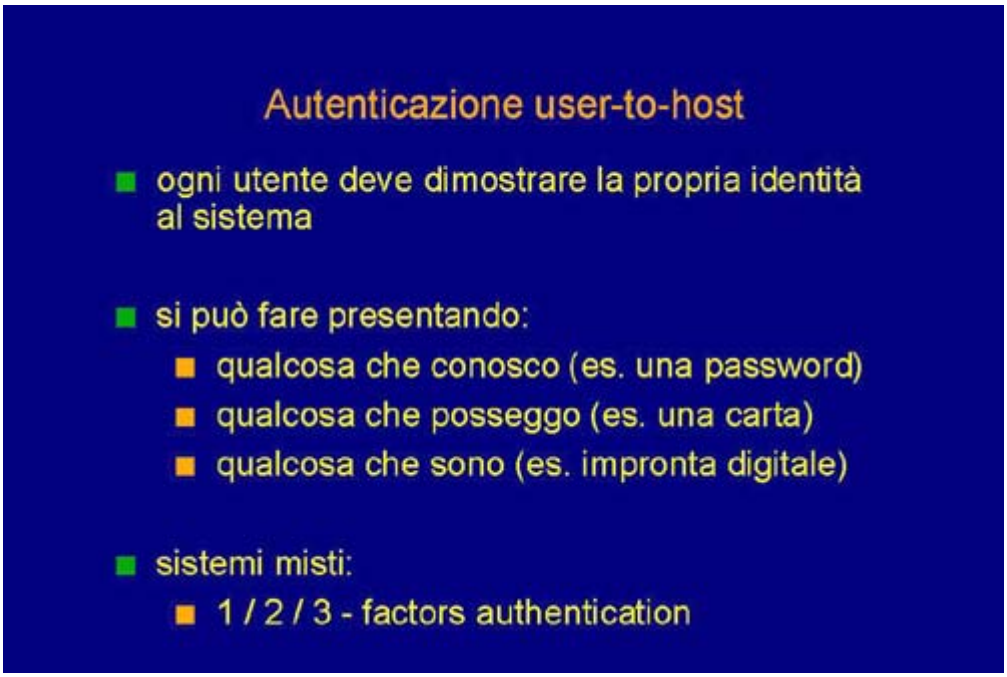
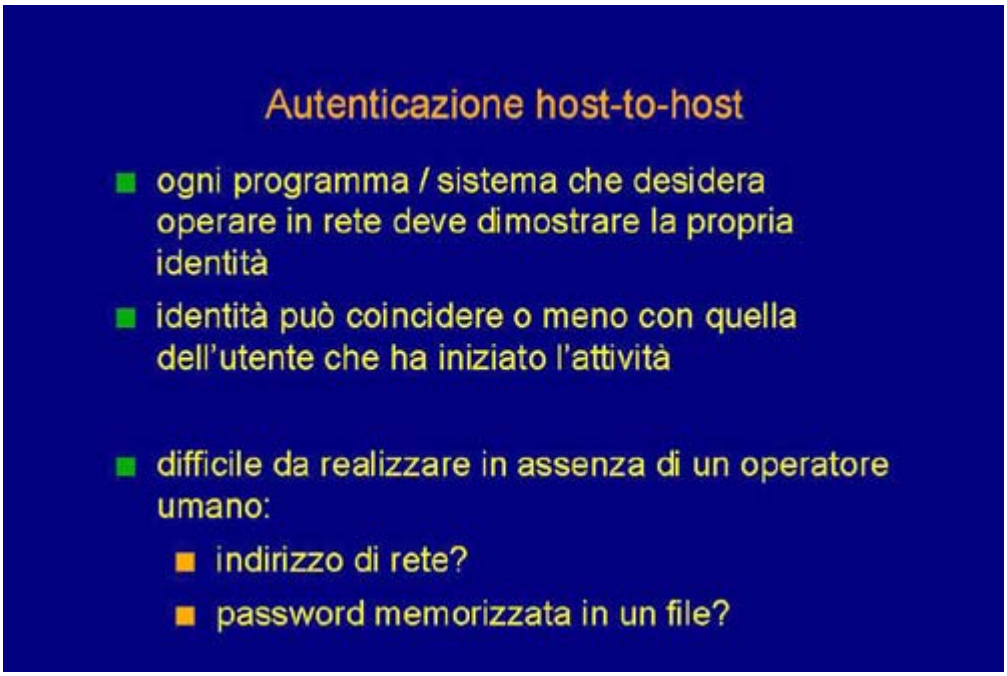
Implementazione di sistemi di autenticazione
Autenticazione user-to-host



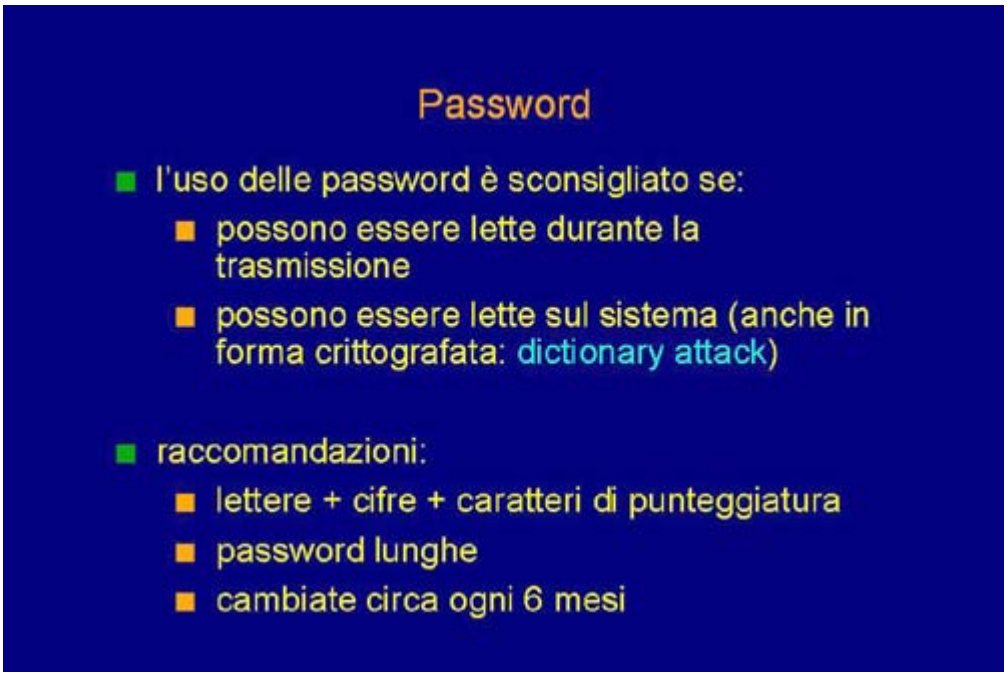
Alla base di tutti quanti i sistemi di sicurezza risiede l'identificazione delle persone e dei programmi che sono abilitati ad operare sul sistema. Abbiamo visto in precedenza che questo prende tecnicamente il nome di autenticazione. In particolare, è possibile avere due tipi di autenticazione all'interno di un sistema informatico. Il primo tipo di autenticazione è quella cosiddetta user-to-host, in cui ogni utente deve dimostrare la propria identità al sistema, ossia deve dimostrare di avere diritto di utilizzare una certa risorsa di calcolo, o di attivare una certa procedura di calcolo. Questo può esser fatto in almeno tre modi diversi. Il modo più banale è quello di dimostrare la propria identità dicendo qualcosa che soltanto quest'utente teoricamente dovrebbe conoscere, banalmente questo è il normale sistema di username e password ed è un sistema non molto sicuro. Per rafforzare la sicurezza di questo sistema potrebbe essere preferibile utilizzare qualcosa che si possiede, ad esempio una carta. Questo significa che l'accesso al sistema viene controllato non soltanto in base allo username e password che vengono digitati su una tastiera, ma viene controllato anche in base ai dati contenuti in una carta di tipo magnetico, o di tipo elettronico, da introdursi nel sistema. Se desideriamo avere delle funzionalità di sicurezza ancora più forte, tipicamente si ricorre a qualcosa che io impersonifico, qualcosa che io sono, ovvero una mia caratteristica fisica. È tipico in questo campo l'uso delle impronte digitali, oppure del tono della voce, o di qualche altra caratteristica fisica: la dimensione della faccia, il colore degli occhi e così via. In generale è possibile avere dei sistemi misti: si parla in questo caso di sistemi a 1, 2, 3 factors authentication, ossia sistemi di autenticazione che, per permettere l'accesso al sistema, richiedono la combinazione di uno, due, oppure tre fattori dei tipi che abbiamo appena menzionato.

