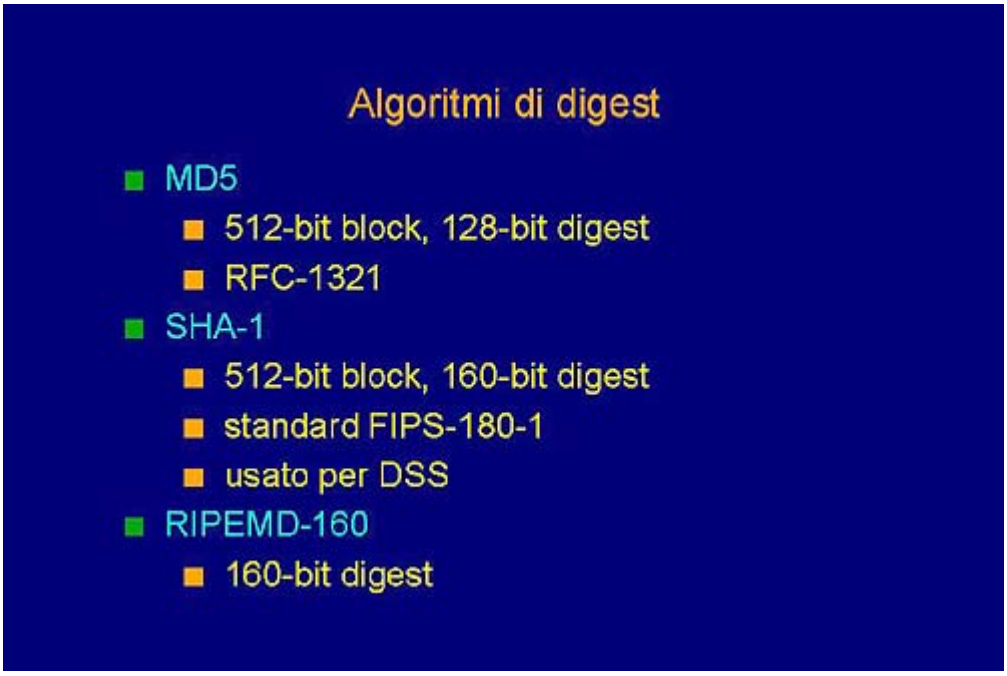
Autenticazione ed integrità dei dati
Integrità dei messaggi

Una delle proprietà di sicurezza che desideriamo avere all'interno di un sistema informatico, è quella che riguarda l'integrità dei messaggi che vengono scambiati e dei dati. Questo perché, anche nel caso che noi cifriamo una comunicazione e quindi le persone che eventualmente le intercettassero non possono leggerlo, possono ciononostante cambiare dei bit della comunicazione cifrata e questo può dar luogo a dei messaggi decifrati di tipo completamente imprevedibile. È chiaro che se chi riceve il messaggio che è stato decifrato in maniera scorretta è un essere umano, probabilmente se ne accorgerà e chiederà che il messaggio venga ritrasmesso. Ma nel caso, invece, che il messaggio sia destinato ad un sistema di elaborazione, che automaticamente deve svolgere delle procedure o delle operazioni, se i dati su cui lavora sono sbagliati, è molto probabile che anche il lavoro che lui cercherà di svolgere sarà di tipo sbagliato e potrebbe addirittura causare dei danni seri a dei sistemi fisici.

Message digest (hash)

Per evitare che i dati vengano danneggiati, sia durante la trasmissione sia mentre sono parcheggiati su disco, si tende ad utilizzare dei codici di protezione basati sui cosiddetti algoritmi di message digest o algoritmi di hash. L'algoritmo di message digest è a tutti gli effetti un riassunto del contenuto del messaggio che si vuole proteggere. È quindi possibile confrontare il riassunto costruito a partire dal messaggio originale con il riassunto ottenuto dal messaggio che è stato trasmesso e vedere se questi due sono identici. Un buon algoritmo di message digest deve essere veloce da calcolare per non sovraccaricare troppo i sistemi su cui viene utilizzato e, soprattutto, deve essere difficile da invertire, ossia deve essere pressoché impossibile, partendo dal digest, ricostruire il messaggio originale. Molto spesso gli algoritmi di digest vengono utilizzati non soltanto per proteggere i dati, ma anche in unione con algoritmi a chiave pubblica, perché, visto che questi algoritmi sono lenti, facendoli operare non sui dati originali ma sul loro riassunto, questi algoritmi diventano più veloci.

