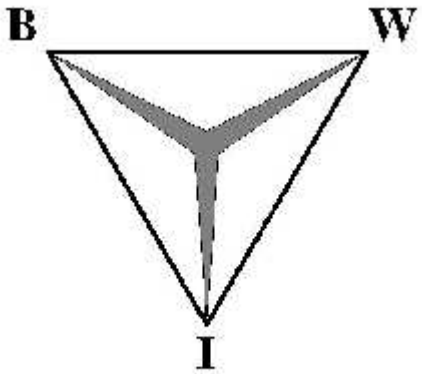
PGP:

Pretty Good Privacy.

Een overzicht.

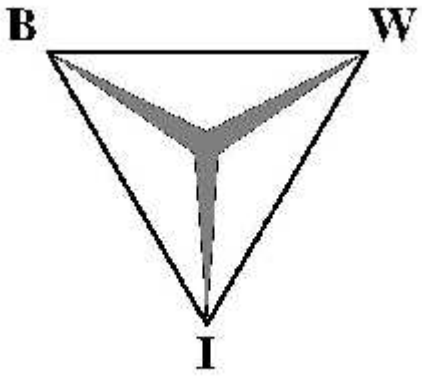


BWI-werkstuk geschreven door:
Aart Valkhof
Maart 2003

PGP:

Pretty Good Privacy.

Een overzicht.



De *vrije* Universiteit
Faculteit der Wiskunde en Informatica
Studierichting Bedrijfswiskunde & Informatica
De Boelelaan 1081a
1081 HV Amsterdam

1 Voorwoord

Een onderdeel van het curriculum van mijn studie Bedrijfswiskunde en Informatica – BWI- aan de Vrije Universiteit Amsterdam is het schrijven van een werkstuk. Het doel van het werkstuk is het beschrijven van een probleem op een heldere manier voor een deskundige manager. Daarbij mag worden uit gegaan dat de deskundige manager een algemene kennis bezit over het onderwerp.

Het onderwerp van mijn werkstuk is PGP. PGP staat voor Pretty Good Privacy en is een programma waarmee digitale opslag en communicatie van gegevens kunnen worden beveiligd. Met behulp van PGP kunnen e-mail en bestanden versleuteld worden en kunnen digitale handtekeningen meegegeven worden.

Het doel van mijn werkstuk is om een heldere beschrijving te geven van PGP zodat de basisprincipes bekend zijn. Het werkstuk is in eerste instantie geschreven voor dr. E. Wattel, die als de eerder genoemde deskundige manager en als begeleider optreedt. Hij geeft onder meer het vak Coderingstheorie en Complexiteit en mag gezien worden als deskundige in het vakgebied waarin PGP opereert. De opdracht voor dit werkstuk luidt als volgt: “Schrijf een document waar een docent op terug kan vallen wanneer in een college PGP behandeld wordt.”.

Toen ik met dit werkstuk begon, stond me goed voor ogen wat me te doen stond. Hierdoor kon ik efficiënt, structureel en doelgericht werken. Heel belangrijk was het om het onderwerp goed af te bakenen. Er is veel over PGP te vinden in de bibliotheek en op het Internet. Als het onderwerp niet goed afgebakend wordt, dan loopt het al snel uit de hand. Hierbij wil ik Dr. E Wattel bedanken voor de begeleiding die hij mij heeft gegeven. Ik wens de lezer even veel plezier toe bij het lezen van dit werkstuk als dat ik had bij het schrijven.

Aart Valkhof.

Den Haag, maart 2003.

2 Samenvatting

PGP is software waarmee e-mail en bestanden beveiligd kunnen worden. PGP is begin jaren negentig ontstaan. Phil Zimmermann was de drijvende kracht hierachter. In de loop van de jaren zijn er verschillende versies verschenen, zowel commerciële als gratis versies. Op het juridische vlak heeft PGP een roerige geschiedenis gehad. Zo heeft PGP patenten geschonden. De FBI heeft onderzoek naar Zimmermann gedaan, omdat export bepalingen in het geding waren. Het onderzoek heeft nooit wat opgeleverd en de patenten zijn sinds 1996 op orde.

PGP gebruikt twee soorten algoritmes: asymmetrische en symmetrische systemen. De asymmetrische systemen die gebruikt worden zijn Diffie-Hellman en RSA en de symmetrische systemen zijn CAST, AES, IDEA, 3DES en TWOFISH. Er wordt ook een hash functie toegepast, namelijk MD5.

De belangrijkste functionaliteit van PGP is het vercijferen en ontcijferen van e-mail. PGP gebruikt hiervoor zowel asymmetrische als symmetrische systemen. Hierdoor combineert het de sterke punten van de twee en vermijdt het de zwakke punten. Het vercijferen en ontcijferen van bestanden werkt nagenoeg hetzelfde als bij e-mail. De digitale handtekening kan gebruikt worden om data te verifiëren.