Autenticazione host-to-host



All'interno di un sistema informativo però non ci sono soltanto utenti che operano, ci sono anche delle procedure, dei processi e dei sistemi che molto spesso operano in modo automatico anche in assenza di un operatore di tipo umano. Anche questi processi, questi sistemi, devono essere autenticati ed in questo caso si parla di una autenticazione di tipo host-to-host. Non c'è più un essere umano che deve autenticarsi nei confronti di un elaboratore, ma c'è un elaboratore che deve autenticarsi nei confronti di un atro elaboratore per ottenere una funzionalità, un servizio o dei dati. In questo caso, l'identità del programma, del processo, che sta cercando di svolgere questa funzione può coincidere o meno con quello dell'utente che ha iniziato l'attività. È in ogni caso un campo molto delicato e molto difficile da realizzare, soprattutto in assenza di un operatore umano, perché tutte le tecniche che noi abbiamo sviluppato in passato sono basate sulla presenza di un operatore umano, non sono quindi adatte ad autenticare programmi o sistemi. Infatti è stato proposto l'utilizzo degli indirizzi di rete, ossia abilitare la funzionalità di un certo tipo solo per processi, programmi, che provengono da una certa classe di rete. Oppure è stato proposto di memorizzare la password da utilizzarsi all'interno di un file, ma in questo caso la password non sarebbe protetta come dovrebbe esserlo, teoricamente, quando risiede all'interno del cervello umano. Quindi, in conclusione, l'autenticazione host-to-host è uno dei campi più delicati e ancora risolti nel modo meno soddisfacente, nell'attuale schema di sicurezza dei sistemi informativi.

Password

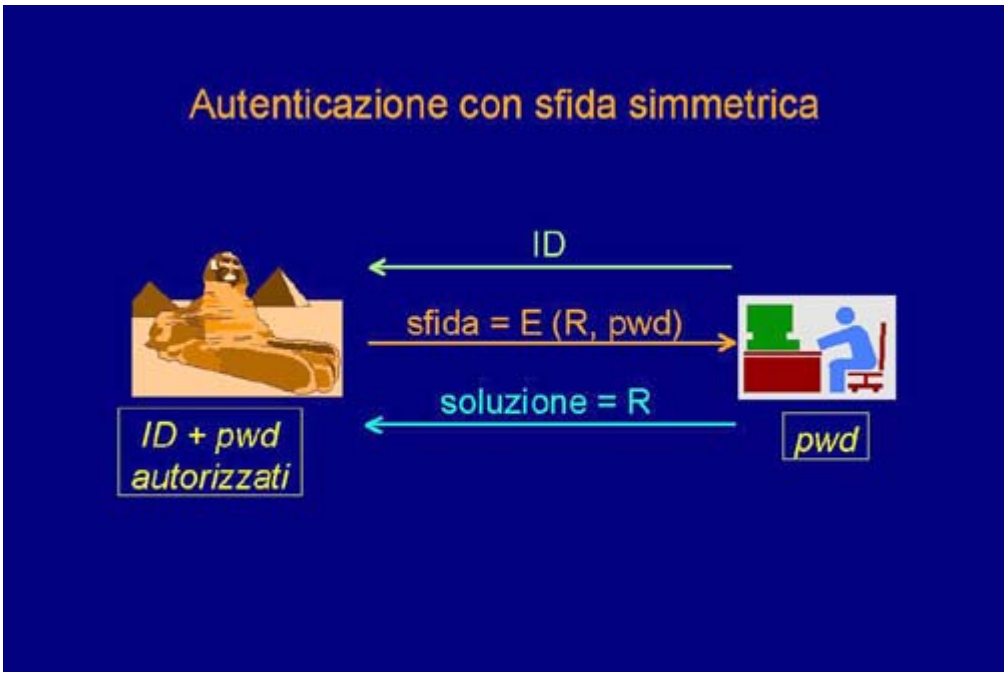


Veniamo adesso ad esaminare un po' più nel dettaglio alcune delle tecniche con cui è possibile autenticare l'utente nei confronti del sistema. La classica tecnica che viene utilizzata è quella della password. L'utilizzo di username e password è fortemente sconsigliabile non in assoluto, ma solo se sussistono alcune possibilità. La prima possibilità è che la password possa essere letta durante la sua trasmissione in rete fra l'utente e il servizio a cui desidera accedere, perché in questo modo può essere facilmente catturata e quindi riutilizzata per fare delle altre sessioni di lavoro non autorizzate. Inoltre le password sono da sconsigliarsi se vengono mantenute su un sistema informativo al cui interno operano delle persone che possono leggere queste password. È chiaro che se le password sono conservate in chiaro in un file è veramente una operazione banale leggerle, ma anche nel caso in cui le password siamo mantenute in forma crittografata, ossia cifrata, ossia segrete, diventano pericolose. È infatti possibile svolgere un attacco di tipo basato su dizionario: questo significa che l'attaccante, dopo aver preso visione di quelle che sono le password crittografate cifrate, prova a cifrare in tanti modi diversi un dizionario di parole più comunemente usate. Siccome gli utenti hanno la tendenza ad usare come password delle parole semplici e di senso compiuto, tipo il proprio nome, il nome dei propri figli, il nome dei propri parenti o del luogo in cui sono stati in villeggiatura durante l'ultimo anno, ecco che questo genere di attacchi sorprendentemente ha sempre un grande successo. Si stima che mediamente qualunque sistema informativo abbia un numero di password banali da indovinare compreso tra il 30 e il 40 per cento delle password totali utilizzate all'interno del sistema. Esistono quindi delle raccomandazioni specifiche da fare agli utenti che siano per qualche motivo obbligati ad usare le password tradizionali. La prima raccomandazione è quella di non utilizzare una password solo di caratteri alfabetici, ma di mischiare al suo interno lettere, cifre e caratteri di punteggiatura. Questo è sufficiente a parare gli attacchi di tipo dizionario, perché ovviamente all'interno di un dizionario ben difficilmente trovano posto parole di questo genere. Inoltre la password deve essere lunga: si consiglia un minimo di 8 caratteri, perché se la password è troppo corta diventa facile cercare di indovinarla. Analogamente non bisognerebbe mantenere la propria password per anni: si dice che mediamente al massimo ogni 4/6 mesi la propria password dovrebbe essere cambiata.

Autenticazione tramite indirizzo IP

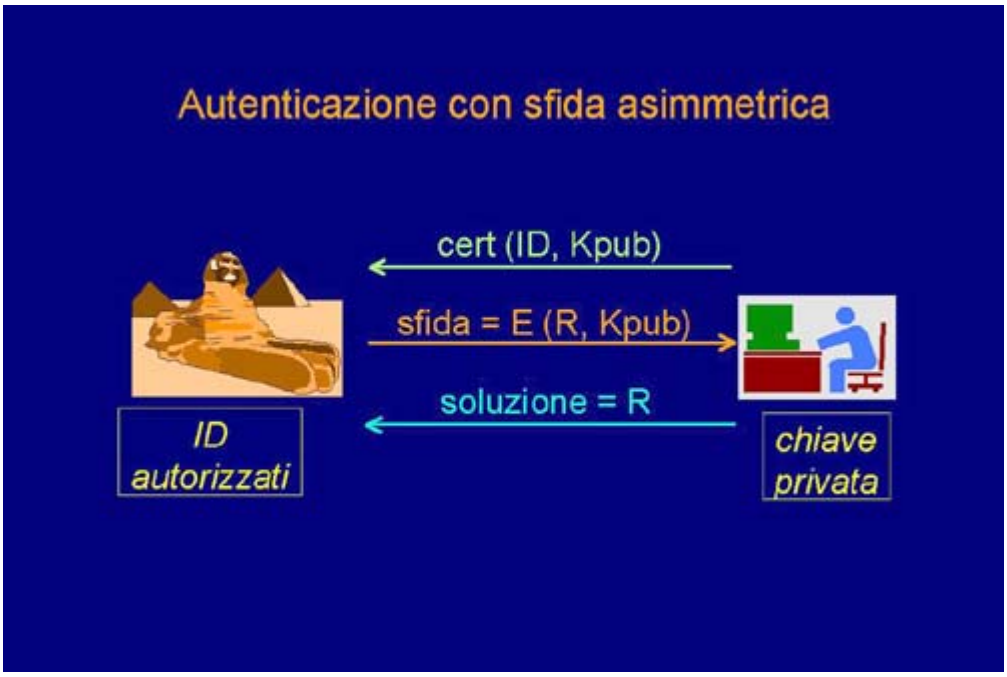
Un'altra possibilità di autenticare sia un utente sia un processo che richiede un servizio, è quello di fare affidamento sugli indirizzi di rete, ossia identificare la postazione di lavoro da cui l'utente sta richiedendo un certo servizio. Questo comunemente viene fatto utilizzando gli indirizzi di rete IP. Ad esempio, in questo schema abbiamo un calcolatore che contiene dati riservati; in particolare questi dati sono riservati alle postazioni di lavoro che hanno indirizzi appartenenti alla classe 130.193. Ciò significa che se un utente è seduto davanti ad un computer che ha un indirizzo di questa classe, come in questo caso (130.193.6.9), allora le sue richieste di servizio verranno esaudite. Ma se il medesimo utente si sedesse di fronte ad un computer appartenente ad una classe diversa, qui abbiamo un computer con indirizzo 140.23.6.9, allora la sua richiesta verrebbe automaticamente rifiutata dal sistema. Questo tipo di autenticazione è sicuramente un tipo di autenticazione debole, perché è molto facile falsificare gli indirizzi IP, è addirittura banale su un normale PC. Un elaboratore dotato del sistema operativo Windows, o anche di un altro sistema operativo, che sia sotto il controllo dell'utente che gli è seduto davanti, permette di impostare a proprio piacimento l'indirizzo IP e a questo punto io potrei impostare quell'indirizzo che mi da accesso a quei dati riservati. Quindi l'autenticazione basata sugli indirizzi di rete è da sconsigliare fortemente in ambienti con grado di sicurezza medio/alto.

