UNIVERZA V LJUBLJANI
Fakulteta za elektrotehniko

PROTOKOLI ZA IZMENJAVO KLJUČEV

Matjaž Pučko

(Seminarska naloga)

Ljubljana, 8.5.2006

Kazalo

Kazalo slik

Povzetek

Varnost je širok pojem in za njegovo razumevanje je potrebno znanje več področij. V najbolj preprosti obliki govori o tem, kako preprečiti nepooblaščenim osebam dostop ali spreminjanje podatkov, ki so namenjeni drugim osebam. Varnost obravnava skrivanje, avtentikacijo, avtorizacijo, ne-tajenje in zagotavljanje integritete podatkov. Algoritmi za šifriranje podatkov so danes že tako razviti, da se jih praktično da razbiti. Šibki člen varnosti predstavlja izmenjava sporočil po prenosnem mediju. Način kako zagotoviti varno izmenjavo sporočil za vzpostavitev povezave predpisujejo protokoli za izmenjavo varnostnih ključev. Kot najbolj preprost protokol izmenjave ključev si lahko zamislimo fizično izmenjavo sporočil, kar pa ni najbolj praktično. V protokolu izmenjave sporočil lahko nastopi posrednik v obliki distribucijskega centra, kateremu brezpogojno zaupamo. Varnostni ključ je lahko simetričen (skrivni) ali asimetričen (javni).
Protokol za izmenjavo ključa Diffie-Hellman (1976) je prisoten v večini današnjih protokolov za izmenjavo ključev IKE, IKEv2, Sigma, jfk,…). D-H protokol običajno nastopi v prvi fazi protokola in se uporablja za vzpostavitev začasne varne povezave, kateri sledi druga faza z izmenjavo sporočil za generiranje simetričnega ključa. Večina današnje komunikacije poteka po principu klient-strežnik. Protokoli za izmenjavo sporočil to dejstvo izkoriščajo.
Protokol Kerberos V4 temelji na protokolu Needham-Schroeder. Kerberos je bil iznajden na inštitutu M.I.T. za varen dostop delovnih postaj do mrežnih virov. Obstajajo tudi novejše različice tega protokola, vendar se V4 uporablja v industriji. Kerberos poleg klientov predpisuje uporabo dodatnih strežnikov.
SSL je protokol, ki se ga uporablja v medmrežju za varno komunikacijo. Razvit je bil s strani podjetja Netscape Communication Corp leta 1995. Njegov namen je bil omogočiti varno izmenjavo sporočil med bančnimi transakcijami, elektronskemu kupovanju, uporabi kreditnih kartic za plačevanje, itd. Protokol je aktualen še danes in se ga uporablja za širok spekter storitev.
Bluetooth vsebuje svoj protokol za generiranje in izmenjavo ključev. Njegova posebnost je, da je varnost povezave in šifrirnih ključev odvisna od uporabljene PIN kode. Na podlagi dolžine PIN kode temeljijo postopki šifriranja.
SIGMA je bil razvit z namenom izboljšanja varnosti protokola IKE in njene zadnje različice IKEv2. SIGMA vnaša novost v današnje protokole saj za potrebe avtentikacije narekuje, da moramo sporočilo podpisati in izračunati njeno avtentikacijsko kodo (SIGn and MAc-your-own-identity).
JFK protokol je preprost, efektiven in varen. Namenjen je za uporabo v IPsec arhitekturi, kjer lahko nadomesti IKE protokol. JFK protokol vsebuje mehanizme proti DoS napadom.
Protokola IKE (internet key exchange) in IKEv2 sta del protokola IPSec. Slednji je nadgradnja prvega, ki ima več pomanjkljivosti in se smatra da ni varen.
802.11i oziroma WPA2 je nov standard in predstavlja najvišjo zaščito za varno komunikacijo v tehnologiji Wi-Fi. Za avtentikacijo in izmenjavo sporočil uporablja arhitekturo, ki temelji na 802.1X protokolu.

Uvod

Varnost je širok pojem in za njegovo razumevanje je potrebno znanje več področij. V najbolj preprosti obliki govori o tem, kako preprečiti nepooblaščenim osebam dostop ali spreminjanje podatkov, ki so namenjeni drugim osebam. Varnost obravnava skrivanje, avtentikacijo, avtorizacijo, ne-tajenje in zagotavljanje integritete podatkov. Skrivanje podatkov govori o preprečevanju dostopa do informacij nepooblaščenim osebam. Avtentikacija govori o ugotavljanju identitete osebe s katero komuniciramo. Avtorizacija govori o dovoljenju, ki omogoča osebi dostop do informacije. Ne-tajenje govori o elektronskih podpisih. Obravnava temo zaščite proti lažnemu zanikanju z dokazilom o izvoru ali sprejemu informacije.
Varnost v elektronskih komunikacijah se lahko izvaja na več nivojih. Na fizičnem nivoju se lahko zaščitimo pred napadalci tako, da prenosni medij postavimo znotraj cevi, ki je pod pritiskom. Če bo kdo želel dostopiti do žice, bo moral prevrtati cev. Posledično bo v cevi padel pritisk in sprožil se bo alarm. Na podatkovnem nivoju se lahko na point-to-point zvezi podatki na vhodu kriptirajo in na izhodu dekriptirajo. Tak način zaščite je transparenten za višje nivoje. Rešitev ima problem, ko podatek potuje preko usmerjevalnikov, saj se mora podatek za potrebe usmerjanja vsakič dekriptirati. V tem času je podatek ranljiv za napade od znotraj usmerjevalnika. Na mrežnem nivoju se z uporabo požarne pregrade (firewall) lahko določi, kateri promet je dovoljen in kateri prepovedan. Na mrežnem nivoju tudi temelji IP varnost. Transportni nivo omogoča, da je celotna povezava oziroma proces enkriptiran od konca do konca (end-to-end). Avtentikacija in ne-tajenje se vršita na aplikacijskem nivoju.

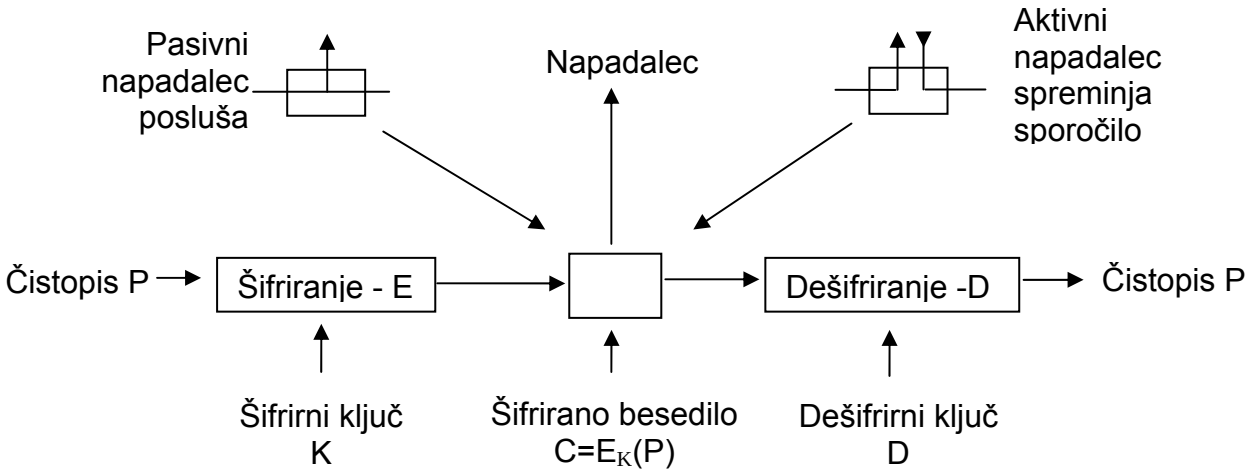
Znano je, da je najslabši del varnosti oziroma njen najšibkejši člen človek. Največ prevar v bankah se zgodi kot posledica slabo načrtovanih mehanizmov varnosti, slabo izobraženih uslužbencev ali pa prevare od znotraj in ne zaradi prisluškovanja (wiretapping) ali dekriptiranja sporočil. Obstajajo primeri, kjer oseba, ki je našla bančno kartica, vstopi v banko in v imenu dobrega poslovanja dobi PIN kodo. Takšne napake v protokolu izničijo vsa prizadevanja za varnost sistema in kriptiranja podatkov. V današnjem času se poslovanje preko interneta eksponento povečuje in ponudniki storitev elektronskega poslovanja (e-commerce) se zavedajo, da je za uporabnike izjemno pomembna zaščita in varnost osebnih podatkov. Tako smo priče nenehnim novitetam na področju varnosti, katerih cilj je vzbuditi prepričanje uporabnika, da je storitev varna.

Kriptografija

Kriptografija je veda o kriptiranju podatkov. V osnovi gre za spreminjanje originalnega sporočil, ki mu pravimo čistopis, v šifrirano obliko (chipertext), ki je za napadalca neberljiva. Obstajata dva načina spreminjanja podatkov in sicer:
- šifra (chiper); transformacija črka na črko (character-for-character) ali bit na bit (bit-for-bit).
- koda (code); transformacija besede za drugo besedo ali simbol.

Koda se danes ne uporablja, saj je za njeno razbitje potrebno manj truda kot za način kodiranja šifra. Najbolj znana uporabe kode je bila v drugi svetovni vojni, kjer so ZDA za prenos informacij uporabljale jezik, ki so ga govorili le Indijanci s plemena Navajo.

V današnjem času se za šifriranje uporablja model prikazan na sliki 1.



Slika 1: Model simetričnega šifriranja

Sporočilo oziroma čistopis, ki ga želimo šifrirati, spremenimo s funkcijo, katera za parameter uporablja šifrirni ključ. Rezultat funkcije imenujemo šifrirano besedilo (chipertext). Šifrirano besedilo se prenese na naslovnika, kateri pozna šifrirni ključ. Prejemnik lahko z uporabo znane funkcije in poznavanjem šifrirnega ključa pretvori šifrirano besedilo nazaj v čistopis. Prenos šifriranega sporočila lahko potega po različnih medijih, ki so ranljivi na prisluškovanje. Vsiljivec lahko podatke prestreže, vendar jih zaradi nepoznavanja šifrirnega ključa le stežka dešifrira. Včasih se zgodi, da lahko vsiljivec aktivno sodeluje na prenosnem mediju. V tem primeru lahko vsiljivec poskuša spremeniti podatke ali sporočilo ponovno pošlje kasneje. Kljub temu, da napadalec ne pozna šifrirnega ključa lahko povzroči škodo (bančno nakazilo na račun, kateri se ponovi n-krat). Proti takim napadom se poskušamo ubraniti s ustreznimi varnostnimi protokoli.
Za razvoj, testiranje in vzpostavitev šifrirne metode je potrebno veliko denarja. Leta 1883 je kriptograf Auguste Kerckhoff izjavil, da morajo biti vsi algoritmi javni, medtem ko so ključi tajni. Kerckhoffov princip (1883) je obveljal in ga uporabljamo še danes.

Šifriranje

Poznamo dva postopka šifriranja in sicer **substitucijsko** ter **transpozicijsko** šifriranje.

Substitucijsko šifriranje pretvori vsako črko ali besedo z drugo in jo zamaskira ter pri tem ohranja vrstni red črk kot v čistopisu. Prvi zapis uporabe takega šifriranje poznamo iz časa Julija Cesarja, ki je uporabljal zamik abecede za k črk (k je šifrirni ključ). Tako šifriranje je zelo preprosto razbiti, zato so se razvili bolj varni sistemi šifriranja. Eden izmed teh je zamenjava črke z drugo črko iz abecede. Kljub temu, da se sistem šifriranja sliši zelo preprost nam preprosta matematična formula prikaže drugačno stanje. Tisti ki bi želel tak sistem razbiti, bi moral poizkusiti 25! = 1.5 X 10^{25} možnih ključev (25 črk v slovenski abecedi). V realnosti se ta številko močno zmanjša, saj besedila ponavadi upoštevajo slovnična pravila, nekatere črke se ponavljalo, itd.

Čistopis	A	B	C	Č	D	E	F	G	H	I	J	K	L	….
Šifrirano	Q	W	E	R	T	Z	U	I	O	P	Š	Č	H	….

Tabela 1: Substitucija

Transpozicijsko šifriranje za razliko od substitucijskega spremeni vrstni red črk in jih ne zamaskira. Eden takih šifrirnih postopkov je prikazan na sliki. Izberemo si ključ – ŠIFRAIRAM, ki ga zapišemo v vrstico. Nato čistopis zapišemo pod njim črko na črko. Ko pridemo do zadnje črke našega ključa preidemo v novo vrstico. Označimo vrstni red ključa po abecedi. Izpišemo stolpce začenši z stolpcem pod črko A, ki je prva v abecedi. Ključ moramo izbrati dovolj velik da zadostuje vsem črkam. S našim ključem lahko šifriramo besedilo dolžine 5X5=25 črk. Ker ima naše besedilo le 22 črk ostanek do 25 zapolnimo za naključnimi črkami. Takemu šifriranju rečemo da je 25 bločno šifriranje.

Š	I	F	R	A
20	10	7	18	6
n	a	L	o	g
a	j	E	t	E
ž	k	A	i	N
z	a	B	a	V
n	a	Š	i	F
r	a	Z	X	Y

Tabela 2: Transpozicija

Čistopis: nalogajetežkainzabavnašifraZXY
Šifrirano: GENVFYLEABŠZAJKAAAOTIAIXNAŽZNR

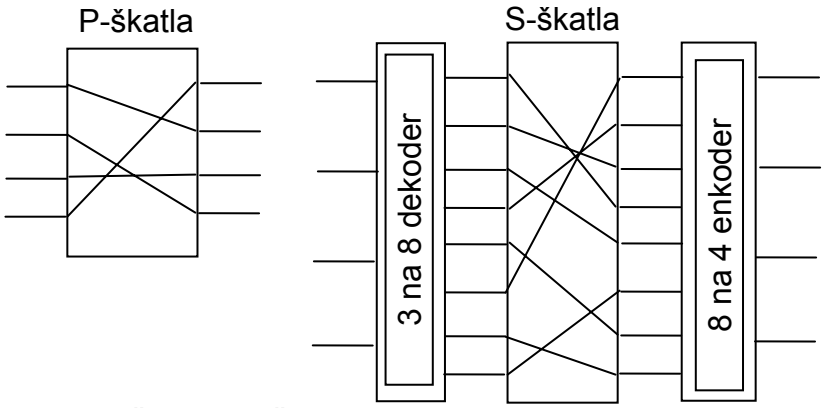
Koda za enkratno uporabo (one-time pad)

Koda za enkratno uporabo je koda, ki se jo ne da razbiti, če se uporablja dovolj dolgo besedilo. Če jo želimo uporabiti potrebujemo dovolj dolg naključen niz, ki predstavlja ključ. Čistopis pretvorimo v binarno obliko, na primer z uporabo ASCII tabele. Nato opravimo XOR operacijo med binarnim zapisom čistopisa in našim naključnim nizom. Ključ mora biti vsaj toliko dolg kot je dolžina čistopisa. Tako šifriran zapis se ne da razbiti. Pri takem šifriranju se moramo držati le enega pravila in sicer, da se ključ uporablja samo enkrat. Problem takega šifriranja je njegova dolžina ključ, katerega morata poznati pošiljatelj in prejemnik.

