je veoma nepraktično. Nepraktičnost se ogleda u velikim dužinama poruka, što iziskuje dosta resursa i troši mnogo vremena za kriptovanje. Kao logično rešenje ovog problema javlja se mogućnost, potpisivanja samo sadržaja umesto potpisivanja cijele poruke. Osoba koja šalje poruku kreira skraćenu verziju poruke tj. njen sadržaj. Tako formiran sadržaj potpisuje i šalje komunikacionim kanalom. Osoba koja primi tako skraćenu poruku provjerava njen potpis. Svaka promjena izvorne poruke izaziva promjenu u sadržaju, što se odražava na promjenu potpisa, čime se minimizuje mogućnost zloupotrebe. Za kreiranje sadržaja poruke se koristi heš funkcija za sažimanje.

Heš funkcije imaju široku primjenu, ali za primjenu u kriptografiji moraju imati i određena svojstva. Primjenom kriptografske heš funkcije [4] na proizvoljan blok podataka dobija se niz bita fiksne dužine koji se naziva heš vrijednost poruke (hash value, message digest, digest). Osnovna karakteristika heš funkcije je da i najmanja promjena u originalnoj poruci dovodi do promjene njene heš vrijednosti. Svojstvo, pri kojem promjena jednog bita ulazne veličine ima veliki uticaj na promjenu izlazne veličine, naziva se efekat lavine (avalanche effect). Nepisano je pravilo da svaka kriptografski sigurna heš funkcija mora imati ovo svojstvo.

Idealna kriptografska heš funkcija treba da posjeduje sledeće karakteristike:

- izračunavanje heš vrijednosti neke poruke je jednostavno,
- nemoguće je (u konačnom broju koraka) pronaći poruku koja ima datu heš vrijednost,
- nemoguće je (u konačnom broju koraka) promijeniti poruku a da ne dođe do promjene i njene heš vrijednosti,
- nemoguće je (u konačnom broju koraka) pronaći dvije različite poruke koje imaju istu heš vrijednost. Tada kažemo da je funkcija slobodna od kolizije (collision-free)

Najjednostavniji oblik funkcije za izračunavanje sadržaja poruke je uzastopna upotreba XOR funkcije (Exclusive OR) na nizu bitova koji se dobijaju dijeljenjem izvorne poruke na dijelove jednake dužine. Najpoznatiji algoritmi

za izračunavanje sadržaja poruke : MD5 (Message Digest 5) sa 128-bitnim sadržajem i SHA-1 (Secure Hash Algorithm 1) sa 160-bitnim sadržajem.

Slika 2. MD5 sažetak poruke

Kada se govori o izvornoj poruci sledeća zapažanja ukazuju na njenu sigurnost: Ako sadržaj dolazi od verifikovanog pošiljaoca, isto tako i poruka dolazi u paketu od verifikovanog pošiljaoca, čime je zadovoljena autentičnost poruke. Zbog karakteristike jednoznačnosti sadržaja (kriptografska heš funkcija bi trebala da bude "1-1" preslikavanje), ukoliko je sadržaj prenešen nepromijenjen, zaključuje se da je i poruka prenešena nepromijenjena čime se zadovoljava integritet.

Način za realizaciju MAC-a je da se koristi kriptografska heš funkcija kao što je SHA-1. HMAC (Hash function-based Message Authentication Code) postao je veoma popularan u praksi tokom posljednje decenije. Na primjer, koristi se kod TLS protokola (što je predstavljeno simbolom katanca u Web pretraživaču), u IP protokolu itd.

HMAC se može koristiti u kombinaciji s bilo kojom iterativnom funkcijom za izračunavanje sadržaja poruke. MD5 i SHA-1 su primjeri takvih funkcija. Sa druge strane, HMAC, kao i MAC, koristi tajni ključ za računanje i provjeru MAC-ova. Glavni ciljevi takve konstrukcije su:

- upotreba, bez promjena, dostupnih algoritama za izračunavanje sadržaja poruke. Posebno, algoritmi za izračunavanje sadržaja poruke koji se dobro izvode u softveru i za koje je kod slobodno i širom dostupan.
- sačuvati izvorne performanse algoritma za izračunavanje sadržaja poruke bez uvođenja značajnih degradacija performansi.
- upotreba i rukovanje ključevima na jednostavan način.
- posjedovanje razumljive kriptografske analize otpornosti mehanizma autentifikacije temeljene na razumnim pretpostavkama o heš funkciji.
- dozvoliti laganu zamjenu algoritma za izračunavanje sadržaja poruke u slučaju da su pronađeni ili zahtijevani brži ili sigurniji algoritmi za izračunavanje sadržaja poruke.

Osnovna ideja koja stoji iza svih šifri autentifikiacije poruke koje su zasnovane na hešu je da se ključ hešira zajedno sa porukom. Moguće su dvije konstrukcije. Prva:

$$m = \mathrm{MAC}_k(x) = h(k \parallel x)$$

zove se tajni prefiks MAC-a, a druga:

$$m = \mathrm{MAC}_k(x) = h(x \parallel k)$$

zove se tajni sufiks MAC-a. Simbol "$\parallel$" označava konkatenaciju (spajanje). Pokazaće se da obje metode imaju svoje slabosti.