Autenticazione con sfida simmetrica



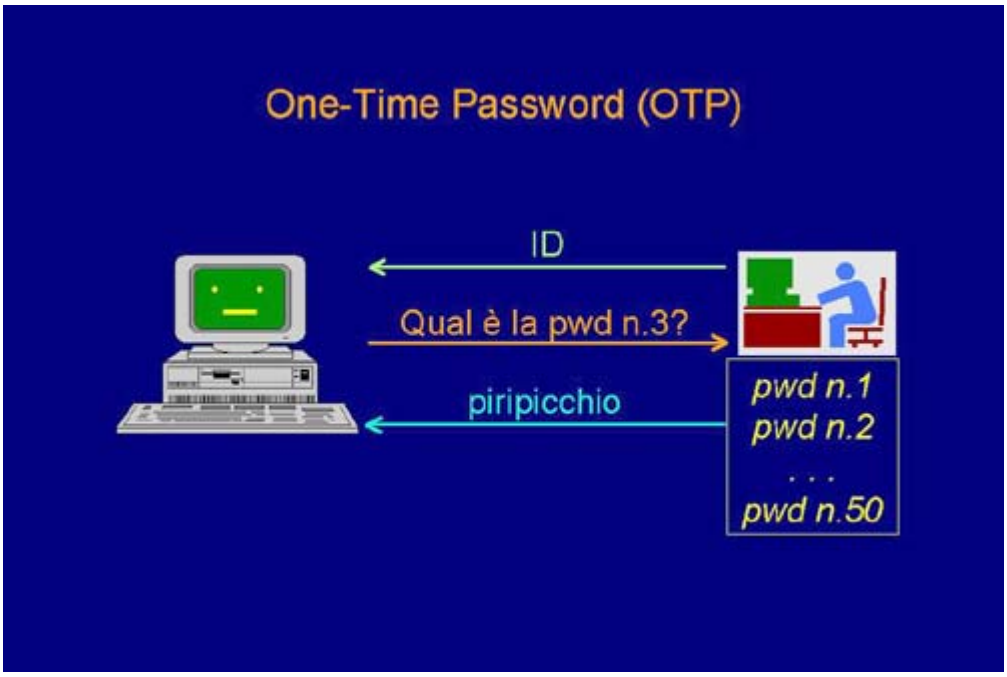
Veniamo ad alcuni sistemi invece che presentano delle caratteristiche di sicurezza molto migliori, per quanto riguarda l'autenticazione. Un modo molto bello di fare autenticazione utilizzando le password senza mostrarle in rete, sono i sistemi cosiddetti di autenticazione a sfida simmetrica. Ipoteticamente l'utente conosce la propria password, che è nota anche al sistema a cui noi vogliamo fare accesso, il quale contiene l'elenco degli identificativi (gli username) e le relative password che sono autorizzate ad accedere al sistema. Quando l'utente desidera accedere al sistema, invia un primo pacchetto in rete che specifica lo username, ossia l'ID, con cui lui desidera accedere al sistema. A fronte di questa richiesta di accesso, il sistema gli invia indietro un messaggio di sfida: banalmente la sfida consiste in un numero random R, un numero casuale, che è stato cifrato (E sta per encrypted) con la password relativa a quell'utente. Se l'utente è veramente chi dice di essere allora è in grado di decifrare questo pacchetto usando la propria password e quindi di rispondere alla sfida mandando la soluzione. La soluzione è semplicemente il numero casuale R che il sistema aveva generato. Se il numero casuale non viene mai più ripetuto in qualunque transazione di tipo futuro, allora questo è un sistema di sicurezza perfetto per quanto riguarda l'autenticazione in rete. Vuol dire che chiunque stia osservando lo scambio di messaggi tra la postazione client e la postazione server, non può da questo scambio di messaggi catturare in alcun modo la password.

Autenticazione con sfida asimmetrica



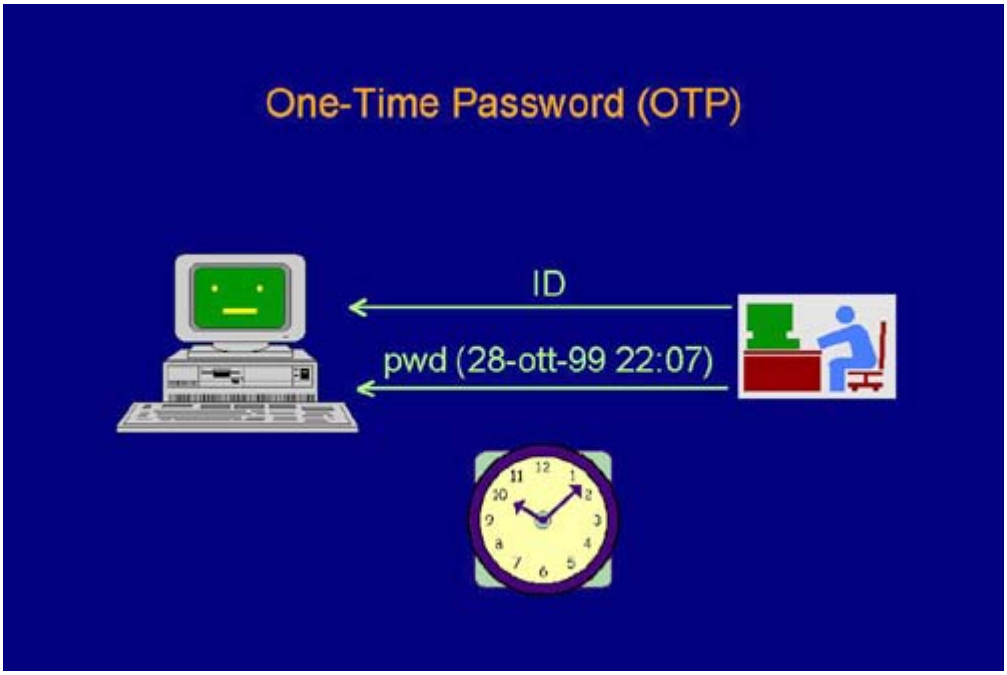
Questo sistema può essere esteso e migliorato utilizzando gli schemi di crittografia asimmetrica. Nello schema di crittografia asimmetrica non è necessario che il sistema cui noi desideriamo collegarci conosca anche lui la nostra password, è sufficiente che lui conosca semplicemente la nostra chiave pubblica. Questi generi di sistemi si basano sul principio che vedete qui illustrato: ad ogni utente deve essere stata distribuita una coppia di chiavi private e chiavi pubbliche, in particolare l'utente dispone sulla sua postazione di lavoro della propria chiave privata. Il sistema cui noi vogliamo accedere memorizza l'elenco degli identificativi (degli username) che sono autorizzati ad accedere al sistema. Quando l'utente desidera accedere manda il proprio certificato a chiave pubblica contenente il proprio identificativo e la propria chiave pubblica. Il sistema, basandosi sulla chiave pubblica estratta da questo certificato, manda una sfida. La sfida consiste nel solito numero casuale R crittografato con la chiave pubblica del destinatario. A questo punto il destinatario, se è veramente chi dice di essere, possiede la chiave privata corrispondente ed è quindi in grado di mandare indietro la soluzione della sfida, che consiste nel decifrare questo pacchetto usando la propria chiave privata e quindi mandare indietro il numero random che era stato generato. Nuovamente, anche in questo caso, chi osserva semplicemente la rete non può in alcun modo desumere quale sia la chiave privata dell'utente e quindi non può in futuro impersonificarlo.

One-Time Password (1/2)



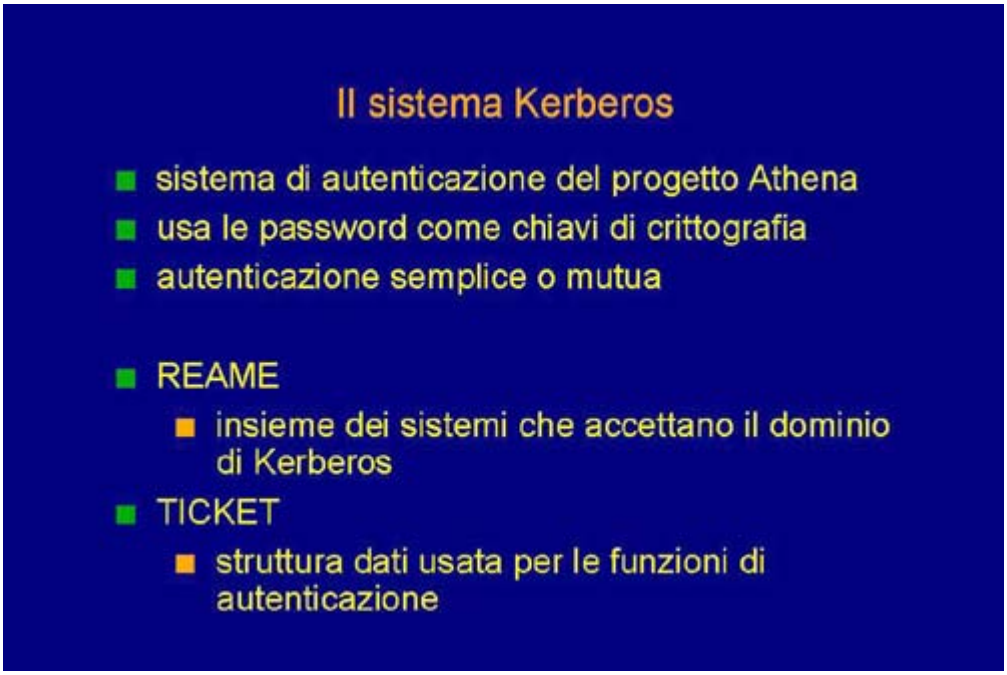
Un altro modo per evitare che le password costituiscano un problema è utilizzare le cosiddette one-time password: le password valide soltanto una volta. Nello schema più semplice di questo tipo, il sistemista, quando abilita un utente ad accedere ad un sistema, gli fornisce un elenco di password numerate. Quando il nostro utente desidera accedere al sistema fornisce il proprio identificativo. Il sistema risponde chiedendo di presentare la password di un certo numero, nel nostro esempio la password numero 3. A questo punto l'utente preleva dalla propria lista la password numero 3 e la comunica in rete. È vero che questa password può essere intercettata e letta da chi ha il controllo della rete, ma se il sistema cui ci stiamo collegando non ci chiederà mai più la password numero 3, questo non ha nessuna influenza. Questo significa che una volta esaurite tutte quante le password che sono state consegnate all'utente, è necessario che l'utente si rechi presso il sistemista per aver un altro elenco di password. Questo fa sì che il sistema sia bello e interessante dal punto di vista teorico ma poco pratico a livello implementativo.

