

DULRITUN

Sjálfstætt verkefni í Tölvuöryggi

Háskólinn í Reykjavík, Tölvunarfræðideild, Haustönn 2004.

Leiðbeinendur: Dr. Gísli Hjálmtýsson og Björn Brynjúlfsson.

Nemandi: Sigurjón Sveinsson.

1 Inngangur

Dulritun (e. encryption) er oftast lýst sem brenglun á texta eða gögnum þannig að þau verða óskiljanleg nema fyrir þá aðila sem hafa svokallaða lykla til að afbrensla gögnin aftur í sitt upprunalega horf.

Eina af fyrstu aðferðum við dulritun má rekja allt til tíma Juliusar Cesars (100-44 BC), svokallað Cesars dulmál. Þar var stöfum hliðrað til í stafrófinu og einungis sendandi og móttakandi vissu hversu mikil hliðrunin var [3 bls. 12]. Þrátt fyrir að dulritun hafi seinna verið notuð, til dæmis í Fyrri heimsstyrjöldinni, var það ekki fyrr en í Seinni heimsstyrjöldinni að fram komu dulritunarvélur af miklum gæðum. Þessar vélar voru hin þýska *Enigma* og hin japanska *Purple Machine*. [4]

Frá lokum Síðari heimsstyrjaldarinnar hafa komið fram nokkrar aðferðir til dulritunar. Má þar nefna DES, RSA, IDEA, PGP og RC5.

Öryggi í gagnasendingum, vistun gagna og auðkenning byggist yfirleitt á dulritun. Helstu aðferðir við dulritun gagna eru dulritun með leynilykli, dulritun með dreifilykli og tætiföll. Þessar aðferðir eru notaðar til mismunandi verka og eru allar sterkar á mismunandi sviðum dulritunar. Aðalmunurinn á leynilykla- og dreifilykla aðferðafræðinni er að í leynilykli er notaður einn lykill til að dulrita og dulráða (e. decrypt) gögnin, en í dreifilykli eru tveir lykilar notaðir, einn til að dulrita og annar til að dulráða. Í tætiföllum er notaður einn leynilykill.

2 Dulritun með leynilykli

Dulritun með leynilykli (e. secret key function) snýst um að dulrita gögn með einum lykli og þarf að nota sama lykilinn til að dulráða gögnin aftur. Helstu notkunarmöguleikar dulritunar með leynilykli eru [1 bls. 46]:

- Að senda gögn yfir óöruggan miðil eins og internetið.
- Vista gögn á ótryggum miðli.
- Auðkenning.
- Heilindi (e. integrity), að gögnum sé ekki breytt í sendingu milli aðila og að sá sem sendir gögn sé ábyggilega sá sem hann/hún segist vera.

Nokkrar aðferðir eru þekktar og má þar helst nefna DES (Data Encryption Standard), IDEA (International Data Encryption Algorithm) og AES (Advanced Encryption Standard).

2.1 Senda gögn yfir öruggan miðil

Í samskiptum milli tveggja aðila er alltaf sú hættu á að einhver komist að inn í samskiptin. M.a. er hægt að hlera síma, lesa póst áður en hann kemst á áfangastað og lesa tölvupóst sem er að fara milli staða.

Ef tveir aðilar nota sama dulritunarlykli er hægt að senda gögn á milli þeirra á öruggan hátt. Gögnin eru dulrituð með lyklinum áður en þau eru send og dulráðin með sama lykli á áfangastað. Þó svo að einhver sjái samskiptin þá eru þau samskipti dulrituð og því óskiljanlegur texti. Þetta er algengasta notkunin á dulritun.

2.2 Vista gögn á ótryggum miðli

Ef vista á gögn á stað sem aðrir geta haft aðgang að er hægt að segja að ekki sé hægt að tryggja að innihald gagnanna haldist leynt. Ef einhver kemst í gögnin er hægt að lesa þau í því formi sem þau eru. Til að tryggja leynd þeirra er hægt að dulrita skránar og gögnin með lykli sem eigandinn einn veit. Svo lengi sem lykillinn er ókunnur öðrum en eigandanum, og lykillinn er góður (nógu langur og ekki fyrirsjáanlegur), eru stjarnfræðilega litlar líkur á því að aðrir geti lesið gögnin.