3.2 Napad protiv tajnog prefiksa MAC-a

Smatramo da je MAC realizovan kao $m = h(k\|x)$. Za napad se pretpostavi da se kriptografski kontrolni zbir m izračunava pomoću heš konstrukcije kao što je prikazano na Slici 3. Ovaj pristup se koristi kod većine današnjih heš funkcija. Poruka x koju Bob hoće da potpiše je niz blokova $x = (x_1, x_2, \ldots, x_n)$, pri čemu dužina bloka odgovara širini ulaza heš funkcije. Bob računa autentifikovanu oznaku kao:

$$m = \mathrm{MAC}_k(x) = h(k\|x_1, x_2, \ldots, x_n).$$

Problem je što MAC za poruku $x = (x_1, x_2, \ldots, x_n, x_{n+1})$, gdje je x_{n+1} proizvoljni dodatni blok, može biti izgrađen iz m bez poznavanja tajnog ključa. Primjetimo da će Alisa primiti poruku $x = (x_1, x_2, \ldots, x_n, x_{n+1})$, kao važeću, iako je Bob autentifikovao $x = (x_1, x_2, \ldots, x_n)$. Poslednji blok x_{n+1} može biti, na primer, dodatak na elektronskom ugovoru, situacija koja može imati ozbiljne posledice.

Slika 3. Napad protiv tajnog prefiksa MAC-a

Napad je moguć jer MAC-u dodatnog bloka poruke je potreban samo prethodni izlaz heša, koji je jednak Bobovom m i x_{n+1} kao ulaz, ali ne i ključ.

3.3 Napad protiv tajnog sufiksa MAC-a

Nakon proučavanja prethodnog napada, čini se da je bezbjedno koristiti druge osnovne metode za konstrukciju, odnosno $m = h(x \parallel k)$. Međutim, drugačija slabost je ovdje prisutna. Pretpostavimo da je Oskar u stanju da napravi koliziju u heš funkciji, tj. može da nađe x i x_0 takve da:

$$\mathrm{h(x)=h}(x_0)$$

Dvije poruke x i x_0 mogu biti, na primer, dvije verzije ugovora koji se razlikuju u nekom ključnom aspektu, npr dogovorene isplate. Ako Bob potpiše x sa šifrom autentifikacije poruke

$$m = h(x \parallel k),$$

m je takođe važeći kontrolni zbir za x_0

$$m = h(x \parallel k) = h(x_0 \parallel k)$$

Razlog za to je opet dat zbog iterativne prirode MAC računanja [1].

3.4 HMAC

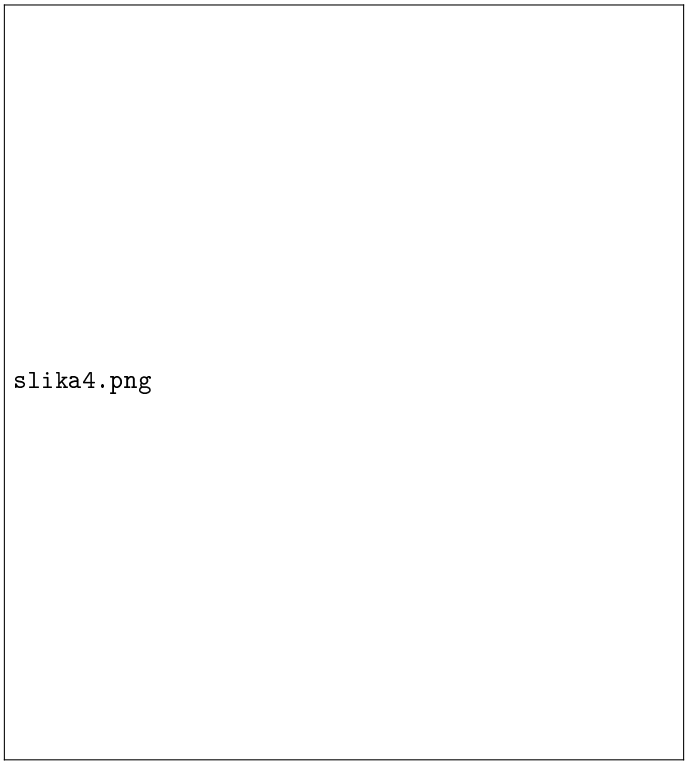
Šifra autentifikacije poruke koja se temelji na hešu koja ne pokazuju bezbjedonosne slabosti koje su gore navedene je HMAC konstrukcija predložena od strane Mihir Bellare, Ran Canetti and Hugo Krawczyk, 1996. godine [2].

MAC računanje počinje sa proširivanjem simetričnog ključa k sa nulama sa lijeve strane tako da je rezultat ključ k^+ dužine b bajtova, gdje je b ulazna širina bloka heš funkcije. Na prošireni ključ i na konstntu ipad=00110110, 00110110, ..., 00110110 je primijenjena operacija XOR. Izlaz XOR-a formira prvi ulazni blok heš funkcije. Sledeći ulazni blokovi su blokovi poruka $(x_1, x_2, \ldots, x_n)$.

Drugo, spoljni heš se izračunava sa ključem i izlazom prvog heša. Ovdje, ključ je ponovo proširen sa nulama, a zatim je opet primijenjena operacija XOR sa spoljnim opad=01011100, 01011100, ..., 01011100. Rezultat XOR operacije formira prvi ulazni blok za spoljni heš. Sledeći ulaz je izlaz unutrašnjeg heša. Nakon što je spoljašnji heš izračunat, izlaz je šifra autentifikacije poruke od x. HMAC konstrukcija može biti izražena kao:

$$\mathrm{HMAC}_k\mathrm{(x)=h}[(k^+ \oplus opad) \parallel h\,[(k^+ \oplus ipad) \parallel x]]\ ,$$

pri čemu h predstavlja heš funkciju, k^+ tajni ključ popunjen nulama do veličine L bitova, x poruku. Simbol $\|$ predstavlja binarnu operaciju spajanja (concatenation) binarnih nizova, dok $\oplus$ predstavlja binarnu operaciju ekskluzivno-ILI, tj gore pominjanu XOR. Ukoliko je L=512, tada su opad i ipad svaki veličine 64 bajta.