De algoritmes die PGP gebruikt zijn stuk voor stuk sterk. Daar zal de veiligheid niet in het geding mee zijn. Een grotere bedreiging vormen de bugs. Zoals ieder groot programma bevat ook PGP bugs. Deze gaan ten koste van de veiligheid. PGP doet zijn naam eer aan. Het is een vrij goed programma om de privacy mee te beschermen, niet om uiterst gevoelige informatie mee te beveiligen.

Sleutelwoorden: PGP, Informatiebeveiliging, Cryptografie, RSA, Zimmermann

Inhoud

3 Inleiding

Tegenwoordig gebruikt bijna iedereen e-mail. Maar niemand versleutelt zijn mail. In de regel vindt ment dat niet nodig. Vanwege de kosten en omslachtigheid zijn cryptografische producten alleen geschikt voor militaire instanties, overheidsinstellingen, diplomatieke diensten, criminelen en heel paranoïde individuen. Phil Zimmermann vond dat ieder recht had op zijn privacy, ook als het een e-mail betreft. Hij vond dat iedere burger toegang moet hebben tot software die zijn privacy beschermt. Hij besloot een crypto product te maken dat makkelijk was te gebruiken en niet al te duur was. Dat werd dus PGP.

In dit werkstuk zal ik zijn programma beschrijven. Dat doe ik op verschillende manieren. In hoofdstuk 4 beschrijf ik de historie van PGP. Ik doe dit niet in een chronologische volgorde, maar naar onderwerp gegroepeerd. In hoofdstuk 5 bespreek ik de algoritmes die in PGP gebruikt worden. Hoofdstuk 6 gaat over het vercijferen en ontcijferen van e-mail en bestanden. Vervolgens wordt in hoofdstuk 7 dieper ingegaan op de digitale handtekening. Hoofdstuk 8 gebruik ik om mijn conclusies over PGP op te maken. De conclusies zijn puur gebaseerd op literatuur onderzoek. Tenslotte komen de literatuurlijst, een begrippenlijst en afkortingen aan bod.

In deze alinea zal ik het onderwerp afbakenen. In dit werkstuk wordt geen algemene introductie in cryptografie gegeven. Ik ben er van uitgegaan dat de doelgroep al enige kennis op dit terrein bezit. In PGP worden verscheidene cryptografische algoritmes gebruikt. Ik zal alleen globaal op deze protocollen in gaan. De source code zal niet bekeken worden. Er zullen geen besturingssysteem afhankelijke aspecten aan bod komen ondanks dat PGP een programma is dat op verschillende besturingssystemen – van Linux tot Windows - functioneert. PGP heeft nogal een geschiedenis op het juridische vlak. Een lange tijd was PGP illegaal omdat patenten en export bepalingen geschonden werden. Deze ‘issues’ zullen beperkt aan bod komen. Ik beëindig deze alinea door aan te geven uit welk oogpunt ik het werkstuk geschreven heb. Ik heb het oogpunt van de manager gebruikt en niet het oogpunt van een programmeur of systeembeheerder. Of zoals dr. E. Wattel het uitdrukt: “Het gaat erom wat je er mee kunt en niet hoe je het kunt.”. Met andere woorden: technische details zullen achterwege blijven.

Op het internet zwerven verschillende versies van PGP rond. Ik ben van de meest up-to-date versie uitgegaan die door de PGP Corporation is uitgebracht, namelijk versie 8.0. Een freeware versie is te downloaden op www.pgp.com. Het is niet te vermijden en zelfs gewenst dat andere versies in dit werkstuk aan bod komen.

4 Historie

De geschiedenis van PGP gaat terug tot het begin van de jaren negentig. Deze begon in 1991 met versie 1.0. Nu in 2003 zitten we op versie 8.0 en er liggen nog vele versies in het verschiet. Ik heb geen chronologische opsomming van de oplopende versies gegeven. Deze is al uitgebreid beschreven in bijvoorbeeld [4] en [5]. In plaats daarvan probeer ik aan de hand van een paar onderwerpen de historie van PGP samen te vatten.

4.1 Het ontstaan van PGP

De man achter PGP is Phil Zimmermann. In juni 1991 bracht hij versie 1.0 uit. Deze versie kwam alleen uit buiten de VS vanwege het RSA patent, dat alleen binnen de VS geldig was. De eerste versie maakte gebruik van de protocollen RSA en Bass-o-Matic. Het laatste is een symmetrisch algoritme dat door Zimmermann zelf ontwikkeld was. Zoals in [4] beschreven staat is een goed algoritme schrijven absoluut niet gemakkelijk en alleen voorbestemd aan de grote cryptografen op aarde. Bass-o-Matic voldeed dan ook niet en werd in versie 2.0 vervangen door IDEA. De tweede versie was wereldwijd ontwikkeld door vele programmeurs. Na versie 1.0 schreef Zimmermann geen regel code meer voor PGP, maar bleef wel altijd betrokken.

4.2 De commerciële versie

Volgens [4] bracht ViaCrypt de eerste commerciële versie uit in augustus 1993, namelijk versie 2.4. ViaCrypt was een bedrijfje dat al een RSA licentie had voordat het in zee ging met Zimmermann. Hiermee kwam een einde aan de illegale status van PGP. ViaCrypt en later PGP Incorporated zouden hierna nog enkele commerciële versies uitbrengen. In 1997 nam NAI – Network Associates Incorporated – PGP Incorporated over en bracht versie 5.5.5 op de markt. De huidige commerciële versies worden door PGP Corporation uitgebracht.