Algoritmi di digest

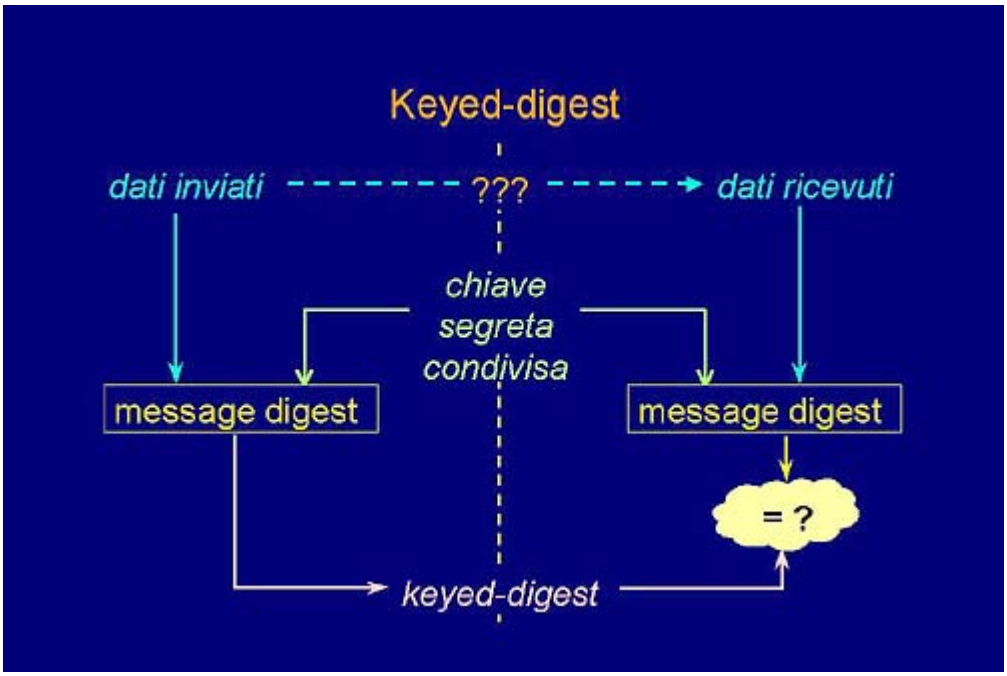


Esistono moltissimi algoritmi di digest, quelli più usati sono attualmente tre: l'algoritmo MD5, opera su blocchi dati da 512 bit e genera un riassunto da 128 bit. L'algoritmo SHA-1, che è quello utilizzato dal governo americano all'interno del suo sistema di firma digitale, opera anch'esso su blocchi da 512 bit, ma genera un digest di dimensione maggiore, da 160 bit. Talvolta viene anche utilizzato il digest RIPEMD-160, che genera digest da 160 bit ed è stato sviluppato in Europa all'interno del progetto RIPE. In generale è opportuno utilizzare algoritmi che generino dei riassunti lunghi, perché si può dimostrare matematicamente che più il riassunto generato è lungo, ossia composto da un maggior numero di bit, e meno sono le informazioni che si perdono, maggiore è la resistenza agli attacchi crittografici che possono essere condotti.

MAC, MIC, MID

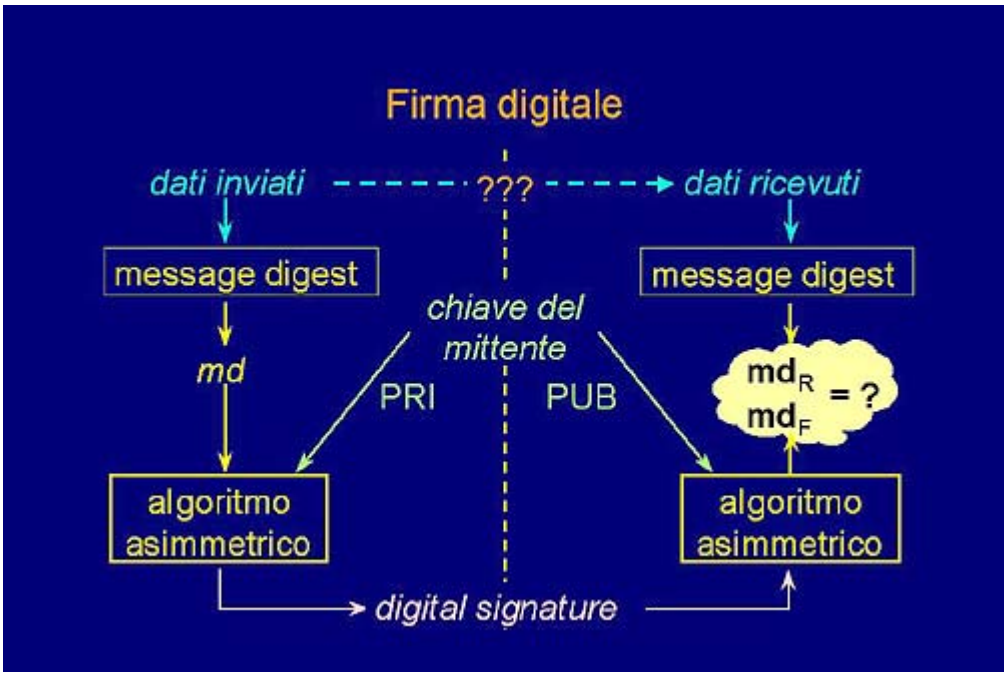
In generale questo riassunto, in una forma opportuna, viene aggiunto al messaggio. Normalmente questa aggiunta prende il nome di MIC, oppure di MAC. Si chiama MIC nel caso in cui chi ha aggiunto questo codice al messaggio, che viene trasmesso o memorizzato, intendesse sottolineare maggiormente il fatto che questo codice permetta di garantire l'integrità del messaggio. Ma poiché spesso l'integrità non è utile senza avere simultaneamente l'autenticazione di chi ha generato i dati, il codice talvolta viene chiamato MAC (Message Authentication Code), volendo così sottolineare che fornisce non solo integrità ma anche autenticazione dei dati. Visto che stiamo aggiungendo un codice ai nostri dati, generalmente quello che capita è di utilizzare questo codice per introdurre anche dei dati aggiuntivi, tipicamente un message identifier, ossia un numero di serie che identifichi questo come il messaggio, ad esempio, numero 27 e poi 28, 29 e 30. In questo modo siamo in grado di parare attacchi di tipo replay, in cui un medesimo messaggio venga inviato più volte, o attacchi di tipo cancellazioni, in cui un messaggio venga cancellato dal normale flusso dei dati.