One-Time Password (2/2)



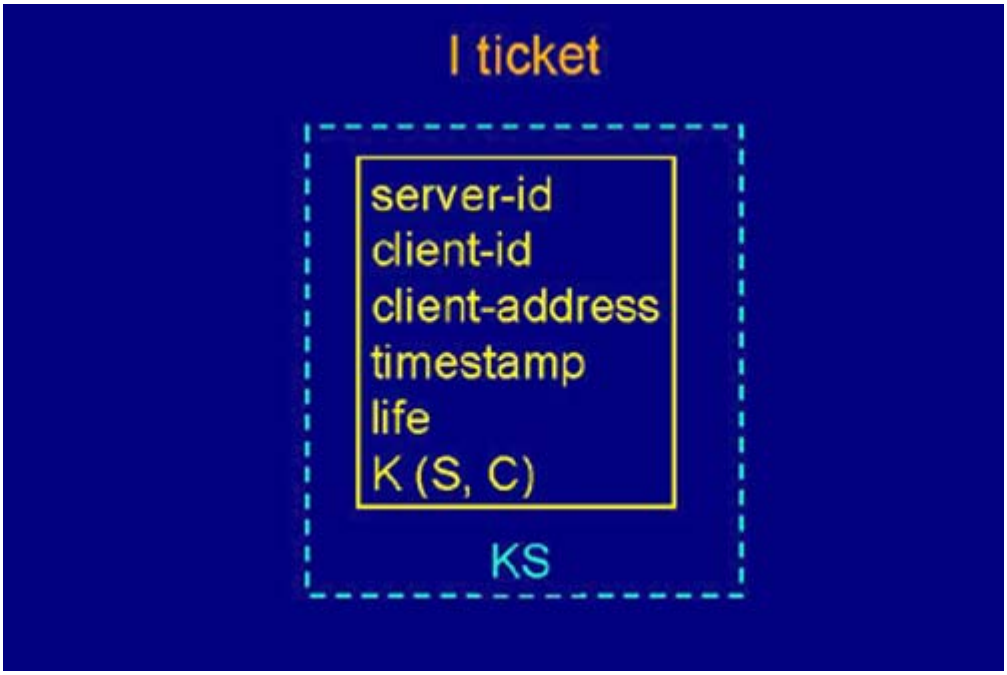
Per questo motivo ne è stata proposta un'altra versione dove è possibile applicare il medesimo schema di one-time password, ossia di password non riutilizzabili, password usa e getta, basate questa volta non su un elenco di password preconfigurato, ma sulla data e ora. In parole povere all'utente viene fornito un programma, o un dispositivo elettronico, che gli permette di inviare in rete non soltanto il proprio identificativo, ma anche la password valida per quel preciso istante di tempo, ovviamente con una certa approssimazione. Vuol dire che, ad esempio, esistono dei dispositivi o dei programmi che cambiano automaticamente la password ogni minuto, una volta ogni 60 secondi. È chiaro che in uno schema di questo genere l'ora a cui fa riferimento il dispositivo o il calcolatore usato dal nostro utente e il sistema di elaborazione a cui noi stiamo cercando di collegarci, deve essere necessariamente la stessa. Se c'è una differenza di orario tra i due sistemi, ovviamente questo tipo di autenticazione fallirà. Visto che la password è relativa unicamente a questo istante di tempo, già un istante di tempo dopo non sarà più valida e quindi non sarà più una password che anche se riutilizzata possa dare accesso al sistema.

Il sistema Kerberos



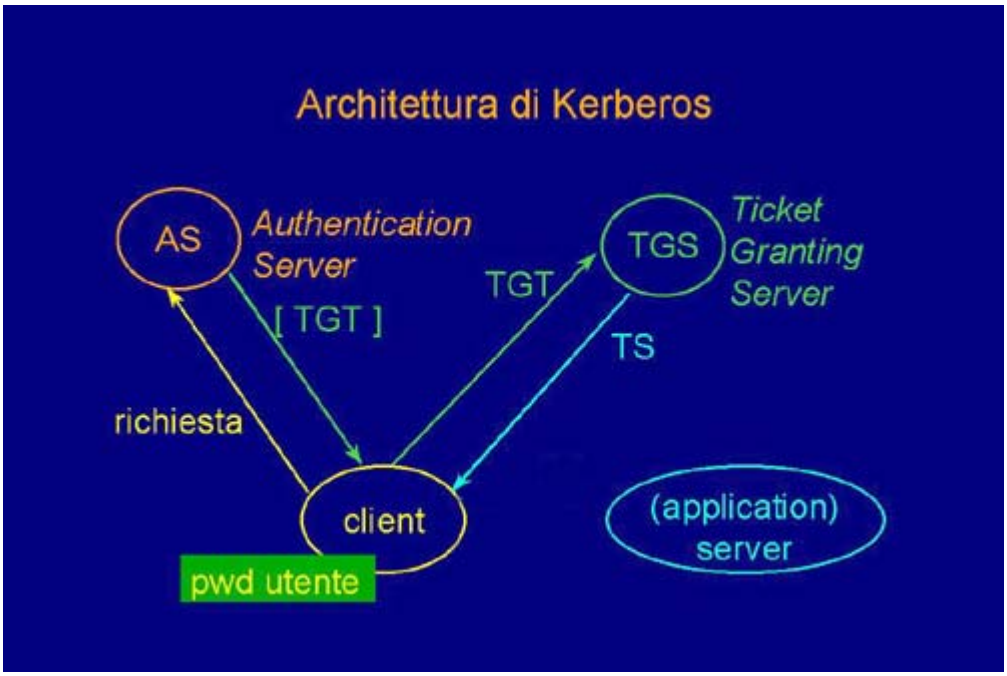
Infine concludiamo questa breve discussione dei sistemi di autenticazione parlando del sistema Kerberos. Kerberos è il sistema di autenticazione che è stato sviluppato al prestigioso MIT di Boston, l'istituto di ingegneria più grosso d'America, all'interno del sistema Athena. Athena è stato il sistema che il MIT di Boston ha messo in piedi quando ha voluto rimpiazzare i suoi grossi mainframe con una rete di workstation Unix. Kerberos costituisce il sistema di autenticazione usato in questa rete e si è poi diffuso oggigiorno ed è presente e realizzato su molti sistemi, non soltanto Unix ma anche Windows. Il concetto base del sistema Kerberos e di assegnare agli utenti delle password usandole solo ed esclusivamente come chiavi di crittografia. Nella versione originale di Kerberos si usavano le password come chiavi per il DES. Visto che oggigiorno il DES è un algoritmo che è stato riconosciuto essere debole, si possono usare le password come chiavi per algoritmi di crittografia più forti, quali 3DES o IDEA. Il sistema Kerberos è in grado di offrirci sia autenticazione semplice, ossia soltanto un utente che dimostra la propria identità al sistema, ma anche mutua autenticazione. Il sistema Kerberos si basa su due concetti fondamentali, il primo dei quali è il reame: l'insieme dei sistemi che accettano il dominio di Kerberos, ossia che offrono i loro servizi solo a chi si è autenticato tramite Kerberos. Il ticket, invece, è la struttura dati usata nel sistema Kerberos per tutte le funzioni di autenticazione.

