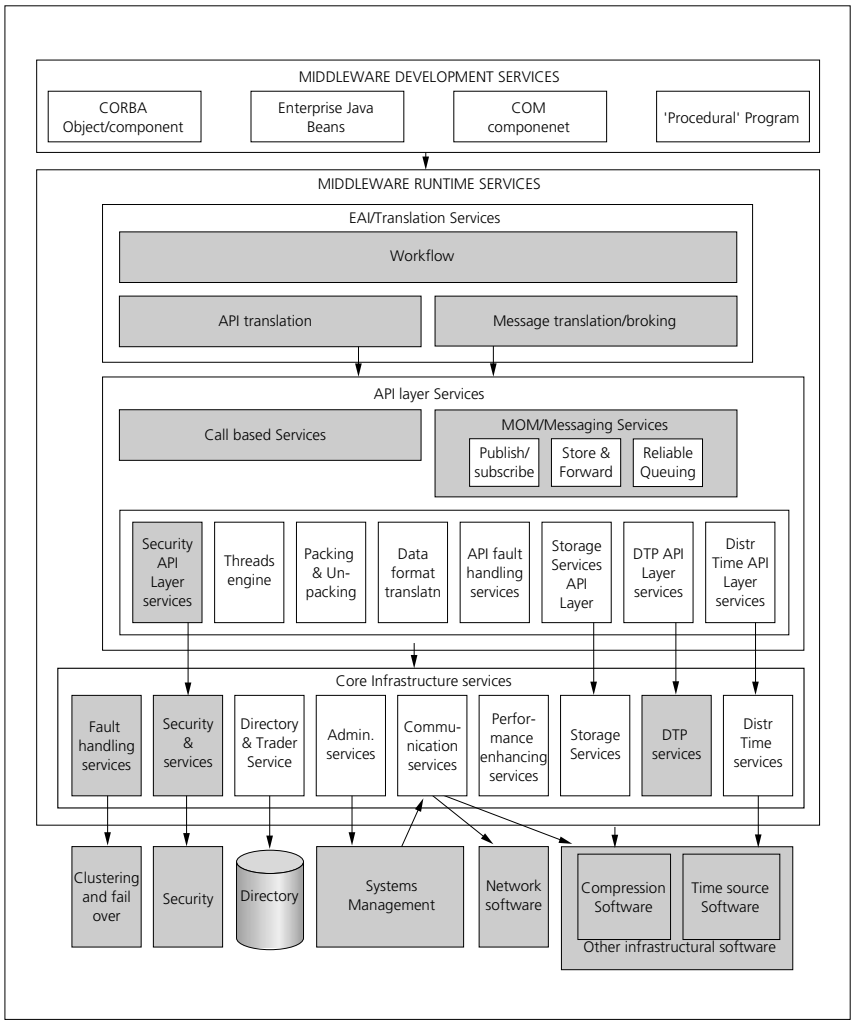
In mijn vorige artikel heb ik de belangrijkste beveiligingsfuncties beschreven zoals een middlewareproduct die biedt. In deze aflevering zal ik wat dieper graven, en laten zien op welke manier middleware een aantal van deze diensten levert.

Middleware en securitydiensten (deel 2)

BEVEILIGINGSMECHANISMEN Authentisering zorgt ervoor dat het proces of de gebruiker die om toegang tot een systeem vraagt, ook inderdaad het proces of de gebruiker is dat of die hij beweert te zijn. In mijn vorige artikel stelde ik dat er verschillende authentiseringmechanismen bestaan. Zo kan authentisering gebaseerd zijn op wat iemand weet (een wachtwoord bijvoorbeeld), wat iemand bezit (bijvoorbeeld een smartcard of een ander fysiek object) of wie iemand is zoals biometrische authentisering op basis van stemherkenning, vingerafdruklezers, netvliesscanners, enzovoort). Op combinaties van gebruikersnaam en wachtwoord gebaseerde mechanismen worden in veel gevallen standaard met de middleware meegeleverd, terwijl voor andere authentiseringmechanismen veelal producten van derden dienen te worden gebruikt.