Keyed digest



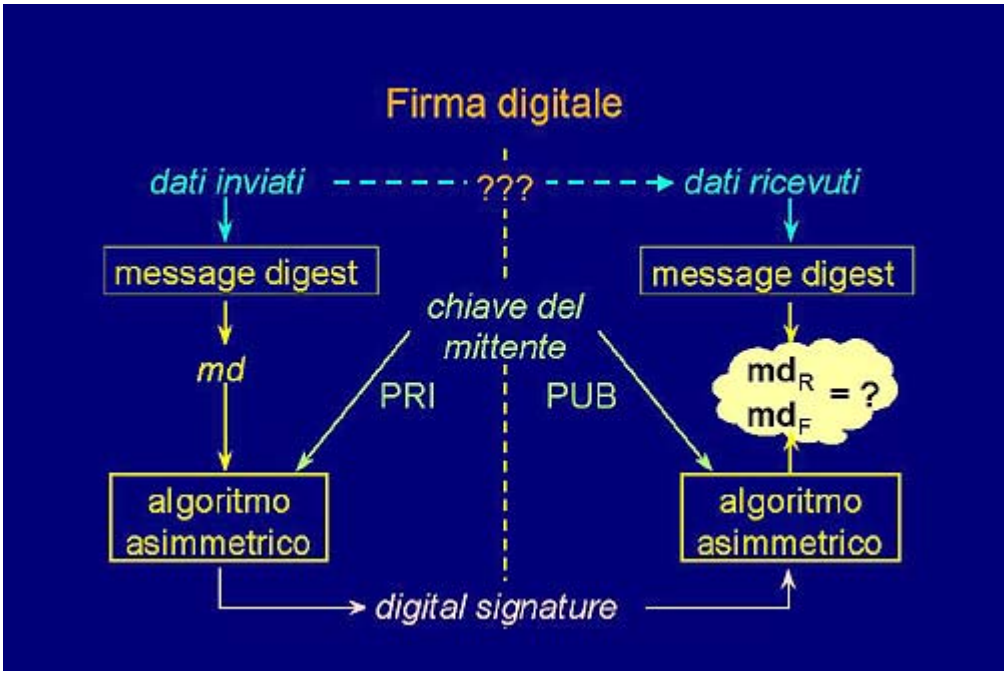
Ovviamente non ha nessun senso spedire direttamente un messaggio e il suo riassunto, perché un eventuale attaccante provvederebbe sia a cambiare il messaggio sia a ricalcolare il riassunto, in modo da non farsi accorgere dell'azione che ha appena svolto. È quindi chiaro che si manda il messaggio, ma bisogna mandare il digest sotto forma protetta, in particolare esistono due principali tecniche per proteggere il digest. Quando le prestazioni sono di grande importanza, ad esempio nella trasmissione dei dati su reti molto veloci, occorre avere delle funzioni di digest altrettanto veloci e delle funzioni di protezione che hanno la medesima proprietà. In questo caso si utilizza il cosiddetto keyed-digest: vengono inviati i dati e una volta ricevuti c'è un ragionevole dubbio se qualcuno ha manipolato i dati durante la loro trasmissione. Per effettuare questo controllo si procede come segue: i dati inviati sono anche passati dentro un algoritmo di digest assieme a un segreto, che è condiviso tra mittente e destinatario. Il risultato dell'algoritmo dipende sia dai dati sia dalla chiave e costituisce il cosiddetto digest con chiave (digest protetto con chiave o keyed-digest). Chi riceve i dati svolgerà la medesima operazione sui dati ricevuti, ossia li introdurrà nell'algoritmo di digest assieme alla chiave segreta nota anche a lui e otterrà il riassunto calcolato sui dati ricevuti. Dopodiché, confronterà se questo riassunto calcolato sui dati ricevuti è uguale al keyed-digest che ha ricevuto, se le due cose sono uguali possiamo essere certi che i dati sono corretti, nessuno li ha manipolati ed arrivano dalla persona che condivide con noi questa chiave segreta. Nel caso, invece, che i due digest non coincidano, purtroppo non possiamo capirne la causa, ossia non riusciamo a distinguere il caso in cui i dati siano stati manipolati dal caso in cui sia stato manipolato, o falsificato, il digest. Non è possibile distinguere queste due cause di fallimento.

Firma digitale 1



Se l'integrità dei dati ci serve non soltanto per nostro uso interno, ma ci serve per poter dimostrare formalmente in una causa legale, in tribunale, o comunque in modo inoppugnabile, che certi dati sono stati prodotti da una certa persona, allora il digest viene normalmente protetto non mediante una chiave condivisa, ma mediante una chiave tipica di ciascun individuo, ossia con la sua chiave privata e quindi utilizzando algoritmi di tipo asimmetrico. Questo costituisce a tutti gli effetti la vera e propria firma digitale secondo gli standard legali esistenti in Italia e in molti altri paese del mondo. La firma digitale opera come segue: prendiamo i dati e li inviamo a qualcuno, questa persona ha il ragionevole dubbio che i dati siano stati manipolati durante la trasmissione. Per rassicurarla quello che possiamo fare è far passare i dati attraverso un algoritmo di digest e così calcolare il suo riassunto. Dopodiché, prendere questo riassunto e cifrarlo mediante la nostra chiave privata. Questo costituisce tecnicamente la firma digitale del messaggio, ossia la firma digitale del messaggio è il suo riassunto cifrato con la chiave privata del mittente.