I ticket



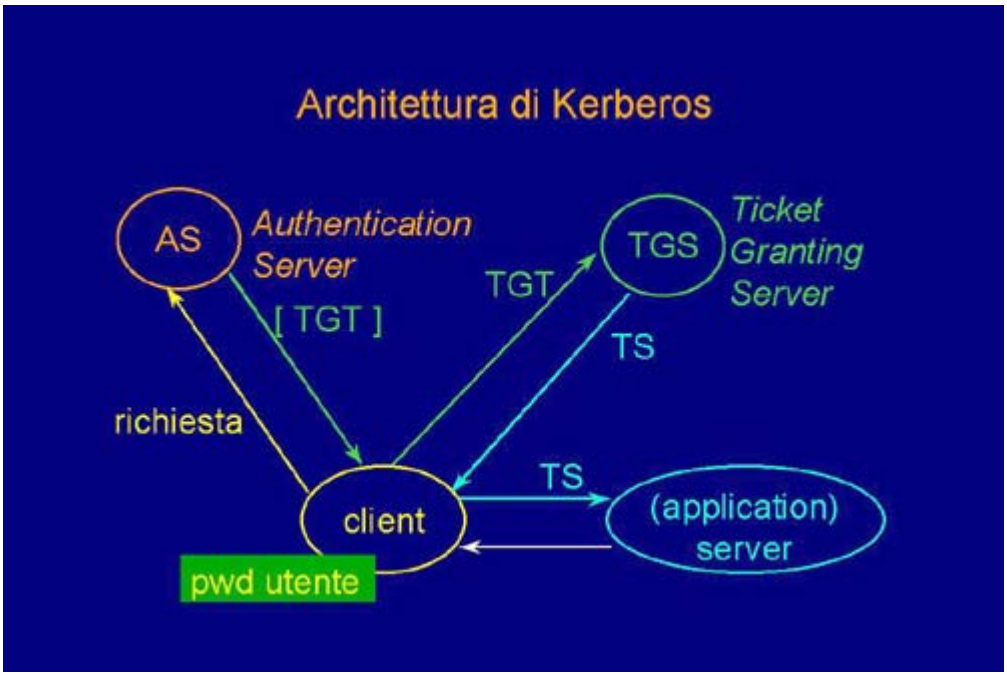
Osserviamo come è fatto un ticket: contiene alcuni dati identificativi, in particolare l'identificazione del server a cui è destinato da parte di un certo client. Il client tipicamente è l'utente che desidera ricevere un servizio da questo server. Il client è identificato sia in base l'identificativo dell'utente, sia in base all'indirizzo IP, ossia l'utente su una ben determinata postazione di lavoro. Inoltre il ticket è valido soltanto per un determinato lasso di tempo: è identificato in base alla sua data e ora di emissione e dalla sua vita, la sua durata. Il ticket contiene al suo interno anche una chiave di crittografia simmetrica condivisa fra il server e il client, questo permette eventualmente al server e al client di effettuare anche delle comunicazioni crittografate. Per evitare che i ticket possano essere manipolati dagli utenti, tutti questi dati vengono presi e cifrati con la chiave KS del server a cui sono destinati. Quindi anche quando un utente riceve un ticket per usufruire di un certo servizio, l'unica cosa che lui può fare è passare questo ticket al server da cui desidera avere questo servizio. Non può in alcun modo manipolarlo perché non conosce la chiave per aprire il ticket. In pratica è una sorta di biglietto messo dentro una cassaforte blindata.

Architettura Kerberos (1/2)



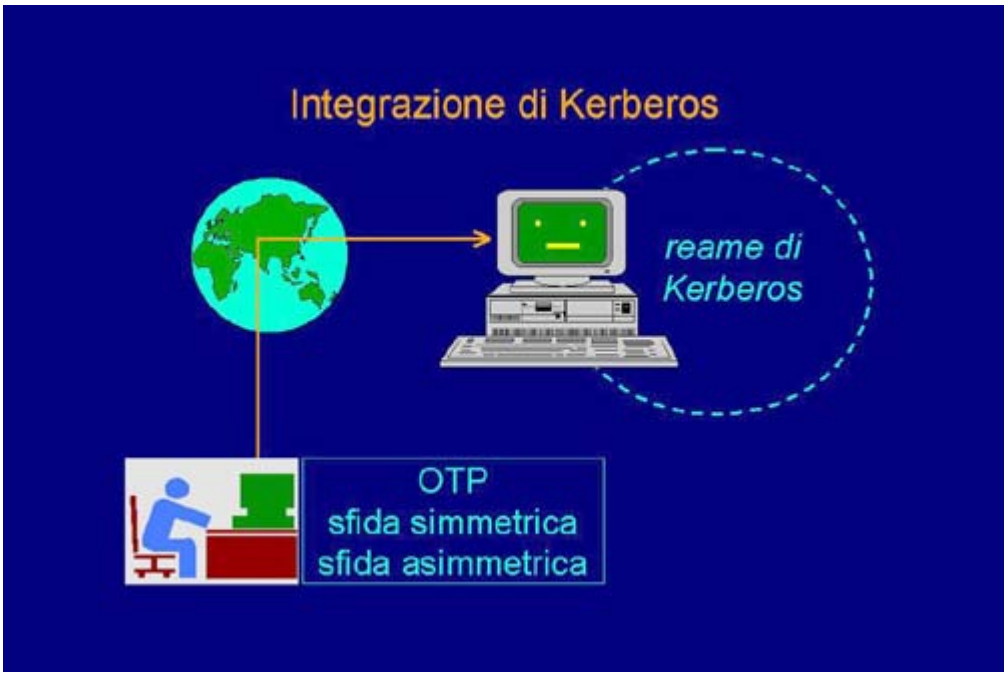
Questa è l'architettura del sistema Kerberos. Il sistema Kerberos richiede vari tipi di elementi, in particolare richiede che esitano dei client, ossia delle postazioni di lavoro, kerberizzate: dotate di un opportuno software di tipo client. Quando un utente accede ad una postazione di tipo client fornisce localmente la propria password, la quale rimane memorizzata all'interno del client senza uscirne mai. Il client per accedere ad un servizio applicativo deve prima dotarsi del necessario ticket di autorizzazione. La procedura segue questo cammino: il client invia in chiaro una richiesta al server di autenticazione. Il server di autenticazione è il cuore di Kerberos e in particolare questo server emette il cosiddetto TGT, o superbiglietto (TGT vuol dire Ticket Granting Ticket), il biglietto che mi da diritto a ricevere gli altri biglietti. Notate che questo biglietto non viene mandato in chiaro, ma viene mandato chiuso dentro una cassaforte che è stata cifrata con la password dell'utente. Quindi se ho fatto una richiesta a nome di un'altra persona, non potrò aprire la busta che contiene il TGT, perché non conoscerò la password. Se invece sono veramente chi ho detto di essere, allora sarò in grado di aprire la busta e di ottenere il TGT. Il TGT deve successivamente essere presentato ad un altro servizio: il Ticket Granting Server, che è il processo demandato ad emettere i biglietti specifici per i servizi applicativi che desideriamo. Quindi, in parole povere, il TGT dimostra la mia identità nei confronti del sistema Kerberos, il TGS controllerà se io ho diritto ad accedere ad una certa applicazione, in caso positivo mi emetterà un ticket specifico per il servizio S che ho richiesto.

Architettura Kerberos (2/2)



Alla fine di tutta questa storia quello che farò sarà mostrare il biglietto specifico per questo server al servizio applicativo che io desideravo ottenere e il server applicativo potrà finalmente erogarmi il servizio. Siccome i ticket hanno una certa durata temporale, posso prendere il ticket in un certo istante di tempo, ad esempio al mattino quando faccio il primo collegamento al sistema, e poi mantenerlo memorizzato a lungo all'interno del mio client. Quindi non è necessario accedere in continuazione all'Authentication Server e al TGS, posso avere dei ticket di lunga durata, 4 o 8 ore, l'unica avvertenza che devo avere è cancellare i ticket dalla mia memoria nel momento in cui io ho finito la mia sessione di lavoro.

Integrazione di Kerberos



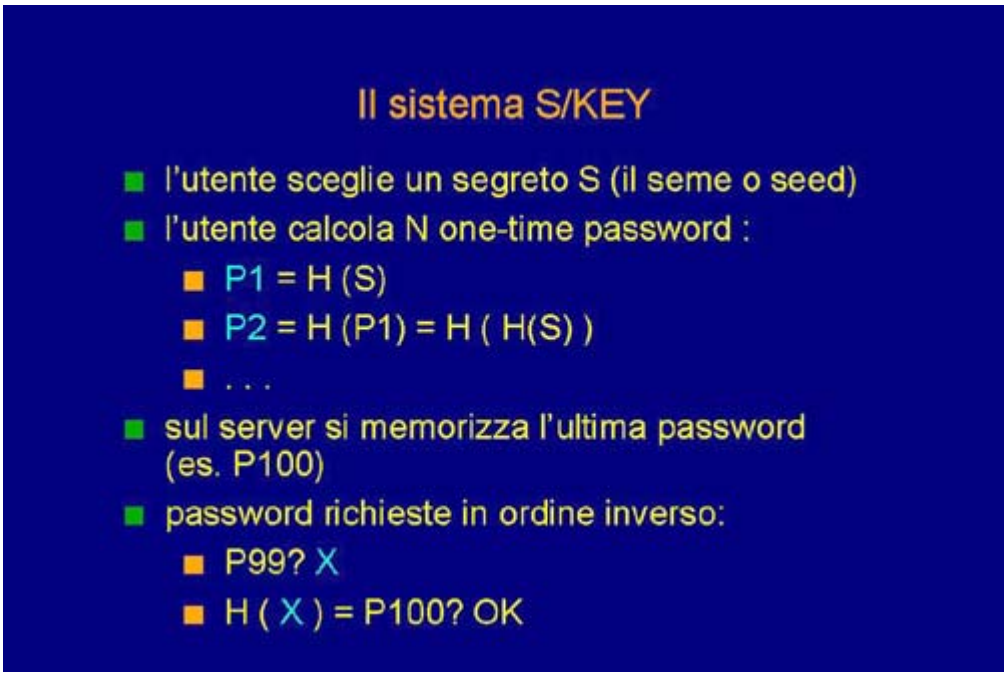
Il sistema Kerberos presuppone che la password sia inserita localmente in un sistema attaccato al reame di Kerberos. Nel caso che io faccia accesso a questo sistema da una postazione remota, ovviamente la mia password potrebbe essere intercettata mentre viaggia, supponiamo, da casa mia

alla macchina kerberizzata. Allora in questo caso, per risolvere questo problema, Kerberos è stato integrato con dei sistemi di autenticazione che non mostrino la password durante il suo transito in Internet. Esistono delle versioni di Kerberos che utilizzano come interfaccia sistemi a one-time password, sistemi a sfida simmetrica o sistemi a sfida asimmetrica. In questo modo si estende l'applicabilità di Kerberos, che di per sé sarebbe ristretta al solo reame che tipicamente coincide con una rete locale, anche all'accesso remoto fatto via Internet tramite reti insicure.