Simetrično šifriranje

Za simetrično šifriranje velja pravilo:
Dk(Ek(P))=P
D in E sta znani matematični funkciji, medtem ko je k šifrirni ključ. Za simetrično šifriranje je značilno da je šifrirni ključ za šifriranje in dešifriranje enak.

V današnjem času velja pravilo, da se naredi algoritem kar se da kompliciran. Algoritem naj bo tako močan, da tudi v primeru, ko ime napadalec velike količine šifriranega besedila, iz njega ne more izluščiti čistopis ne da bi imel ključ. Osnovni gradniki šifriranja so šifrirni bloki, ki n-bitni vhod pretvorijo v n-bitni izhod. Šifrirni elementi so lahko realizirani v obliki strojne opreme (hitro izvajanje) ali kot programska oprema (fleksibilnost). Osnovni gradniki za strojno šifriranje substitucije in transpozicije so preprosta elektronska vezja. Eden takih je tako imenovana P-škatla, ki opravlja funkcijo transpozicije, in S-škatla, ki opravlja funkcijo substitucije. S kombinacijo P-škatel in S-škatel gradimo šifrirni algoritem, ki ga imenujemo produkt šifriranja. Produkti šifriranja imajo ponavadi k-vhodov in k-izhodov, kjer k je 64 ali 265. Strojna rešitev kodiranja uporablja 18 stopenj P in S-škatel, medtem ko se v programski rešitvi to rešuje z programskimi zankami.



Slika 2: P-škatla in S-škatla

Kodiranja, ki so zgrajena na tak način so DES, Triple DES, AES (Rijndael), Serpent, Twofish, RC6, MARS

Šifrirni algoritem	Avtor	Dolžina ključa [bit]	Komentar
Blowfish	Bruce Schneier	1-448	Počasen in star
DES	IBM	56	Slaba zaščita
IDEA	Massey in Xuejia	128	Dober vendar patentiran
RC4	Ronald Rivest	1-2048	Pozor, določeni ključi s šibki
RC5	Ronald Rivest	128-256	Dober vendar patentiran
Rinjdael	Daemen in Rijmen	128-256	Najboljša izbira
Serpent	Anderson, Biham in Knudsen	128-256	Močna zaščita
Triple DES	IBM	168	Druga najboljša izbira
Twofish	Bruce Schneier	128-256	Močna zaščita, množična uporaba

Tabela 3: Pregled šifrirnih postopkov

Asimetrično šifriranje (algoritem z uporabo javnega ključa)

Uporaba simetričnega šifriranja ima samo eno pomanjkljivost in sicer pošiljatelj in prejemnik morata poznati ključ. Kljub temu, da je šifriranje močno, predstavlja prenos ključa šibki člen varnosti. Tako se je rodila zamisel o algoritmi, ki bi omogočal varen prenos šifrirnih ključev.
Leta 1976 sta Diffie in Hellman na Univerzi v Stanfordu predlagala nov način šifriranja, pri katerem bi bila šifrirni in dešifrirni ključ različna in bi se dešifrirni ključ zelo težko ali praktično nemogoče pridobil iz poznavanja šifrirnega ključa. Njene zahteve so bilo:

1. D(E(P))=P
2. Težko ali praktično nemogoče določiti D iz E
3. E se ne da razbiti z poljubnim napadom s čistopisom

Predlog pod številko 3 omogoča koncept javnega ključa.

Opis izmenjave sporočil z uporabo javnega ključa.
Oseba A, ki želi sprejemati varna sporočila iznajde dva algoritma, ki ustrezata zgoraj omenjenim pogojem. Šifrirni algoritem – E in javni ključ osebe A se objavi na spletu in je javno dostopen. Če želi oseba B uporabljati varno povezavo in poslati sporočilo osebi A, mora sporočilo šifrirati z šifrirnim algoritmom E s parametrom, ki predstavlja javni ključ osebe A. Za potrebe označevanja se uporabi zapis E_A (algoritem E s parametrom A). B šifrirano sporočilo pošlje osebi A. Oseba A sprejme sporočilo in ga s pomočjo privatnega ključ, ki je znan samo osebi A in javno dostopnega dešifrirnega algoritma in spremeni v čistopis.

$D_A(E_A(P))=P$

$D_B(E_A(P))\neq P$
$D_A(E_B(P))\neq P$

$D_B(E_B(P))=P$

Javni in privatni ključ sta par, ki vedno nastopata skupaj. Oseba B, ki želi sprejemati varna sporočila, mora ustvariti svoj par ključev.

RSA
Algoritem, ki zajema vse pogoje, katere omenjata Diffie in Hellman, so leta 1978 ugotovili trije raziskovalci. To so bili Rivest, Shamir in Adleman in po začetnicah njihovih imen se imenuje algoritem RSA. Algoritem je do danes prestal vse poizkuse razbitja in je obveljal kot varen. Edini problem algoritma je dolžina ključa, kateri potrebuje vsaj 1024 bitov za varno komunikacijo. Za primerjavo se spomnimo, da potrebujemo za simetrično šifriranje ključe dolžine 128 bitov.

Algoritem RSA:
1. Izbereš dve praštevili p in q (veliki vsaj 1024 bitov)
2. Izračunaš n = p X q in z = (p-1) X (q-1)
3. Izberi si številko, ki s številom z nima skupnega faktorja, in ga imenuj d
4. Najdi tak e, za katerega velja e x d = 1 mod z

Čistopis P, ki ga želimo šifrirati, se nato razbije na bloke, katerih dolžina mora biti v intervalu $0 \leq P < n$. Čistopis grupiramo v bloke k bitov, kjer je k največje celo število za katerega velja $2^k < n$.

Pripravljeno besedilo šifriramo s funkcijo:
$C = P^e \pmod n$
Za dešifriranje uporabimo funkcijo:
$P = C^d \pmod n$

Tako smo prišli do para (e,n), ki predstavljata javni ključ, in para (d,n), kateri predstavlja privatni ključ.
Varnost algoritma temelji na problemu faktoriziranja velikih števil. Če napadalec iznajde način, kako priti do številke n, lahko potem ugotovi p in q in iz njih z. S poznavanjem števila z in e, ki sta javno objavljen, bi lahko s pomočjo evklidske matematike ugotovil d. Na srečo je problem faktoriziranja velikih števil tako kompleksen, da bi za razbitje 500 bitnega števila trajalo 10^{25} let napada s silo (brute force attack). Če bi se močno povečala procesorska moč današnjih računalnikov, bi z uporabo daljšega ključe vse to izničili.

Dolžina ključa posledično upočasni celoten algoritem. Zato se danes uporabljalo hibridni modeli za varno sporočanje, pri čemer se z uporabo javnega ključa prenaša le ključ, katerega uporabimo za simetrično šifriranje sporočila, ki je veliko hitrejše.

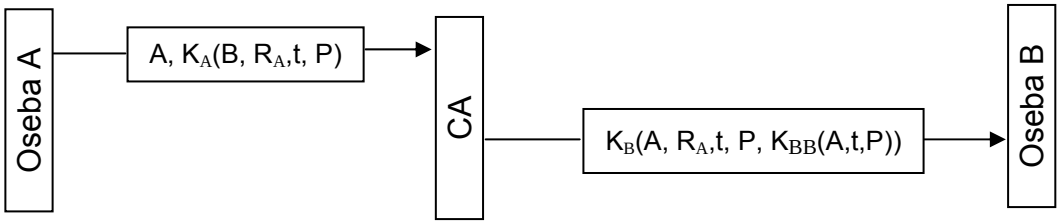
Digitalni podpisi

Poslovanje se počasi seli iz papirne oblike v elektronsko obliko. V papirni obliki obstaja podpis, ki jamči identiteto posameznika. Če želimo podobno jamstvo v elektronski obliki se moramo zateči k ideji o digitalnem podpisu. Digitalni podpis mora izpolnjevati naslednje pogoje:

1. Sprejemnik lahko preveri identiteto pošiljatelja
2. Pošiljatelj ne more kasneje zanikati, da je sporočilo poslal.
3. Sprejemnik ne more falsificirati sporočila od nekoga drugega in ga poslati sebi.

Digitalni podpis z uporabo simetričnega ključa

Če želimo uporabljati simetrični ključ za potrebe digitalnega podpisa, potrebujem centralno avtoriteto (CA), kateri brezpogojno zaupamo. Oseba A si izbere svoj simetrični ključ in ga po varni poti dostavi CA. CA in oseba A sta edina, ki poznata ključ osebe A. Oseba B opravi isti postopek, kot ga je opravila oseba A, le da v tem primeru le oseba B in CA poznata simetrični ključ osebe B.



Slika 3: Digitalni podpis

Oseba A želi poslati podpisano sporočilo osebi B. Sporočilo najprej pošlje centralni avtoriteti CA, ki preveri identiteto pošiljatelja. CA dešifrira sporočilo s ključem osebe A in

ga ponovno šifrira s ključem osebe B. Nato takšno sporočilo pošlje osebi B. Ključna stran take komunikacije je CA, kateremu zaupajo vsi udeleženci.
Za potrebe digitalnega podpisa in varne komunikacije so se vpeljala dodatna sporočila, ki so lahko le redundanca ali pa imajo nek pomen. Tako sporočilo osebe A vsebuje naslednje podatke:

A – identiteta pošiljatelja (izvor)
K_A (B, R_A, t, P) – šifrirano sporočilo s ključem A, ki ga pošiljamo osebi B preko CA
B – identiteta prejemnika (ponor)
R_A – naključno število, ki si ga izbere oseba A
t – časovni žig
P – sporočilo osebe A

Sporočilo osebe A centralna avtoriteta CA dešifrira in ustvari novo sporočilo namenjeno osebi B, ki vsebuje naslednje podatke:

K_B (A, R_A, t, P, K_{CA} (A,t,P)) – šifrirano sporočilo s ključem B, ki ga CA pošlje osebi B v imenu osebe A
A – identiteta pošiljatelja (izvor)
R_A – naključno število, ki si ga izbere oseba A
t – časovni žig
P – sporočilo osebe A
K_{CA} (A,t,P) – podpisano, šifrirano sporočilo osebe A s strani CA

Oseba B dokaže, da je poslano sporočilo od osebe A in sicer:
1. CA je sprejel sporočilo od A, ker le oseba A lahko ustvari sporočilo K_A, šifrirano s ključem A
2. Oseba B lahko pokaže K_{CA} (A,t,P), ki predstavlja podpisano sporočilo od CA in je dokaz da je sporočilo P osebe A poslano osebi B
3. V primeru nesoglasij lahko CA dešifrira K_{CA} (A,t,P) in dokaže trditev pod točko 2.

Protokol ima zaščito proti ponovnemu pošiljanju sporočila v obliki časovnega žiga in naključnega števila R_A, ki si ga izbere oseba A. Časovni žig predstavlja zaščito za primer ponovne uporabe sporočila na dolga obdobja. Uporaba naključnega števila predstavlja zaščito proti napadom, ki ponovno uporabljajo sporočilo v zelo kratkem času.

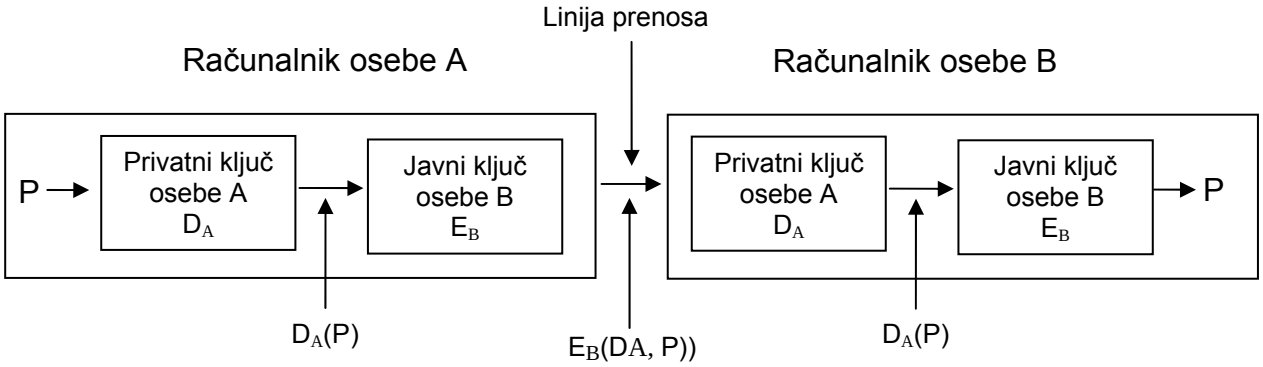
Digitalni podpisi z uporabo javnega ključa

Problem digitalnega podpisa z uporabo simetričnega ključa je v tem, da potrebujemo osrednji organ, kateremu lahko vsi zaupamo. Kandidati za osrednji organ bi lahko bila državna institucija ali banka, katerim vsi državljani ne zaupajo. Poleg tega bi imel osrednji organ vpogled v vsa sporočila.

V primeru uporabe javnega ključa se potreba po osrednjem organ izniči. Algoritmi šifriranja in dešifriranja z uporabo javnih ključev, kot je RSA, imajo lepo lastnosti, da je vrstni red operacije nepomemben:
E(D(P))=P
D(E(P))=P

Če oseba A sporočilo podpiše s svojim privatnim ključem, lahko z uporabe javnega ključa osebe A sporočilo D_A(P) dešifriramo E_A (D_A (P))=P. Ker sta javni in privatni ključ unikaten par, lahko tako preprosto dokažemo izvor sporočila.

Kadar želi oseba A poslati sporočilo osebi B na varen način uporabi postopek $E_B(D_A(P))$: Oseba A svoje sporočilo podpiše (operacija šifriranja s privatnim ključem A), nato pa sporočilo šifrira še s javnim ključem osebe B. Ko oseba B sprejme sporočilo, jo lahko dešifrira s svojim privatnim ključem. Osebi B ostane podpisano sporočilo osebe A, katero dešifrira z uporabo javnega ključa osebe A.