Slika 4. HMAC konstrukcija

Ne predstavlja problem da izlaz unutrašnje heš funkcije ne odgovara veličini ulaza spoljašnjeg heša jer heš funkcije imaju korake predobrade da izjednače ulazni string i širinu bloka. U slučaju HMAC-a može da se pokaže da ako napadač Oskar može slomiti HMAC, on takođe može razbiti heš funkciju koja se ovdje koristi. Razbijanje HMAC-a znači da iako Oskar ne zna ključ, on može da konstruiše važeće autentifikovane oznake za poruke. Razbijanje heš funkcije znači da on ili može da nađe koliziju ili da može da izračuna izlaz heš funkcije iako on ne zna početnu vrijednost.

3.5 HMAC algoritmi

HMAC-MD5 je HMAC algoritam temeljen na MD5 heš funkciji.
Opis algoritma: Neka niz znakova 'text' označava podatke na koje će biti primijenjen HMAC-MD5 algoritam, a K neka označava tajni ključ autentifikacije poruke kojeg dijele klijenti. Ključ K može biti proizvoljne dužine do dužine bloka algoritma za izračunavanje sadržaja poruke, tj. do 64 bajta za MD5 (međutim, 16 bajta je minimalna preporučena dužina za ključeve). Definišimo dva stalna i različita niza ipad i opad kao što slijedi ('i' i 'o' mnemonici za unutrašnji i spoljni):

ipad = bajt 0x36 ponovljen 64 puta,
opad = bajt 0x5C ponovljen 64 puta

Da izračunamo HMAC-MD5 od podataka 'text' izvodimo:

MD5(K XOR opad, MD5(K XOR ipad, text))

To jest potrebno je:

1. dodati nule na kraj K tako da se dobije 64-bajtni niz znakova (npr. ako je K dužine 16 bajta, dodaju se 48 nula bajta 0x00)
2. izračunati vrijednost XOR funkcije između niza dobijenog u koraku 1. i ipad -a
3. dodati niz podataka 'text' 64-bitnom nizu dobijenog u koraku 2.
4. primijeniti MD5 algoritam na niz generisan u koraku 3.
5. izračunati vrijednost XOR funkcije između niza dobijenog u koraku 1. i opad -a
6. pridodati rezultat MD5 algoritma iz koraka 4. na 64-bajtni niz dobijen u koraku 5.
7. primijeniti MD5 algoritam na niz generiran u koraku 6. i uzeti rezultat sa izlaza

Ključevi za HMAC-MD5 mogu biti proizvoljne dužine (ključevi duži od 64 bajta se prvo heširaju koristeći MD5 i zatim rezultujućih 16 bajta koristimo kao ključ za HMAC-MD5). Međutim, ključevi kraći od 16 bajta se ne preporučuju, jer bi smanjili sigurnost algoritma. Ključevi duži od 16 bajta su prihvatljivi, ali dodatna dužina ne bi značajnije povećala snagu algoritma (duži ključ je preporučljiv ako se smatra da je slučajnost ključa slaba). Ključevi trebaju biti izabrani slučajno, ili koristeći kriptografski jak generator pseudoslučajnih brojeva sa slučajnom početnom vrijednošću. Preporučuje se periodička i što učestalija promjena ključeva.

Algoritam MD5 funkcije je razvio Ron Rivest 1991. godine. Nakon pet godina otkriveni su mali nedostatci u algoritmu te su kriptografi preporučivali upotrebu drugih heš funkcija. U nekoliko narednih godina su otkriveni dodatni

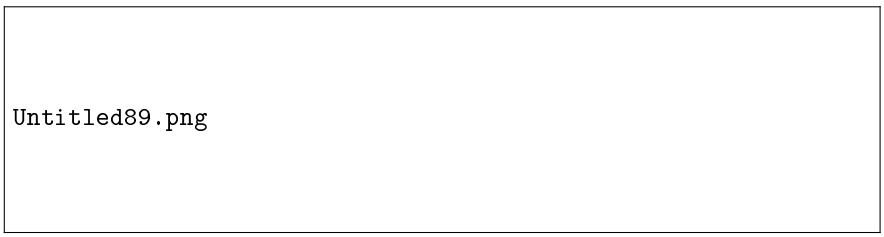
nedostaci te je upotreba ovog algoritma dovedena u pitanje. Tokom 2005. godine grupa istraživača je uspela da formira isti sadržaj primenjujući MD5 na dva različita dokumenta. Zbog pronađenih nedostataka, danas se ovaj algoritam sve ređe koristi za autentikaciju dokumenata i poruka, ali je našao primjenu u provjeri integriteta fajlova, gdje se koristi za izračunavanje kontrolnih suma. Dužina sadržaja koji se formira na osnovu MD5 funkcije je kratka (128 bita).

SHA (Secure Hash Algorithm) se pojavio 1993. godine (u literaturi se često koristi termin SHA-0) od strane američke vladine agencije nacionalnog instituta za standarde i tehnologiju (NIST), kao zvanični standard za sažimanje poruka. Zbog pronađenih propusta u sigurnosti ubrzo je povučen, a nasledio ga je SHA-1 1995. godine . Obe varijante SHA-0 i SHA-1 formiraju sadržaje dužine 160 bita, a maksimalna dužina izvorne poruke može biti 264. Nešto kasnije su se pojavile varijante ovog algoritama koje formiraju i duže sadrzaje. To su SHA-256, SHA-224, SHA-384 i SHA-512 a nazive su dobile po dužini sadržaja. Zbog dužeg sadržaja ove varijante su sigurnije rešenje u odnosu na SHA-0 i SHA-1. Ovi algoritmi čine grupu nazvanu SHA-2. Iako je ova grupa algoritama po pitanju sigurnosti bolje rešenje, danas je zbog jednostavne implementacije i brzine najviše u upotrebi SHA-1. U osnovi SHA-1 je baziran na idejama MD4 i MD5 algoritama. SHA-1 našao je primjenu u mnogim aplikacijama kao sto su TLS, SSL, PGP, SSH, S/MIME i IPSec.