4.3 De gratis versie

In [3] is te vinden dat de eerste legale gratis versie verscheen in 1994 onder de autorisatie van het MIT – Massachusetts Institute of Technology. Het MIT is houder van het PGP patent. Deze versie maakte gebruik van een RSA library die gratis was te gebruiken voor wetenschappelijke doeleinden. Deze zogenaamde freeware versies worden gebruikt om het gebruik en ontwikkeling van RSA en PGP te stimuleren. Er is ook een GNU versie van PGP, GNU Privacy Guard –GPG- genaamd. GNU is een beweging, die gratis software ontwikkelt en uitbrengt. Sinds 1999 brengen zij versies uit van PGP die geheel gratis te gebruiken zijn.

4.4 Juridische kwesties

[3] geeft aan dat de eerste versies van PGP gebruik maakte van het RSA algoritme De patenthouder van RSA, RSA Data Security Inc, had hier echter geen toestemming voor gegeven. Dit leverde PGP een illegale status op. Latere versies maakten gebruik van de RSAREF –RSA Reference- library. Deze was gratis te gebruiken en was speciaal uitgegeven voor non-commerciële producten. Bij versie 5.0 eindigde het ondergrondse bestaan van PGP. Vandaag de dag zijn er zowel gratis als commerciële versies beschikbaar, die te gebruiken zijn zonder dat daarbij patenten geschonden worden.

In 1993 begon de FBI een onderzoek naar Phil Zimmermann. Het was volgens de wet verboden om crypto producten te exporteren zoals in [4] te lezen staat. Versie 1.0 werd alleen buiten de VS uitgebracht en moest dus geëxporteerd zijn. De FBI verdacht Zimmermann ervan dat hij PGP via het Internet verspreid had. Het onderzoek zou drie jaar duren en nooit enig bewijsmateriaal tegen Zimmermann opleveren.

Ondanks al het juridische getouwtrek leverde het Zimmermann nooit een veroordeling op. PGP kreeg wel een geuzen naam: Guerrillaware. In 1996 hield Zimmermann een toespraak voor een commissie van het Amerikaanse Congres over cryptografie. Met de speech rehabiliteerde Zimmermann zich. De speech is terug te vinden op [10].

5 Algoritmes

Deze paragraaf beschrijft de voornaamste cryptografische algoritmes die PGP gebruikt. Er wordt onderscheid gemaakt tussen twee soorten systemen: asymmetrische systemen en symmetrische systemen.

5.1 Asymmetrische systemen

Bij een asymmetrisch systeem wordt voor het ontcijferen een andere sleutel gebruikt dan voor het vercijferen. In 1976 introduceerden Diffie en Hellman dit begrip in de wereld van de cryptografie. Ze ontwikkelden een asymmetrisch systeem dat ook in PGP gebruikt wordt. Omdat PGP het RSA protocol vanaf versie 1.0 al gebruikt heb ik dit hier beschreven.

5.1.1 RSA

Dit asymmetrische systeem werd in 1977 gepubliceerd door Ronald Rivest, Adi Shamir en Leonard Adleman. Het is gebaseerd op het idee dat het gemakkelijk is om twee priemgetallen met elkaar te vermenigvuldigen, maar dat de weg andersom –factoriseren- moeilijk zo niet onmogelijk is. RSA staat uitvoerig beschreven in [4]. De algoritme heeft twee willekeurige priemen P en Q nodig. De vercijfer modulus N is gelijk aan het product van P en Q. Daarnaast is ook een vercijfer sleutel E nodig. E is een getal dat groter is dan drie en kleiner (P-1)*(Q-1). Bovendien heeft E geen factor gemeenschappelijk heeft met (P-1)*(Q-1). N en E zijn de public key die een ieder mag weten. Als private key is de ontcijfersleutel D nodig. Deze ontcijfersleutel is zo gekozen zodat D*E rest 1 heeft bij deling door (P-1)*(Q-1). In formule vorm ziet dit er als volgt uit.

$$D * E = 1 \bmod ((P-1)*(Q-1))$$

Nu komt het vercijferen. Boodschap X wordt omgezet in een getal M. De geëncrypte tekst C is nu M^E modulo N. Ontcijferen gaat met behulp van de private key D. De geëncrypte tekst C wordt tot de macht D verheven. De uitkomt modulo N is gelijk aan het oorspronkelijke getal M. Dit getal M is gemakkelijke om te zetten in de boodschap X. Voor de overzichtelijkheid volgen hier de formules voor vercijferen en ontcijferen nog eens.

$$C = M^E \bmod N \qquad M = C^D \bmod N$$

RSA wordt als een zeer sterk algoritme gezien. Volgens [3] zijn de meeste crypto-analisten het er over eens dat er één manier is om RSA te breken: het factoriseren van de vercijfer modulus N in de twee priemen P en Q. Dit factoriseren is een uiterst moeilijk en tijdrovend proces. Hoe groter het product N is, des te groter is de uitdaging voor crypto-analisten. Zowel [3] als [4] merken op dat getallen met een lengte van 512 bits in bepaalde tijd te factoriseren zijn en dus als onveilig betiteld worden. Daarom adviseren ze om producten met een lengte van 1024 te gebruiken.