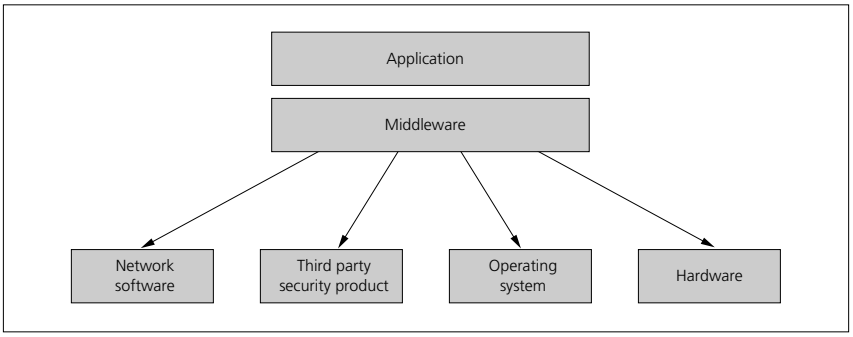
GEBRUIKERSNAMEN EN WACHTWOORDEN Hoewel ze op grote schaal worden toegepast, zijn gebruikersnamen en wachtwoorden als identificatiemiddel gevoelig voor misbruik. Ze kunnen worden vergeten, zijn soms al te voor de hand liggend en daardoor gemakkelijk te raden, worden door gebruikers vaak achteloos gebruikt (en bijvoorbeeld op een Post-It briefje aan hun beeldscherm gehangen), en zijn niet noodzakelijkerwijs uniek binnen de organisatie of tussen organisaties. Ondanks deze bezwaren is de gebruikersnaam/ wachtwoord-aanpak in organisaties en middlewareproducten een van de meest verbreide methoden.

Voor dit type identificatie wordt door de systeembeheerder handmatig een



FIGUUR 1: Overzicht van middleware-diensten

beveiligd bestand met daarin de gebruikersnamen, groepen en wachtwoorden opgezet. Om de beveiliging te verbeteren kunnen de in het bestand aanwezige wachtwoorden eventueel worden versleuteld. Het bestand kan onderdeel uitmaken van de directory service of een separaat bestand zijn. Om een gebruiker te kunnen authentiseren, neemt de authentiseringdienst aan de hand van de gebruikers- of groepsnaam een kijkje in het bestand en vergelijkt hij het opgegeven wachtwoord met het in het bestand aanwezige wachtwoord.



FIGUUR 2: Middleware vervult een 'koppelingsrol' tussen de applicatie en (bijv.) beveiligingsproducten

DIGITALE CERTIFICATEN Digitale certificaten bieden een elektronisch bewijs van identiteit en doen dit met behulp van kryptografische sleutels die worden gebruikt om het bericht te 'ondertekenen' of te 'verzegelen'. Het voordeel van dit soort certificaten is dat ze het mogelijk maken om zowel de verzender de ontvanger te vragen om zijn of haar identiteit aan te tonen, en op die manier ook de client een zekere mate van garantie te bieden dat de benaderde server inderdaad de server is die hij beweert te zijn.

Digitale certificaten worden in dit geval uitgegeven door een derde partij, die vaak wordt aangeduid als Trusted Third Party (TTP) of Digital Certificate Authority.

TTP's zijn verantwoordelijk voor het intrekken van certificaten zodra die niet langer geldig zijn, voor het in de directory publiceren van een lijst met ingetrokken certificaten en voor het hervalideren en opnieuw uitgeven van certificaten die verlopen zijn. Hoewel onafhankelijke TTP's op dit moment ietwat dun gezaaid zijn, is er geen enkele reden waarom een bedrijf niet als zijn eigen TTP kan optreden als het een netwerk van (interne en externe) geregistreerde geauthentiseerde gebruikers wil opzetten. Sterker nog, dit zou in bepaalde gevallen wel eens de beste methode kunnen zijn, aangezien elk bedrijf er dan belang bij heeft om ervoor te zorgen dat zijn gebruikers op de correcte wijze fysiek zijn geauthentiseerd, alvorens hen een certificaat te verstrekken. Als bedrijf kun je ook je smartcards persoonlijk overhandigen en heb je een grotere kans dat certificaatgegevens ook actueel worden kunnen gehouden.

Het systeem waarin elk bedrijf als zijn eigen TTP optreedt kent echter ook nadelen. Het eerste nadeel is de extra administratieve belasting die deze aanpak met zich meebrengt. Daarnaast loop je het risico dat je 'onbekende gebruikers', de gebruikers die je als bedrijf bijvoorbeeld juist wil binnenlokken voor e-commerce over het Internet, met honderden certificaten kunnen komen te zitten die de digitale tegenhanger zijn van al het plastic waar onze portefeuilles bol van staan. Dit zou in potentie een serieuze barrière kunnen vormen voor mensen die van Internet gebruikmaken. Als ze elke keer dat ze via Internet willen bestellen moeten zien te weten welk certificaat ze ook al weer moeten gebruiken, en elke keer om een certificaat moeten vragen als ze iets willen bestellen, zullen ze dat hoogstwaarschijnlijk al gauw te veel moeite vinden.

De zojuist genoemde bezwaren zijn, met name in het Verenigd Koninkrijk, voor een aantal industrieconsortia reden geweest om de koppen bij elkaar te steken om te komen tot standaarden voor hun eigen branche, waaronder de detailhandel en transportsector en levensverzekeraars en pensioenverzekeraars. Laatstgenoemde bedrijven hebben, althans in het Verenigd Koninkrijk, een nieuw beveiligingsraamwerk opgezet voor online communicatie tussen henzelf en de zelfstandig financieel adviseurs waarmee ze werken.