4 MAC iz blok kripto-sistema

4.1 Blok kripto-sistemi

Simetrična kriptografija je podijeljena na blok šifre i niz šifre (šifre toka), koje je lako razlikovati. Slika 5. opisuje operativne razlike između niz šifri (slika 5.a) i blok šifri (slika 5.b) kada želimo da enkriptujemo b bitova u datom trenutku, gdje je b širina blok šifre.



Slika 5. Princip enkripcije b bita sa niz (a) i blok (b) šiframa

Blok šifre enkriptuju jedan cijeli blok bitova otvorenog teksta u datom trenutku sa istim ključem. Ovo znači da šifrovanje bilo kog bita otvorenog teksta u datom bloku zavisi od svih ostalih bitova u istom bloku. U praksi, većina blok šifri imaju dužinu bloka od 128 bitova (16 bajtova) poput standarda za napredno šifrovanje (AES) ili dužinu bloka od 64 bita (8 bajtova) poput standarda za šifrovanje podataka (DES) ili trostrukog DES (3DES) algoritma. U praksi, a naročito prilikom šifrovanja računarske komunikacije na Internetu, blok šifre se koriste češće od šifri toka. Moderne blok šifre poput AES veoma su efikasne u softverima. Štaviše, i za hardver postoje veoma efikasne blok šifre poput PRESENT-a.

Blok šifra je više od algoritma za šifrovanje. To može da bude višestruko upotrebljiv blok za građenje sa kojim raznoliki set kriptografskih mehanizama može da bude realizovan. Na primjer, možemo da ih koristimo za građenje različitih tipova šema za šifrovanje koje su zasnovane na blokovima, a možemo i da koristimo blok šifre za izvođenje šifri toka. Različiti načini šifrovanja se nazivaju režimi rada. Blok šifre se mogu koristiti za konstruisanje kodova za autentifikaciju poruke (MAC).

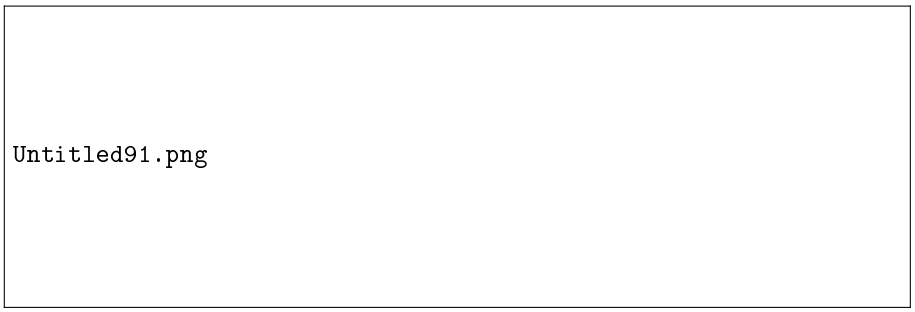
Postoji više načina za šifrovanje dugih otvorenih tekstova sa blok šifrom. Predstavićemo režim ulančavanja šifrarskih blokova (CBC). CBC šifruje podatke i time obezbjeđuje povjerljivost poruke koju je Alisa poslala Bobu. U praksi, ne samo da želimo da podatke učinimo povjerljivim, već Bob želi i da sazna da li

poruka zaista dolazi od Alise. To je u stvari autentifikacija i modalitet brojača Galois (GCM), koji ćemo takođe predstaviti, jeste režim rada koji omogućava da primalac (Bob) utvrdi da li je poruka zaista poslata od strane osobe koja sa njim dijeli ključ (Alisa).

4.2 CBC

Postoje dvije glavne ideje iza režima ulančavanja šifrarskih blokova (CBC). Prvo, šifrovanje svih blokova koji su 'vezani zajedno' tako da šifrat y_i zavisi ne samo od bloka x_i već i od svih prethodnih blokova otvorenog teksta. Drugo, šifrovanje je randomizirano vektorom inicijalizacije (IV). Evo i detalja CBC režima.

Šifrovani tekst y_i, koji je rezultat šifrovanja bloka otvorenog teksta x_i, vraćen je do inputa šifre i na njemu je primjenjena XOR operacija sa sledećim blokom otvorenog teksta x_{i+1}. Ova XOR vrijednost je zatim šifrovana, propušta sledeći šifrat y_{i+1}, koji zatim može da se koristi za šifrovanje x_{i+2}, itd. Ovaj proces je prikazan na lijevoj strani slike 6. Za prvi blok otvorenog teksta x_1 ne postoji prethodni šifrat. Zbog toga je IV dodat prvom otvorenom tekstu, koji nam takođe omogućava da učinimo svaku CBC enkripciju nedeterminističkom. Obratite pažnju da prvi šifrat y_1 zavisi od otvorenog teksta x_1 (i IV). Drugi šifrat zavisi od IV, x_1 i x_2. Treći šifrat y_3 zavisi od IV i x_1, x_2, x_3, itd. Poslednji šifrat je funkcija svih blokova otvorenog teksta i IV.