Firma digitale 2

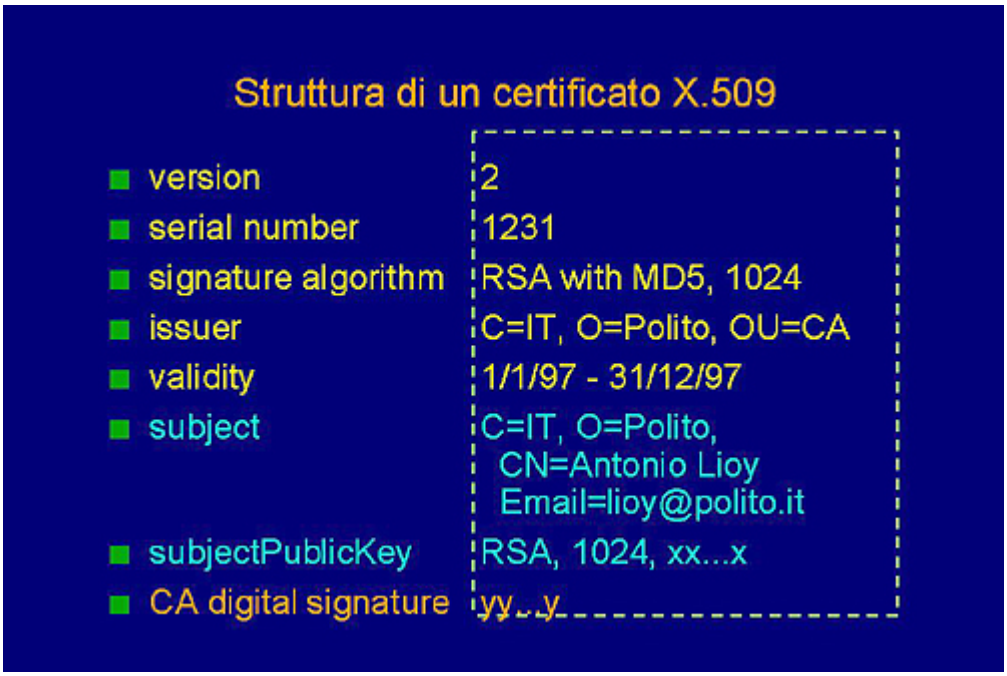


Chi riceve i dati e riceve anche la firma digitale, può verificare se i dati sono corretti attuando il seguente procedimento: prende i dati e ne calcola il digest, ottenendo il riassunto calcolato sui dati ricevuti. Dopodiché, a partire dalla firma digitale ottiene il digest estratto dalla firma, perché in grado di decifrarla utilizzando la chiave pubblica del mittente. Se questi due digest coincidono, allora abbiamo la certezza di due cose: che i dati ricevuti sono integri e che questi dati sono stati generati dalla persona che possiede la chiave privata corrispondente alla chiave pubblica da noi utilizzata. Se invece i due digest non coincidono, non abbiamo modo di capire la causa: può essere stato causato da manipolazioni durante la trasmissione dei dati, da una manipolazione durante la trasmissione della firma digitale, oppure semplicemente dal fatto che abbiamo usato la chiave pubblica sbagliata, ossia i dati non arrivano da chi noi crediamo essere l'originatore del messaggio. Sottolineiamo una volta di più che la firma digitale dipende sia dai dati che dal mittente. In particolare, se per caso i dati vengono cambiati anche soltanto in una parte non significativa, come ad esempio un aumento del numero di spazi tra due caratteri, la firma digitale immediatamente ci avverte di questa manipolazione.

Certificato a chiave pubblica

Per sapere a chi appartiene una certa chiave pubblica, per poter ad esempio attribuire con certezza una firma digitale ad una persona, bisogna disporre di un certificato a chiave pubblica. Il certificato a chiave pubblica è definito come una struttura dati che lega in modo sicuro una chiave pubblica con alcuni attributi. Siccome tipicamente questi documenti con firme digitali vengono usati da esseri umani, è abbastanza tipico che il certificato leghi la chiave con l'identità della persona. Ma poiché le chiavi pubbliche e private possono essere utilizzate da qualunque elemento attivo di un sistema di elaborazione, sono possibili anche altre associazioni. Ad esempio si può dire che una certa chiave pubblica corrisponde alla chiave privata utilizzata da un nodo di elaborazione che ha un certo indirizzo IP. Per evitare che il certificato a chiave pubblica venga manipolato dopo che è stato emesso, viene firmato tramite firma digitale dall'emettitore. L'emettitore si chiama autorità di certificazione ed è l'autorità che si fa garante dell'associazione fra chiave pubblica e attributi. Un certificato digitale ha delle proprietà simili a quelle di molti altri certificati, tipo carta d'identità o passaporto che dir si voglia. Ad esempio, ha una scadenza temporale. Inoltre è revocabile sia da chi ha emesso il certificato sia dall'utente, nel caso che qualcosa sia cambiato, ad esempio nel caso di furto della chiave privata.