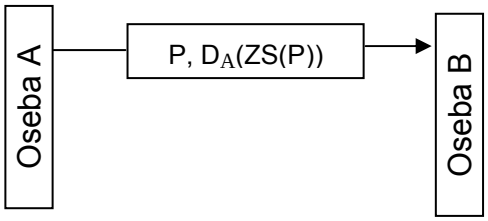


Slika 4: Zaščita in podpis sporočila

Zgoščeno sporočilo (Hash)

Zgoščeno sporočilo je matematična operacija, ki jo uporabljamo za kratek opis ali identifikacijo sporočila. Njegove lastnosti so opisane v štirih točkah:
1. Iz sporočila je enostavno izračunati ZS(P), kjer je ZS matematična operacija zgoščevanja
2. Če imamo ZS(P) iz nje ne moremo ugotoviti P
3. Če imamo P, ne moremo najti P' za katero bi veljalo ZS(P')=ZS(P)
4. Če v sporočilu spremeni le 1 bit, nam ZS operacija vrne popolnoma drug rezultat

ZS nad sporočilom operacija mora vrniti vsaj 128 bitov, če želimo da lastnost pod točko 3 drži. ZS operacije so veliko hitrejše kot operacije šifriranja. Zgoščevanje se uporablja za dokazilo, da izvorno sporočilo ni bilo spremenjeno. Njegovo uporabo lahko najdemo tudi pri digitalnih podpisih.



Slika 5: Zgoščevalna funkcija

Oseba A želi poslati podpisano sporočilo osebi B. Ker je sporočilo javne narave, ga ni treba šifrirati. Oseba A potrebuje dokaz, da je njeno sporočilo pristno in da se ni spreminjalo med pošiljanjem. Tako oseba A pošlje sporočilo kot čistopis, zraven pa pripne še dokazilo o pristnosti. Pristnost predstavlja podpisano sporočilo z privatnim ključem osebe A, ki vsebuje ZS(P).

Popularne ZS funkcije, ki se uporabljajo danes so MD5 in SHA-1

Upravljanje z javnimi ključi

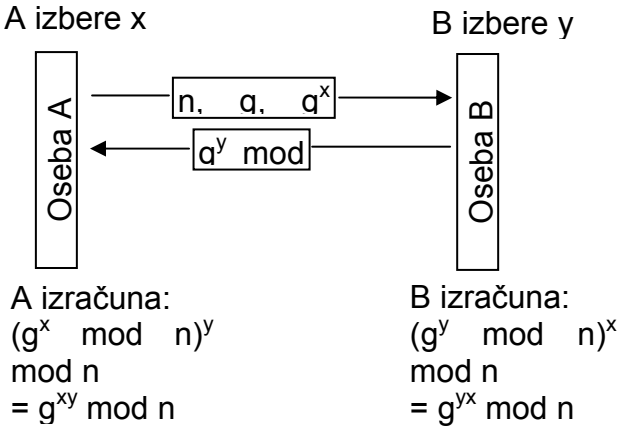
Koncept javnih ključev omogoča varno sporočanje med ljudmi, ki ne uporabljajo skupnega ključa. Omogočajo podpisovanje dokumentov brez prisotnosti tretjega organa. Podpisano zgoščeno sporočilo omogoča preprost način ugotavljanja integritete sporočila.
Glavni problem celotnega postopka je vprašanje, kje objaviti naš javni ključ. Preprost odgovor bi bil, da ga objavimo na spletni strani, katera je vedno dosegljiva. Internet je podvržen številnim napadom, kateri močno omajajo varnost. Napad, ki se lahko pojavi opisuje naslednja situacija:
1. Oseba A želi pridobiti informacijo o javnem ključu osebe B, zato vtipka v brskalnik URL osebe B
2. Brskalnik osebe A pogleda v DNS zapis URL strani, kateri mu pošlje ukaz GET zahteve.
3. Oseba C (napadalec) prestreže ukaz GET zahteve in spremeni sporočilo, tako da se osebi A odpre spletna stran osebe C.
4. Oseba A ne ve, da je dobila napačno stran. Oseba A uporabi javni ključ, ki ga je pridobila z napačne strani.
5. Oseba A pošlje sporočilo, ki je šifrirano s javnim ključem osebe C. Oseba C dešifrira sporočilo s svojim privatnim ključem. Sporočilo lahko prebere ali spremeni.
6. Oseba C spremenjeno sporočilo šifrira s javnim ključem osebe B in mu jo pošlje.
7. Oseba B dobi sporočilo za katero misli da je od osebe A, v resnici pa je spremenjeno od osebe B.

Preprost primer nam prikaže dejansko sliko varnosti. Problem ni algoritem za šifriranje, ki je dovolj močan da prestane napad s silo – brute force attack, ampak **protokol za izmenjavo ključev**.

Protokoli za izmenjavo ključev

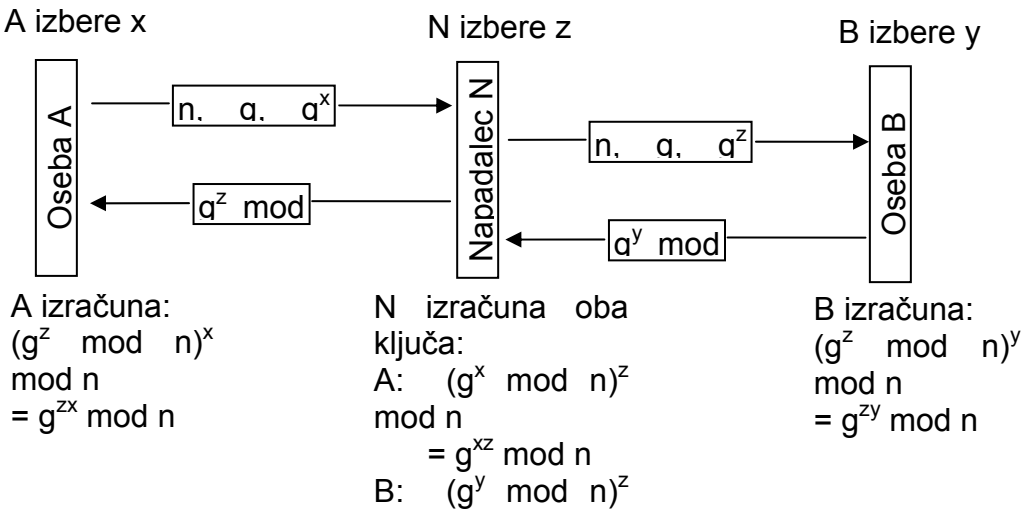
Izmenjava ključev Diffie-Hellman

Protokol Diffie-Hellman key exchange (1976) opisuje varno izmenjavo šifrirnega ključa po nevarnem komunikacijskem kanalu. Tako pridobljeni ključ se lahko kasneje uporabi a simetrično šifriranje sporočil med osebama.
Protokol deluje po naslednjem postopku. Oseba A in B si izbereta dve veliki številki n in g, kjer je n praštevilo, (n-1)/2 je prav tako praštevilo. Podobni pogoji veljajo tudi za število g. Številki ni potrebno skrivati, saj to ni njun namen. Številki sta lahko javno objavljeni, saj njeno poznavanje še ne predstavlja tveganja za varnost protokola. Oseba A si izbere veliko številko, ki naj ima vsaj 512 bit in jo skrije. Prav tako si oseba B izbere številko in jo obdrži zase. Oseba A začne vzpostavljati izmenjavo ključev tako, da osebi B pošlje (n, g, g^x mod n). Oseba B odgovori osebi a s (g^y mod n) sporočilom. Oseba A izvrši operacijo (g^y mod n)^x mod n. Oseba B izvrši operacijo (g^x mod n)^y mod n. Obe operaciji nam vrneta isti rezultat g^{xy} mod n, ki predstavlja skrivni simetrični ključ.
Napadalec, ki želi ugotoviti skrivni ključ, ima na voljo le podatke, ki se prenašajo preko komunikacijskega kanala. Podatki s katerimi razpolaga so n,g, g^x mod n in g^y mod n. Če bi poznal x ali y bi lahko razbil protokol. Ker je napadalcu informacija o x dana samo v obliki g^x mod n iz nje ne more izluščiti x, saj ne obstajajo praktični algoritmi, ki bi to omogočali. Podobno velja za y.



Slika 6: Protokol Diffie-Hellman

Protokol ima napako, saj je lahko podvržen man-in-the-middle napadu. Oseba A ne ve ali se pogovarja z osebo B ali s kakšno tretjo osebo, saj protokol ne vsebuje avtentifikacije. Napadalec lahko to izkoristi in izvede napad. Izbere si številko z, ki ustreza pogojem x in y. Napadalec prestreže sporočilo osebe A osebi B in jo zamenja s svojim sporočilom. Osebi A in B se ne zavedata za napadalca in vzpostavita skrivni ključ z napadalcem. Vsaka komunikacija med osebo A in B tako poteka preko napadalca, ki lahko sporočila prebere in spreminja. Protokol ima še eno slabost. Če poznamo veliko oseb s katerimi potrebujemo varno povezavo, moramo za vsako imeti svoj ključ.



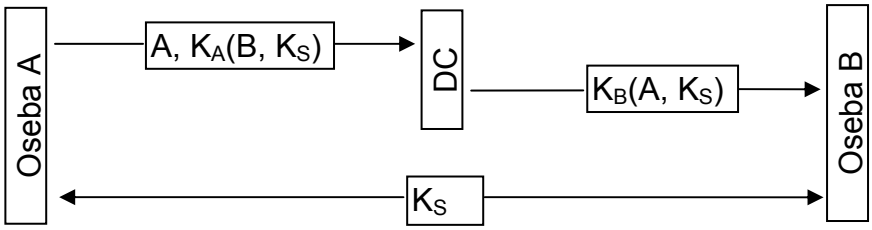
Slika 7: Prikaz napada na protokol D-H

Uporaba distribucijskega centra DC za izmenjavo varnostnih ključev

Uporaba DC za izmenjavo simetričnih ključev

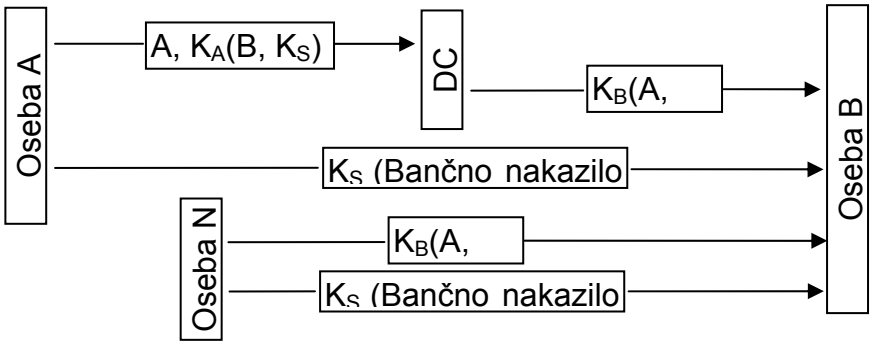
Uporaba distribucijskega centra (DC) poenostavi komuniciranje s veliko osebami. Potrebujemo le en varni ključ, s katerim se pogovarjamo s DC. Avtentikacijo in upravljanje s sejnim ključem opravlja DC.

Oseba A želi varno komunicirati s osebo B. Oseba A si izbere sejni (šifrirni) ključ K_S in pošlje zahtevo DC, da želi komunicirati z osebo B z uporabo ključa K_S. Sporočilo osebo A šifrira s ključem, katerega poznata le A in DC. DC dešifrira sporočilo osebe A in preveri njegovo vsebino. DC nato skonstruira novo sporočilo, ki vsebuje sejni ključ in identiteto osebe A. Sporočilo DC šifrira s skrivnim ključem osebe B in jo pošlje osebi B. Ko oseba B dobi sporočilo, ga dešifrira in iz njega izlušči informacije. Sporočilo vsebuje informacijo o osebi A, katera želi z njim komunicirati in sejni ključ, s katerim naj oseba B direktno komunicira s osebo A. Avtentikacija se v tem primeru vrši na DC, saj DC ve da je sporočilo osebe A lahko prišlo le od osebe A, ker nihče drug ne pozna ključa osebe A.



Slika 8: Uporaba DC za izmenjavo simetričnih ključev

Protokol ima napako, saj je občutljiv na replay-attack. Oseba A želi bančno nakazilo za osebo N in jo pošlje osebi B, ki dela z bančnimi nakazili. Napadalec (N) lahko kopira sporočilo, katero je poslano osebi B in sporočilo, ki temu sledi in predstavlja bančno nakazilo. Napadalec oba sporočila večkrat ponovi in si pridobi neupravičena denarna sredstva.

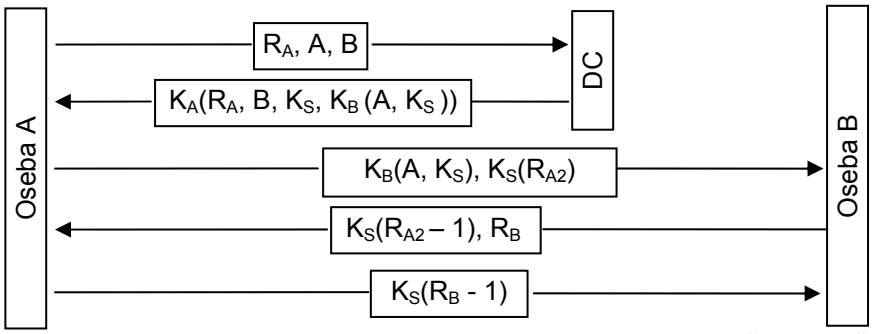


Slika 9: Napad s ponavljanjem sporočila

Proti replay-attack napadi se borimo z časovnimi žigi (timestamps) in naključnimi števili (nonce). Časovni žigi preprečujejo ponovitev sporočila v daljših obdobjih. Na kratke čase se na časovni žig ne moremo zanašati, saj vsi računalniki niso časovno sinhronizirani in tako obstaja interval, ko je protokol ranljiv. To luknjo v varnosti zapolnimo s naključnim številom, ki ga dodamo sporočilu. Če bi prišlo do ponovitve sporočila v kratkem času, bi sprejemnik napad preprosto ugotovil, saj bi se naključno število prehitro ponovilo.

Needham-Scroeder protokol za izmenjavo simetričnih ključev z uporabo DC

Needham-Scroeder protokol (1978) uporablja za izmenjavo ključev DC, obenem pa protokol upošteva časovni žig in naključno število. Oseba A želi komunicirati z osebo B. Svojo namero sporoči DC v sporočilu, ki vsebuje identiteto osebe A, osebe B in veliko naključno število R_A. DC odgovori osebi A s sporočilom šifriranim s ključem A, ki vsebuje R_A, sejni ključ K_S in vstopnico (ticket) $K_B(A, K_S)$, ki predstavlja dovoljenje za komunikacijo z B. Bistvo R_A je v tem, da oseba A ve, da je odgovor na sporočilo sveže in ne gre za kakšno ponovitev starega sporočila. V sporočilu osebe A je vključena informacija o osebi B, tako da napadalec ne more spremeniti sporočilo s svojo identiteto in prepričati DC da šifrira vstopnico s K_N namesto s K_B. Vstopnica je zapisana znotraj šifriranega sporočila K_B da se ga ne more zamenjati s drugo vstopnico.
Oseba A pošlje novo sporočilo osebi B, ki vsebuje vstopnico $K_B(A, K_S)$ in novo naključno število R_{A2}, katero šifriramo s sejnim ključem K_S. Oseba B pošlje nazaj $K_S(R_{A2} -1)$ in R_B in s tem potrdi svojo identiteto osebi A. Če bi B poslal nazaj $K_S(R_{A2})$ s tem ne bi dokazal svoje identitete, saj bi napadalec lahko sporočilo kopiral iz prejšnjega sporočila. Oseba A nato pošlje nazaj sporočilo $K_S(R_B-1)$ in s tem potrdi osebi B, da komunicira s osebo A in da ne gre za ponovitev starega sporočila.