De TTP genereert eerst een sleutelpaar, dat bestaat uit een public key en een private sleutel voor de persoon die om het certificaat heeft gevraagd. De public key wordt in een bestand gezet. Dit bestand moet toegankelijk zijn voor elk proces of elke gebruiker waarmee de persoon in kwestie moet kunnen communiceren. Het zal duidelijk zijn dat deze informatie idealiter ligt opgeslagen in de directory service van de middleware. De private sleutel wordt uitgereikt aan de persoon zelf.

Wanneer deze persoon nu een dienst wil aanvragen van een server die zich elders bevindt, wordt zijn bericht versleuteld met behulp van de private sleutel. Deze fungeert op dezelfde manier als een handtekening. Aangezien alleen de persoon zelf de private sleutel kent, kan het alleen die persoon zijn die het bericht heeft verstuurd. Aan de andere kant 'ontsluit', of eigenlijk 'ontcijfert', de ontvanger het bericht met behulp van de public key die hij uit de directory heeft gehaald.

Er bestaat een standaard, het X590 Authentication Framework, die zich richt op digitale certificaten en PKCS (Public Key Certificate Standard) en de verschillende vormen beschrijft die verzoeken om certificaten kunnen hebben. Krachtens de standaard ontvangt de Digital Certificate Authority een initieel verzoek van een

gebruiker en wijst hij vervolgens aan elke gebruiker een unieke naam toe, vergezeld van diens sleutels. De standaard beschrijft ook hoe daaropvolgende verzoeken van een verzender om de public keys van andere ontvangers moeten worden afgehandeld.

SMARTCARDS Er moet uiteraard voor worden gewaakt dat de private key niet in verkeerde handen terechtkomen. Als vuistregel geldt dat ze niet op een voor anderen toegankelijke machine moeten worden opgeslagen. Een van de manieren om er zeker van te zijn dat de sleutel veilig is opgeslagen, is de private sleutelroutine bijvoorbeeld op een diskette te bewaren, of op een ander opslagmedium dat je veilig bij je kunt dragen.

Een van de belangrijkste nieuwe methoden om private keys te bewaren is echter met behulp van de smartcard. Deze hebben als bijkomend voordeel dat, om ze te kunnen gebruiken, iemand ook de pincode van de kaart moet hebben. Dit houdt in dat, zelfs als de kaart zoek raakt of wordt gestolen, de op de kaart aanwezige gegevens –de private sleutel van de smartcardbezitter- toch beschermd zouden moeten zijn.

Sinds 1996 is een Personal Computer/Smartcard Workgroup (PC/SC Workgroup) actief, die is opgericht door Microsoft, Groupe Bull, Hewlett-Packard, Schlumberger en Siemens Nixdorf. De werkgroep is in het leven groepen om een antwoord te vinden op interoperabiliteitsvraagstukken en specificaties te ontwikkelen voor de koppeling van tussen smartcardlezers en smartcards en pc's. Men kijkt ook naar manieren om apparatuur-onafhankelijke applicatieprogrammeerinterfaces te kunnen leveren die de ontwikkeling van op smartcards voorbereide applicaties moeten stimuleren. Inmiddels is er ook een eerste ontwerpversie verschenen van een standaard voor contactloze smartcardsystemen. Daarnaast zijn er concurrerende smartcardspecificaties gepubliceerd.

Dat er standaarden voor smartcards nodig zijn heeft dezelfde reden die we eerder zagen bij certificatieautoriteiten. Bij een overvloed aan smartcards, die zonder enige twijfel een rem zou leggen op elektronisch handelsverkeer, is niemand gebaat. Ook in het geval van smartcards zien we echter dat standaardisatie vooral een zaak lijkt te zijn van branchegerichte organisaties. De Britse overheid, bijvoorbeeld, wilde een standaard voor smartcards om smartcards te kunnen gebruiken voor het betalen van belastingen en toegang tot andere overheidsdiensten. Men was bijvoorbeeld van plan om ook de uitbetaling van uitkeringen aan een smartcard te koppelen, een initiatief dat inmiddels echter in de ijskast is beland.