2.3 Auðkenning

Í auðkenningu (e. authentication) fellst að sanna að maður sé sá sem maður segist vera. T.d. byggist aðgangur að tölvukerfum á því að auðkenning aðila sé tryggð. Sterk auðkenning (e. strong authentication) þýðir að einhver getur sannað vitneskju á leyndarmáli án þess þó að gefa það upp. Auðkenningar aðferðir eru yfirleitt á þann veg.

2.4 Heilindi

Til að tryggja að rétt gögn komist á leiðarenda þarf að athuga hvort gögnin séu eins og þau voru þegar þau voru send. Það getur komið upp að einhver breyti gögnunum á leiðinni og sendi þau áfram. Ef engar varúðarráðstafanir eru teknar mun móttakandi líta á breyttu gögnin sem þau upprunalegu.

Hægt er að sjá hvort að gögn hafi breyst á leiðinni með því að reikna prófsummu (e. checksum) af gögnunum með leynilykli og aðferðafræði og senda þá tölu með. Á áfangastað er prófsumman reiknuð með gögnunum og sama leynilykli. Ef talan úr þeim útreikningi er ekki sú sama og kom með gögnunum þýðir það að gögnin hafa breyst á leiðinni, hvort sem um ásetning er að ræða eða ekki.

2.5 DES (Data Encryption Standard)

DES dulritunarstaðallinn var þróaður af IBM og birtur árið 1977 af Staðlastofnun Bandaríkjanna (e. National Bureau of Standards). DES notar 56 bita lykil, tekur 64 bita innlag og sendir frá sér 64 bita frágag. Lykillinn virðist reyndar vera 64 bita, en einn af hverjum bitum er notaður fyrir bitavillutékk (e. odd parity check) og því er lykillinn sjálfur í raun ekki nema 56 ($8 \cdot 7$) bitar.

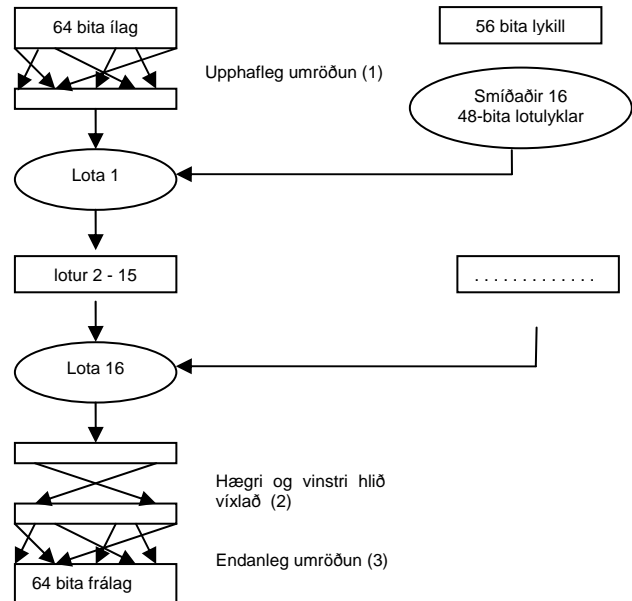
DES dulritun hentar vel fyrir vélbúnaðarútfærslur en síður fyrir dulritun í hugbúnaði og stafar það af umröðun á bitum við upphaf og enda dulritunar. Milli umraðananna fara fram 16 lotur þar sem hin raunverulega dulritun fer fram.

2.5.1 Aðferðafræði DES við dulritun.

Orð með tölustaf ⁽¹⁾ er að vísa í sama tölustaf í teikningu. Textinn á því við þann stað í teikningunni.