5.2 Symmetrische systemen

Bij een symmetrisch systeem wordt voor versleutelen en ontsleutelen dezelfde sleutel gebruikt. In versie 1.0 werd Bass-0-Matic gebruikt. Deze algoritme was door Zimmermann zelf bedacht. De algoritme voldeed niet en werd in versie 2.0 vervangen door IDEA. In de latere versies kan uit verschillende systemen gekozen worden. Deze zijn hieronder beschreven. De beschrijvingen komen voornamelijk uit [6].

5.2.1 CAST

Dit protocol is ontworpen door Carlisle Adams en Stafford Tavares. CAST lijkt veel op DES. CAST gebruikt een sleutelgrootte van 128 bits en 64-bits blok versleuteling. PGP gebruikt CAST5-128 waarbij gebruik wordt gemaakt van een blokgrootte van 128 bits. CAST kan gebruikt worden op militair niveau en is volgens [7] betrouwbaarder dan DES.

5.2.2 AES/ Rijndael

Joan Daemen en Vincent Rijmen zijn de uitvinders van het algoritme Rijndael. In 2000 werd Rijndael tot Advanced Encryption Standard verheven. AES verving hiermee de DES standaard. AES maakt gebruik van blok versleuteling. De blokgroottes zijn 128, 192 en 256 bits. De standaard is vrij jong. Tot op heden zijn er weinig aanvallen op bekend.

5.2.3 IDEA

IDEA is de afkorting voor International Data Encryption Algorithm. Het algoritme werd ontwikkeld door ETH in Zurich. Het maakt gebruik van 64 bits blokversleuteling met een sleutelgrootte van 128 bits. Het wordt binnen PGP gebruikt om RSA sleutels te vercijferen. IDEA staat als vrij sterk bekend. Sinds het ontstaan in 1992 zijn er geen succesvolle aanvallen bekend geworden. De sleutellengte van 128 bits is te groot om simpelweg alle mogelijke sleutels te proberen.

5.2.4 3DES

Dit algoritme is geënt op de Data Encryption Standard. Het DES algoritme werd door de Amerikaanse overheid geïntroduceerd in 1977, maar komt oorspronkelijk bij IBM vandaan. DES maakt gebruik van een blokgrootte van 64 bits. De sleutelgrootte van 56 bits voldeed lange tijd. Toen DES niet meer veilig bleek te zijn, kwam men met 3DES op de proppen. 3DES is drie keer DES versleuteling met drie verschillende sleutels. 3DES heeft een sleutelgrootte van 168 bits. Het is een tijd vretend algoritme en wordt volgens [5] gezien als het zwakste algoritme binnen PGP.

5.2.5 TWOFISH

Dit algoritme is door Counterpane Labs ontworpen in 1998 aldus [9]. Het maakt gebruik van 128 bits blokversleuteling en een sleutelgrootte van 128, 192 en 256 bits. Het is ongepatenteerd en gratis voor alle gebruikers.

5.3 Hash functie

Een hash functie is een functie die een input van willekeurige lengte neemt en die transformeert naar een string van vaste lengte. Het is een zogenaamde one-way functie. Dit houdt in dat de terug berekening heel moeilijk, zo niet onmogelijk is. Dit maakt de functie uitermate geschikt voor de verificatie van data. In hoofdstuk 7 wordt hier ook op ingegaan.

5.3.1 MD5

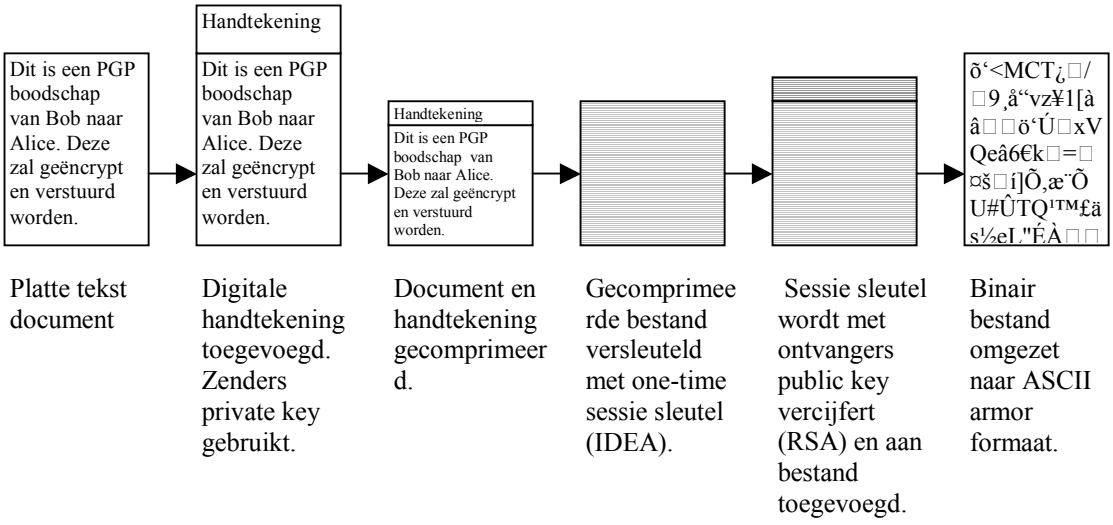
Dit is de vijfde versie van het Message Digest algoritme zoals uit [8] blijkt. Het is ontwikkeld door Ronald Rivest, die we nog van RSA kennen. De algoritme heeft als input een boodschap van willekeurige lengte en als output een handtekening met een lengte van 128 bits. Het is erg onwaarschijnlijk om bij twee verschillende boodschappen als input dezelfde handtekening te verkrijgen.