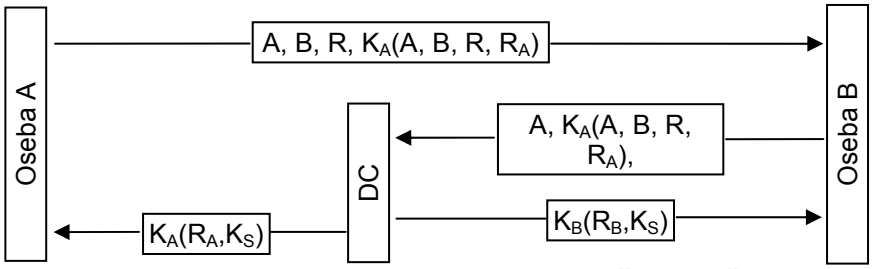


Slika 10: Needham-Scroeder protokol za izmenjavo simetričnih ključev z uporabo DC

Protokol ima napako, in sicer če napadalec dobi v roke star sejni ključ v čistopisu, lahko vzpostavi novo sejo z osebo B tako, da ponovi sporočilo $K_B(A, K_S)$, $K_S(R_{A2})$ osebe A. S tem lahko napadalec N prepriča osebo B da je oseba A.

Otway-Rees protokol izmenjave simetričnih ključev z uporabo DC

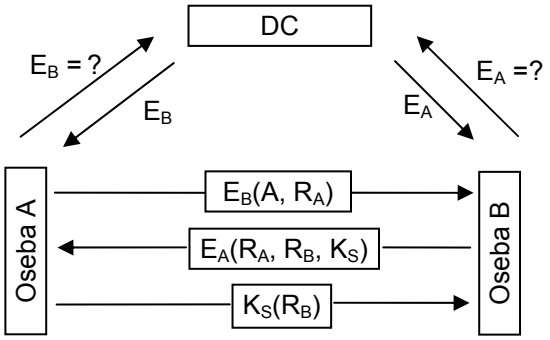
Protokol, kateri je odporen na vse do sedaj opisane napade se imenuje Otway-Rees protokol (1987). V tem protokolu oseba A generira naključni števili R in R_A. Ko oseba B sprejme sporočilo, ustvari novo sporočilo iz šifrirnega dela sporočila osebe A in iz dela, ki predstavlja osebo B. Sporočilo predstavlja identiteto osebe A in B. DC preveri vsebino sporočila in preveri naključna števila, ki se morajo ujemati. V nasprotnem primeru DC ve da je sporočilo spremenjeno od tretje osebe. Če je sporočilo veljavno, DC ustvari sejni ključ in ga pošlje obema udeležencema v sporočilu, ki je šifriran s posameznikovim ključem. Po tem koraku imata oseba A in B sejni ključ, ki ga lahko uporabita za varno komuniciranje.



Slika 11: Otway-Rees protokol izmenjave simetričnih ključev z uporabo DC

Izmenjava javnih ključev z uporabo DC

Postopek izmenjave ključev, kjer nastopajo javni ključi. Avtentikacijo omogoča uporaba distribucijskega centra, kateremu vsi udeleženi zaupajo.
Protokol izmenjave ključev poteka po naslednjem vrstnem redu. Oseba A za komunikacijo z osebo B potrebuje javni ključ osebe B. Oseba A dobi informacijo o ključu od centralnega organa, ki poseduje vse javne ključe. Distribucijski center pošlje osebi A certifikat X.509, ki vsebuje javni ključ osebe B. Oseba A preveri podpis sporočila in preveri njegovo veljavnost. Oseba A nato uporabi javni ključ osebe B in z njim šifrira sporočilo, ki vsebuje identiteto A in naključno število R_A. Oseba B sprejme sporočilo in ga dešifrira s svojim privatnim ključem. V tem trenutku oseba B ne ve od koga je dobila sporočilo. Zato v naslednjem koraku pošlje sporočilo DC, v katerem ga prosi za javni ključ osebe A. DC odgovori na zahtevo in pošlje informacijo o javnem ključu osebe A osebi B. Oseba B uporabi to informacijo in pošlje sporočilo osebi A šifrirano s javnim ključem osebe A. Sporočilo vsebuje naključno število R_A, katero je prejel od osebe A, novo naključno številko R_B, katero si izmisli sam in sejni ključ K_S, katerega predlaga za varno povezavo. Oseba A dešifrira sporočilo s svojim privatnim ključem. Ker sporočilo vsebuje K_A, oseba A ve da je sporočilo dobila od osebe B. Oseba A sprejme predlagani sejni ključ osebe B. Oseba A nato pošlje osebi B sporočilo, katerega šifrira s dogovorjenim sejnim ključem K_S in vsebuje naključno število osebe B. Oseba B sprejme sporočilo šifrirano s sejnim ključem in vsebuje njegovo R_B število. S tem sporočilom se postopek dogovora in identificiranja konča, nadaljnje komuniciranje poteka s sejnim ključem K_S.



Slika 12: Izmenjava javnih ključev z uporabo DC

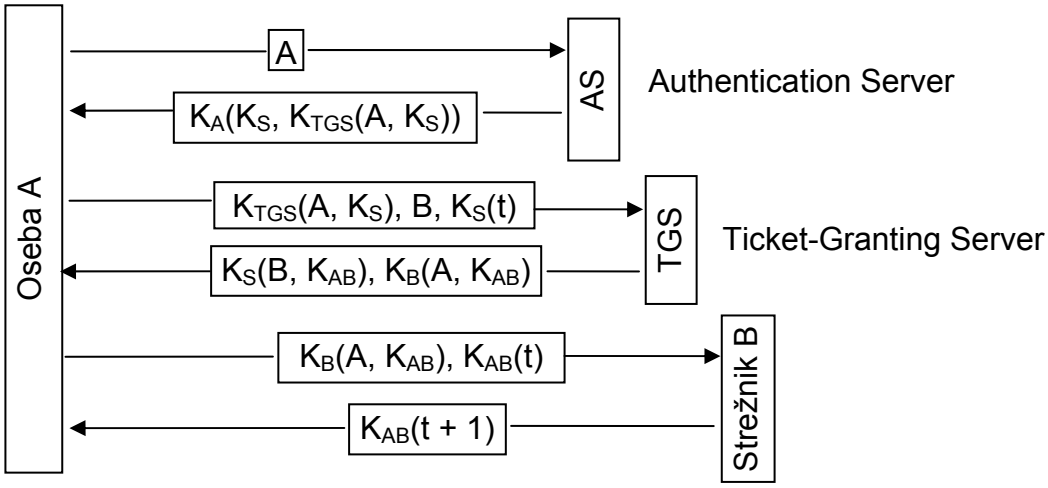
Kerberos

V današnjem času se v veliko sistemih (npr. OS Windows 2000) uporablja protokol Kerberos V4, ki temelji na protokolu Needham-Schroeder. Kerberos je bil iznajden na inštitutu M.I.T. za varen dostop delovnih postaj do mrežnih virov. Obstajajo tudi novejše različice tega protokola, vendar se V4 masovno uporablja v industriji.
Kerberos poleg klientov predpisuje uporabo treh dodatnih strežnikov:

- Authentication Server (AS): preverja uporabnike med prijavljanjem v omrežje
- Ticket-Granting Server (TGS): obravnava in dodeljuje vstopnice za omrežje
- Server: strežnik, ki dejansko opravlja naloge, katere si je klient zaželel

AS ima podobno vlogo kot distribucijski center, saj vzpostavlja varno povezavo s vsakim uporabnikom. TG izdaja vstopnice, s katerimi se klient identificira strežniku.

Za vzpostavitev varne seje oseba A pošlje svojo identifikacijo (uporabniško ime) v čistopisu AS strežniku. AS odgovori s sporočilom šifriranim s ključem K_A, ki vsebuje sejni ključ K_S in vstopnico $K_{TGS}(A, K_S)$. Po prejetju sporočila se osebo A vpraša za geslo, s katerim se generira K_A, katerega oseba A uporabi da dešifriranje sporočila. V trenutku, ko se ustvari K_A se geslo osebe A prepiše oziroma spremeni. Če bi se napadalec želel identificirati kot oseba A, bi ga AS strežnik zavrnil, saj bi napadalec uporabljal neveljavno, staro geslo. Osebe A nato želi dostopati do strežnika B in pošlje TGS strežniku vstopnico $K_{TGS}(A, K_S)$, identifikacijo strežnika B in časovni žig, katerega šifrira s sejnim ključem K_S. Ključni element sporočila je vstopnica, s katerim se oseba A identificira strežniku TGS. Strežnik TGS ustvari nov sejni ključ K_{AB}, kateri bo omogočal varni komunikacijo med osebo A in strežnikom B. TGS odgovori s sporočilom, ki je sestavljen iz dveh delov. Prvi del vsebuje ključ K_{AB} in je šifriran s ključem K_S, katerega zna dešifrirati oseba A. Drugi del vsebuje ključ K_{AB} in je šifriran s ključem K_B, katerega zna dešifrirati strežnik B. Napadalec je tukaj brez moči, saj sporočilo osebe A strežniku TGS vsebuje časovni žig, katerega napadalec ne zna dešifrirati in spremeniti. Če napadalec ponovi to sporočilo dovolj hitro, ko časovni žig pade v interval tolerance, ne doseže nič, saj le sprejme sporočilo strežnika TGS, ki ga prav tako ne zna dešifrirati. Oseba A pošlje strežniku B sporočilo šifrirano s ključem strežnika B, ki vsebuje identiteto in varni ključ dodeljen tej seji K_{AB}, ter časovni žig šifriran s ključem K_{AB}. Strežnik B dešifrira sporočilo in dobi informacijo o ključu dodeljenem tej seji K_{AB}. Strežnik B nazadnje pošlje sporoči osebi A, kateri je dokaz, da oseba A komunicira s strežnikom B. Sporočilo je šifrirano s sejnim ključem K_{AB} in vsebuje časovni žig povečan za ena. Po tem koraku lahko oseba A in strežnik B varno komunicirata s uporabo ključa K_{AB}. V primeru, da želi oseba A dostop do drugih virov omrežja, mora ponoviti postopek pridobivanja sejnega ključa pri strežniku TGS. Oseba A lahko na tak način varno dostopa do vseh vsebin na mreži. Njeno geslo se nikoli ne prenaša po komunikacijskem kanalu.



Slika 13: Kerberos

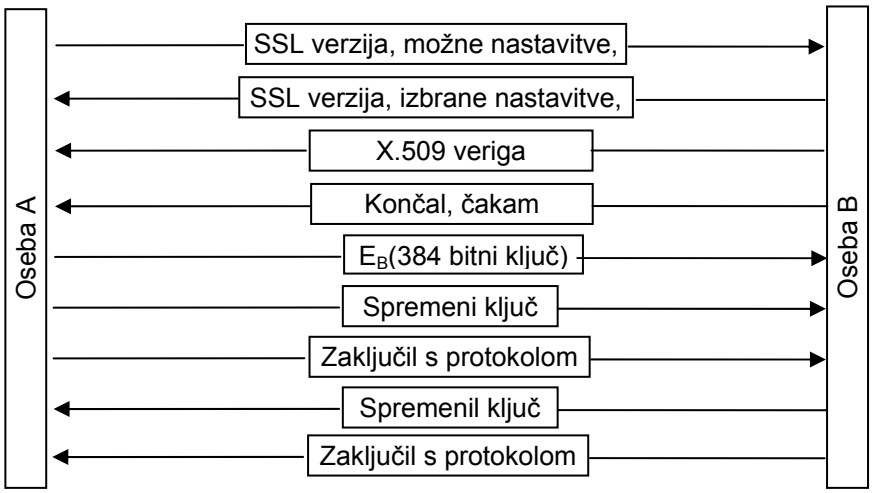
SSL izmenjava ključev

SSL (Secure Sockets Layer) je protokol, ki se ga uporablja v medmrežju za varno komunikacijo. Razvit je bil s strani podjetja Netscape Communication Corp leta 1995. Njegov namen je bil omogočiti varno izmenjavo sporočil med bančnimi transakcijami, elektronskemu kupovanju, uporabi kreditnih kartic za plačevanje, itd. Protokol je aktualen še danes in se ga uporablja za širok spekter storitev.

SSL vzpostavi varno komunikacijo med dvema točkama in vključuje:

- dogovor o parametrih povezave
- avtentikacija sodelujočih
- skrivanje povezave
- ohranjanje integritete podatkov

SSL je sestavljen iz dveh protokolov. Prvi protokol vzpostavi varno povezavo, drugi protokol uporablja varno povezavo. Vzpostavitev varne povezave poteka z izmenjavo sporočil. Oseba A pošlje osebi B prošnjo za začetek vzpostavitve zveze. Sporočilo vsebuje naključno število R_A in SSL verzijo, katero oseba A poseduje, ter njene možnosti nastavitve glede kompresije in šifrirnih algoritmov. Oseba B odgovori s sporočilom, ki vsebuje naključno število R_B in izbrane nastavitve za SSL sejo. Oseba B pošlje tudi sporočilo, ki vsebuje certifikat in predstavlja njegovo identiteto. Nadaljnje osebe B pošlje sporočilo osebi A da je končal in da čaka na odgovor. Oseba A si izbere 384 biten naključni ključ (premaster key) in ga pošlje osebi B v sporočilu šifriranem s javnim ključem osebe B. Oseba B sprejme ključ osebe A. Oseba A in oseba B iz ključa in naključnih števil R_A in R_B na kompleksen način izračunata sejni ključ K_S. Oseba A sporoči osebi B, da naj preklopi na nov sejni ključ K_S in ga obvesti, da je s tem končan protokol za vzpostavitev varne povezave. Oseba B sporoči osebi A, da je sprejel sporočilo in preklopil na nov ključ K_S. S tem dejanjem se konča protokol za izmenjavo varnostnih ključev.



Slika 14: SSL

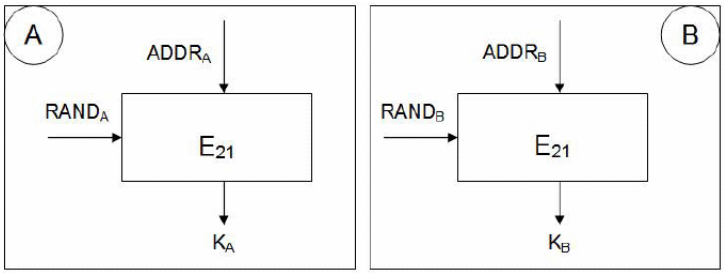
Oseba B ponavadi predstavlja strežnik na medmrežju, oseba A predstavlja uporabnika storitve. Ker oseba B predhodno ne pozna osebo A, se mora oseba A na nek način osebi predstaviti. To je lahko realizirano z prijavo na internetni strani z uporabo uporabniškega imena in gesla.