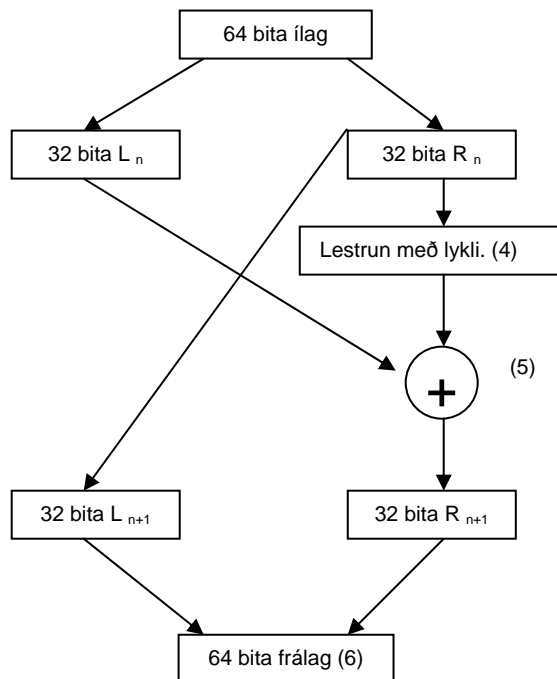
Í upphafi er 64 bita ílagið tekið og bitunum er umræðað með fyrirfram ákveðnu mynstri, ekki tilviljunarkenndu⁽¹⁾. Þannig er hverjum bita varpað í annað sæti innan 64 bita og hver vörpun er "one-to-one", sem þýðir að ef, sem dæmi, búið er að varpa bita 5 í sæti 62 þá getur enginn annar biti komið í sæti 62.

Þegar bitunum hefur verið stokkað upp eru keyrðar lotubundnar dulritunaraðgerðir 16 sinnum (sjá mynd og útskýringu). Þegar þessar lotur hafa keyrt á enda er ílaginu skipt í tvo 32 bita hluta og þeim víxlað⁽²⁾ og skeytt saman og þeim stokkað aftur upp⁽³⁾ og er þá komið endanlegt dulritað frágag.



2.5.2 DES lota

Í hverri lotu er ílaginu skipt upp í tvo 32 bita hluta, köllum þá L og R . L_{n+1} er í raun R_n en L_{n+1} er afurð n.k. lestrunar⁽⁴⁾ (e. mangler function) á R_n og XOR⁽⁵⁾ aðgerðar með L_n . R_{n+1} og L_{n+1} er svo skeytt saman og er sú útkoma þá frágag þeirrar lotu⁽⁶⁾.



2.5.3 Lestrunaraðgerð (e. mangler function)

R_n og lykillinn K_n eru notaðir í þessari lestrun. R_n (32 bita) er skipt í átta 4 bita einingar. Við hverja einingu er skeytt þeim bita af næstu einingum sem liggur næst henni og verður þá hver eining 6 bitar við þetta. K_n er einnig skipt upp í átta einingar en hver þeirra er 6 bitar. Hverri einingu af K_n og R_n er síðan XOR-að saman og frágag þeirrar aðgerðar er 4 bita eining, E_n . Allar E einingarnar eru síðan settar saman og er það frágag lestrunaraðgerðarinnar [1 bls. 68, 2 bls. 272].

2.6 Margföld dulritun með DES

Eins og kom fram hér að framan þá er lykill DES ekki nema 56 bita langur og því margfalt veikari en t.d. IDEA sem hefur 128 bita lykil. Til að auka öryggi DES og bæta fyrir þann veikleika sem felst í þessum stutta lykli hefur verið þróuð aðferðafræði við að dulrita gögn nokkrum sinnum er sú aðferð kölluð EDE (encryption-decryption-encryption).

Til verksins þarf tvo lykila, K1 og K2. Gögnin eru dulrituð einu sinni með lykli K1, dulráðin með lykli K2 og svo síðast dulrituð með lykli K1. Til að dulráða gögnin er dulritunin keyrð aftur-á-bak og því er aðgerðin í raun DED (decryption-encryption-decryption).

En af hverju að dulrita-dulráða-dulrita og það með tveimur lykllum? Af hverju er t.d. ekki nóg að dulrita tvisvar með sama lyklinum? Ástæða þess er að við það að dulrita með sama lyklinum tvisvar tvöfaldast einungis sá tími sem tekur að leita línulega að öllum mögulegum lykllum, sem eru 2^{56} möguleikar. [1 bls. 94, 2 bls. 359]

En ef dulritað er tvisvar með sittthvorum lyklinum? Það mætti halda að þá væri lykillinn orðinn nokkurs konar 112 bita lykill og línuleg leit að lyklinum því löng eftir því. En vandamálið er að það er til aðferðafræði sem finnur lykilinn með sömu afköstum og leit að 56 bita lykli og því er þessi aðferð ekki nógu góð.