6 Vercijferen en ontcijferen

Het vercijferen en ontcijferen is de voornaamste taak van PGP. Als eerste zal het vercijferen en ontcijferen van e-mail beschreven worden. Vervolgens komt het vercijferen en ontcijferen van bestanden in dit hoofdstuk aan bod.

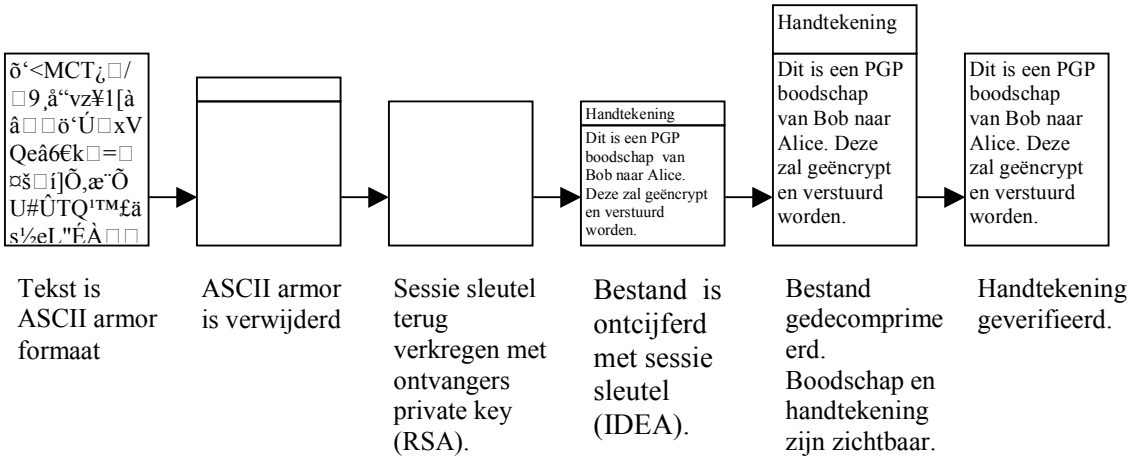
6.1 Hoe vercijfer je een e-mail

Het proces van het vercijferen van een e-mail staat uitvoerig beschreven in [1]. De afbeelding hieronder komt ook uit dit boek. Aan de hand van de stappen in de afbeelding zal het vercijferen van e-mail uitgelegd worden. Stel er wordt een e-mail verstuurd naar een ontvanger met behulp van PGP. Voordat de e-mail vercijferd wordt, vinden er nog enige stappen plaats. Zo kan er een digitale handtekening aan de e-mail toegevoegd worden. Deze worden samen gecomprimeerd om vervolgens door middel van een symmetrisch systeem –bijvoorbeeld IDEA- versleuteld te worden. Hiervoor wordt een one-time sessie sleutel gebruikt. Deze sleutel is van belang voor het ontcijferen en moet zodoende met de e-mail meegestuurd worden. Aangezien deze sleutel ook over het Internet verstuurd wordt, moet deze sleutel ook versleuteld worden. Dit doet PGP met een asymmetrisch systeem –bijvoorbeeld RSA. De zender gebruikt de public key van de ontvanger om de sleutel te vercijferen. Vervolgens voegt PGP de sleutel bij de reeds versleutelde e-mail. Wat we nu hebben is een binair bestand. Omdat e-mail een op tekst gebaseerd systeem is, moet het binaire bestand naar ASCII omgezet worden. Het resultaat wordt het ASCII armor bestand genoemd. Dit is het bestand dat naar de ontvanger wordt verstuurd.