Bluetooth

Leta 1998 je bila ustanovljena skupina z imenom Bluetooth Special Interest Group. Ustanovili so je največja imena v telekomunikacijski industriji, kot so : Ericsson, IBM, Intel, Nokia in Toshiba. V kasnejših letih so se skupini pridružili številni partnerji, tako da je danes več kot 3000 pridruženih partnerjev. Glavni cilj skupine je bil iznajdba protokola za povezovanje mobilnih telefonov, dlančnikov, slušalk in ostalih mobilnih naprav.
Bluetooth je tehnologija za povezovanje naprav po brezžičnem mediju, ki deluje na frekvenci 2.4 GHz , ima domet do 10 m in omogoča prenos podatkov z maksimalno hitrostjo 1 Mbit/s. Tehnologija Bluetooth vsebuje varnostne mehanizme povezave, med katerimi je prisoten tudi protokol za izmenjavo varnostnih ključev.

Ključen element za varno povezavo v tehnologiji Bluetooth je njen protokol za izmenjavo ključev. Naprava A in B želita komunicirati varno. V začetni fazi se napravi ne poznata in nimata predhodno nameščenega varnostnega ključa. Ob vzpostavitvi povezave se zažene protokol za izmenjavo ključev in rezultat sta ključa povezava (link) in šifriranje (encryption). Proces generiranja ključev se imenuje parjenje (pairing). Dve napravi, ki sta par, med seboj komunicirata z varnim ključem.

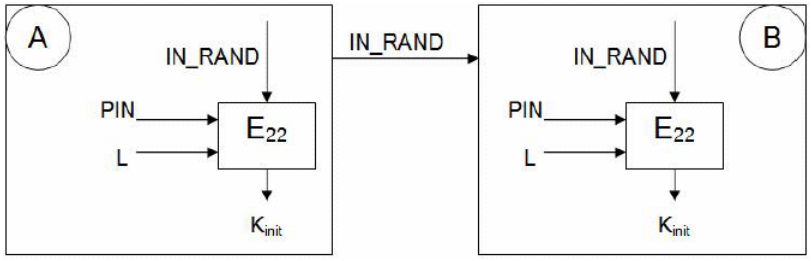
Ob prvem vklopu Bluetooth naprava samodejno izračuna ključ enote (unit key). Ta ključ je unikaten za vsako Bluetooth napravo in se v praksi nikoli ne spreminja. Ključ enote se shrani v NVRAM. Za potrebe komunikacije se ključ enote uporablja le v primeru, da naprava ne poseduje dovolj spomina za shrambo sejnega ključa. Ključ enote se generira iz naključnega števila RAND, ki ga naprava sama generira, in Bluetooth adrese (strojna adresa, ki je unikatna za vsako napravo). Algoritem, ki ustvari ključ enote se imenuje E_{21}.



Slika 15: Izračun ključa enote

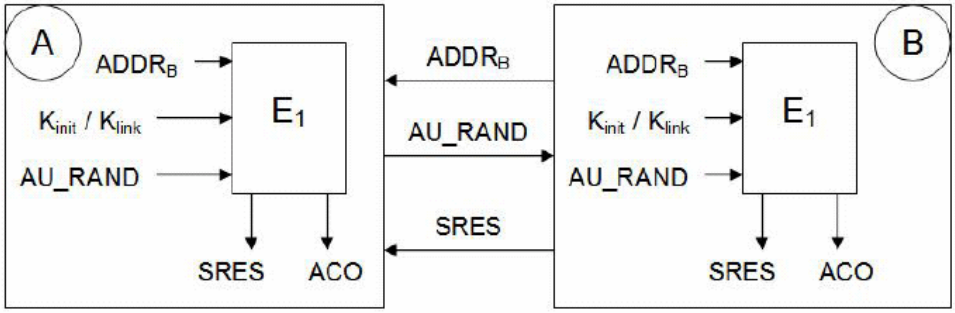
V naslednjem koraku se ustvari ključ vzpostavitve (initialization key). Ključ vzpostavitve je začasen ključ, ki je ustvarjen is kombinacije naključnega števila IN RAND (ustvari ga naprava A, ki želi vzpostaviti povezavo, in ga pošlje napravi B), skupne PIN kode in dolžine L kode PIN. Rezultat operacije algoritma E_{22} je začasen skupni ključ, ki se imenuje ključ vzpostavitve.

PIN se vnese na obeh napravah, ki se želita povezati. Dolžina PIN kode je lahko med 8 in 128 bitov. Tipično je sestavljena iz 4 decimalk. Če naprave ne vsebuje mehanizma za vnašanje PIN kode, je koda nastavljena na 0000.



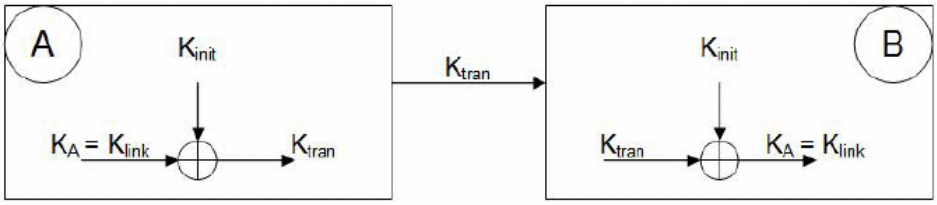
Slika 16: Generiranje ključa vzpostavitve

Ob vsaki novi uporabi skupnega ključa, ki je lahko ključ vzpostavitve ali povezave, se izvede protokol avtetikacije. Protokol temelji na rokovanju in je izveden dvakrat. Najprej se B identificira napravi A, in če je uspešna, se A identificira napravi B. Identifikacije poteka z izmenjavo sporočil. Naprava B pošlje napravi A svojo unikatno Bluetooth adreso. Ko naprava A sprejme sporočilo, generira naključno število AU RAND in jo pošlje napravi B. Naključno število ima vlogo izziva (challange). Obe napravi izračunata funkcijo SRES = E_1(ADDR$_B$,K_{link},AU RAND). K_{link} predstavlja skupni ključ, ki je odvisno od situacije lahko ključ vzpostavitve ali ključ povezave. Avtentikacijska funkcija E_1 je računsko operacija, ki uporablja šifrirno funkcijo SAFER+. Celoten algoritem je izboljšana verzija javno dostopne 64 bitne naprave za bločno šifriranje SAFER-SK128. Naprava B pošlje rezultat funkcije napravi A, katera ga primerja s svojim rezultatom. Če sta izračuna identična, je avtorizacija uspela in začne se postopek pridobivanja sejnega ključa oziroma ključa povezave (link key). Stranski produkt računske operacije E_1 je ACO (Authenticated Ciphering Offset), katerega se nadaljnje uporabi pri ustvarjanju ključa za šifriranje podatkov.



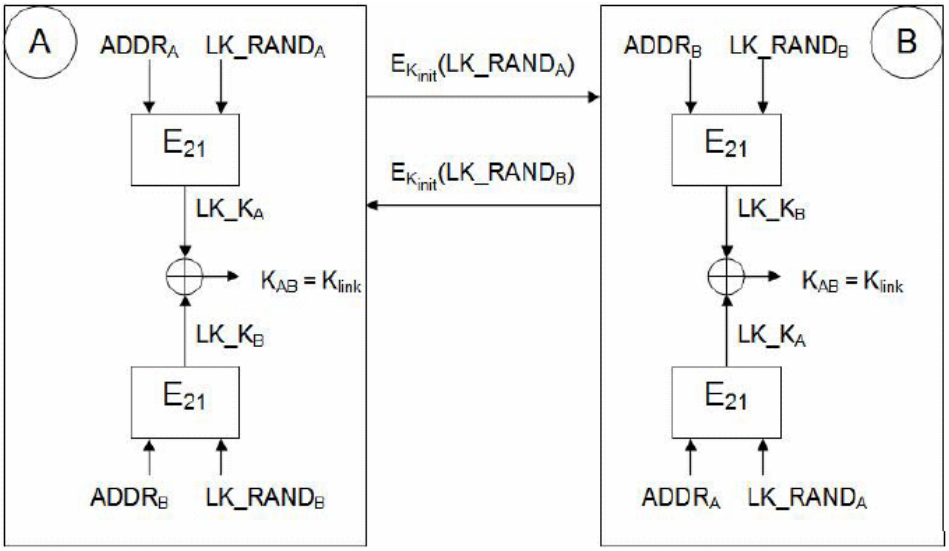
Slika 17: Izračun SRES in ACO

Napravi v tem trenutku posedujeta ključ vzpostavitve. Ključ se nadaljnje uporabi za dogovor o novem delno-stalnem ključu, ki ga imenujemo ključ povezave (link key). Ključ povezave se shrani na obe napravi in se ga lahko uporabi za bodoče povezave. Če naprava nima kapacitete za shrambo ključa povezave, se uporabi namesto njega ključ enote. V tem primeru se ključ povezave, ki je enak ključu enote, prenese iz naprave A v B. Prenos je šifriran z ključem vzpostavitve z operacijo XOR



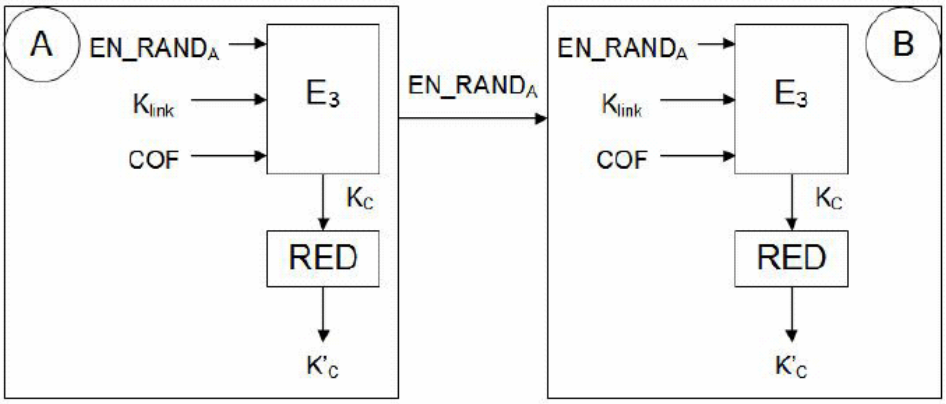
Slika 18: Ključ povezave je ključ enote

V primeru, da ima naprava dovolj spomina, se izračuna ključ povezave iz kombinacije vhodnih podatkov obeh naprav. Obe napravi ustvarita naključno število LK RAND. Napravi svoji števili šifrirata s ključem vzpostavitve in si jih med sabo izmenjata. Nadaljnje napravi izračunata števili LK K_A in L_K KB. Števili uporabimo v algoritmu ustvarjanja ključa povezave E_{21}, ki enak algoritmu ustvarjanja ključa enote. Rezultat algoritma je ključ povezave K_{AB} = K_{LINK}. V naslednjem postopku poteka ponovni protokol identifikacije naprave A in B, kakršna je že potekala po vzpostavitvi ključa vzpostavitve.



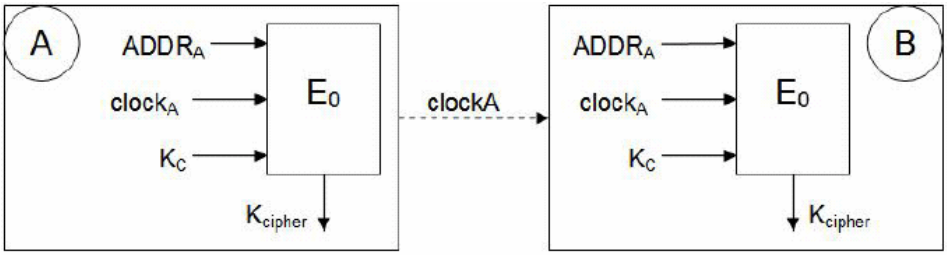
Slika 19: Ključ povezave generiran iz kombinacije vhodnih podatkov obeh naprav

Sledi postopek izmenjave ključev za šifriranje podatkov. Naprava A ustvari naključno število EN RAND$_A$ in ga pošlje napravi B. Obe napravi izračunata šifrirni ključ $K_C = E_3$(EN RAND$_A$, K_{link}, COF). COF (Ciphering Offset Number) je ACO število, ki smo ga izračunali med postopkom avtentikacije. Dolžino šifrirnega ključa lahko po potrebi zmanjšamo na šifrirni ključ K_{0C}.



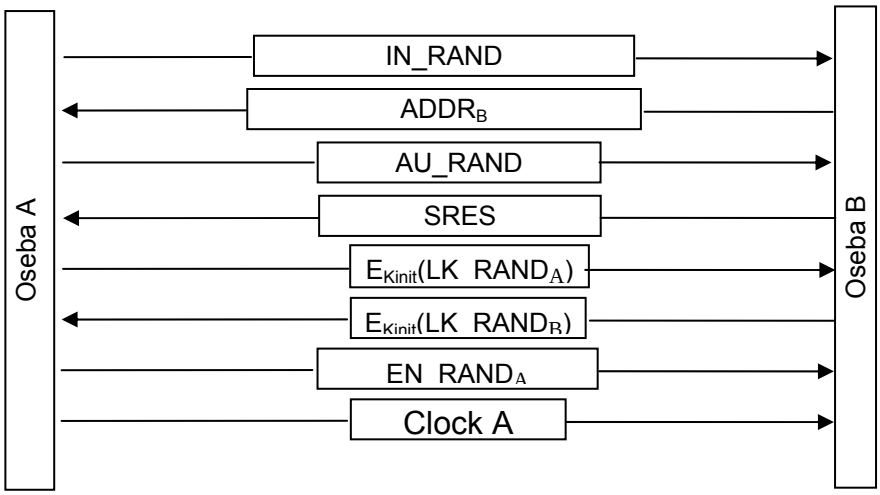
Slika 20: Izračun šifrirnega ključa Kc

V zadnjem koraku se šifrirni ključ uporabi za izračun varnega pretočnega ključa K_{CIPHER}. Izračuna se ga z algoritmom E_0, v katerega vnesemo K_C, ter Bluetooth adreso in notranjo uro naprave A. S pretočnim ključem K_{CIPHER} in funkcijo XOR šifriramo podatke, ki jih želimo prenašati.



Slika 21: Izračun pretočnega šifrirnega ključa

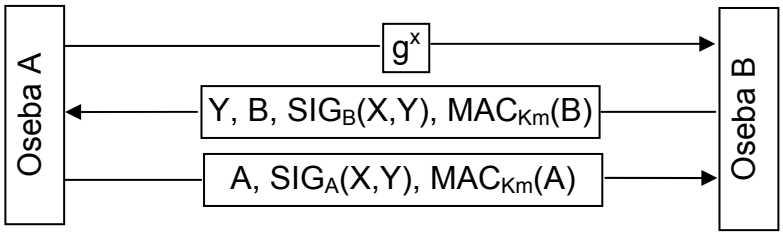
Največja slabost protokola je uporaba PIN kode, ki se prenaša med napravami kot čistopis. Ponavadi se uporablja le kratka PIN koda (4 znaki) ali koda »0000«. Celoten postopek in varnost je odvisna od dolžine PIN kode, zato je priporočljiva uporaba čim daljše PIN kode.