2.7 IDEA (International Data Encryption Algorithm)

IDEA var þróað af Xuejia Lai og James L. Massey og var upphaflega kallað IPES (Improved Proposed Encryption Standard). IDEA var þróað með það í huga að það yrði hagkvæmt að útfæra í hugbúnaði, öfugt við DES sem er þungt í vöfum þegar það er útfært í hugbúnaði.

Þó svo að IDEA sé vel þekkt og aðferðafræðin kunn og aðgengileg öllum sem vilja kynna sér hana, þá hefur engin aðferð enn verið birt til að brjóta þessa dulritun, önnur en að reyna alla mögulega lykila. Sú aðgerð krefst gífurlegs tölvubúnaðar og mikils tíma.

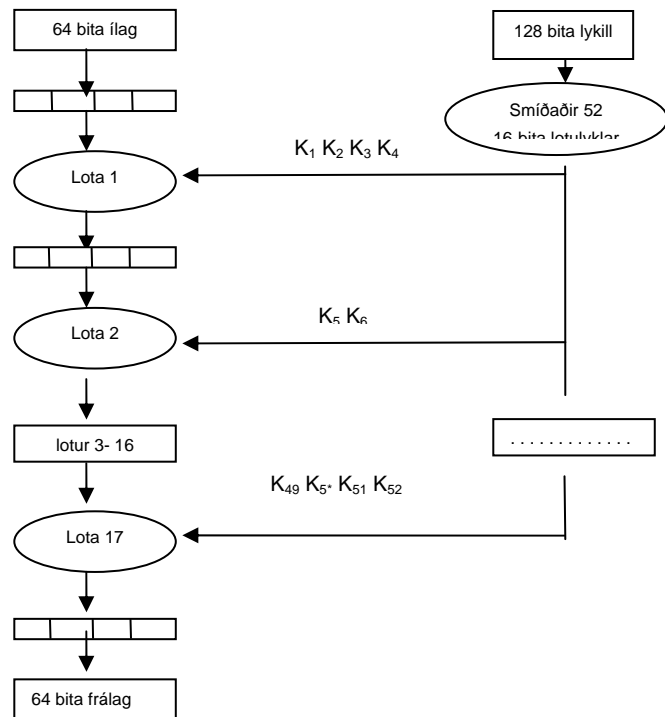
IDEA er að mörgu leyti líkt DES. Báðar aðferðinar vinna í lotum og báðar nota flóknar lestrunaraðgerðir sem þurfa ekki að virka í öfugri röð til að þær geti dulráðið gögn.

2.7.1 Aðferðafræði IDEA við dulritun

IDEA hefur 64 bita ílag og 64 bita frálag. Ílaginu er skipt upp í fjóra 16 bita hluta. Dulritunin fer fram í 17 lotum og er umismunandi aðgerðir framkvæmdar í oddatölulotum og slétttölulotum [1 bls. 75].

Lykillinn er 128 bita langur. Við dulritunina eru búnir til 52 16 bita lotulyklar. Lotulyklunum er svo beitt í hverri lotu og er hver þeirra einungis notaður einu sinni. Lotulyklar sem notaðir eru eru tveir í lotu sem ber sléttatölunúmer og fjórir í lotu sem ber oddatölunúmer. Þannig fara fjórir lotulyklar í lotu nr. 1, tveir í lotu nr. 2 o.s.frv.

Þessir lotulyklar eru gerðir þannig að 128 bita lyklinum er skipt upp í átta 16 bita lykla. Síðan er farið að bita 25 og haldið áfram að smíða 8 lykla í viðbót. Þegar komið er að bita 128 er byrjað aftur á bita 1. Við næstu 8 bita er aftur hliðrað til um 25 bita til að byrja og svona haldið áfram þar til 52 lotulyklar hafa verið smíðaðir [1].