6.2 Hoe ontcijfer je een e-mail

Stel de e-mail die in de vorige paragraaf verstuurd is, wordt ontvangen. Aan de hand van onderstaande afbeelding probeer ik uit te leggen hoe PGP de e-mail ontsleuteld. Voordat de e-mail gelezen kan worden, moet PGP de nodige bewerkingen maken. Als eerste moet het ASCII armor formaat verwijderd worden zodat er een binair bestand ontstaat. Dit binaire bestand bestaat uit een vercijferde e-mail en een vercijferde sleutel. Door middel van de ontvangers private key ontcijfert PGP de sleutel volgens het asymmetrische systeem – zoals voorheen RSA. Met deze sleutel ontcijfert PGP de e-mail gebruik makend van een symmetrisch systeem – zoals voorheen IDEA. De verkregen file wordt gedecomprimeerd. Afhankelijk of er een digitale handtekening gebruikt is, wordt de data geverifieerd of niet. Klopt de handtekening –of is deze niet gebruikt - dan is de e-mail leesbaar.



6.3 Hoe vercijfer en ontcijfer je bestanden

Er zijn twee gevallen waarbij het vercijferen van bestanden gewenst is. Er wordt een bestand overgedragen door middel van het Internet of een gegevens drager. Of er wordt een bestand op je harddisk bewaard op een manier zodat niemand anders dat bestand kan lezen. Beide gevallen worden in deze paragraaf besproken.

In het eerste geval gaat het vercijferen nagenoeg gelijk als het vercijferen van een e-mail. Het enige verschil is het ontbreken van de ASCII armor stap. Het bestand kan overgedragen worden per diskette of als bijlage bij een e-mail. Daar het geen e-mail betreft, hoeft het binaire bestand niet in tekst omgezet te worden.

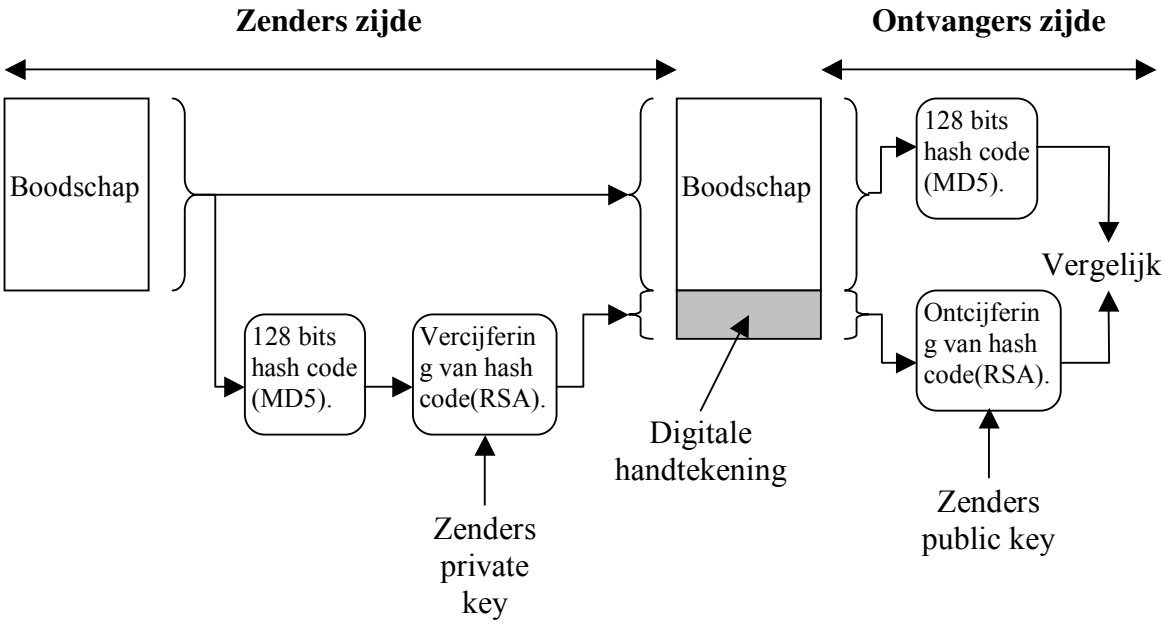
Het tweede geval betreft het vercijferen van een bestand op een locale computer. Omdat er geen tweede partij in het spel is, is het asymmetrische systeem overbodig. Eerst wordt het bestand gecomprimeerd. Vervolgens wordt het bestand met behulp van een symmetrisch systeem vercijferd. Door middel van een wachtwoord voorziet de gebruiker het algoritme van een sleutel. Het ontcijferen gaat met hetzelfde wachtwoord, sleutel en algoritme.

In het tweede geval moet het originele bestand verwijderd worden. Anders heeft het vercijferen geen zin. Gewoon verwijderen zoals een gebruiker gewend is, is niet voldoende. Het besturingssysteem verwijdert alleen de verwijzing naar het bestand. Met een simpel trucje als “undelete” kan het bestand terug getoverd worden. PGP verwijdert het bestand echt door hem te overschrijven. Volgens [1] wordt het originele bestand overschreven met nullen of met random data. Dit verschilt per versie.

7 De digitale handtekening

Ook bij het uitleggen van de digitale handtekening maak ik gebruik van een afbeelding die ik in [1] heb gevonden. Een digitale handtekening wordt gebruikt om de data te verifiëren. Bij PGP werkt het als volgt. De hash functie MD5 heeft de data als input nodig. Dit levert een hash code als output op. Deze hash code is 128 bits groot. PGP gebruikt de private key van de zender om deze hash code te versleutelen volgens de RSA algoritme. Dit levert de digitale handtekening die aan de data wordt toegevoegd. De data is klaar om verzonden of overgedragen te worden.