Slika 6. Enkripcija i dekripcija u CBC režimu

Kada se dešifruje blok šifrata y_1 u CBC režimu, moramo da promijenimo redosled dvije operacije koje smo izvršili na strani šifrovanja. Prvo, moramo da preokrenemo enkripciju blok šifre primjenjujući funkciju dešifrovanja $e^{-1}()$. Nakon ovoga moramo da poništimo XOR operaciju tako što ćemo je opet izvršiti na tačan blok šifrovanog teksta. Ovo može biti izraženo za opšte blokove y_1 kao ${e_k}^{-1}(y_i) = x_i \oplus y_{i-1}$. Desna strana Slike 6. predstavlja ovaj proces. Zatim, ako je prvi blok šifrata y_1 dešifrovan, na rezultat mora da se izvede XOR operacija sa vektorom inicijalizacije IV radi određivanja bloka otvorenog teksta x_1, tj. $x_1 =$ IV $\oplus$ ${e_k}^{-1}(y_i)$. Cijeli proces šifrovanja i dešifrovanja se može opisati kao:

Definicija. Režim ulančavanja šifrarskih blokova (CBC)
Neka e() bude blok šifra bloka veličine b; neka x_1 i y_1 budu nizovi bita dužine b; i neka je IV dužine b.
Šifrovanje (prvi blok): $y_1 = \mathrm{e}_k(x_1 \oplus \mathrm{IV})$
Šifrovanje (opšti blok): $y_i = \mathrm{e}_k(x_i \oplus y_{i-1})$, i ≥ 2
Dešifrovanje (prvi blok): $x_1 = \mathrm{e}_k{}^{-1}(y_1) \oplus \mathrm{IV}$
Dešifrovanje (opšti blok): $x_i = \mathrm{e}_k{}^{-1}(y_i) \oplus y_{i-1}$, i ≥ 2

Sada verifikujemo režim, tj. pokazujemo da dešifrovanje ustvari preokreće šifrovanje. Za dešifrovanje prvog bloka y_1, dobijamo:

$$d(y_1) = e_k^{-1}(y_1) \oplus IV = e_k^{-1}(e_k(x_1 \oplus IV)) \oplus IV = (x_1 \oplus IV) \oplus IV = x_1$$

Za dešifrovanje svih sledećih blokova y_i, i ≥ 2, dobijamo:

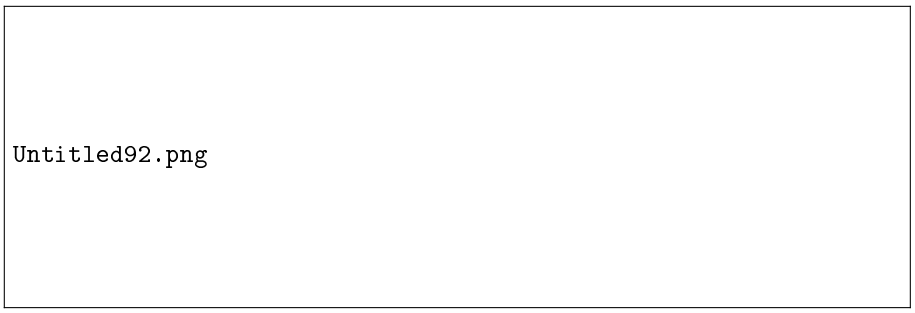
$$d(y_i) = e_k^{-1}(y_i) \oplus y_{i-1} = e_k^{-1}(e_k(x_i \oplus y_{i-1})) \oplus IV = (x_i \oplus y_{i-1}) \oplus y_{i-1} = x_i$$

Ako izaberemo novi IV svakog puta kada šifrujemo, CBC režim postaje šema vjerovatnoće enkripcije. Ako šifrujemo niz blokova x_1 ,..., x_t jednom sa prvim IV i drugi put sa različitim IV, dvije sekvence šifrata koje dobijamo kao rezultat izgledaju potpuno nepovezano za napadača. Obratite pažnju da ne moramo da čuvamo IV u tajnosti. Ali, u većini slučajeva želimo da IV bude za trenutnu upotrebu, tj. broj koji se upotrebljava samo jednom. Postoji mnogo različitih načina za dobijanje i usaglašavanje vrijednosti inicijalizacije. U najprostijem slučaju, nasumično izabran broj je transmitovan između dvije strane koje komuniciraju, prije šifrovane sesije. Alternativno, to je brojna vrijednost koja je poznata Alisi i Bobu, i povećana je svaki put kada nova sesija počinje (koja zahtijeva da brojna vrijednost bude sačuvana između sesija). Može se izvesti iz vrijednosti poput Alisinog i Bobovog identifikacionog broja, npr. njihove IP adrese, zajedno sa trenutnim vremenom. Postoje takođe i neki napredni napadi koji zahtijevaju da IV bude nepredvidiv.

Najvažnija primjena blok šifri u praksi, zajedno sa enkripcijom podataka, su kodovi za autentifikaciju poruke (MAC). Šeme CBC-MAC, OMAC i PMAC su konstruisane sa blok šifrom [6].

4.3 CBC-MAC

Ranije smo vidjeli da se heš funkcije mogu koristiti za realizaciju MAC-a. Alternativna metoda je da se izgradi MAC iz blok šifri. Na Slici 7. prikazana je kompletna postavka za primjenu MAC-a koja se temelji na blok šifri u CBC režimu. Lijeva strana prikazuje pošiljaoca, desna strana primaoca.