2.8 Hvernig á að dulrita stór gögn?

Gögn sem þarf að dulrita eru ekki alltaf 64 bita stór heldur eru þau yfirleitt stærri. Það eru til aðferðir sem nota dulritun með leynilykli sem aðferðafræði.

2.8.1 ECB (e. Electronic Code Book)

Gögnunum er skipt upp í 64 bita blokkir og hver um sig er dulrituð. Þessi aðferð er mjög gölluð að því að leyti að það er hægt að víxla blokkum á leiðinni án þess að þess verði vart. Einnig geta tvær eða fleiri dulritaðar blokkir verið alveg eins og getur það gefið upplýsingar um innihald gagnanna. Vegna þessara galla er ECB nær aldrei notað [1 bls. 80].

2.8.2 CBC (e. Cipher Block Chaining)

Til að losna við galla ECB er búið til 64 bita tilviljunarkennd tala sem er XOR-uð við ódulritaðan textann (e. plain text) og er tilviljunarkennda talan send með. En til að fækka þessum tilviljunarkenndu tölum eru dulritaðar blokkir notaðar sem tilviljunarkennd tala [1 bls. 84]. Þannig að fyrir ódulritaða blokk i er dulrituð blokk i-1 notuð sem tilviljunarkennd tala. Með þessu er komið í veg fyrir að blokkir sem innihalda sama ódulritaða textann séu eins eftir dulritun auk þess sem komið er í veg fyrir að gögnin tvöfaldist við að búa til tilviljunarkennda tölu fyrir hverja blokk og senda með.

2.8.3 OFB (e. Output Feedback Mode)

Byrjað er með því að búa til nokkurs konar gervitviljunarkenndan streng. Strengurinn er búinn til með 64 bita tilviljunarkenndri tölu sem er kölluð IV. Sú tala er dulrituð með lyklinum til að búa til blokk b_0 . b_0 er síðan dulrituð til að búa til b_1 , b_1 er dulrituð til að búa til b_2 o.s.frv., eins oft og þurfa þykir. Til að dulrita gögn eru þau XOR-uð við eins marga bita af b_0 b_1 b_2 b_3 b_4 ... og þurfa þykir. Dulrituðu gögnin eru svo send áfram ásamt IV. Þar er sama ferlið framkvæmt með sama lykli og var notaður af sendanda.

Langur gervitviljunarkenndur strengur sem notaður er til að XOR-a við skilaboð til að dulrita þau er svokallaður "one-time-pad" [1 bls. 86, 3 bls. 16].

3 Dulritun með dreifilykli

Til eru nokkrar aðferðir við að dulrita með svokölluðum dreifilykli (e. public key) og eru aðferðirnar nokkuð ólíkar sín á milli. En nokkrir eiginleikar eru sameiginlegir. Aðferðirnar byggjast á tveimur stærðfræðilega skyldum lykllum. Annar er öllum aðgengilegur og er kallaður dreifilykill. Hinn er kallaður einkalykill og er leynilegur, s.s. er ekki dreift. Gögn eru dulrituð með öðrum lyklinum og ekki er hægt að dulráða gögnin aftur með sama lykli. Einungis skyldi lykillinn getur ráðið gögnin [1 bls. 129].

Nytsemi hvernar aðferðafræði fyrir sig er mismunandi.

- Að senda gögn yfir óöruggan miðil eins og internetið (RSA).
- Stafrænar undirskriftir (El Gamal, RSA og DSS)
- Auðkenning (zero knowledge proof systems)

3.1 RSA

RSA er nefnt eftir höfundum sínum, Ronald L. Rivest, Adi Shamir og Leonard M. Adleman. RSA er dreifilyklaaðferð sem dulritar og dulræður. Lyklalengd er breytileg og hægt er að velja langan lykil til að auka öryggi eða styttri lykil til að auka afköst og hraða. Algengasta lengd á lykli er 512 bitar [1 bls. 134].

Stærð blokkarinnar (gögnum er skipt í blokkir) sem er dulrituð verður að vera minni en lykillinn en þegar dulritun fer fram er blokkir orðin jafnlöng og lykillinn.