Struttura di un certificato X.509

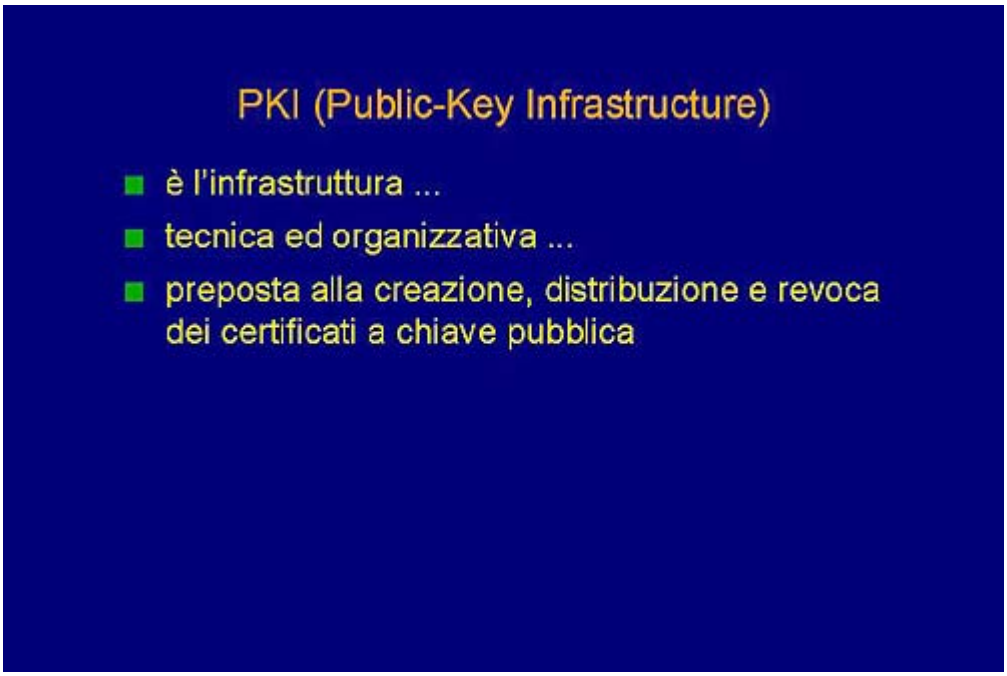


I certificati a chiave pubblica oggigiorno più utilizzati sono i certificati che seguono lo standard X.509. In particolare lo standard X.509 versione 3. Qui viene riportato un esempio di alcuni dati contenuti in un certificato X.509 versione 3. All'inizio ci sono alcuni dati tecnici: tipo la versione (nonostante qui ci sia scritto 2 questo è un certificato versione 3, perché la numerazione parte da 0), alcuni dati tecnici sull'algoritmo con cui è stato firmato questo documento (algoritmo RSA applicato ad un digest MD5) e la chiave RSA era da 1024 bit. Questo certificato ha numero di serie 1231 per distinguerlo da altri certificati emessi dallo stesso emettitore. Gli emettitori molto spesso sono identificati tramite una notazione che inizia con C, ad indicare country = nazione (Italia), prosegue con O per dire organizzazione (Politecnico di Torino) e termina con OU, unità organizzativa, in particolare qui si sta parlando dell'autorità di certificazione del politecnico di Torino. Ogni certificato ha una validità, che è espressa tramite la data di inizio e la data di fine, durante la quale la CA si fa garante dell'associazione. L'associazione è specificata subito dopo. In questo caso la CA si fa garante che questa chiave pubblica RSA da 1024 bit, i cui bit sono elencati qui di seguito, corrisponde alla persona chiamata Antonio Lioy (CN indica nome comune del politecnico di Torino in Italia) e si fa anche garante che questa persona corrisponde all'indirizzo di posta elettronica lioy@polito.it. Per evitare che dopo l'emissione qualcuno cambi uno qualunque di questi dati, la CA appone la sua firma digitale e questo rende il certificato inalterabile, o meglio, se il certificato venisse alterato la manipolazione verrebbe immediatamente rivelata.

X.509 CRL (Certificate Revocation List)

Può capitare che a una persona venga rubata, o la persona smarrisca, la propria chiave privata. In questo caso non è più vero quanto attestato dal certificato, non è più vero che il certificato contiene la chiave pubblica corrispondente alla chiave privata di una certa persona. In questo caso il certificato deve essere dichiarato non più valido. Il metodo con cui i certificati vengono dichiarati non più validi è quello di includerli in una lista chiamata CRL. La CRL è la lista dei certificati revocati, ossia dei certificati non più validi. Le CRL sono numerate, anzi gli viene attribuito una data, questa è la CRL del 30 settembre del 1999 e dice che il certificato numero 104 non è più valido a partire dal 25 aprile 1998, perché si è rotta la smart-card entro cui la chiave privata era contenuta. Invece il certificato 97 non è più valido a partire dal 30 agosto 1999, a causa di un furto del portatile che conteneva la chiave privata. Per evitare che qualcuno emetta delle CRL false, anche la CRL viene firmata dalla autorità di certificazione che ha emesso la CRL. Si noti che l'autorità di certificazione agisce in modo abbastanza piratesco, nel senso che l'autorità di certificazione emette questo elenco di certificati non più validi, ma demanda agli utilizzatori dei certificati il controllo. Ogniqualvolta si riceve una firma digitale accompagnata da un certificato a chiave pubblica, è compito di chi riceve questa firma andare a verificare che il certificato associato sia ancora valido, ossia non sia inserito all'interno di una lista di certificati revocati.

PKI (Public-Key Infrastructure)



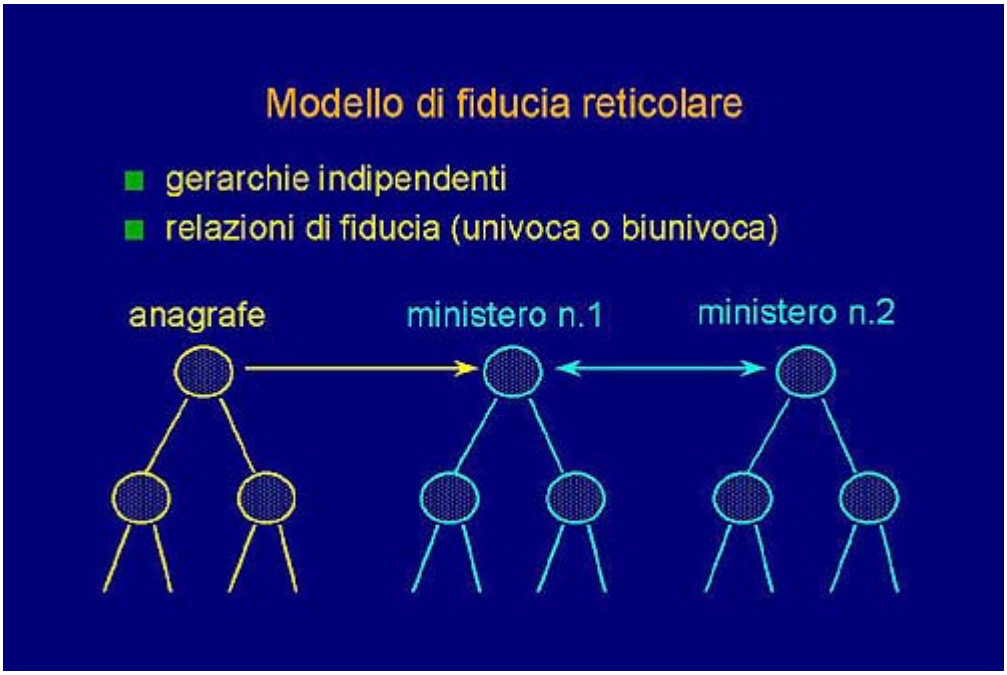
Ovviamente bisogna creare una infrastruttura, chiamata PKI (infrastruttura a chiave pubblica), che è una infrastruttura tecnica ed organizzativa per creare, distribuire e revocare questi certificati. Questo ci permetterà di sfruttare a pieno tutti i benefici degli algoritmi asimmetrici, ossia degli algoritmi a chiave pubblica. In particolare una PKI è costituita da una o più autorità di certificazione.