BIOMETRICA Biometrische technieken identificeren een individu op basis van een 'biologische' sleutel. Er zijn inmiddels verschillende technologieën op de markt waarmee aan de hand van biometrische gegevens de identiteit van de gebruiker kan worden vastgesteld:

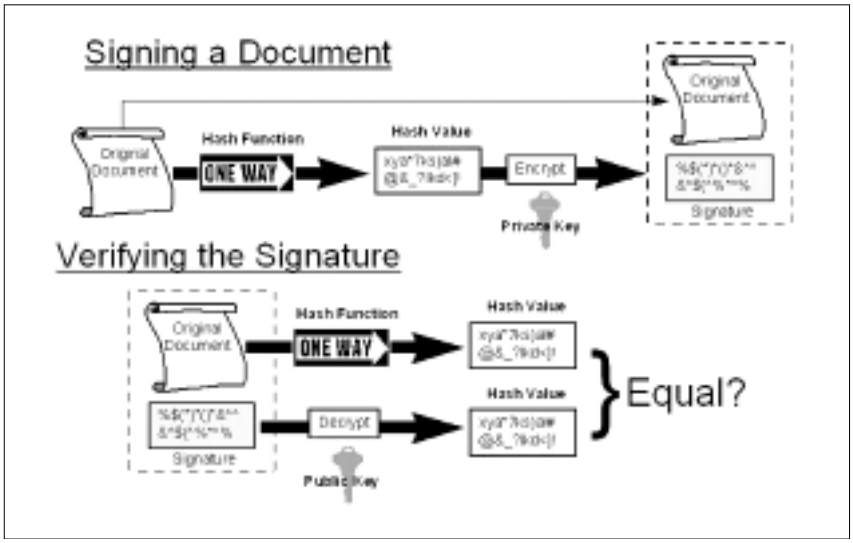
Vingerafdruk – vingerafdrukken, of eerder nog duimafdrukken, worden door een scanner 'afgenomen'. De scanner reflecteert het door het vingeroppervlak weerkaatste licht via een prisma naar een reeks lichtgevoelige cellen (CCD's zoals die ook in een videocamera zitten). Het beeld wordt vervolgens vertaald naar een verzameling datapunten. Algemeen beschouwd als een van de veiligere apparaten, kan de huidige generatie scanners niet meer worden misleid door bijvoorbeeld op rubberen handschoenen afgedrukte neppatronen, door vingerbewegingen of -buitengewoon gruwelijk- vingers die niet langer aan een lichaam vastzitten of vingers die juist aan een ander dood lichaam zijn vastgemaakt (de vinger wordt namelijk gecontroleerd op warmte en de aanwezigheid van stromend bloed).

Retinascan – deze netvliesscans bieden een met een vingerafdruk vergelijkbaar identificatiemiddel dat gebaseerd is op het feit dat het netvlies van iedere mens een uniek patroon heeft. Deze oplossing is tamelijk vergaand. Slechts weinig mensen vinden het prettig om hun ogen aan een scanner bloot te stellen, hetgeen verklaart waarom we deze technologie slechts in een handjevol producten terugvinden.

Handtekening – hoewel iemands statische handtekening kan worden vervalst, is dit voor de schrijfbeweging waarmee de handtekening tot stand komt, en de versnellingen en drukverschillen die zich daarbij voordoen, vrijwel uitgesloten. Apparaten die het biometrisch gebruik van de handtekening ondersteunen, maken gebruik van bewegingssensoren die zijn ingebouwd in een gewone pen die draadloos aan een computer is gekoppeld.

Stemherkenning – de belangstelling voor stemherkenning als identificatiemiddel is enerzijds te verklaren door het feit dat deze methode ook over de telefoon kan worden gebruikt (en de gebruiker lokaal dus geen speciale apparatuur nodig heeft), en anderzijds door het feit dat deze methode kan worden gecombineerd met spraakherkenningssoftware. Stemherkenningssoftware registreert stembuigingpatronen en pieken en dalen in de toonhoogte van de spreker. Voice prints zijn betrekkelijk simpel te maken. Daar staat uiteraard wel tegenover dat iemands stem in de loop van de tijd kan veranderen (met name tijdens verkoudheid en keelontstekingen), en kan worden geïmiteerd.

Gezichtsherkenning – mensen herkennen andere mensen op basis van de kleur en vorm van hun gelaatstrekken. Gezichtsherkenningssoftware doet dat anders, en deelt het beeld op in lichtere en donkerdere gedeelten waarmee de vorm in een meer contour-gerelateerde manier wordt geanalyseerd, en waarbij aspecten worden gebruikt die het onder-



FIGUUR 3. Middleware maakt onder meer gebruik van asymmetrische of public key-encryptie en symmetrische of private key-encryptie.

scheid tussen gezichten helpen maken. Op deze manier maken variaties qua haardracht, een andere bril of variaties in gelaatsuitdrukkingen geen verschil meer. Plotselinge gewichtstoename soms wel, terwijl de huidige technologie nog kan worden misleid met behulp van levensgrote fotografische maskers waar de eigen neus doorheen steekt!

Er zijn nog meer technologieën die op bovenstaande lijst hadden kunnen staan, waaronder technieken voor het analyseren van bewegingspatronen, de unieke lichaamsbewegingen die iemand maakt (bijvoorbeeld iemands 'loopje').