De ontvanger van de data ontcijfert de handtekening met de public key van de zender. Hij gebruikt de data als input voor dezelfde MD5 hash functie. Dit levert een hash code op die een lengte van 128 bits heeft. Deze hash code vergelijk hij met de ontcijferde handtekening. Zijn ze gelijk dan is de data niet veranderd. Zijn ze ongelijk dan is de data wel veranderd.



8 Conclusies

In deze paragraaf probeer ik een oordeel te geven over de veiligheid die PGP biedt. In ogenschouw zijn niet genomen veiligheid risico's zoals voor de hand liggende wachtwoorden, virussen of trojan horses. De conclusies zijn niet gebaseerd op eigen onderzoek, maar zijn gefundeerd vanuit de literatuur.

De algoritmes zijn stuk voor stuk sterke algoritmes. De zwakheid van PGP zal hier niet in schuilen. [5] ziet 3DES als het zwakste protocol. In [6] is te lezen dat de gebruiker van PGP zelf aan kan geven aan welk algoritme hij de voorkeur geeft en welk algoritme niet toegestaan is. Als 3DES niet vertrouwd wordt, dan wordt het algoritme niet toegestaan. Er zijn genoeg alternatieven beschikbaar.

[5] heeft een lijst van bugs gepubliceerd. De opgesomde bugs en de beveiliging van de computer waarop PGP wordt gebruikt zijn verontrustender dan de gebruikte algoritmes. Het is vanzelfsprekend dat zo'n groot programma veel bugs bevat. Hierdoor kan PGP niet als onveilig betiteld worden. [5] keurt het gebruik van PGP vanwege de bugs niet af, maar adviseert voorzichtig te zijn al PGP gebruikt wordt.

[3] geeft aan dat PGP makkelijk is te verkrijgen, te verspreiden en te gebruiken. Er is geen complexe infrastructuur, zoals veilig kanalen om de sleutels te verspreiden, voor nodig. [3] vindt het een voordeel dat één partij verantwoordelijk is voor het beheer van PGP. Hierdoor heeft PGP relatief weinig last van de bureaucratie waar andere standaarden onder lijden.

PGP heeft altijd in de belangstelling gestaan. Of dit nu komt door patent kwesties, het FBI onderzoek of de veiligheid. PGP is een programma dat wereldwijd bekend is en waar vele programmeurs mee bezig zijn. Zij houden zich bezig met de source code, het vinden van bugs, het testen en nog veel meer. PGP is uitgebreid beschreven in de bibliotheek en op het Internet en heeft een open karakter. Als een bug gerapporteerd wordt, dan zal daar ook passend actie op genomen worden. Al deze factoren leiden tot een betere kwaliteit van het product.

PGP doet zijn naam eer aan. Vandaar ook de titel van dit werkstuk: *"PGP: Pretty Good Privacy*". Het is inderdaad vrij goed voor het beschermen van de privacy. Het programma beantwoordt behoorlijk aan de behoeften van de doelgroep: de doorsnee e-mail lezer die zijn privacy wil beschermen. PGP veiligheid is echter niet absoluut zoals [5] stelt. PGP moet ook niet gebruikt worden om zware geheimen te beveiligen.

Ieder product is voor verbetering vatbaar. In PGP zitten veel bugs. Iedere bug die wordt opgelost, maakt PGP beter. [5] adviseert om het programma te simplificeren. Verkort de source code en laat PGP alleen de basis dingen doen. Zo worden bugs voorkomen.

9 Literatuur

[1] William Stallings. Protect Your Privacy – A Guide for PGP Users (Prentice Hall 1995).
[2] Simon Singh. The Code Book (Doubleday 1999).
[3] Bruce Schneier. E-mail Security – How to keep your electronic messages private (John Wiley & Sons 1995).
[4] Simson Garfinkel. PGP – Pretty Good Privacy (O'Reilly & Associates 1995).
[5] Sieuwert van Otterloo. A security analysis of Pretty Good Privacy, graduation thesis (2001).
[6] PGP Corporation. PGP 8.0 for Windows User's Guide (2002).
[7] J. Lee, H.M. Heys and S.E. Tavares. Resistance of a CAST-Like Encryption Algorithm to Linear and Differential Cryptanalysis (1997).
[8] R. Rivest. The MD5 Message-Digest Algorithm, rfc 1321 (1992).
[9] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, N. Ferguson. Twofish: A 128-Bit Block Cipher (1998).
[10] Phil Zimmermann. www.mit.edu/~prz. Homepage.

10 Begrippenlijst

Hier volgt een lijst van gebruikte begrippen. Ze zijn gesorteerd op alfabetische volgorde. De definities zijn voornamelijk uit [3] afkomstig.