Certification Authority (CA)

L'autorità di certificazione è formalmente, diciamo tecnicamente, un'autorità in cui un insieme di utenti ripone la propria fiducia per creare e assegnare certificati a chiave pubblica. Nello standard X.509 non viene specificata nessuna relazione fra l'autorità di certificazione, l'emettitore e il soggetto specificato nel certificato. Esistono però delle norme di legge che dicono che se una firma digitale deve avere valore legale nei confronti di chiunque, questa deve essere emessa non da una generica autorità di certificazione, ad esempio interna ad un'azienda, ma ad un'autorità di certificazione che risponde a certi requisiti legali. È ovviamente possibile creare sia delle autorità di certificazione indipendenti, che permetteranno una comunicazione sicura solo fra tutti gli utenti che si fidano di

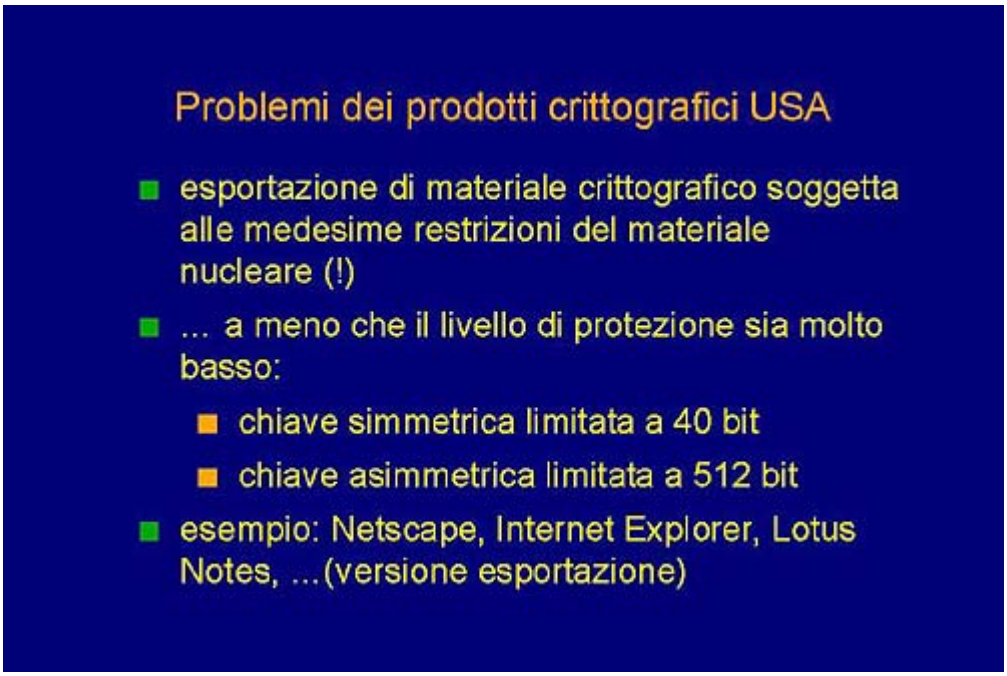
quella particolare autorità di certificazione, oppure creare delle reti di autorità di certificazione, ossia delle autorità di certificazione che si riconoscono reciprocamente. Il modo più semplice di organizzare queste reti è in strutture gerarchiche: ossia permettere a un'entità suprema, una CA di primo livello, di certificare quindi di asserire la sua fiducia in autorità di certificazione di secondo livello e così a scendere fino alle autorità di certificazione di più basso livello, che emetteranno certificati per i nostri utenti.

Modello di fiducia reticolare



Una autorità di certificazione gerarchica sarebbe l'ideale dal punto di vista tecnico. Purtroppo, a causa degli innumerevoli litigi che esistono fra gli esseri umani, non sempre è possibile accordarsi su un'unica struttura di certificazione. Oggigiorno la situazione reale delle infrastrutture di certificazione corrisponde piuttosto allo schema che si vede qui illustrato. Esistono delle autorità di certificazione gerarchiche limitatamente a certe funzionalità o certe aziende. Dopodiché esisteranno tante gerarchie: nel caso che un messaggio dotato, ad esempio, di firma digitale viaggi da una gerarchia all'altra, non verrebbe riconosciuto automaticamente. Per farlo riconoscere bisogna che i gestori di queste infrastrutture di certificazione stabiliscano delle relazioni di fiducia. Le relazioni di fiducia possono essere univoche o biunivoche. Per esempio, l'anagrafe potrebbe fidarsi univocamente dei certificati emessi dal ministero n.1, ma questo ministero potrebbe non aver fiducia nei certificati emessi dall'anagrafe. Viceversa, fra due ministeri appartenenti al medesimo governo si potrebbe avere una relazione di fiducia reciproca e quindi, teoricamente, potrebbero essere collocati sotto una unica autorità di certificazione centralizzata.