Slika 22: Bluetooth

Sigma Σ_0

Protokol Sigma je protokol za izmenjavo varnostnih ključev, kateri vključuje mehanizem za avtentikacijo. Razvit je bil z namenom izboljšanja varnosti protokola IKE in njene zadnje različice IKEv2 in temelji na protokolu Diffie-Hellman ob uporabi digitalnih podpisov. V protokolu Sigma je obrazloženo, da samo znanje ključa varne povezave $K_{XY} = g^{xy} \bmod n$, kot ga poznamo iz D-H protokola ni zadosten, saj direktno ne povezuje ključa povezave z njenima uporabnikoma. Kratico SIGMA si lahko razlagamo kot zloženko dveh besed, ki zapovedujeta da moramo sporočilo podpisati in izračunati njeno avtentikacijsko kodo (SIGn and MAc-your-own-identity). Avtentikacijska koda MAC (Message authentication code) vzame šifrirni ključ in sporočilo in nato izračuna zgoščevalno funkcijo.



Slika 23: SIGMA

Protokol izmenjave ključev se začne s izbiranjem velikega primarnega števila g. Nato si oseba A izbere veliko število x in izračuna $X=g^x$. Podobno operacijo $Y=g^y$ izvrši oseba B s uporabo svojega števila y. Oseba A pošlje X osebi B. Oseba B izračuna $K=X^y=g^{xy}$ in $k_m=MAC(K)$. Oseba B nato pošlje osebi A sporočilo, ki vsebuje:
- $Y = g^y$
- B; identiteta osebe B
- SIGN $k_b(X,Y)$; podpisano sporočilo s privatnim ključem osebe B, katero vsebuje X in Y
- MAC $k_m(B)$; šifrirano sporočilo s ključem k_m , ki vsebuje identiteto osebe B

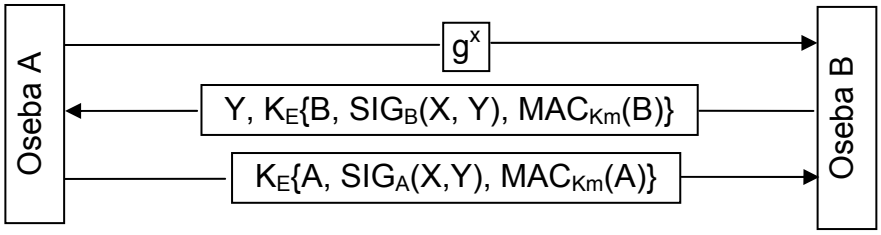
Oseba A sprejme sporočilo in izračuna $K=Y^x$ in $k_m=MAC(K)$. Nadaljnje dešifrira oba sporočila z ustreznim ključem in preveri, če se podatki ujemajo:
SIGN $k_B(X,Y)$ ➔ Y ➔ preveri, če se ujema s Y, ki ga je dobil v čistopisu
MAC $k_m(B)$ ➔ B ➔ preveri, če se ujema s B, ki ga je dobil v čistopisu

Če se oba podatka ujemata, oseba A ve, da se pogovarja z osebo B. Oseba A v odgovoru osebi B pošlje sporočilo, ki vsebuje:
- A; identiteta osebe A
- SIGN $k_A(X,Y)$; podpisano sporočilo s privatnim ključem osebe A, katera vsebuje X in Y
- MAC $k_m(A)$; šifrirano sporočilo s ključem k_m , ki vsebuje identiteto osebe A

Oseba B sprejme sporočilo, ga ustrezno dešifrira in preveri ujemanje podatkov:
MAC $k_m(A)$ ➔ A ➔ preveri identiteto A in uporabi javni ključ osebe A za dešifriranje podpisanega sporočila SIGN $k_A(X,Y)$
SIGN $k_A(X,Y)$ ➔ X ➔ preveri, če se ujema s X, ki ga je dobil v čistopisu v prvem koraku

Sigma ima dve verziji svojega protokola Sigma-i in Sigma-r. Obe različici imata lastnost varovanja identitete. Pri Sigma-i je varovana identiteta osebe, ki želi vzpostaviti povezavo, medtem ko je pri Sigma-r varovana identiteta prejemnika prošnje za vzpostavitev povezave.

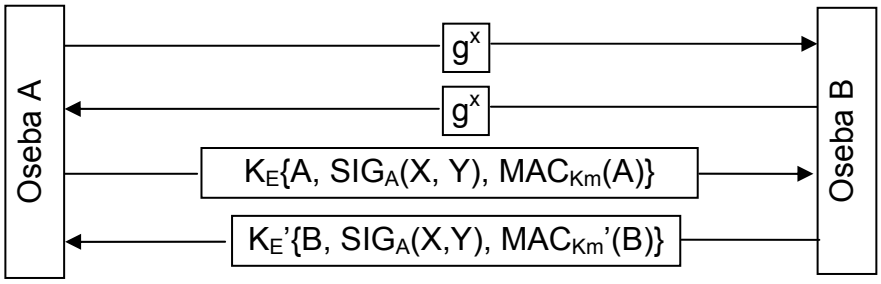
Sigma-i



Slika 24: SIGMA-i

K_e and K_m sta izpeljana iz g^{xy} preko psevdonaključne funkcije.

Sigma-r



Slika 25: SIGMA-r

K_m, K_m' in K_e, K_e' se uporabljajo proti napadu s ponavljanjem.

JFK (Just Fast Keying)

JFK je protokol za izmenjavo ključev preko interneta, ki je bil od začetka načrtovan da izpopolnjuje pogoje:

- Varnost: samo udeleženca komunikacije imata dostop do sejnega ključa
- PFS: kar se da se mora približati Perfect Forward Secrecy
- Privatnost: Obvarovati mora identiteto vsaj ene od strani komunikacije
- DoS: protokol mora biti odporen proti DoS napadom
- Efektivnost: protokol mora biti varčen z procesorsko močjo, potrebno pasovno širino in številom izmenjave sporočil
- Preprostost: protokol ne sme imeti veliko število nastavitev

JFK protokol je preprost, efektiven in varen. Namenjen je za uporabo v IPsec arhitekturi, kjer lahko nadomesti IKE protokol.

Obstajata dve verziji JFK protokola. Oba protokola uporabljata štiri sporočila, oziroma vsak po dva sporočila za vzpostavitev varne povezave. Obe verziji protokola sta enako odporni proti DoS napadom.
V prvi varianti, ki jo zapišemo JFKi, je proti aktivnemu napadalcu varovana identiteta tistega, ki želi povezavo vzpostaviti. Varnost identitete tistega, ki sprejema zahtevo za vzpostavitev povezave ni zaščitena. JFKi se uporablja za scenarij klient-strežnik, kjer klient oziroma tisti, ki povezavo vzpostavlja, želi obvarovati svojo identiteto, medtem ko je strežnik javen.
JFKr je druga verzija protokola. V tem primeru se proti aktivnemu napadalcu obvaruje identiteta tistega, ki sprejema željo po povezavi. Obenem protokol omogoča zaščiti obeh strani povezave proti pasivnemu prisluškovanju. Protokol JFKr se uporablja v situaciji, kjer poteka peer-to-peer komunikacija in sprejemnik zahteve želi ohraniti svojo identiteto skrito. JFKr v nasprotju z JFKi omogoča tajenje o izmenjavi ključev. Če se kdo dokoplje do ključa v primeru JFKr seje, ne more dokazati, kdo je uporabljal ključ za povezavo.
Oba protokola temeljita na osnovi protokola SIGMA, katerega modificirata in si prirejata. Ob vsaki uporabi JFK protokola morata obe strani povezave generirati svežo naključno število. Naključni števili se uporabljajo za izračun novega sejnega ključa in so zaščita za primer, da se za izračun želi uporabiti staro eksponentno število za D-H protokol. Sejni ključ bo drugačen, če bo vsaj eno naključno število drugačno.

JFKi

V prvem sporočilu se predvideva, da oseba A, ki želi vzpostaviti povezavo, pozna grupo veljavnih številk g^A,oziroma $g^A \bmod(p)$, s katerimi se lahko poveže z osebo B.Oseba A generira naključno število g^A, ki je unikatna za to sejo. Te številke so lahko javne in se lahko večkrat uporabijo. Poleg g^A oseba A pošlje identiteto A', s katero se želi avtenticirati osebi B oziroma želi sporočiti s čigavim ključem nad oseba B nadaljnje šifrira sporočila. Iz identitete oseba B ugotovi certifikat osebe A in njen javni ključ. Sporočilo je poslano v čistopisu.
Odgovor osebe B na prošnjo po vzpostavitvi veze osebe A je bolj kompleksen. Osebe B najprej preveri veljavnost g^A. Nato generira svojo eksponentno število g^B in izračuna $g^B \bmod(p)$, naključno število R_B in svojo identiteto, s katero želi osebi A sporočiti, s katerim ključem naj odgovor šifrira. V sporočilu prav tako vključi podatke, katere šifrirne protokole podpira in kakšne nastavitve naj oseba A uporabi. Sporočilo še vključuje rezultat zgoščevalne funkcije HK_B, ki je generiran iz podatkov g^B, naključnih števil R^A , R^B in IP adrese osebe A. Edina procesorska operacija v odgovoru je računanje MAC funkcije dela

sporočila. Zgoščevalna funkcija MAC ni procesorsko požrešna in če osebe B (strežnik) ni procesorsko zelo obremenjena, lahko oseba B izračuna novo eksponentno število in podpis le za to izmenjavo sporočil. V tem primeru mora strežnik shraniti nekatere podatke (D-H) osebe A za nadaljnjo operacijo.
Oseba A sprejme sporočilo in ustvari odgovor nanj. Oseba A preveri rezultat zgoščevalne operacije HK_B in preveri identiteto osebe B. Znotraj HK_B je IP adresa osebe A. Oseba A lahko preveri če sta IP adrese v odgovoru in IP adresa osebe A identična. IP podatek v sporočilu je zaradi ugotavljanja DoS napada »cookie jar«. Odgovor vsebuje potrdilo, da sta predhodni dve fazi izmenjanja sporočil bili uspešni in veljavni. V odgovoru je prav tako identiteta osebe A, informacija o storitvi, do katere želi dostopati osebe A. Šifrirani oziroma podpisani del sporočila vsebuje naključni števili R^A , R^B, identiteto strežnika B in obe eksponentni števili g^A, g^B.
Strežnik B sprejme sporočilo in ustvari novo sporočilo, ki vsebuje informacije za vzpostavitev IPSec povezave, podpisane podatke R_A, R_B, g^A, g^B in identiteto osebe A.
Šifrirni ključi, ki so uporabljeni v sporočilih so ustvarjeni iz podatkov R_A, R_B in g^A, g^B. Šifrirne operacije in algoritmi so opisani v $grpinfo_B$, kjer najdemo podatke o šifrirnih postopkih, simetričnih algoritmih in zgoščevalnih funkcijah, katere uporabljamo v izmenjavi sporočil.

“cookie jar” DoS napad: napadalec pridobi piškotek s vzpostavitev seje in razkritjem IP adrese kompomitiranega klienta. Napadalec na tak način pridobi veliko število veljavnih piškotkov. Vse te podatke nadaljnje uporabi za DDoS napad.

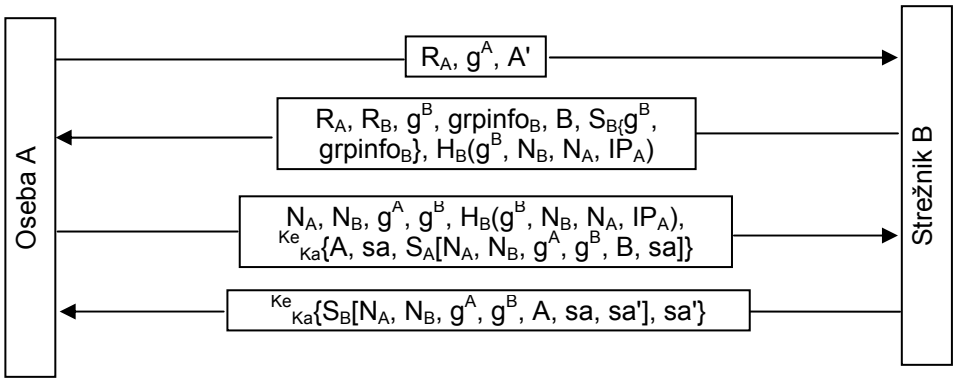
Razlaga uporabljenih kratic:

$H_k(M)$ zgoščevalna funkcija s ključem k sporočila M. H je psevdonaključna funkcija in opravlja nalogo varnostne avtentikacijske sporočilne funkcije (MAC).

$\{M\}^{Ke}_{Ka}$ Šifriranje s uporabo šifrirnega ključa Ke, kateremu sledi MAC avtentikacija (funkcija zgoščevanja) z uporabo simetričnega ključa Ka sporočila M. MAC je izračunan nad šifriranim sporočilom in ima pripono v obliki ASCII besedila »A« ali »B«, odvisno od tega, kdo sporočilo pošilja (A = oseba A, B = strežnik B).

$S_x(M)$ Digitalno podpisano sporočilo M s ključem x, kjer je x lahko A ali B.

sa Šifrirne in storitvene lastnosti varnostnega dogovora SA, katere oseba A želi doseči.
sa' SA podatki strežnika B, ki so potrebne za vzpostavitev IPSec seje.

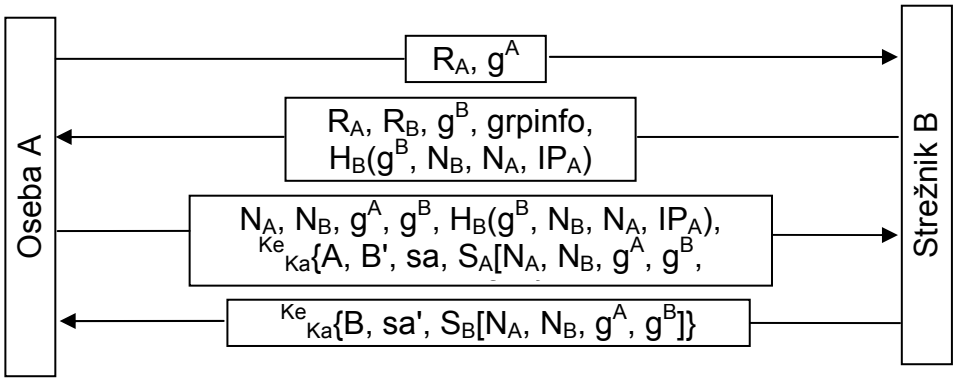


Slika 26: JFK-i

Ob vsaki uporabi g^A in g^B je mišljena uporaba funkcije $g^A(\text{mod } n)$ oziroma $g^B(\text{mod } n)$.