Il concetto di OTP

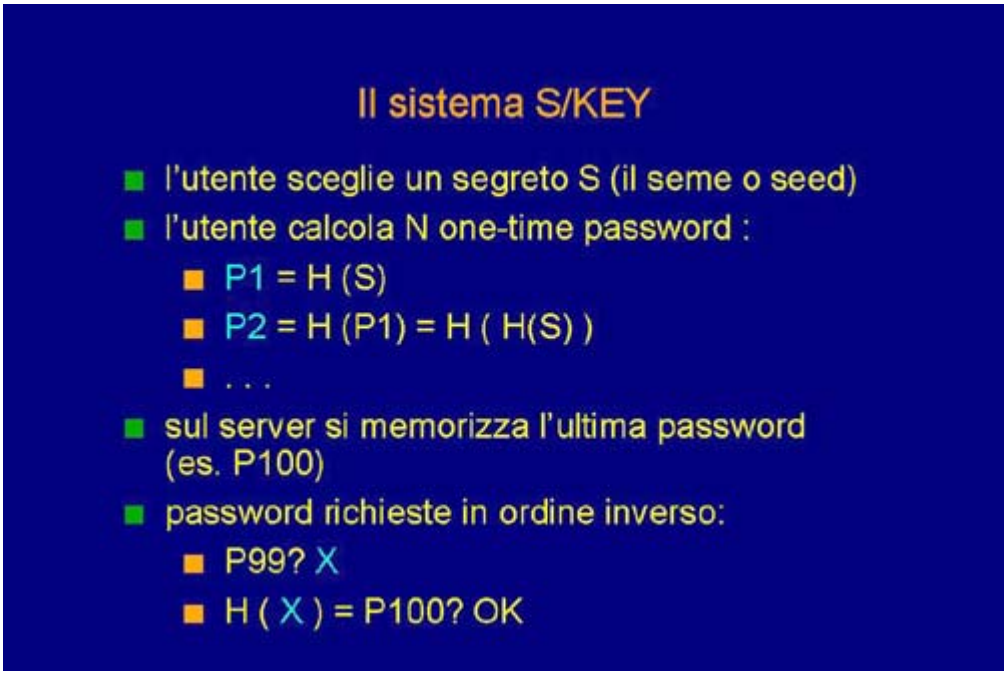
È noto che le password sono un mezzo insicuro per dimostrare la propria identità ad un sistema di elaborazione a cui si desidera accedere. Nelle precedenti lezioni abbiamo visto che è possibile non utilizzare le password per autenticarsi nei confronti di un sistema di elaborazione, usando al loro posto delle tecniche alternative, per esempio le one-time password oppure i sistemi a sfida. In questa lezione quello che faremo sarà esaminare alcune tecniche per implementare le one-time password o i sistemi a sfida. Cominciamo con la prima di queste. In particolare, ricordo che le one-time password sono delle password utilizzabili una ed una sola volta; il problema che ci si pone, allora, è come fornire all'utente sempre nuove password, perché chiaramente queste verranno esaurite in brevissimo tempo. Ci sono delle soluzioni più artigianali, altre più tecnologiche: entrambe sono tecnicamente valide, quale applicare dipende semplicemente da qual è il tipo di convenienza, economica o organizzativa, verso la quale si punta. In particolare, la soluzione più semplice è quella di fornire all'utente le password scrivendogliene tante elencate su un foglietto ed aspettare che il sistema ci richieda una di queste password. Una soluzione più tecnologica consiste invece nel dare al nostro utente uno strumento automatico, che effettui il calcolo della password necessaria a collegarsi al sistema in un determinato istante di tempo.

Il sistema S/KEY (1/2)



Un'implementazione del primo tipo è quella fatta dal sistema chiamato S/KEY. Nel sistema S/KEY l'utente sceglie un segreto, il quale non è altro che un numero binario casuale ed è anche chiamato seme o seed (in inglese); dopodiché l'utente calcola autonomamente N, numero a piacere, di password utilizzabili una volta sola. La modalità con cui le calcola è quella indicata qui: la password numero 1 viene calcolata applicando una funzione di hash al seme S. La password numero 2 è data dalla stessa funzione di hash applicata alla password numero 1, ossia la funzione di hash applicata due volte al segreto S. In questo modo si può generare un numero di password lungo a piacere. Per poter poi accedere al server, sul server deve essere memorizzata l'ultima password. Ad esempio, supponendo di averne generate 100, verrà memorizzata sul server la password numero 100.

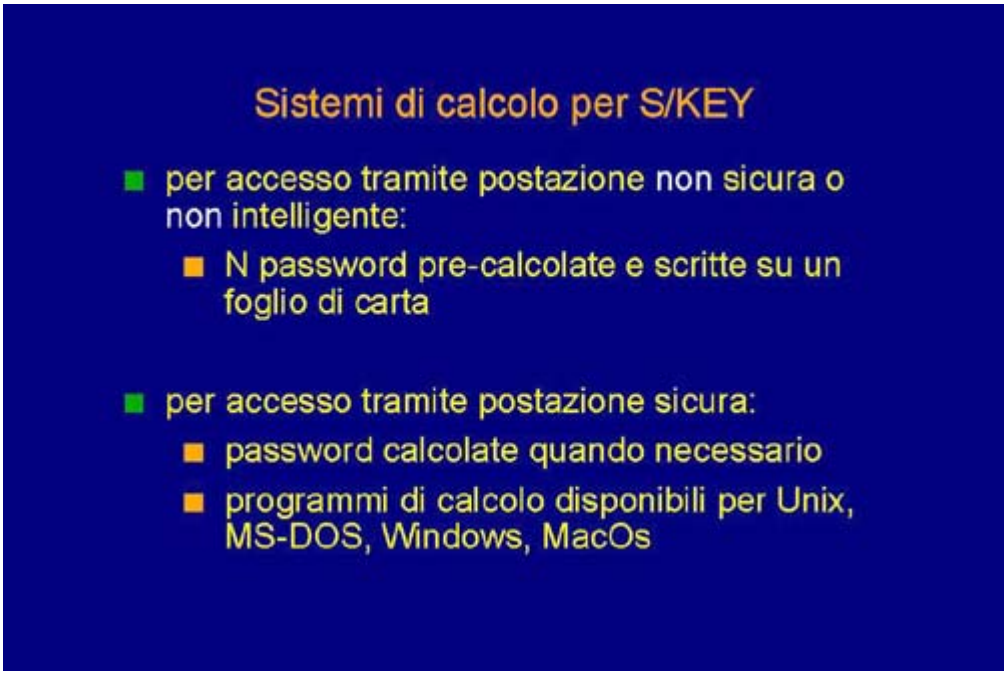
Il sistema S/KEY (2/2)



Ogni volta che noi desidereremo fare login o accedere a un servizio fornito da questo server, lui ci

chiederà le password in ordine inverso a quello con cui sono state generate. Ad esempio, ci chiederà la password numero 99, perché conteneva al suo interno la numero 100. Alla password numero 99 risponderemo con un certo numero X. Il server può verificare che questo numero X sia effettivamente la password numero 99, effettuando il calcolo che vedete indicato: applicando la funzione di hash al numero X fornito, si deve ottenere la password numero 100. La password numero 100 era memorizzata sul server, quindi confrontando il risultato di H(X) con la password già memorizzata, può decidere se l'utente ha il diritto ad accedere al servizio oppure no. Ovviamente ogni volta che una password è stata utilizzata viene buttata via, in questo caso al posto della password numero 100 verrà memorizzata la numero 99 e la prossima volta all'utente verrà richiesta la numero 98. Questo è uno schema mosto semplice: una volta che siamo arrivati alla password numero 0, ovviamente quello che occorre è che l'utente si generi di nuovo altre 100, altre 1.000, altre 10.000 password a suo piacimento e successivamente dia l'ultima di queste al server con cui desidera effettuare il collegamento.