Slika 7. MAC zasnovan na blok šiframa u CBC režmu

Generisanje MAC-a
Za generisanje MAC-a, moramo podijeliti poruku x na blokove x_i, i = 1,..., n. Sa tajnim ključem k i početnom vrijednošću IV, možemo izračunati prvu iteraciju MAC algoritma na sledeći način:

$$y_1 = e_k(x_1 \oplus IV)$$

gdje IV može biti javna ali nasumično odabrana vrijednost. Za naredne blokove poruke koristimo XOR bloka x_i i prethodni izlaz y_{i-1} kao ulaz za algoritam enkripcije:

$$y_i = e_k(x_i \oplus y_{i-1})$$

Konačno, MAC poruke x = $x_1x_2x_3 \ldots x_n$ je izlaz y_n poslednje serije:

$$m = \mathrm{MAC}_k(x) = y_n$$

Vrijednosti y_1, y_2, y_3, ..., y_{n-1} se ne šalju. Oni su samo intervalne vrijednosti koji se koriste za izračunavanje konačne MAC vrijednosti m=y_n.

Verifikacija MAC-a
Kao i sa svakim MAC-om, verifikacija obuhvata ponavljanje operacija koje su se koristile za generisanje MAC-a. Tako dobijeni m' moramo da uporedimo sa primljenom MAC vrijednošću m. U slučaju da je m'=m, poruka je verifikovana kao tačna. U slučaju m'≠m, poruka i/ili MAC vrijednost m su promijenjene tokom prenosa. Napominjemo da se MAC verifikacija razlikuje od CBC dekripcije, koja je zapravo obrnuta operacija od operacije enkripcije. Dužina izlaza MAC-a je određena veličinom bloka šifre koja se koristi. Istorijski gledano, DES je u širokoj upotrebi (na primer, za bankarske aplikacije). Od nedavno, AES se češće koristi, on daje MAC dužine 128 bita.

5 Galois Counter Message Authentication Code (GMAC)

5.1 GCM

GCM je režim šifrovanja koji takođe određuje šifru za autentičnost poruka (MAC). Dvije funkcije GCM-a nazivaju se autentična enkripcija i autentična dekripcija. Svaka od ovih funkcija je relativno efikasna i paralelna; samim tim, visokopropusne implementacije su moguće i u hardveru i u softveru. GCM ima još nekoliko korisnih karakteristika, među kojima su sledeće: GCM funkcije su 'online' u smislu da dužine tajnih podataka i dodatnih, javnih podataka, nijesu zahtijevane unaprijed; umjesto toga, dužine se mogu izračunati u toku pristizanja i procesuiranja podataka. GCM funkcije zahtijevaju jedino proslijeđivanje osnovne blok šifre (tj. inverzni pravac nije potreban). Autentičnost zaštićenih podataka može da bude verifikovana nezavisno od vraćanja tajnih podataka iz njihove šifrovane forme. Ako je jedinstveni inicijalizovani niz podataka predvidljiv, a dužina povjerljivih podataka poznata, onda primjenjivanje blok šifri u okviru GCM mehanizama za enkripciju može da bude unaprijed izračunato. Ako su neki ili svi dodatni, javni podaci nepromjenjivi, onda odgovarajući elementi GCM mehanizama za autentifikaciju mogu da budu unaprijed izračunati.

GCM štiti tajnost otvorenog teksta x koristeći šifrovanje u modalitetu brojača. Dalje, GCM štiti ne samo autentičnost otvorenog teksta x već i autentičnost niza podataka AAD tj. dodatnih autentičnih podataka. Ovi autentični podaci su, nasuprot otvorenom tekstu, ostavljeni u početnom stanju u ovom režimu rada. U praksi, niz podataka AAD može da uključi adrese i parametre u mrežnom protokolu.

GCM se sastoji iz osnovne blok šifre i Galois multiplikatora polja sa kojim su izvedene dvije GCM funkcije: autentična enkripcija i autentična dekripcija. Šifra treba da ima veličinu bloka od 128 bita poput AES-a. Na strani pošiljaoca, GCM šifruje podatke sa modalitetom brojača (CTR) nakon čega slijedi računanje MAC vrijednosti. Za šifrovanje, prvo se inicijalni brojač izvede iz IV i serijskog broja. Zatim inicijalna vrijednost je povećana, i ova vrijednost je šifrovana i izvedena je XOR operacija sa prvim blokom otvorenog teksta. Za sledeće otvorene tekstove, brojač je uvećan a zatim i šifrovan. Obratite pažnju da je osnovna blok šifra korištena jedino u režimu šifrovanja. GCM omogućava da se unaprijed odredi funkcija blok šifre ukoliko je vektor inicijalizacije poznat unaprijed.

Za autentifikaciju, GCM izvodi lančanu Galois multiplikaciju polja. Za svaki

otvoreni tekst x_i posredni parametar autentifikacije g_i je izveden. g_i se dobija kao XOR zbir datog šifrata y_1 i g_1, i pomnožen je konstantom H. Vrijednost H je heš potključ koji se dobija šifrovanjem svih nula inputa sa blok šifrom. Sve multiplikacije su u 128-bitnom Galois polju (2^{128}) sa ireducibilnim polinomom P (x) = x^{128}+ x^7 + x^2 + x +1. Pošto je samo jedna multiplikacija potrebna po enkripciji blok šifre, GCM režim dodaje veoma malo računskih dodataka šifrovanju.

Definicija. Osnovni Galois modalitet brojača (GCM)
Neka e() bude blok šifra bloka veličine 128 bita; neka x bude otvoreni tekst koji se sastoji iz blokova x_1, x_2, x_3, ... , x_n; i neka AAD budu dodatni autentifikovani podaci.