Problemi dei prodotti crittografici USA



Terminiamo questa chiacchierata segnalando un particolare problema, che più che essere un problema di tipo tecnico è un problema di tipo politico e organizzativo. Gli stati Uniti, come ben si sa, sono i leader mondiali nel campo della tecnologia informatica e anche nel campo della sicurezza offrono sicuramente alcuni fra i migliori prodotti al mondo. Purtroppo il governo americano ritiene che l'esportazione di prodotti crittografici possa danneggiare gravemente la propria sicurezza e quindi impedisce alle proprie aziende di esportare prodotti che contengano sistemi di sicurezza forti. Le aziende sono libere di esportare questi prodotti solo ed esclusivamente se abbassano il livello di protezione. In particolare si impone un limite di 40 bit sulla chiave simmetrica utilizzata da questi prodotti e un limite di 512 bit sulle chiavi asimmetriche. Questo significa che anche quei prodotti di largo uso in tutti i paesi del mondo, tipo i browser Netscape o Explorer e i sistemi di workflow quali Lotus Notes, che contengono al loro interno dei sistemi di sicurezza, sono tutti quanti in versione esportazione. Non è la versione identica a quella che si trova negli Stati Uniti, ma sono delle versioni molto deboli perché, per quanto abbiamo detto nelle lezioni precedenti, delle chiavi simmetriche inferiori a 64 bit, o delle chiavi asimmetriche inferiori a 1024 bit, sono da reputarsi altamente insicure.

Step-up (o gated) cryptography

Nel dicembre 1998 il governo americano ha leggermente cambiato la propria opinione in materia e ha permesso l'esportazione di prodotti di sicurezza forti, ossia pari a quelli disponibili in America, nel caso che questi prodotti non realizzino funzionalità di cifratura dati in dispositivi di memorizzazione, ma solo funzionalità di sicurezza durante il transito di dati in rete. Quindi ci permettono di proteggerci dagli attacchi condotti in rete, non dagli attacchi o dalle indagini condotti direttamente sui nostri calcolatori. Il vincolo è caduto e quindi ora è possibile esportare prodotti ad alta sicurezza di canale per proteggere la privacy dei dati, o i siti di commercio elettronico. Tutto questo però, solo con destinatari specifici, come filiali estere di aziende statunitensi, organizzazioni finanziarie, banche, assicurazioni e organizzazioni sanitarie. Esiste però un ulteriore vincolo: questi prodotti vengono venduti e diventano operativi solo se utilizzano certificati X.509 emessi dal fiduciario del governo statunitense. Il fiduciario del governo statunitense è l'autorità di certificazione chiamata VeriSign. Quindi, se qualcuno appartiene a una di queste categorie qui elencate e desidera comprare un prodotto di questo genere, non può dotare quel prodotto di un certificato X.509 emesso dalla propria autorità di certificazione, ma deve necessariamente acquistare, pagandolo, un certificato X.509 generato da VeriSign, la quale si farà carico della verifica del rispetto delle regole di esportazione imposte dal governo statunitense.

Il formato PEM

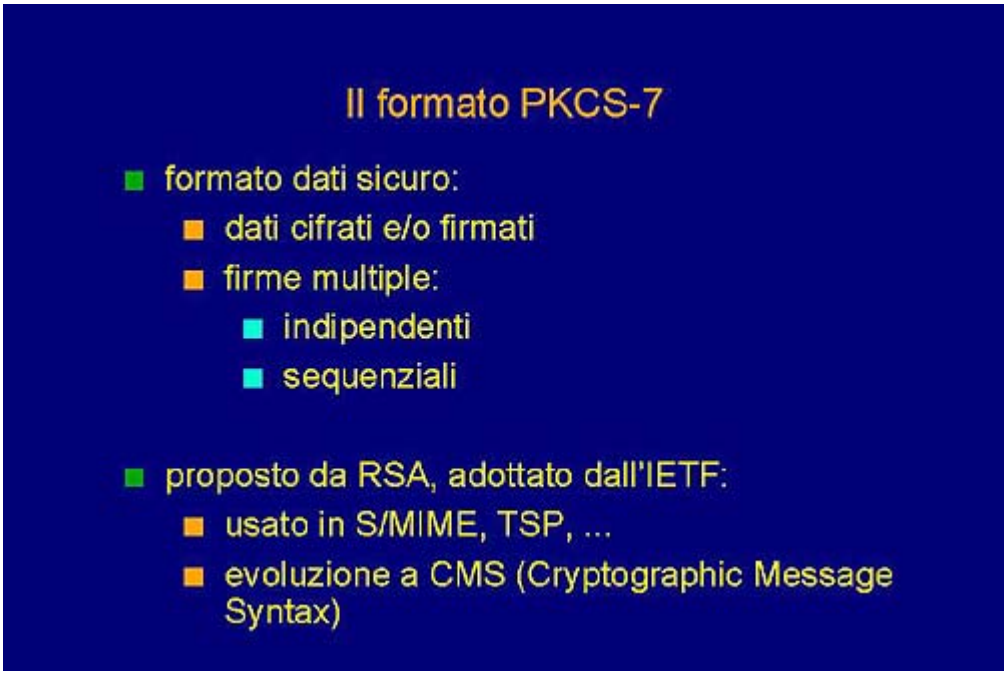
In generale, quando si parla di firma digitale si sta parlando di qualcosa di astratto. Se vogliamo applicare la firma digitale a dei documenti elettronici, quindi a dei file, tipicamente bisogna anche definire il formato con cui la firma digitale viene introdotta all'interno. Esistono vari formati, il primo definito nel passato è stato il formato PEM, che è stato sviluppato per rendere sicuri i messaggi di posta elettronica, ossia per generare messaggi firmati oppure messaggi con firma digitale ed anche cifratura, ossia segretezza. Questo è un formato molto vecchio, attualmente non viene più usato come formato dati, per trasmettere messaggi, ma solamente per salvare e proteggere sui dischi locali le chiavi pubbliche e private di alcuni processi. Ad esempio, il suo tipico utilizzo è all'interno dei server Web sicuri.