Sistema di calcolo per S/KEY

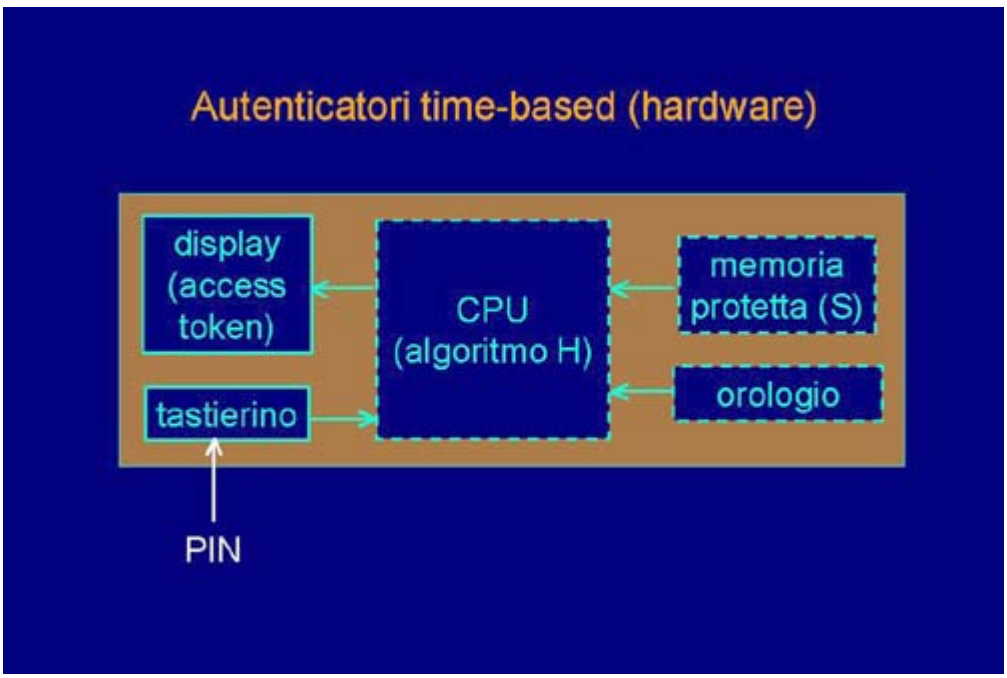


Allora, nel caso che noi desideriamo accedere a questo server tramite delle postazioni di lavoro non sicure, quindi delle postazioni di lavoro in cui non possiamo esser certi del software installato o del hardware, oppure vogliamo accedere tramite una postazione non intelligente non in grado di effettuare il calcolo della funzione H, l'unica soluzione è quella di usare queste N password precalcolate, ossia di scriverle su un foglio di carta. Questo ci da completa indipendenza dalla postazione di lavoro e dal suo livello di sicurezza. Se invece possiamo accedere al nostro sistema tramite una postazione sicura, quindi una postazione in cui noi ci fidiamo del software installato e del hardware tramite cui noi interagiamo con questa postazione, allora è possibile non doversi portar dietro questo foglietto ma semplicemente, fornendo il segreto alla postazione di lavoro, far si che sia lei ad effettuare il calcolo. In questo caso le password vengono calcolate, quando necessario, da un programma residente sulla postazione sicura. Sono disponibili in rete programmi di calcolo per sistemi Unix, ma anche per MS-DOS, Windows, MacOs. S/KEY è un sistema di public domain e come tale è stato ampiamente implementato su varie architetture.

Sistemi di autenticazione time-based

Basandoci sul medesimo principio di avere delle password non riutilizzabili più volte, si può scegliere una filosofia diversa: quella di non calcolare una password dopo l'altra, ma di calcolare una password in funzione dell'istante di tempo in cui essa stessa debba essere utilizzata. Nei sistemi one-time password basati sul tempo, le password dipendono dal seme iniziale scelto dall'utente, ma anche dall'istante di tempo. Quindi la password non è più numerata, ma è la password dell'istante di tempo t ed è calcolata applicando una funzione di hash al seme e al tempo. In questo caso, per far si che il server a cui noi effettuiamo il collegamento possa essere convinto che la password che abbiamo fornito sia quella giusta, il server e l'utente devono condividere tre cose: il segreto S, il riferimento di tempo e l'algoritmo di calcolo H. In questo caso c'è una richiesta in più rispetto alle one-time password precedenti: anche il server deve conoscere il segreto S.

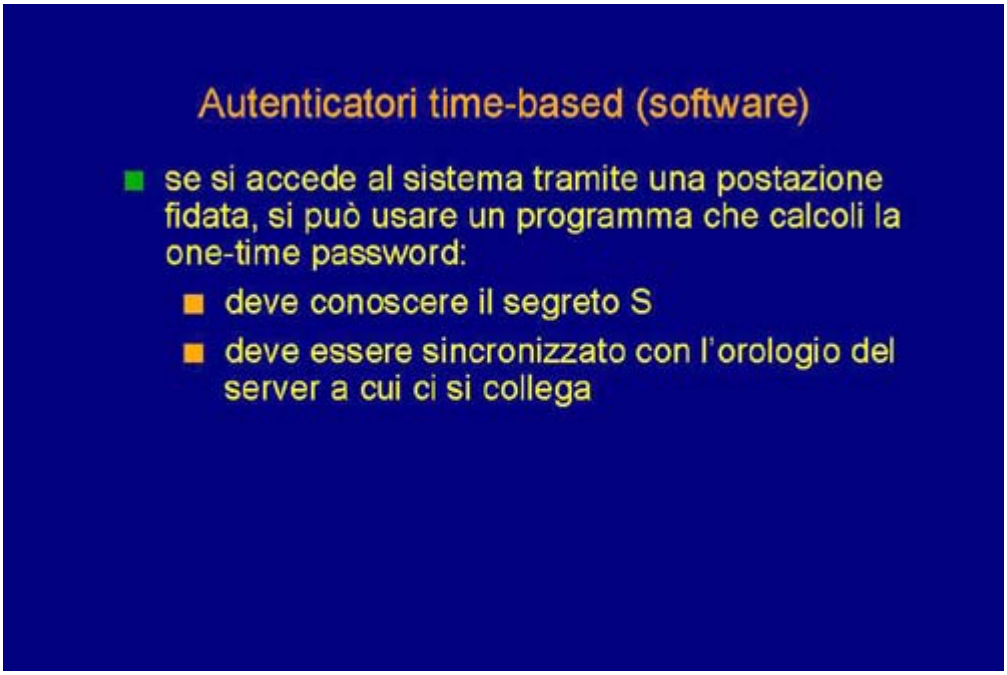
Autenticatori time-based (hardware)



Nel caso in cui vogliamo raggiungere completa indipendenza dalla postazione di lavoro e

simultaneamente non avere tra le mani un ingombrante foglietto di carta contenente qualche decina se non centinaia di password, alcune ditte commerciali hanno sviluppato i cosiddetti autenticatori hardware. Sono delle minicalcolatrici portatili che effettuano per noi il calcolo necessario per il sistema di autenticazione one-time password. In questo diagramma vediamo lo schema hardware di un autenticatore basato su tempo. Da un punto di vista esterno questo assomiglia a una normalissima carta plastica, formato carta di credito, soltanto leggermente più spessa e dotata tipicamente di un tastierino alfanumerico, o anche solo numerico, su cui l'utente deve battere un PIN per abilitare la carta alle sue funzioni. Il PIN abilita il calcolo dell'algoritmo H sulla CPU presente all'interno di questo autenticatore. La CPU prenderà i suoi dati da una memoria protetta, sempre interna alla carta stessa, che contiene il segreto S e da un riferimento temporale interno, quindi un orologio autonomo della carta. Il risultato del calcolo sarà la one-time password valida per un certo istante di tempo e sarà visualizzata su un piccolissimo display alfanumerico che quindi visualizzerà il cosiddetto token di accesso, che ci permetterà di accedere al sistema.

Autenticatori time-based (software)



Se si accede invece al sistema tramite postazioni fidate, anche per sistemi basati su one-time password differenti per i diversi istanti di tempo, è possibile utilizzare una soluzione basata su software e non su hardware. Le soluzioni su hardware di solito sono molto più care di quelle basate su software, perché richiedono di avere questi oggetti, che essendo non facilmente manipolabili o forzabili dall'esterno, sono oggetti molto cari. La soluzione software si basa sull'utilizzo di un programma che calcoli la one-time password sul sistema fidato. Ovviamente a questo programma deve essere fornito il segreto S ed inoltre il PC, la postazione di lavoro su cui il programma esegue, deve avere il suo orologio sincronizzato con quello del server a cui ci si collega. Questo è il requisito essenziale affinché entrambi i sistemi calcolino la medesima password nel medesimo istante di tempo.