Met het oog op een snellere verwerking kan biometrica worden gecombineerd met smartcards. In het meer algemene geval geeft de gebruiker het systeem eerst zijn identificatie (naam of andere code), waarna het systeem in een database het biometrische patroon (retinascan, duimafdruk, handtekening, stempatroon, enz.) van de genoemde persoon opzoekt. Het uit de database afkomstige patroon wordt vervolgens vergeleken met het patroon dat de beveiligingssoftware van de gebruiker heeft 'afgenomen'. Als de biometrische database zich elders bevindt, kan dit de verwerkingstijd nadelig beïnvloeden, terwijl de gegevensoverdracht in dat geval zelf mogelijkerwijs niet goed is beveiligd.

Gecombineerd met smartcards in plaats van een certificaat, bevat de smartcard het biometrische patroon en kan de vergelijking en verdere verwerking lokaal door het beveiligingsapparaat worden uitgevoerd. Dit kan een aanzienlijke snelheidswinst opleveren, en bovendien veiliger gebeuren. De Association for Biometrics (www.afb.org.uk), het Biometric Consortium (www.biometrics.org), het Biometrics Journal en het International Biometric Industry Consortium (www.ibia.org) zijn stuk voor stuk goede bronnen van meer gedetailleerde informatie over technologieën en leveranciers.

AUTORISATIE Autorisatie is de dienst die er voor zorgt dat de gebruiker of het proces, als de authenticiteit eenmaal is vastgesteld, toestemming krijgt om de specifieke dienst of systeembron ook inderdaad aan te vragen. Autorisatie stelt dus vast of een eenmaal geauthentiseerde persoon gerechtigd is om een dienst of applicatie te gebruiken, of toegang te krijgen tot bepaalde gegevens.

Een van de manieren om autorisatieregels te implementeren is om de policy-regels uit te geven met digitale certificaten. Op deze manier vastgelegde policies worden Digital Certificate Privileges genoemd. In het vorige artikel zagen we echter dat de meeste autorisatie in de praktijk wordt geïmplementeerd met behulp van Access Control Lists (ACL's).

PRIVACYMECHANISMEN Deze dienst is bedoeld om de vertrouwelijkheid te beschermen van gegevens die onderweg zijn of zich in een bestand bevinden. In de praktijk wordt deze doelstelling gerealiseerd door gebruik te maken van encryptie. In het vorige artikel hebben we gezien dat asymmetische of public key-encryptie en symmetrische of secret key-encryptie de twee encryptiemethoden zijn die door middleware worden gebruikt.

PUBLIC KEYS Als een bericht van het ene naar het andere proces moet worden versleuteld, maakt de verzender voor het versleutelen van het bericht gebruik van de public key van de ontvanger. Op deze manier kan worden gegarandeerd dat de ontvanger de enige is die het bericht kan ontcijferen, aangezien alleen deze over de private sleutel beschikt.

Als eerste stap haalt de verzender de public key van de ontvanger op uit de directory met public keys. Als tweede stap gebruikt de verzender de public key om het bericht te vercijferen. Als derde stap wordt het bericht verzonden, om uiteindelijk door de ontvanger met behulp van zijn private sleutel te worden ontcijferd.

Om de verzender er zeker van te kunnen laten zijn dat de public key die hij op het punt staat te gebruiken ook inderdaad die van de beoogde ontvanger is, wendt de verzender zich tot de Trusted Third Party/Digital Certificate Authority en vraagt deze aan de hand van een of andere identificator van de ontvanger om diens public key. De Digital Certificate Authority vercijfert de public key met behulp van zijn eigen private sleutel. De verzender beschikt over een per-

manent exemplaar van de public key van de Digital Certificate Authority, en is daarmee in staat het bericht te ontcijferen en de public key te extraheren van degene aan wie hij het bericht wil sturen.

Wat hiermee wordt bereikt is dat het bericht alleen afkomstig kan zijn van de Digital Certificate Authority, aangezien alleen deze de private sleutel heeft waarmee het bericht wordt versleuteld, zodat de verzender er op kan vertrouwen dat er geen binnendringer is geweest die in de conversatie heeft ingebroken en de public key van de ontvanger door een andere heeft vervangen. Waar dit alles op neerkomt, is dat de verzender er zeker van kan zijn dat alleen de beoogde ontvanger het bericht zal kunnen lezen. Berichten die een met de private sleutel van de DCA vercijferde public key van een bepaald proces bevatten, staan bekend als 'Public Key Certificates'.