Virgni RSA í dulritun og dulráðningu er mun hægari en leynilyklaaðferðir á borð við DES og IDEA. Þetta gerir RSA mjög óhagkvæma sem aðferð þegar verið er að dulrita mjög stór gögn. Yfirleitt er RSA notað til að dulrita leynilykil til að senda milli aðila og svo er leynilykillinn notaður í þau samskipti sem á eftir koma.

3.1.1 Aðferðafræði RSA við dulritun

Til að búa til lykila í RSA þarf að finna tvær stórar prímtölur p og q . Þær eru margfaldaðar saman og útkoman er n . Tölurnar tvær p og q eru áfram leynilegar.

Til að búa til dreifilykilinn þarf að velja tölu sem er ósambátta (e. relatively prime) $\phi(n)$ [1] köllum hana e . Þar sem við vitum p og q þá vitum við $\phi(n)$ (sem er talan e); $(p-1)*(q-1)$. Dreifilykillinn er því $\langle e, n \rangle$ [1 bls 140, 2 bls. 467].

Til að búa til einkalykilinn er fundin talan d sem er margföldunarumhverfa (e. multiplicative inverse) við e modulus $\phi(n)$. $\langle d, n \rangle$ er þá einkalykilinn.

Til að dulrita gögn m með dreifilykli þarf að búa til dulritaðan texta (e. ciphertext) c sem er $m^e \bmod n$. Einungis er hægt að dulræða þennan dulritaða texta með samstæðum einkalykli $c^d \bmod n$.

Auk þess getur handhafi einkalykils búið til stafræna undirskrift $s = m^d \bmod n$ fyrir skilaboð m en hver sem er getur staðfest þessa undirskrift með því að athuga hvort $m = s^e \bmod n$.

3.1.2 Af hverju er RSA öruggt?

Það er ekki hægt að sanna að RSA sé öruggt. Það er aðeins hægt að treysta á undirstöðukenninguna í dulritun (e. Fundamental tenet of cryptography) sem er þannig að fullt af mjög gáfuðu fólki hefur reynt að brjóta RSA og ekki tekist það ennþá. Aðal galdurinn við RSA er sú ályktun að það sé mjög erfitt að þátta stórar tölur og bestu aðferðirnar við þáttun stórra talna eru frekar hægar. Til að þátta tölu sem er 512 bitar þarf um 30,000 MIPS ár [1 bls. 135]. MIPS ár er sá fjöldi aðgerða sem örgjörvi, sem framkvæmir eina milljón aðgerða á sekúndu, framkvæmir á ári.

3.2 Diffie-Hellman

Diffie-Hellman (hér eftir kallað DH) dreifilykla aðferðin er eldri en RSA og er reyndar elsta dreifilyklakerfið í notkun í dag. DH hvorki dulritar né býr til stafrænar undirskriftir og hefur því takmarkaðri nytsemi en RSA. En DH sinnir aftur á móti sínu hlutverki með meiri afköstum en önnur kerfi [1 bls. 147].

Það sem DH gerir er að leyfa tveimur aðilum að fallast á sameiginlegan lykil, sem enginn annar veit, þó svo að samskipti þeirra um lykilinn fari fram á ótryggum miðli og allir geti hlustað á samskiptin. Með þessu móti geta tveir aðilar, eftir að hafa komið sér saman um leynilykil eins og t.d. DES, hafið dulrituð samskipti [2 bls. 514].

3.2.1 Aðferðafræði Diffie-Hellman

Í upphafi eru tvær tölur, p og g , sem þurfa ekki að vera leyndarmál og geta þess vegna verið kunnugar öllum. Talan p er stór prímtala, um það bil 512 bitar, og talan g er minni en p . Þegar báðir aðilar, A og B, hafa komist að samkomulagi um sama p og g velja báðir aðilar sitt hvora tilviljunarkenndu töluna S_A og S_B og halda þeim leyndum. Síðan reikna A og B, hvor fyrir sig, $g^{S_A} \bmod p$. Þannig fær A út töluna T_A og B fær T_B .

- A og B velja sér tölu af handahófi, S_A og S_B
- A reiknar út $T_A \bmod p$ og B reiknar út $T_B \bmod p$.
- A og B skiptast á T -tölum.
- A og B bera T -tölnar saman við sína S tölu.