1. **Šifrovanje**
   a. Izvesti vrijednost CTR_0 iz IV i izračunati $CTR_1 = CTR_0 + 1$.
   b. Izračunati šifrat: $y_i = e_k(CTR_i) \oplus x_i$, i >1.
2. **Autentifikacija**
   a. Generisati potključ autentifikacije $H = e_k(0)$
   b. Izračunati g_0 = ADD× H (Galois multiplikacija polja)
   c. Izračunati $g_0 = (g_{i-1} \times y_i) \times H$, $1 \leq i \leq n$ (Galois multiplikacija polja)
   d. Konačna oznaka autentifikacije: $T = (g_n \times H) \oplus e_k(CTR_0)$

Primalac paketa $[(y_1, y_2, y_3, \ldots, y_n), T, ADD]$ dešifruje šifrat takođe primjenjivanjem modaliteta brojača. Da bi provjerili autentičnost podataka, primalac takođe određuje oznaku autentifikacije T' koristeći primljeni šifrat i ADD kao input. Upotrebljava potpuno iste korake kao i pošiljalac. Ako se T i T' poduda-raju, primalac je siguran da šifratom (i ADD) nije manipulisano u komunikaciji i da je jedino pošiljalac mogao da napravi poruku.

5.2 GMAC

GMAC je varijanta modaliteta brojača Galois (GCM). GMAC je režim rada za osnovne blok šifre sa simetricnim kljucem. Suprotno GCM režimu, GMAC ne šifruje podatke već samo računa šifru za autentifikaciju poruke. GMAC je lako paralelabilan, što je privlačno za high-speed aplikacije. Upotreba GMAC-a u IPsec Encapsulating Security Payload (ESP) i Authentification Header je veoma znacajna [7] . Navedena literatura opisuje kako treba koristiti AES u GMAC-u da bi se dobila autentifikacija porijekla podataka bez tajnosti u okviru IPsec ESP i AH. GMAC se može efikasno ugraditi u hardver i dostići brzinu od 10Gbit/s i višu.

Druge MAC konstrukcije. Drugi tip šifri za autentifikaciju poruke je zasnovana na univerzalnom heširanju i naziva se UMAC [5]. UMAC je podržan formalnom analizom bezbijednosti, i jedina interna kriptografska komponenta je blok šifra korištena da se dobije pseudoslučajna sekvenca i način da odredemo ključ. Univerzalna heš f-ja je korištena da se dobije kratka heš vrijednost definisane dužine. Na ovaj heš je zatim primjenjena XOR operacija sa pseudoslučajnom sekvencom izvedenom iz ključa. Univerzalna heš funkcija je osmišljena da bude veoma brza u softveru i uglavnom je zasnovana na dodacima 32-bitnim i 64-bitnim brojevima i multiplikacijama 32-bitnim brojevima.

6 Primjer

Predstavićemo primjer u kojem ćemo vidjeti da enkripcija teksta nije dovoljna za očuvanje integriteta podataka.

Enkripcija sa blok šiframa šifruje podatke i na taj način obezbjeđuje tajnost poruke koju je Alisa poslala Bobu. U praksi, ne samo da često želimo da podatke učinimo povjerljivim, već Bob želi da zna i da li je poruku zaista poslala Alisa. Knjiga elektronskih šifri (ECB) je najneposredniji način za šifrovanje poruke. U onome što slijedi, neka $e_k(x_i)$ označava šifrovanje bloka otvorenog teksta x_i sa ključem k upotrebom neke arbitrarne blok šifre. Neka ${e_k}^{-1}(y_i)$ označava dekripciju bloka šifrata y_i sa ključem k. Pretpostavimo da blok šifra šifruje (dešifruje) blokove veličine b bita. Kako je prikazano na Slici 8., u ECB režimu svaki blok je šifrovan zasebno. Blok šifra može, npr., da bude AES ili 3DES.

Slika 8. Enkripcija i dekripcija u ECB režimu

Šifrovanje i dešifrovanje u ECB režimu je formalno opisano kao:

Definicija. Knjiga elektronskih šifri (ECB)
Neka e() bude blok šifra veličine bloka b, i neka x_i i y_i budu nizovi bitova dužine b.
Enktipcija: $y_i = e_k(x_i)$, $\quad i \geq 1$
Dekripcija: $x_i = {e_k}^{-1}(y_i) = {e_k}^{-1}(e_k(x_i)), \quad i \geq 1$

Neposredno se verifikuje tačnost ECB režima:

$$e(y_i) = e_k^{-1}(e_k(x_i)) = x_i$$

Glavni problem ECB režima je da šifruje veoma deterministički. Ovo znači da identični blokovi otvorenog teksta za rezultat imaju identične blokove šifrata, dok god ključ ostaje nepromijenjen. ECB režim se može posmatrati kao gigantska knjiga kodova koja preslikava svaki input na određeni output. Naravno, ukoliko se ključ promijeni cijela knjiga kodova se mijenja, ali koliko god je ključ

isti, knjiga je nepromijenjena. Ovo ima nekoliko nepoželjnih posledica. Prvo, napadač prepoznaje kada je ista poruka poslata dva puta jednostavno gledajući šifrat. Izvođenje informacija iz šifrata na ovaj način naziva se analiza protoka (saobraćaja). Na primjer, ukoliko postoji definisani heder koji uvijek prethodi poruci, on za rezultat ima isti šifrat. Iz ovoga napadač može, na primjer, da otkrije kada je poslata nova poruka. Drugo, blokovi otvorenog teksta su šifrovani nezavisno od prethodnih blokova. Ako napadač promijeni redosled blokova šifrata, ishod toga može da bude validan otvoreni tekst, i izmjene možda i ne budu primjećene. Predstavljamo napad koji istražuje navedene slabosti ECB režima.

ECB režim je podložan supstitucionim napadima, jer ako je jednom poznat određeni par $x_i \rightarrow y_i$ (otvoreni tekst, blok šifrat), nizom blokova šifrata se može lako manipulisati. Predstavljamo kako bi supstitucioni napad mogao da funkcioniše u stvarnom svijetu. Zamislite sledeći primjer elektronskog transfera između banaka.