Versleuteling met behulp van public keys is een buitengewoon nuttige vorm van encryptie, aangezien hij over het Internet kan worden gebruikt, zolang de directory met public keys maar toegankelijk is voor de verzender. Dit betekent dat elke gebruiker, waar hij of zij zich ook bevindt, een bericht kan versleutelen alvorens het de lijn op te sturen, zonder daarvoor zelf sleutels over het netwerk te hoeven sturen. Het belang van dit laatste punt zal duidelijk zijn: door public key-encryptie te gebruiken, wordt het over de lijn uitwisselen van sleutels overbodig en kun je, gesteld dat je dat wilt, toegang geven aan gebruikers die jou tot dan toe onbekend waren. Dit betekent dat een bedrijf bijvoorbeeld bestellingen kan accepteren van klanten die nog niet als klant bekend staan (maar waarvan de identiteit via een TTP en de public key van die persoon kan worden vastgesteld), en de bestellingen kunnen worden versleuteld als ze over het netwerk gaan. Dit houdt in dat de nieuwe klant creditcardgegevens en andere gevoelige informatie met de bestelling kan meesturen, in de wetenschap dat de vertrouwelijkheid van die informatie gewaarborgd is.

Aan public keys kleven echter enkele problemen. Op public keys gebaseerde algoritmen zijn bijvoorbeeld notoir traag, tot wel duizend keer langzamer dan de symmetrische algoritmen waarop secret key-cryptografie is gebaseerd. Een ander bezwaar is dat het beheer van public keys, waarvoor een beveiligde, honderd procent actuele en voor zowel verzender als ontvanger toegankelijke directory is vereist, een behoorlijke inspanning vergt.

> 'Onbekende gebruikers' van e-commerce kunnen met honderden certificaten komen te zitten

Tot de bekendere public key-algoritmen die momenteel in gebruik zijn behoren:

- **RSA** ontwikkeld door Ron Rivest, Adi Shamar en Leonard Adleman van het MIT. Het algoritme wordt inmiddels op grote schaal onder licentie gebruikt. Export van producten die RSA bevatten is toegestaan, maar voor export voor encryptiedoeleinden gelden restricties. De beide sleutels waarvan RSA gebruik maakt, zijn afgeleid van een paar zeer grote priemgetallen.
- **DSA** (Digital Signature Algorithm) ontwikkeld door David Kravitz van de NSA.

Secret keys Deze benadering houdt in dat elke deelnemer aan de conversatie toegang heeft tot dezelfde secret key, die zowel voor het vercijferen als het ontcijferen van het bericht wordt gebruikt. Een bericht van een verzender wordt vercijferd met behulp van de secret key en het encryptie-algoritme, Aan de andere kant van de lijn wordt het bericht door de ontvanger ontcijferd met behulp van het bijbehorende decryptie-algoritme en dezelfde secret key. De algoritmen zijn vrij verkrijgbaar, in de zin dat ze door onafhankelijke leveranciers of standaardisatieorganisaties ter beschikking worden gesteld.

De secret key is dus het belangrijkste onderdeel van het proces, aangezien het deze door beide partijen overeengekomen sleutel is die bepaalt hoe veilig het bericht eigenlijk is. Doorgaans is de lengte van de sleutel de bepalende factor voor de mate van beveiliging die kan worden geboden. Hoe langer de gebruikte sleutel, des te moeilijker het bericht door een inbreker te kraken is. Vandaar ook de discussies over 40 bits en 56 bits sleutels.

De Verenigde Staten hanteren nog altijd exportbeperkingen voor producten die het gebruik van sleutels boven een bepaalde lengte ondersteunen. Voor financiële instellingen en multinationals zijn enkele uitzonderingen gemaakt, zodat het op bedrijfsniveau zin heeft om na te gaan of het middlewareproduct dat je wilt gaan gebruiken het vereiste beveiligingsniveau biedt en ook gebruikt mag worden in het land waar je werkzaam bent.

Tot de belangrijkste symmetrische algoritmen die op dit moment in beveiligingsproducten worden toegepast behoren:

- **DES** (Data Encryption Standard) een geheimschrift-algoritme van 64 bits grote blokken met een sleutellengte van 56 bits (in voorkomende gevallen kan dit lager zijn vanwege exportrestricties die gelden voor het land waar je je bevindt).
- **Triple DES** – zoals de naam al aangeeft wordt DES hier drie keer achter elkaar toegepast.

NON-REPUDIATION – EEN VOORBEELD

Iemand wil een bestelling plaatsen bij een bedrijf. De persoon in kwestie stuurt eerst een bericht waarin hij de dienst vraagt de bestelling te plaatsen,. Het bericht wordt ondertekend met de private sleutel van de aanvrager. Deze wordt vervolgens geauthentiseerd door de server, die daarna een met behulp van zijn private sleutel vercijferd bericht terugstuurt dat hij bereid is de bestelling te accepteren. De persoon authentiseert de server door de directory in te gaan om daar de public key op te halen waarmee hij het bericht van de server kan ontcijferen. Als het bericht zich correct laat ontcijferen, is de server de juiste, zoniet dan is de server, bijvoorbeeld, een oplichter.