3DES - Triple DES. 3DES is een variant op het DES algoritme dat iedere boodschap drie keer met het DES algoritme vercijfert;
ASCII - American Standard Code for Information Interchange. Dit is een standaard om letters, cijfers en andere karakters in nummers om te zetten;
ASCII armor - dit formaat wordt door PGP gebruikt om een binaire file in ASCII karakters weer te geven;
Asymmetrisch systeem - bij dit cryptografisch systeem wordt het vercijferen en ontcijferen door middel van een sleutelpaar gedaan. De sleutels van het sleutelpaar zijn onderling verschillend;
Bass-o-Matic - dit symmetrisch systeem is ontwikkeld door Phil Zimmermann en is gebruikt in PGP versie 1.0;
Blokversleuteling - algoritme dat data met een vaststaande lengte versleutelt;
Boodschap - een blok informatie dat van een zender naar een ontvanger verstuurd wordt;
Bug - een fout in de software;
CAST - dit is een symmetrisch systeem dat is ontwikkeld door Carlisle Adams en Stafford Tavares. De initialen van de makers vormen de naam van het algoritme;
Cryptografie - kunst en wetenschap van het versleutelen van data;
Crypto-analyse - kunst en wetenschap die zich richt op het breken van cryptografie;
Cryptologie - het geheel van cryptografie en crypto-analyse;
DES - Data Encryption Standard. Dit symmetrisch systeem is ontwikkeld door IBM. Later geadopteerd als standaard door het National Bureau of Standards;
Digitale handtekening - een blok data dat aan een boodschap is toegevoegd of dat een binaire file vergezelt. De handtekening staat in voor de authenticiteit van het bestand;
Encryptie algoritme - een bepaald wiskundig proces dat door een computer gebruikt wordt om een boodschap te vercijferen of ontcijferen;
Encryption key - vercijfer sleutel. Een woord, nummer of zin dat door een encryptie algoritme gebruikt wordt om een boodschap te vercijferen of ontcijferen;
Factoriseren - het ontbinden van een getal in priemen. Bijvoorbeeld: het getal 91 kan gefactoriseerd worden in de priemgetallen 7 en 13.
Gegevens drager - apparaat waar digitale data op bewaard kan worden. Bijvoorbeeld een diskette;
Hash-functie - een functie die een input van willekeurige lengte neemt en die transformeert naar een string van vaste lengte
IDEA - International Data Encryption Algorithm. Dit symmetrisch systeem uit Zwitserland is ontwikkeld door James L. Massey en Xuijia Lai;
Lengte van een sleutel - het aantal cijfers in een sleutel;
Library - een stuk software dat een verzameling van functies bevat;

MD5 - Message Digest vijfde editie. Dit message digest algoritme is ontwikkeld door Ron Rivest;

Message digest - een nummer dat is afgeleid van een boodschap;

Patent - een monopolie verleent door de overheid. Een patent verleent de patent houder het recht om andere mensen ervan te weerhouden van het maken, uitvoeren, gebruiken, verkopen of importeren van een uitvinding;

PGP - Pretty Good Privacy. Met behulp van PGP kunnen e-mail en bestanden versleuteld worden en digitale handtekeningen meegegeven worden;

Private key - een sleutel die gebruikt wordt voor zowel vercijferen als ontcijferen en die strikt vertrouwelijk is;

Priem - een nummer dat niet geheel gedeeld kan worden anders dan door zichzelf en door het nummer 1. Bijvoorbeeld het nummer 7;

Public key - een sleutel die wijd verspreid bekend mag zijn;

Random data - een onvoorspelbaar rijtje nullen en enen;

RSA - een asymmetrisch systeem uitgevonden door Ronald Rivest, Adi Shamir en Leonard Adleman. De eerste letters van hun achternaam vormen de naam van dit protocol;

RSAREF - De RSA reference implementatie is ontwikkeld door RSA Data Security en gratis beschikbaar. RSAREF geeft non-commerciële programma's de mogelijkheid om het RSA algoritme te gebruiken zonder het gevaar inbreuk te maken op het patent;

Symmetrisch systeem - cryptografisch systeem waarbij het vercijferen en ontcijferen met dezelfde sleutel gebeurt;

Trojan horse - een, meestal kwaadwillend, programma dat zich hecht aan een ander programma en zo ongemerkt een computer systeem binnendringt;

Virus - een virus is een stukje software dat zichzelf vermenigvuldigt door een besturingssysteem of programma of document te infecteren;

11 Afkortingen

Hier volgt een lijst van gebruikte afkortingen. Ze zijn gesorteerd op alfabetische volgorde.

3DES	Triple DES
AES	Advanced Encryption Standard
ASCII	American Standard Code for Information Interchange
BWI	Bedrijfswiskunde en Informatica
CAST	Carlisle Adams en Stafford Tavares
DES	Data Encryption Standard
DH	Diffie-Hellman
FBI	Federal Bureau of Investigation
GNU	Gnu is Not Unix
GPG	GNU Privacy Guard
IDEA	International Data Encryption Algorithm
MD5	Message Digest fifth edition
MIT	Massachusetts Institute of Technology
PGP	Pretty Good Privacy
RSA	Rivest, Shamir and Adleman
RSAREF	RSA Reference