Primjer: supstitucioni napad na elektronski transfer između banaka
Pretpostavimo koji je protokol za elektronski transfer između banaka (slika ispod). Postoje pet polja koja određuju transfer: ID i broj računa banke koja vrši slanje, ID i broj računa banke koja prima novac, i količinu novca. Pretpostavimo sada (a ovo je glavno pojednostavljivanje) da svako od ovih polja je potpuno iste dimenzije širine blok šifre, npr. 16 bajtova u slučaju AES. Dalje, ključ između dvije banke se ne mijenja veoma često. U skladu sa prirodom ECB, napadač može da iskoristi deterministička prirodu ovog režima rada jednostavnom zamjenom blokova. Detalji napada su sledeći:

blok #	1	2	3	4	5
	Slanje Banka A	Slanje Račun #	Primanje Banka B	Primanje Račun #	Količina $

1. Napadač, Oskar, otvara jedan račun u banci A i još jedan u banci B.
2. Oskar prati šifrovanu liniju bankovne komunikacione mreže.
3. Vrši transfer od 1.00$ sa svog računa iz banke A na svoj račun u banci B u više navrata. Posmatra šifrat koji prolazi kroz komunikacionu mrežu. Iako ne može da dešifruje nasumice posmatrane blokove šifrata, može da traži blokove šifrata koji se ponavljaju. Nakon određenog vremena može da prepozna 5 blokova svog transfera. Sada čuva blokove 1, 3 i 4 iz tih transfera. To su šifrovane verzije ID brojeva obje banke kao i šifrovana verzija njegovog računa u banci B.
4. Zna da dvije banke ne mijenjaju ključeve često. Ovo znači da je isti ključ korišten za nekoliko drugih transfera između banke A i banke B. Upoređivanjem blokova 1 i 3 svih sledećih poruka sa onim koji je sačuvao, Oskar prepoznaje sve transfere koji su izvršeni sa nekog računa u banci A na neki račun u banci B. Sada jednostavno zamjenjuje blok 4 – koji

sadrži broj računa na koji se šalje novac – sa blokom 4 koji je prethodno sačuvao. Ovaj blok sadrži Oskarov broj računa u šifrovanoj verziji. Kao posledica, svi transferi sa nekog računa u banci A izvršeni prema nekom računu u banci B su preusmjereni na Oskarov račun B! Obratite pažnju na to da banka B sada ima mogućnosti da otkrije da je blok 4 zamijenjen u toku nekog od transfera.

5. Brzo podiže novac iz banke B i leti u drugu zemlju.

Ono što je zanimljivo u ovome napadu je da funkcioniše u potpunosti bez napada na samu blok šifru. Pa čak i ako bi koristili AES sa 256-bitnim ključem i ako bi smo šifrovali svaki blok, recimo, 1000 puta, ni to ne bi spriječilo napad. Poruke koje su nepoznate Oskaru i dalje su tajne. On jednostavno zamjenjuje djelove šifrata sa nekim drugim šifratima.

Došlo je do narušavanja integriteta poruke. Vidjeli smo da sama enkripcija nije dovoljna za očuvanje integriteta poruke. Ovakvi problemi se rešavaju korišćenjem MAC algoritama.

7 Zaključak

Sve češća upotreba interneta dovodi do potrebe sigurnog i pouzdanog utvrđivanja autentičnosti i integriteta dokumenata. Enkripcija obezbjeđuje tajnost podataka, ali često ne omogućava i integritet podataka.

MAC pruža dvije sigurnosne stavke, integritet poruke i autentifikaciju poruke, koristeći simetričnu tehniku. MAC je funkcija koja zavisi od simetričnog ključa k i poruke x. Motivacija za korišćenje MAC-a ja da Alisa i Bob žele da detektuju bilo koju manipulaciju poruke x prilikom prenosa. Autentifikacija se postiže korišćenjem tajnog dijeljenog ključa, jer samo pošiljalac i primalac imaju tajni ključ. Na prijemu se ponovo računa MAC vrijednost primljene poruke i poredi se sa vrijednošću koja je poslata uz poruku. Ako se te dvije vrijednosti ne poklapaju, došlo je do neovlašćene promjene sadržaja originalne poruke.

MAC predstavlja niz bita koji se dodaju na originalnu poruku. Integritet i autentifikacija poruke su takođe obezbijeđeni kod digitalnog potpisa, ali kod MAC-a su mnogo brže. MAC se široko koristi u protokolima. MAC ne pruža neopovrgavanje. U praksi, MAC je zasnovan ili na blok šiframa ili heš funkcijama. HMAC je popularni MAC koji se koristi u mnogim praktičnim protokolima kao što je TLS.

Literatura

[1] Christof Paar, Jan Pelzl. Understanding Cryptography: A Textbook for Students and Practitioners. Springer-Verlag Berlin Heidelberg 2010.

[2] H. Krawczyk, M. Bellare, R. Canetti. HMAC: Keyed-Hashing for Message Authentication. February 1997.

[3] John R. Black. Message Authentication Codes. California State University at Hayward 1988.

[4] Bart Van Rompay. Analysis and design of cryptographic hash functions, MAC algorithms and block ciphers. Universiteit Leuven June 2004.

[5] T. Krovetz. UMAC: Message Authentication Code using Universal Hashing. CSU Sacramento March 2006.

[6] Morris Dworkin. Recommendation for Block Cipher Modes of Operation: The CMAC Mode for Authentication. Computer Security National Institute of Standards and Technology May 2005.

[7] D. McGrew and J. Viega.: The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH. Corporation for National Research Initiatives, Network Working Group, May 2006.