Zowel de persoon als de server heeft de identiteit van de andere partij vastgesteld. Vanaf dat moment kan elk onderling bericht op dezelfde manier worden behandeld. De besteller kan zijn bestelling met de private key vercijferen, terwijl elke ontvangstbevestiging ook door de server kan worden ondertekend. Op deze manier hebben beide partijen dan een methode gebruikt waarmee ze ontkenning van ontvangst of verzending kunnen voorkomen.

- **RC5** een geheimschrift-algoritme dat is ontwikkeld door Ron Rivest, de mede-oprichter van RSA Data Security, en dat variabele parameters kent.
- **IDEA** een geheimschrift-algoritme van 64-bits blokken en een 128-bits sleutel (een lengte die plaatselijk kan afwijken). Ontwikkeld door Xuejia Lai en James Massey.
- **RC4** een streaming geheimschrift-algoritme met variabele sleutellengte dat door Ron Rivest voor RSA Data Security is ontwikkeld. Kreeg een speciale exportstatus toegekend nadat de sleutellengte tot 40 bits of minder werd beperkt.

Het is misschien aardig om te weten dat het RC5-algoritme in combinatie met een sleutellengte van 40 bits in drieënhalf uur werd gebroken, terwijl RC5 met een sleutellengte van 48 bits 13 dagen stand hield. In dit geval blijkt grootte er dus wel degelijk toe te doen.

Secret keys kunnen worden gebruikt in combinatie met Diffie-Hellman, een algoritme dat kan worden toegepast voor het distribueren van sleutels, maar niet voor encryptie. Dit algoritme werd midden jaren '70 van de vorige eeuw ontwikkeld door Whitfield Diffie en Martin Hellman en stelt twee partijen in staat om door het uitwisselen van berichten in real-time een tijdelijke sessiesleutel te genereren. Het algoritme is gebaseerd op het gebruik van een zeer groot priemgetal. Is de tijdelijke sessiesleutel eenmaal vastgesteld, dan kan deze worden gebruikt voor conventionele secret key-encryptie. Voor een derde partij die de berichtenuitwisseling volgt, is het niet eenvoudig om te bepalen welke sessiesleutel wordt gehanteerd. Het Diffie-Hellman algoritme is beschikbaar in het public domein, waarmee het een voor de hand liggende kandidaat is voor gebruik in e-commerce oplossingen.

INTEGRITEITSCONTROLE Integriteitscontrole zorgt er voor dat de informatie onderweg niet is veranderd. In het vorige artikel hebben we gezien dat hiervoor doorgaans checksums of hashing-functies worden gebruikt.

NON-REPUDIATION Non-repudiation werkt, zodra de persoon is geautoriseerd en geauthentiseerd, op het niveau van het individuele bericht. Non-repudiation biedt een manier om elk bericht te authentiseren en de deelnemers te beschermen tegen vervalste berichten of ontkenning van verzending of ontvangst van een bericht.

In het vorige artikel zagen we dat er op dit moment slechts één breed gebruikt mechanisme is voor het implementeren van non-repudiation, en dat dit mechanisme gebruikmaakt van op public en private key-gebaseerde encryptiemethoden.

In dit geval gebruikt de persoon zijn private sleutel om elk bericht te vercijferen. Dit vormt dan het bewijs dat het bericht alleen van die persoon afkomstig kan zijn, en dient dus als handtekening onder het bericht. De ontvanger ontcijfert het bericht met behulp van de public key van de verzender.

Aangezien de persoon die het bericht verstuurt ook kan vragen om authentisering van de server, kan hij of zij elk bericht van de server ook behandelen alsof het ondertekend is.

AUDITS EN LOGBOEKFUNCTIES

Audit-mechanismen zijn, als ze al worden meegeleverd, veelal specifiek voor de middlewareleverancier in kwestie. Veel middlewareleveranciers gebruiken bijvoorbeeld een audit-logboek om alle aan beveiliging gerelateerde activiteiten te kunnen vastleggen. Dit levert een audit-spoor op van beveiligingsgebeurtenissen waarin gegevens zoals datum, tijd, gebruiker, object en gebeurtenis zijn vastgelegd.

DE SECURITY 'API' Eigenlijk zou het niet nodig moeten zijn de beveiligingsdienst te 'programmeren'. Het enige wat nodig is, is communicatie tussen de beveiligingsdiensten en de applicatie, teneinde deze diensten te kunnen voorzien van de informatie die ze nodig hebben en ze de resultaten van hun controles terug te kunnen laten rapporteren. Als het product dat u gebruikt over een omvangrijke beveiligings-API beschikt, is er iets grondig mis.

Waar hebben we een API nodig?

Als de applicatieclient wordt geauthentiseerd, moeten gebruikersnaam en wachtwoord (of welk ander mechanisme er ook als identificatie wordt gebruikt) ter controle aan het beveiligingsmechanisme worden doorgegeven, iets dat aan het begin van de verwerking of op het moment van inloggen kan plaatsvinden. Het resultaat van de controle –OK of niet-OK- moet weer aan het programma worden geretourneerd. Dit komt neer op een uiterst simpele aanroep.