JFKr

Obe strani komuniciranja pošiljata svojo identiteto šifrirano in avtentificirano s uporabo K_E in K_A ključa in se s tem zaščitita proti pasivnim prisluškovalnim napadanjem. Prvi, ki razkrije svojo identiteto je tisti, ki zvezo vzpostavlja – oseba A. Oseba B razkrije svojo identiteto šele potem, ko preveri identiteto osebe A. Današnja tehnologije ne omogoča popolne zaščite za obe strani komunikacije (DoS zaščita, pasivno varovanje identitete obed strani, aktivno varovanje identitete vzpostavljajoče osebe). Posledično je potreben dogovor o vrsti povezave. Če želimo varno povezavo proti DoS napadom, potem mora odzivna stran prenesti prvo sporočilo preden pošlje drugo, procesorsko bolj obremenjujoče. To pomeni, da mora odzivna stran – oseba B, poslati svojo identiteto v zadnjem sporočilu, če želi biti zaščitena proti pasivnemu prisluškovanju. Ker je identiteta osebe B poslana v zadnjem sporočilu, mora oseba A razkriti svojo identiteto pred osebo B.



Slika 27: JFK-r

IKE (Internet Key Exchange)

IPSec je varni mrežni protokol, ki temelji na konceptu skrivnega simetričnega ključa. Pošiljatelj in prejemnik želita izmenjavo sporočil, katere lahko prebereta le sama. Proti napadom in spremembo sporočila uporablja protokola ESP (Encapsulating Security Payload) in AH (Authentication Header). ESP omogoča šifriranje podatkov sporočila. AH vsebuje sisteme za ohranjanje nepovezavne integritete in informacije o izvoru IP datagramov ter nudi zaščito proti ponavljajočem napadu (replay attack). Za delovanje IPSec protokola potrebujeta njegova varnostna mehanizma ESP in AH skrivni ključ. IKE je protokol za izmenjavo ključev preko interneta in je del protokola IPSEC.

IKE je hibridni protokol, ki implementira dva starejša protokola Oakley in SKEME, ki rahlo prilagojena delujeta v okviru protokola ISAKMP (Internet Security Association and Key Management Protocol). ISAKMP predpisuje osnovno zgradbo za izmenjavo ključev in avtentikacijo; Oakley opisuje sekvenčni potek izmenjave ključev in opisuje storitve za identifikacijo in avtentikacijo, SHEME opisuje dejanski potek izmenjave ključev. Za delovanje storitve IPSec ni nujno potreben IKE. Uporablja se ga, ker nudi dodatne ugodnosti, kot so: avtomatična vzpostavitev in avtentikacija zveze, avtomatično uporabo certifikatov (CA) in možnost izmenjave ključev med IPSec sejo. IKE omogoča implementacijo in uporabo storitev MD5, SHA in Tiger za zgoščevanje, Triple-DES za šifriranje, digitalno podpisovanje z uporabo RSA algoritma, itd.

Glavne lastnosti IKE:
- izmenjava in dogovarjanje o varnostni politiki
- vzpostavitev varnih sej, ki jih imenujemo SA (Security Association)
- izmenjava ključev
- upravljanje s ključi
- se lahko uporablja tudi drugje, ne samo znotraj IPSec

Security Association (SA) je skupek atributev, ki opisujejo varno povezavo. Vsebuje podatke o metodah šifriranja, ključih in življenjski dobi seje. V okviru ISAKMP potrebujemo za vzpostavitev varne povezave dva koraka SA sej.

ISAKMP korak 1

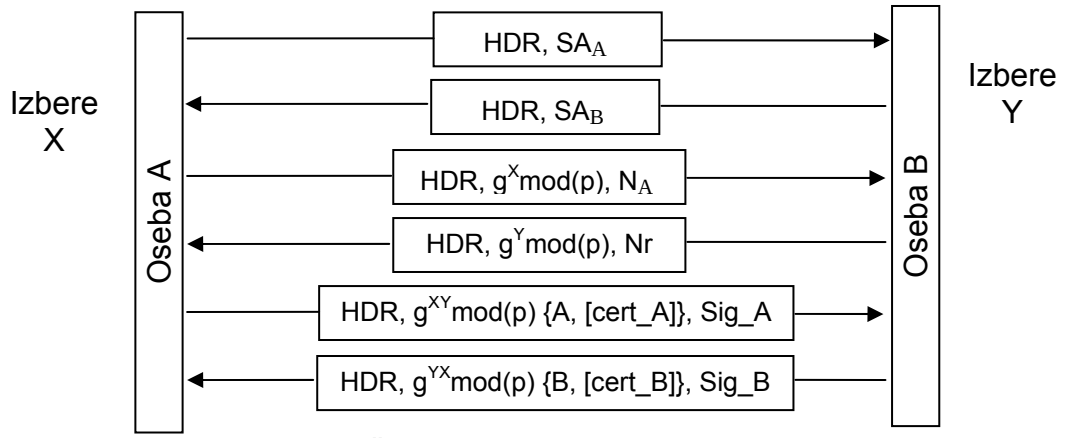
1. Dogovor in vzpostavitev ISAKMP Security Association (SA); varen komunikacijski kanal za nadaljnje potrebe IKE komunikacija.

   Oba konca povezave generirata skupno vrednost z uporabo Diffie-Hellman algoritma, katerega uporabimo kot osnovo simetričnega ključa. Nadaljnje IKE komuniciranje poteka šifrirano s tem ključem.

2. Preverjanje identitete sistema na nasprotni strani (osnovna avtentikacija).

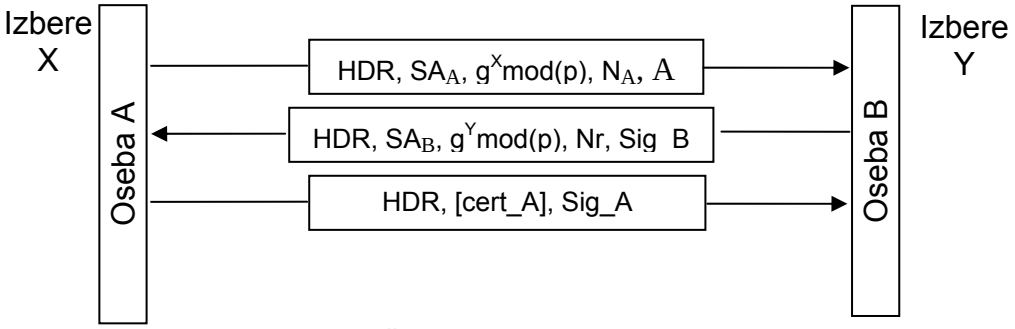
   - Med dvema točkama se vzpostavi samo enkrat.
   - Lahko poteka v dveh načinih:
     Osnovni način – Izmenjava 6 sporočil, omogoča več možnosti dogovora
     Hitri način – Izmenjava 4 sporočil, omogoča manj možnosti dogovora

Osnovni način



Slika 28: IKE faza 1, osnovni način

Agresivni način

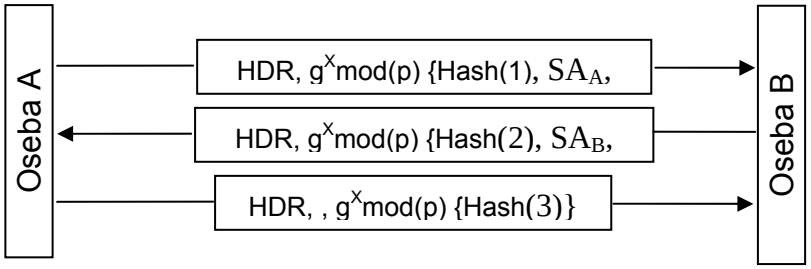


Slika 29: IKE faza 1, agresivni način

Sig_A in Sig_B predstavljata dokaz identitete. Računsko to pomeni, da se s dogovorjenim ključem šifrira zgoščevalno funkcijo Hash_A oziroma Hash_B. Hash je funkcija g^X, g^Y, SA_A, SA_B, A, B.

ISAKMP korak 2

Z uporabo ISAKMP SA varne povezave se vzpostavi ena ali več SA varnih sej za IPSec (ESP, AH, …). Vsaka storitev lahko vzpostavi svojo SA sejo



Slika 30: IKE faza 2

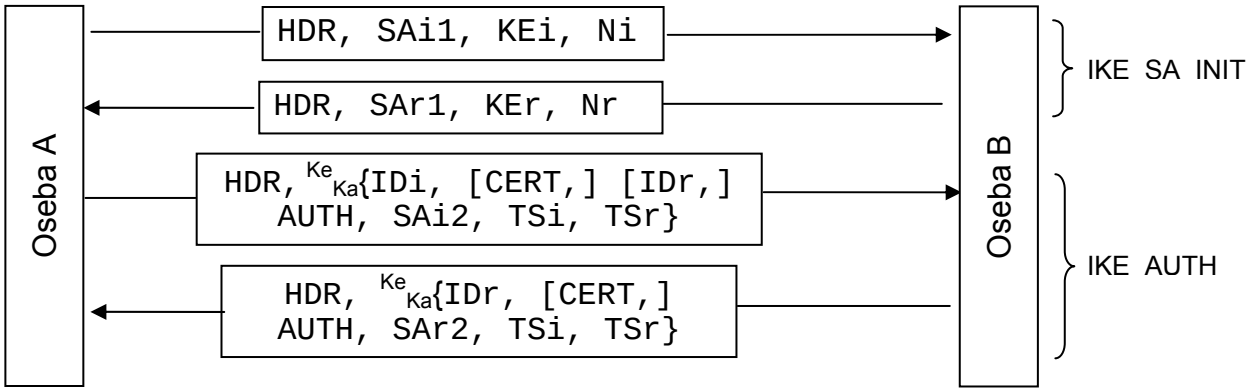
SA_A, SA_B; kriptografski postopki, katere podpirata
Hash(x); verzija zgoščevalne funkcije s parametrom x

IKEv2

Protokol IKEv2 je nadgradnja protokola IKE. Uporablja se ga za izmenjavo sporočil za vzpostavitev varne IPSec internetne povezave.

Komunikacija s protokolom IKEv2 se vedno začne z sporočili IKE_SA_INIT in IKE_AUTH, bolj znanimi kot faza 1 v IKEv1. Začetna izmenjava sporočil v normalnih pogojih zajema izmenjavo štirih sporočil, čeprav obstaja možnost, da se število sporočil poveča. Vsa komunikacija v IKEv2 protokolu deluje po načelu parov zahteva/odgovor. Glava sporočila vsebuje Security Parameter Indexes (SPIs), verzijo, in razne zastave. Prvi par sporočil IKE_SA_INIT vsebuje možnosti za dogovor o šifrirnih algoritmih, izmenjavo naključnih števil in izmenjavo sporočil za zagon Diffie-Hellman [D-H] protokola.

Drugi par sporočil označimo z IKE_AUTH in ima funkcijo avtentikacije, izmenjajo identitete, certifikate in vzpostavijo varno povezavo (CHILD_SA). Sporočila so delno šifrirana in imajo zaščito integritete, ki omogočajo varno povezavo pred pasivnimi napadi in prisluškovanjem.



Slika 31: IKEv2

Uporaba kratic:

AUTH	avtentikacija
CERT	Certifikat
HDR	IKE glava sporočila
IDi	Identiteta tistega, ki povezavo vzpostavlja (Initiator)
IDr	Identiteta tistega, ki odgovarja na prošnjo (Responder)
KE	Izmenjava varnostnih ključev
Ni, Nr	Naključna števila
SA	Varnostni dogovor - Security Association
TSi	Izbira prometa - Initiator
TSr	Izbira prometa - Responder

V SAi1 je zapis o šifrirnih postopkih, katere oseba A podpira za vzpostavitev IKE_SA. V KEi sporočilu je poslana informacija za D-H algoritem. Ni predstavlja naključno število osebe A.

Odgovor na sporočilo vsebuje informacije o šifrirnih postopkih, ki jih oseba B podpira v SAr1 sporočilu. V KEr je podatek, ki omogoča osebi A izračun D-H ključa. Poleg tega oseba B pošlje še naključno število Nr.

Po sprejetju sporočila obe strani izračunata ključe z uporabo operacije SKEYSEED za nadaljnje komuniciranje. Vsa sporočila, ki sledijo so šifrirana in imajo zaščito integritete. Obe strani izračunata svoje varnostne ključe.

SKEYSEED=prf(Ni | Nr, g^ir)
SK_d=prf(SKEYSEED, Ni | Nr | SPIi | SPIr | 0x01)
SK_ai=prf(SKEYSEED, SK_d | Ni | Nr | SPIi | SPIr | 0x02)
SK_ar=prf(SKEYSEED, SK_ai | Ni | Nr | SPIi | SPIr | 0x03)
SK_ei=prf(SKEYSEED, SK_ar | Ni | Nr | SPIi | SPIr | 0x04)
SK_er=prf(SKEYSEED, SK_ei | Ni | Nr | SPIi | SPIr | 0x05)
Naključna števila dodajajo svežino pri izračunu ključev.

Po izmenjavi prvih dveh sporočil se vzpostavi IKE-SA, ki je varna ne pa tudi avtenticirana povezava.

Nadaljnja sporočila so namenjena avtentikaciji obeh strani. Obe strani pošljeta drugemu svojo identiteto in pripadajočo skrivnost. Avtentikacija poteka z uporabo digitalnih podpisov in izračunov zgoščevalne funkcije ali z uporabe vnaprej dogovorjenega skrivnega ključa.

Informacija za vzpostavitev prve SA varne seje oziroma Child_SA se prenaša tretjem in četrtem sporočilu. V tretjem sporočilu oseba A predlaga IPSec SAi2. V četrtem sporočilu je odgovor osebe B, v katerem se strinja z predlagano sejo IPSec SAi2. Oblika prometa, katere je dovoljena da se uporablja skozi Child_SA sejo se dogovori preko sporočila izbire prometa TSi in TSr.

802.11i (WPA2)

S prihodom Wi-Fi komunikacije po zraku je prišla tudi potreba po njeni varnosti. Razvili so se različni protokoli, ki so imeli nalogo, da bi le-to omogočili. Prvi je bil razvit protokol WEP (Wired Equivalent Privacy), ki se je izkazal za neučinkovitega, in je odsvetovan za uporabo. WEP ne omogoča varno povezavo, saj je bil njegov protokol razbit. Proizvajalci opreme so se združili in začeli razvijati nov, boljši protokol, ki bi poleg varnosti omogočal tudi avtentikacijo in integriteto podatkov. Nov protokol, ki so ga poimenovali 802.11i, je potreboval veliko časa za razvoj in ratifikacijo. Posledično so se nekateri proizvajalci opreme združili in izdali nov standard WPA (Wi-Fi Protected Access), ki predstavlja vmesni korak med WEP in 802.11i.