- A reiknar út $T_B^{S_A}$ mod p og B reiknar út $T_A^{S_B}$ mod p
- A og B munu fá sömu tölu.

Sem þýðir að A og B hafa skipst á leyndarmáli án þess þó nokkurn tíma að hafa gefið það upp og enginn annar getur komist að því.

3.3 DSS

NIST (e. National Institute of Standards and Technology) hefur komið fram með aðferð fyrir stafrænar undirskriftir. Aðferðin heitir DSA (e. Digital Signature Algorithm) en sem staðall er hún kölluð DSS (e. Digital Signature Standard).

DSS sinnir tveimur mikilvægum hlutverkum samtímis. Skjal, sem er undirritað með einkalykli og hvers heilindi eru staðfest með notkun hliðstæðs dreifilykils, er hægt að líta á sem öruggt. Öruggt að því leyti að því hefur ekki verið breytt síðan það var undirritað og að sá sem á undirskriftina er örugglega sá sem undirritaði skjalið. Með þessari vitneskju er hægt að slá föstu að innihaldið sé rétt og að höfundur/eigandi/sendandi sé virkilega sá sem hann/hún segist vera. Það má því segja að með stafrænni undirskrift sé skjalið einnig orðið óhrekjanlegt því sendandinn getur heldur ekki þrætt fyrir það að hafa sent skjalið þar sem líkurnar á því að einhver annar hafi skrifað undir það eru stjarnfræðilegar, þ.e. ef einkalykill sendandans er ekki þekktur öðrum en honum sjálfum.

DDS er nokkuð hraðvirk við að búa til undirskrift vegna þess að þeir útreikningar sem fara fram eru gerðir á 160 bita prímtölu í stað 512 bita. Að reikna "modulus" af 160 bita tölu er þrisvar sinnum hraðvirkara en 512 bita prímtölu. Í staðinn er DSS hægvirkara þegar staðfesta þarf undirskriftina hvort sem það er gert af þeim sem bjó hana til eða af viðtakanda gagnanna [1 bls. 152].

3.3.1 Af hverju er DSS talið öruggt?

Þó svo að DSS staðallinn sé öllum kunnur sem vilja þekkja hann hefur engum tekist að brjóta hann, þ.e. komast að því hver einkalykillinn að gögnunum er án þess að þekkja hann. Þar að auki hefur hin bandaríska NSA (e. National Security Agency), almennt taldir bestu aðilar í dulritun í heimi, gefið staðlinum blessun sína.

3.3.2 Ein tala fyrir hverja sendingu

Fyrir hverja sendingu og hverja stafrænu undirskrift þarf sendandinn að koma með nýja leynilega tölu. Ef hann gerir það ekki er hætt á að einhver komist að því hver einkalykill sendandans er.

3.4 Zero knowledge proof systems

Zero knowledge proof systems (hér eftir kölluð ZKPS) sinna einungis auðkenningu. Þau hjálpa aðilum með að auðkenna sig með leyndarmáli, hvort sem það er lykilorð eða önnur aðferð, án þess nokkurn tíma að gefa upp það leyndarmál. RSA getur sinnt þessu hlutverki einnig með því að sjá samhengið milli leynilykils og dreifilykils en ZKPS gera þetta með mun betri afköstum án þess þó að geta dulritað né búið til stafrænar undirskriftir.

4 Tætiföll

Tætiföll (e. hash function) eru einstefnuföll og eru skilgreind sem föll því þau taka ílag (færibreytu) og skila frálagi (skilagildi). Litið er á tætiföll sem einstefnuföll því að það er ekki hægt að reikna út ílagið með því að setja frálagið í fallið.

Til þess að hægt sé að líta á tætiföll sem örugg í dulritun verða þau að vera þannig gerð mjög ólíklegt sé að hægt sé að finna gögn sem skila sama frálaginu og einhver önnur en, einnig að það sé ekki hægt að finna tvö skilaboð sem skila sama frálagi úr tætifalli.