Autorisatie dient automatisch te verlopen. Telkens wanneer een aan-

roep wordt gedaan of een bericht wordt verstuurd, zou de client moeten worden gecontroleerd op zijn bevoegdheid om de server te benaderen, een bericht naar de wachtrij te versturen, enzovoort. Er zouden geen rechtstreekse aanroepen richting de beveiligingsdienst moeten plaatsvinden, hoewel in de SEND of CALL API voorzien zou moeten zijn in een foutmelding die aangeeft dat de autorisatiecontrole is mislukt.

Encryptie dient eveneens automatisch te zijn. De beveiligingsdiensten moeten echter wel een signaal krijgen dat een bericht of aanroep moet worden vercijferd. Dit zou kunnen gebeuren via een parameter in de CALL of SEND berichtopdracht zelf, of in een aankruisvakje in de grafische tools waarmee de applicatie wordt ontwikkeld.

Integriteitscontrole moet volkomen automatisch verlopen, terwijl mislukking van de integriteitscontrole moet resulteren in het opnieuw verzenden van het bericht, een functie die de middleware voor zijn rekening neemt. Tussenkomst van een programmeur zou overbodig moeten zijn.

Non-repudiation kan eveneens automatisch verlopen. Hoewel de door mij gegeven beschrijving de indruk zou kunnen wekken dat een persoon al dit digitale handtekeningenwerk doet, zou de programmeur in de praktijk een bericht digitaal moeten ondertekenen met behulp van, bijvoorbeeld, een met de SEND-opdracht meegegeven parameter of een aankruisvakje in de grafische tools waarmee de applicatie wordt gebouwd.

De audit-functie is eveneens geheel automatisch.

In werkelijkheid is er dus vrijwel geen programmeerwerk vereist.

Samenvatting In de afgelopen twee artikelen hebben we gezien dat de beveiligingsdiensten worden gebruikt om de gedistribueerde applicatie te beschermen tegen ongeautoriseerde toegang en gebruik. Er zijn zes hoofdfuncties die algemeen worden erkend als de primaire functies waarover een beveiligingsproduct moet beschikken. Misschien wel sterker dan bij welke andere middlewaredienst dan ook, kan beveiliging worden gerealiseerd met behulp van producten van derden (hoewel het kan zijn dat de middlewareleverancier hiervoor bruikbare standaarddiensten levert):

- **Autorisatie** waarbij het authentiseringmechanisme gebaseerd kan zijn op:
- **Gebruikersnamen en wachtwoorden** functie die door de middlewareleverancier doorgaans standaard wordt meegeleverd.
- **Digitale certificaten** de middleware biedt toegang tot externe producten.
- **Smartcards** de middleware biedt toegang tot externe producten.
- **Biometrica** de middleware biedt toegang tot externe producten.
- **Authentisering** een van de manieren om authentiseringregels te implementeren is om de policyregels uit te geven in combinatie met digitale certificaten (extern aangeleverd). In de praktijk wordt autorisatie in de meeste gevallen echter geïmplementeerd met behulp van Access Control Lists (ACL's), die als regel door de middlewareleverancier als onderdeel van het product zelf worden aangeboden.
- **Privacy** We hebben gezien dat middleware gebruik maakt van asymmetrische of public key-encryptie en symmetrische of secret key-encryptie, die beide waarschijnlijk ondersteund zullen worden door vanuit de middleware toegang te bieden tot producten van anderen.
- **Integriteitscontrole** Dit wordt over het algemeen gerealiseerd met behulp van checksums of hashing-functies. Hiervoor kunnen producten van derden of interne diensten worden gebruikt.
- **Non-repudiation** Er is slechts één breed gebruikt mechanisme voor de implementatie van non-repudiation, dat gebruikmaakt van public en private key-encryptiemethoden waarvoor producten van derden worden ingezet.

> Het systeem waarin elk bedrijf als zijn eigen TTP optreedt kent echter ook nadelen

Bedenk wel dat beveiligingsdiensten automatisch zijn, en de mechanismen waarmee deze dienst wordt geleverd de programmeur in het algemeen geen hoofdbrekens zouden hoeven te bezorgen. Alleen de beheerder, die de schone taak heeft om een harmonieuze verzameling beveiligingsproducten bijeen te brengen, hoeft zich bezig te houden met de vraag welke diensten precies worden ondersteund en op welke manier dat gebeurt.

Rosemary Rock-Evans
is zowel zelfstandig consultant als associate van Ovum en Xephon. Ze is de auteur van meer dan tien boeken over middleware, waaronder 'Ovum Evaluates Middleware' (www.ovum.com). Ze heeft meer dan twintig jaar ervaring in de computerwereld als consultant, schrijver en spreker op congressen.

Nederlandse vertaling van dit artikel door Joost Mulder.