WEP
Uporablja statični varnostni ključ za komunikacijo, ki je lahko 64 ali 128 biten. Za povečanje varnosti uporablja 24 biten incializacijski vektor, ki je delno naključen. WEP ne vsebuje protokola za izmenjavo varnostnih ključev. Ključe je potrebno ročno vnašati na obeh straneh komunikacije.
Potrebni čas za razbitje 64bitnega ključa je dve sekundi, za 128 bitnega je 10 minut.

WPA
Vmesni korak na poti proti 802.11i. WPA ohranja združljivost z WEP protokolom in svojim naslednikom. Za izboljšanje varnosti uporablja dinamično spreminjanje šifrirnega ključa z uporabo protokola TKIP. WPA je za potrebe avtentikacije vpeljal protokola EAP (Extensible Authentication Protocol).

802.11i (WPA2)
WPA2 je nov standard in predstavlja najvišjo zaščito za varno komunikacijo v tehnologiji Wi-Fi. Za avtentikacijo uporablja arhitekturo, ki temelji na 802.1X protokolu (EAP, avtentikacijski strežnik). WPA2 za šifriranje podatkov uporablja AES (Advanced Encryption Standard). Za potrebe zagotavljanja integriteta sporočila WPA2 uporablja MIC (Message Integrity Control ~ Michael).

Oblika sporočila EAP (EAP over LAN):
EAPOL-Key(S, M, A, I, K, KeyRSC, ANonce/SNonce, MIC, RSNIE, GTK[N])

S pomeni, da je prvotna izmenjava ključev končana. Varnostni bit informacije o ključu
M pomeni, da je MIC možen v vseh sporočilih.
A pomeni, da je potreben odgovor na to sporočilo
I bit nalaganja. Naloži pairwise ključ
K tip ključa: P (Pairwise), G (Group/STAKey)
KeyRSC ključ RSC
ANonce/SNonce sta naključni števili klienta in AP
MIC preverjanje identitete, ki je narejena z uporabo KCK
RSNIE informacija o RSN elementu
GTK enkapsulacija GTK.
N je indikator ključa, ki se uporabi z GTK

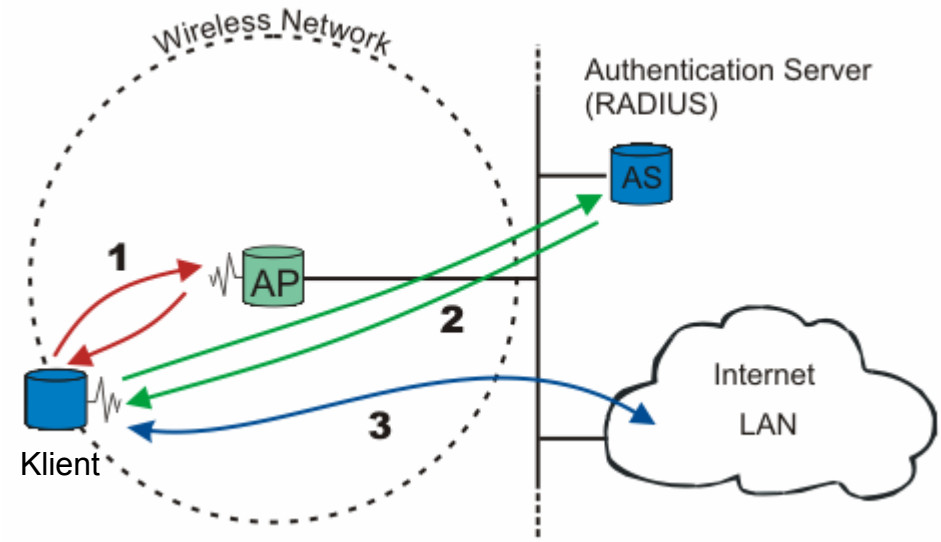
Vzpostavitev varne povezave - avtentikacija

1. Vzpostavitev WPA2 varne povezave in izmenjava ključev se začne z prošnjo(request) access point-a (AP) po identiteti WiFi klienta (WN). Sporočilo je poslano v EAP obliki.

2. Klient (WN) pošlje informacijo o svoji identiteti AS in s tem začne proces avtentikacije. AS sporočilo razpakira, ga ponovno enkapsulira v sporočilo formata RADIUS in ga po varni povezavi pošlje Avtentitakorju (Authentication Server – AS [RADIUS strežnik]). AP v tem procesu deluje kot tranzitni strežnik in ne sodeluje v avtentikaciji.

Kot AS se ponavadi uporablja RADIUS strežnik (Remote Authentication Dial-In User Service). Primarno je uporabljen pri ponudnikih internetnih storitev (ISP) za avtentikacijo uporabniškega imena in gesla. 802.1X ne specificira uporabo RADIUS-a za avtentikacijski strežnik. Kljub temu se smatra RADIUS kot "de-facto" standard v 802.1X. Obstajajo tudi drugi AAA (Authentication, Authorization, and Accounting) strežniki, eden takih je strežnik DIAMETER.

3. Klientu se odobri ali zavrne dostop. V primeru odobritve AS se nadaljuje proces vzpostavljanja varne povezave.



Slika 32: Vzpostavitev varne povezave - avtentikacija

Vzpostavitev varne povezave – 4 smerno rokovanje za vzpostavitev varnega ključa

V primeru odobritve v sporočilu 3, AS poleg odobritve pošlje klientu v sporočilu informacijo o glavnem ključu (Master Key – MK). Informacija o MK je dostopna samo klientu in AS. MK je unikaten in je vezan na sejo med klientom in AS.

Klient in AS na podlagi MK izračunata nov ključ, ki se imenuje Pairwise Master Key (PMK). Samo AS in klient znata izračunati PMK.

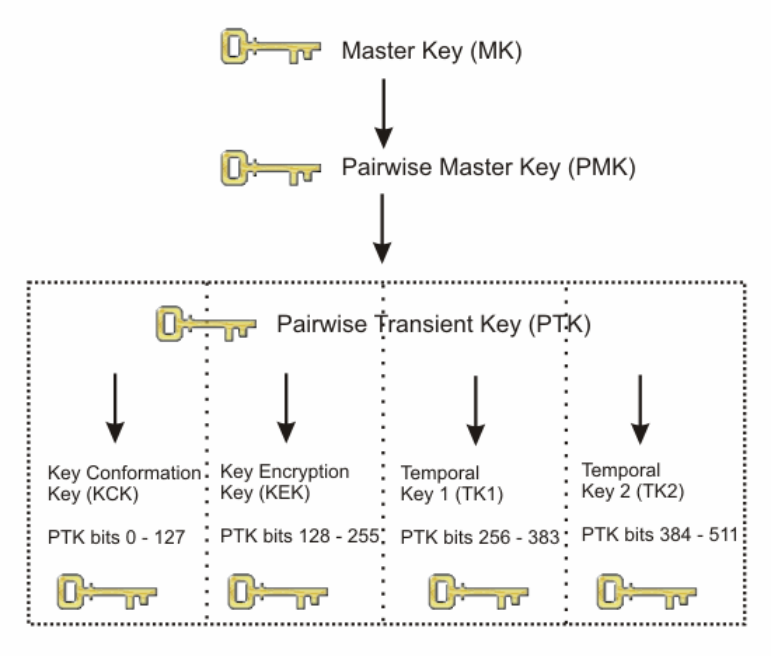
AS pošlje AP informacijo o PMK. PMK je svež simetrični ključ, ki se uporablja za komunikacijo med klientom on AP.
Med klientom in AP se zažene 4-smerni protokol rokovanja, katerega rezultat je nov ključ, ki se imenuje Pairwise Transient Key (PMK). Izračun PTK temelji na psevdonaključni funkciji, ki ima za vhodne podatke MAC adresi in naključni števili klienta in AP.

AS => Klient: EAPOL-Key(0,0,1,0,P,0,ANonce,0,0,0)
Klient => AS: EAPOL-Key(0,1,0,0,P,0,SNonce,MIC,RSNIE,0)
AS => Klient: EAPOL-Key(1,1,1,1,P,KeyRSC,ANonce,MIC,RSNIE,GTK[N])
Klient => AS: EAPOL-Key(1,1,0,0,P,0,0,MIC,0,0)

Psevdonaključna fukncija za izračun PTK:
PRF- X(PMK, “Pairwise key expansion”,
Min(AA,SPA) || Max(AA,SPA) ||
Min(ANonce,SNonce) ||
Max(ANonce,SNonce))

PRF – X = psevdonaključna (pseudorandom) funkcija, ki ima X bitov na izhodu
SPA = MAC adresa klienta, AA = MAC adresa AP
SNonce = naključno število klienta, ANonce = naključno število AP

PMK ni samostojen ključ, ampak je sestavljen iz več ključev:
Key Confirmation Key (KCK); uporablja se za dokazilo o posedovanju PMK in za povezavo PMK na AP.
Key Encryption Key (KEK); uporablja se v postopku distribucije Group Transient Key (GTK).
Temporal Key 1 in 2 (TK1/TK2); uporablja se jih za šifriranje.

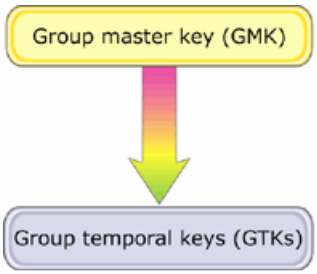


Slika 33: Generiranje PTK ključev

Nadaljnje se zažene rokovanje za prenos GTK ključa od AP do klienta. GTK je ključ, katerega uporabljajo vsi klienti trenutno na mreži za varni prenos multicast in broadcast sporočil. Zahtevo po prenosu začne klient.
Klient => AS: EAPOL-Key(1,1,1,0,G,Key RSC,0, MIC, 0,GTK[N])
AS => Klient: EAPOL-Key(1,1,0,0,G,0,0,MIC,0,0)

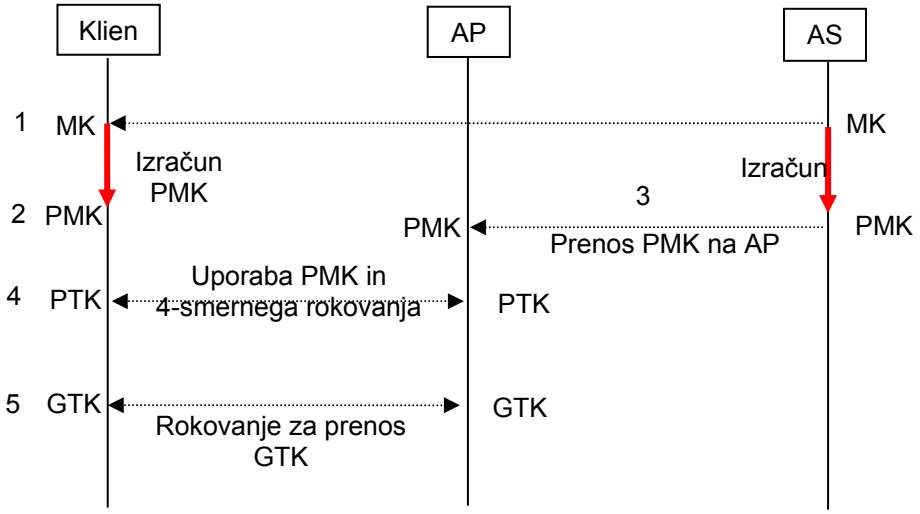
Operacija za izračun GTK:
PRF-X(GMK, 'Group key expansion', AA || GNonce)
PRF – X = psevdonaključna (pseudorandom) funkcija, ki ima X bitov na izhodu

AA = MAC adresa AP
GNonce = naključno število, ki si ga izbere AP



Slika 34: Generiranje GTK

Diagram izmenjave varnostnih sporočil za izračun sejnega ključa:



Slika 35: Vzpostavitev varne povezave – 4 smerno rokovanje za vzpostavitev varnega ključa

Zaključek

Potreba po varni komunikaciji se je pojavila že v preteklosti. Varnost in integriteta podatkov je izjemnega pomena za vojsko, policijo, državo, itd... Veliko denarja se vlaga v razvoj algoritmov za šifriranje podatkov. Današnji algoritmi so že tako izpopolnjeni, da se jih praktično ne da razbiti.
Izkaže se, da je šibki člen varnosti protokol za izmenjavo ključev. Obstaja cela vrsta napadov na prenos sporočil med dvema entitetama. Napadalec je lahko pasiven in samo posluša, po drugi strani pa je lahko aktiven in spreminja sporočila v prenosu. Sporočila so najbolj občutljiva na napad, ko se prenašajo po prenosnem mediju. Problem nastane, ko se obe strani, ki želita vzpostaviti varno povezavo, ne poznata. Princip javnega ključa omogoča varno izmenjavo sporočil za vzpostavitev varne povezave. Javni ključ se lahko objavi in njegovo poznavanje napadalcu ne koristi. Večina današnjih protokolov za izmenjavo varnostnih ključev temelji na Diffie-Hellman protokolu. Za zaščito integritete sporočil se uporabljajo zgoščevalne funkcije. Proti napadom s ponavljanjem sporočila protokoli za izmenjavo ključev uporabljajo naključna števila in časovne žige. Za avtentikacijo se uporablja kombinacija zgoščevalne funkcije, osebnega certifikata in digitalnega podpisa.
Večina današnjega razvoja na področju varnosti je namenjena internetu in brezžičnim komunikacijam. Gonilo razvoja je zahteva uporabnikov po varnosti, kateri se je odzvala industrija. Osebni podatki so za uporabnika sveti in za zaupnost je pripravljen plačati.

Literatura

(1) Anrew S. Tenenbaum. 2003. Computer Networks

(2) Configuring IPSec/IKE, http://www.Securityfocus.com/infocus/1616/ 21.5.2006.

(3) RFC 2409, http://www.networksorcery.com/enp/rfc/rfc2409.txt. 8.5.2006.

(4) IPSec, http://www.ietf.org/html.chapters/ipsec-chapter.html. 8.5.2006

(5) 802.11i-2004, http://standards.ieee.org/getieee802/download/802.11i-2004.pdf. 6.5.2006

(6) The Cryptography of the IPSec and IKE Protocols, http://www.ee.technion.ac.il/~hugo/sigma.ppt. 1.5.2006

(7) Efficient, DoSResistant, Secure Key Exchange for Internet Protocols, http://www.crypto.com/papers/jfk-ccs.pdf

(8) Internet Key Exchange (IKEv2) protocol, http://community.roxen.com/developers/idocs/rfc/rfc4306.html

(9) Bluetooth. Security, http://www.niksula.hut.fi/~jiitv/bluesec.html. 28.4.2006

(10) Security Overview of Bluetooth, www.cosic.esat.kuleuven.be/publications/article-565.pdf 27.4.2006

(11) Building a secure wireless network, http://www.atheros.com/pt/atheros_security_whitepaper.pdf 29.4.2006

(12) Kerberos protocol, http://en.wikipedia.org/wiki/Kerberos_(protocol). 26.4.2006

(13) Cryptographic Origins – SIGMA

http://www.jabber.org/jeps/jep-0116.html#foundations. 20.4.2006