Web of Trust

Naast certificaten bestaat nog een andere manier waarop je een soort garantie kunt krijgen dat een ontvangen public key inderdaad afkomstig is van de juiste eigenaar. Dat werkt zonder 'certificate authority' en heet *Web of Trust* (WoT). Het wordt o.a. gebruikt in GPG (*GNU Privacy Guard*) waarmee je bestanden kunt encrypten en/of signeren om ze bijv. als email-attachment te versturen. GPG is een implementatie van de OpenPGP standaard, waarbij PGP staat voor 'Pretty Good Privacy'.

Twee partijen die over en weer berichten willen sturen moeten elk een eigen public/private keypaar hebben. Als jij geëncrypte berichten ontvangt dan heeft de zendende partij die geëncrypt met jouw public key. Voor decrypten gebruik je je eigen private key; die heb je zelf gemaakt dus die kun je vertrouwen. Maar als jij berichten wil encrypten voor *verzending* dan heb je de public key van de ontvangende partij nodig. Die key moet je van tevoren ergens hebben opgehaald. Web of Trust is bedoeld om jou zekerheid te geven dat die public key inderdaad afkomstig is van de bedoelde andere partij, en dat er niet mee is gerommeld voordat jij 'm in handen kreeg.

Bij het maken van een GPG-keypair wordt een identificatie van de maker/eigenaar -meestal zijn email adres- aan de public key gekoppeld. De maker kan daarna zijn public key op een *public key server* plaatsen, en iedereen die 'm nodig heeft kan 'm downloaden. Maar hoe weet de ontvanger dat die key niet gemanipuleerd is?

Als jij iemand een geëncrypt bericht wil sturen dan haal je zijn public key van een keyserver, of je vraagt of hij zijn key wil opsturen. Die key moet je daarna toevoegen aan jouw 'sleutelbos' (key ring); dat wordt *importeren* genoemd. Als je op dat moment die key al voldoende vertrouwt dan kun je vervolgens je gang gaan met het encrypten en versturen van je bericht.

Bij GPG-keys hoort een standaard hashberekening: die levert een z.g. *fingerprint* van de key. Die kun je uitrekenen en dan verifiëren bij de eigenaar, bijvoorbeeld in een telefoongesprek. Dat is een goede manier om vertrouwen in de ontvangen key te krijgen, maar het is wel omslachtig.

Het Web of Trust is hiervoor een alternatief, en dat werkt als volgt. Elke maker/eigenaar kan zijn public key laten *signeren*[1] door mensen uit zijn kennissenkring. Dat kan op individuele basis, en soms worden ook *key signing*

---

1. Een ondertekenaar *signeert* een bericht door een hash daarvan uit te rekenen, die te encrypten met zijn private key, en dat als handtekening bij het bericht te voegen. De ontvanger rekent zelf ook de hash uit, gebruikt de public key van de ondertekenaar om de handtekening te decrypten, en vergelijkt beide hashes. Als dat decrypten van de handtekening lukt dan staat vast dat de encryptie is gedaan met de private key van de ondertekenaar, en die persoon is, (als het goed is) de enige die die private key heeft. Als beide hashes ook nog identiek zijn, dan is alles OK.

parties georganiseerd, bijvoorbeeld tijdens congressen. Dat betekent dus dat aan iemands public GPG-key een aantal digitale handtekeningen komt te hangen van andere mensen die daarmee zeggen: “ik kan je verzekeren dat deze public key inderdaad van mijn kennis ..... is”. Houd wel in gedachten dat je zonder de public key van de ondertekenaar niks hebt aan zo’n handtekening.

Bij importeren moet je ook een oordeel geven over hoe zorgvuldig de eigenaar van die nieuwe key, volgens jou, de identiteit zal controleren van een derde persoon die aan hém een key-signing verzoek voorlegt. Dat gaat op een schaal met een aantal niveaus zoals 'ik weet het niet', 'hij is een allemansvriend', tot 'ik vertrouw zijn zorgvuldigheid blindelings'. Het oordeel blijft vertrouwelijk binnen jouw sleutelbos. Het gaat pas later een rol spelen, wanneer jij keys van andere personen ontvangt.

Na verloop van tijd bevat je GPG-sleutelbos dus meerdere geïmporteerde sleutels (d.w.z. public keys die jij zodanig vertrouwt dat je ze hebt geaccepteerd), met bij elke sleutel een oordeel van jou over de eigenaar van die sleutel. En daarnaast kan, zoals hierboven uitgelegd, elke sleutel nog handtekeningen met zich meedragen van kennissen van de eigenaar. Maar met die handtekeningen kun je nog niks.

Stel dat jij later de sleutel van een nieuwe relatie ontvangt en wil importeren. Op dat moment wordt gekeken of mede-ondertekenaars van die nieuwe sleutel al in jouw sleutelbos aanwezig zijn. Anders gezegd: bevat jouw sleutelbos al sleutels (= public keys) van kennissen van die nieuwe relatie. Is zo’n kennis inderdaad al aanwezig, en heb jij van die kennis indertijd genoteerd dat je blindelings vertrouwt hoe hij identiteiten van signing-verzoekers controleert, dan wordt de sleutel van de nieuwe relatie ook meteen geaccepteerd. Dat is natuurlijk op voorwaarde dat de handtekening van die kennis aan de nieuwe sleutel klopt, maar dat wordt ter plekke gecontroleerd met diens public key want die zat dus al aan je sleutelbos. De nieuwe sleutel wordt ook geaccepteerd als er al drie andere sleutels aan je bos zitten waarvan je de eigenaren als 'matig zorgvuldig' hebt gelabeld. En zo zijn er nog meer combinaties die het automatisch accepteren van die nieuw binnengekomen sleutel laten afhangen van wat jij eerder hebt geoordeeld over degenen die die nieuwe sleutel mede hebben ondertekend, voor zover de eigen sleutels van die mede-ondertekenaars al eerder in jouw sleutelbos waren terechtgekomen.

Nog even recapituleren: als jij een nieuwe sleutel ontvangt en wil importeren, en die nieuwe sleutel draagt handtekeningen van kennissen van de eigenaar, dan spelen die alleen een rol als de sleutels van die kennissen zelf al eerder in jouw bos terecht waren gekomen. Is dat niet het geval dan moet je op een andere manier besluiten of je die nieuwe sleutel voldoende vertrouwt om ’m te importeren aan je sleutelbos.

In dit hele verhaal gingen we ervan uit dat jij iemands sleutel (public key) nodig hebt om berichten te encrypten die jij naar die persoon wilt sturen. Maar dezelfde maatregelen gelden ook als jij een gesigneerd bericht ontvangt, en jij de (public) sleutel van de ondertekenaar nodig hebt om de handtekening te controleren. Want ook die sleutel heb je moeten ophalen en importeren, en daarbij moest je zeker weten dat die te vertrouwen was.

Web-of-Trust implementeert op die manier een soort “de vrienden van mijn vrienden zijn ook míjn vrienden”. Maar het blijft omslachtig, en daarom wordt het niet zo veel gebruikt.