Il formato PGP

Un formato alternativo è il cosiddetto formato PGP. PGP è un programma che permette di fare crittografia e firma digitale sia dei messaggi di posta elettronica sia di generici file. È un programma molto utilizzato da persone non strutturate: come gli hacker o le persone appartenenti

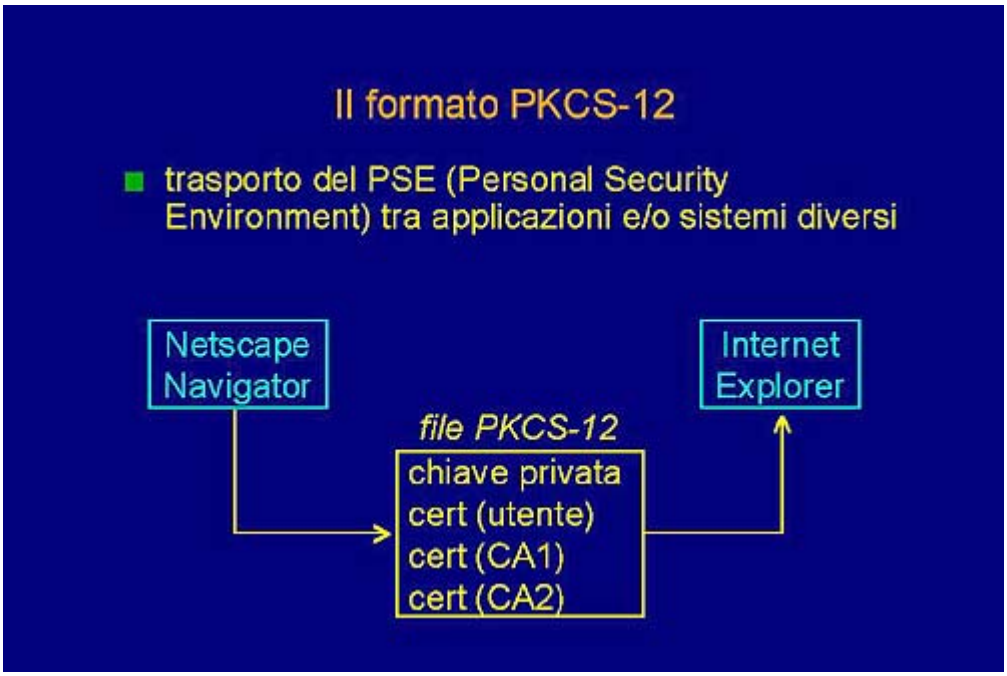
all'underground di Internet o semplicemente coloro che hanno particolarmente a cuore la privacy dei propri dati. Il formato PGP ha delle funzionalità analoghe a quelle di PEM, ossia è in grado di generare dei dati firmati ed eventualmente cifrati. PGP è passato da una versione public domain a una versione sviluppata da una ditta. Nella prima versione gli algoritmi utilizzati erano IDEA, RSA, MD5. Nella nuova versione, che è incompatibile con la precedente, viene utilizzato l'algoritmo 3DES, lo standard di firma digitale DSS e lo scambio chiavi D-H. Uno dei maggiori problemi che si frappongono all'adozione su larga scala del formato e dei sistemi di sicurezza basati su PGP, è il fatto che non utilizza certificati X.509 e questo causa difficoltà nella distribuzione delle chiavi pubbliche usate da PGP.

Il formato PKCS-7



Attualmente il formato dati più utilizzato per apporre firme digitali o per cifrare i dati è il cosiddetto formato PKCS-7. È un formato molto flessibile che permette non soltanto di cifrare o firmare in modo semplice i dati, ma anche di apporre firme multiple, ossia il medesimo dato, il medesimo messaggio, può contenere le firme di più persone. Queste persone possono aver firmato il messaggio ciascuna in modo indipendente dall'altro, oppure possono averlo fatto sequenzialmente, ossia in pratica avendo visto chi aveva già firmato il messaggio in precedenza. Il formato PKCS-7 è stato sviluppato e proposto dalla ditta RSA, ma è stato recentemente adottato da Internet, ossia dall'IETF, l'ente che sviluppa tecnicamente le procedure per Internet. In particolare il formato PKCS-7 è usato nella posta elettronica sicura S/MIME nei protocolli per fornire ora e data certa. Il formato PKCS-7 sta evolvendo alla sintassi chiamata CMS (Cryptographic Message Syntax), che dovrebbe diventare la sintassi standard per la firma digitale e la segretezza dei documenti nei prossimi anni.

Il formato PKCS-12



Infine, nel caso che noi abbiamo ottenuto una chiave privata e una chiave pubblica per utilizzo all'interno di una particolare applicazione o su un determinato calcolatore, si potrebbe porre il problema di come utilizzare questa nostra identità digitale anche su altri sistemi. In questo caso ci viene in aiuto il formato cosiddetto PKCS-12. PKCS-12 è un formato dati che è stato definito per permettere il trasporto del PSE, ossia del mio ambiente di sicurezza personale, fra applicazioni o sistemi diversi. Questo vuol dire, ad esempio, che se io utilizzo le funzionalità di firma digitale in Netscape, posso usare le medesime funzionalità anche in Explorer semplicemente esportando il mio ambiente di sicurezza in un file PKCS-12 protetto. Metto qua dentro, ad esempio, la mia chiave privata, il mio certificato di utente e i certificati delle CA che hanno emesso il mio certificato d'utente. Il tutto è poi fornito in pasto al sistema Explorer, che assumerà quindi la medesima identità digitale già disponibile sul mio sistema Netscape.