Það eru til nokkrir tætifallastaðlar. NIST hefur gefið út staðal fyrir tætifall sem heitir SHS (e. Secure Hash Standard). Einnig eru til MD2, MD4 og MD5. Allar þessar aðferðir gera í raun það sama, taka gögn sem ekki eru háð neinni lengd, framkvæma útreikninga á þeim gögnum og enda með streng af ákveðinni lengd. Sá strengur er yfirleitt 128 bitar (MD2-5) eða lengri eins og SHS sem skilar 160 bita frálagi [1 bls. 101].

4.1 Nytsemi tætifalla

Það er hægt að gera margt nýtsamlegt með tætiföllum. Mörg tætiföll nota leynilykil til að reikna frágag. En þó svo að leynilykill sé notaður er ekki um dulritun með leynilykli að ræða því að tætiföll virka bara í eina átt og ekki er hægt að reikna með tætifalli ódulritaða textann frá dulritaða textanum með aðstoð leynilykilsins. Ef það væri hægt væri um dulritun að ræða.

4.1.1 Auðkenning

Tveir aðilar sem eiga saman eitthvað leyndarmál geta sannreynt auðkenni hvors annars. A sendir B skilaboð. B setur saman skilaboðin frá A og leyndarmálið sem A og B deila og setur það allt saman í tætifall. Útkomuna sendir B til A. A framkvæmir nákvæmlega sama útreikning og B og ef frálagið frá A er það sama og frá B þá veit A að B var sá sem svaraði og að B svaraði rétt [1 bls. 107].

4.1.2 Reikna út heilindakóða gagna.

Til að tryggja heilindi gagna er hægt að reikna út svokallaðan heilindakóða þeirra (e. message integrity code). Það sem þessi kóði gerir í raun er að segja til um hvort gögnin séu þau sömu og þau voru þegar kóðinn var reiknaður.

Segjum sem svo að A vilji senda B skilaboð og að A vilji að B geti sannreynt það að skilaboðunum hafi ekki verið breytt á leiðinni. A skrifar skilaboðin og með tætifalli og leynilykli, sem A og B einir vita, reiknar A út frágag. Þetta frágag sendir A með skilaboðunum þegar hann sendir þau til B. B tekur síðan skilaboðin og lykilinn og reiknar út frágag með tætifalli. Ef frágag B er það sama og frágag A þá hafa skilaboðin komist í gegn án þess að vera breytt. B getur líka verið viss um að það var A sem sendi skilaboðin því til þess að C geti sent skilaboð sem B þarf C að hafa sama einkalykil og A og B deila.

4.1.3 Dulritun með tætifalli

Það er ekki hægt að beita tætifalli til að dulræða gögn en það er hægt að búa til svokallaða einnota blokk (e. one-time-pad) með tætifalli og svo nota blokkina til að dulrita og dulræða gögn.

Þessi aðgerð býr til gervitilviljunarkenndan streng (sjá OFB á bls. 6). Til þess að búa til strenginn þarf í upphafi leynilykil. Leynilykillinn er notaður til að búa til fyrsta fralagið B_1 . B_1 er sett sem leynilykill í tætifallið til að fá B_2 , B_3 sett í tætifallið til að fá B_4 og svo koll af kolli þar til kominn er strengur samsettur af öllum B_i nógu langur til að geta XOR-a við skilaboðin. Báðir aðilar framkvæma þessa aðgerð á sama hátt. A býr til streng S_i sem og B. A XOR-ar skilaboðin með strengnum og sendir yfir til B, sem aftur XOR-ar skilaboðin til að fá upphaflega textann.

Það er ekki öruggt að nota sama S_i strenginn tvisvar og því þarf að búa til IV á sama hátt og var gert í OFB og senda til móttakanda áður en skilaboðin eru dulrituð svo að móttakandi geti búið til sinn S_i streng áður en skilaboðin koma.

5 Heimildaskrá.

- [1] Charlie Kaufman, Radia Perlman og Mike Speciner, 1995. *Network Security: Private Communication in a Public World*.
- [2] Bruce Schneier, 1996. *Applied Cryptography, Second Edition*.
- [3] Brett C. Tjaden. 2004. *Fundamentals of secure computer systems*.
- [4] Shireen Hebert, 2001, *A Brief History of Cryptography*.
<http://cybercrimes.net/Cryptography/Articles/Hebert.html>