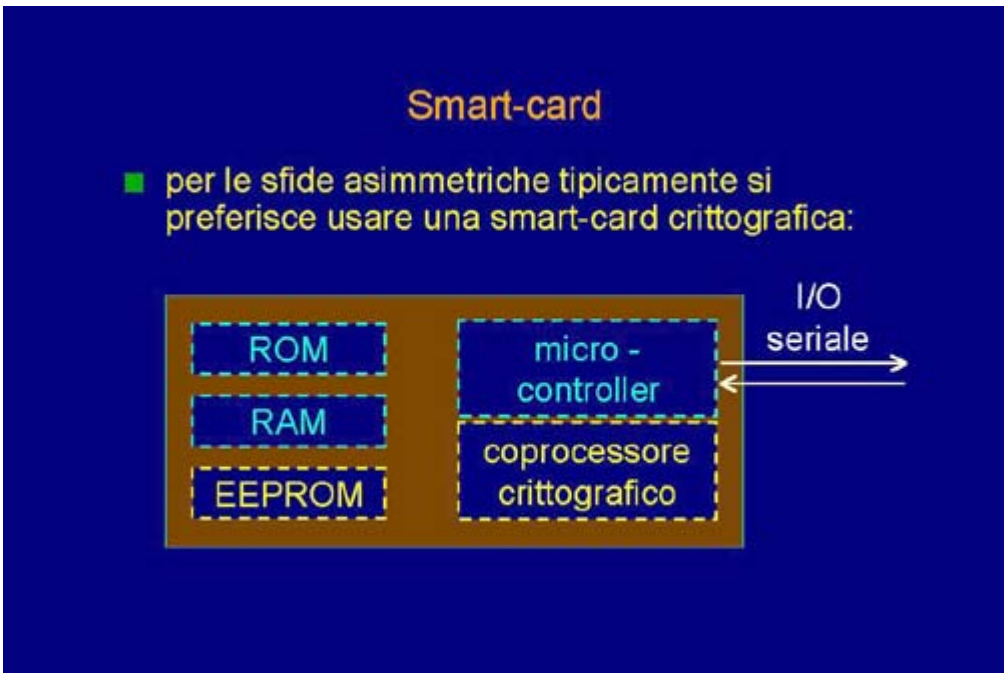
I sistemi a sfida

Veniamo ora all'altra categoria di sistemi che possono aiutarci a rimpiazzare le password nell'autenticazione nei confronti di un sistema di elaborazione. Questi sono i sistemi a sfida. Nei sistemi a sfida il concetto è che la password sia una password non riutilizzabile, perché calcolata come risposta a una sfida fornita dal sistema di elaborazione e basata anche su un segreto noto soltanto all'utente e al sistema di elaborazione. Il problema, in questo caso, è come permettere all'utente di conservare, indipendentemente dalla postazione di lavoro, il segreto suo personale e come calcolare la funzione F, quantomeno il risultato della funzione F. Una prima soluzione non ci rende indipendenti dalla postazione di lavoro, ossia dice: il segreto viene memorizzato utilizzando il disco del mio personal computer, supposto sicuro, e la funzione F viene calcolata utilizzando la CPU del mio sistema di elaborazione. Questa è una soluzione valida, ma lega l'accesso alla postazione di lavoro. Una soluzione migliore e più indipendente è quella che utilizza un dispositivo autonomo di memoria e calcolo. Quindi, qualcosa di simile agli autenticatori hardware che abbiamo visto un attimo fa per le one-time password basate sul tempo.

Calcolatori per sfide simmetriche

Normalmente, per i sistemi a sfida si utilizzano quelle che si chiamano le criptocalcolatrici. I sistemi a sfida possono basarsi su algoritmi di crittografia simmetrici o asimmetrici. Per i sistemi basati su sfida simmetrica si usano delle criptocalcolatrici tipicamente basate sull'algoritmo DES. Quindi, in questo caso, il segreto corrisponde con la chiave di cifratura, ad esempio la chiave di cifratura DES, e la funzione F è l'algoritmo di cifratura simmetrico presente a bordo della carta. L'autenticatore hardware, in questo caso, contiene al suo interno la chiave di cifratura, implementa l'algoritmo di cifratura stesso, aspetta l'introduzione della sfida fornita dal sistema all'utente e calcola poi la risposta. È compito dell'utente battere la risposta che è stata ottenuta sulla tastiera, per fornirla al sistema che ci ha sfidato.

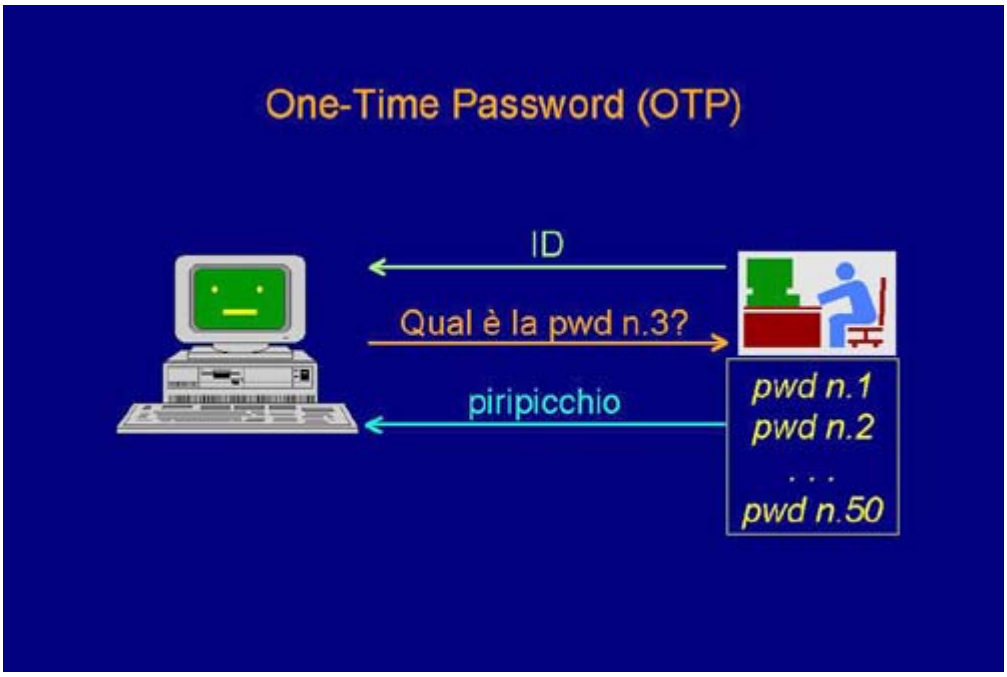
Smart-card



Nel caso di sfide asimmetriche, al posto che utilizzare una criptocalcolatrice vera e propria si utilizzano dei dispositivi particolari chiamati smart-card (che significa carta intelligente), che non sono altro che criptocalcolatrici normalmente di tipo asimmetrico. Possono esistere anche atri tipi di smart-card: quelle che sono comunemente utilizzate per darci i premi nei concorsi, ad esempio per chi acquista più benzina, o per chi fa più spesa al supermercato, sono delle normalissime carte dotate soltanto di un chip di memoria. Nel caso invece di applicazioni alla sfida asimmetrica, la smart-card è dotata di un chip in grado di effettuare i calcoli crittografici asimmetrici. Questo è lo schema di una smart-card utilizzabile per sfide asimmetriche. La smart-card è racchiusa in un contenitore plastico, che vedremo tra poco, e contiene al suo interno un microcontroller. Un microcontroller non è nient'altro che una piccola CPU in grado di svolgere varie funzionalità, tra cui anche semplici funzioni di input/output di tipo seriale. Normalmente le smart-card colloquiano con il calcolatore a cui sono collegate tramite un dispositivo di input/output seriale a 9.600 b/s. La smart-card contiene anche un po' di ROM, contenente il programma di controllo della smart-card, e un po' di RAM, per permettere alla CPU microcontroller di effettuare alcuni semplici calcoli. Affinché una smart-card sia utilizzabile per risolvere delle sfide asimmetriche deve essere dotata di due cose addizionali: in particolare deve essere dotata di un coprocessore crittografico. Il coprocessore crittografico è del hardware aggiuntivo che realizza la funzione di crittografia asimmetrica, tipicamente la funzione RSA, talvolta anche la funzione DSA. Inoltre, è presente all'interno della smart-card della memoria aggiuntiva, chiamata EEPROM o E2PROM. Si tratta di una memoria di sola lettura che però è anche scrivibile tramite funzioni elettriche. Questa a tutti gli effetti agisce come un piccolo hard disk, come un piccolo disco magnetico, pur trattandosi di memoria puramente elettronica. Ci permette di memorizzare dei dati sulla smart-card in modo

permanente, ossia dei dati che rimangono memorizzati anche quando la smart-card non viene collegata al computer e quindi non è alimentata elettricamente.

Standard per le smart-card



Ovviamente, per permettere alle smart-card di essere utilizzate su tanti sistemi diversi e da tante applicazioni diverse (cosa che oggi è molto richiesta, perché abbiamo visto che le funzioni asimmetriche sono valide non soltanto per l'autenticazione a sfida, ma anche per la firma digitale), anche per la sicurezza delle reti sono stati definiti una serie di standard. In particolare, gli standard definiti dall'ISO, raggruppati sotto il numero 7816, definiscono il formato fisico, elettrico e logico, diciamo di basso livello, delle smart-card. Nel caso che però la smart-card effettivamente sia in grado di effettuare delle operazioni crittografiche, è desiderabile anche avere un'interfaccia verso queste funzioni. Quest'interfaccia normalmente è chiamata interfaccia di tipo CSP (Crypto Service Provider = fornitori di servizi crittografici). I servizi crittografici che fornisce sono due: il calcolo di alcuni algoritmi di crittografia e un database sicuro per le chiavi private dell'utente. Non c'è accordo su quale deve essere l'interfaccia CSP standard da utilizzare. Attualmente esistono due standard in competizione fra di loro e con funzionalità simili. Lo standard PKCS-11 è l'interfaccia CSP utilizzata da alcuni sistemi, tipo i browser Netscape, o i sistemi di workflow forniti da Lotus. L'interfaccia CAPI (Crypto API) è invece l'interfaccia CSP fornita standard con i sistemi Microsoft. Sono interfacce equivalenti ma purtroppo incompatibili. Lo standard PC/SC è invece lo standard de facto per l'interfacciamento dei lettori di smart-card, ossia per permettere ad una smart-card di essere utilizzata su lettori di tipo diverso.

Smart Card

Questa è una smart-card grezza; come vedete si tratta semplicemente di un rettangolo di plastica su cui è stato incastrato a forza un chip sottilissimo. Questa è una smart-card crittografica in grado di contenere 8 Kbyte di memoria e in grado di svolgere funzioni di crittografia asimmetrica di tipo RSA. Queste smart-card cominciano ad essere utilizzate in vari progetti, anche a livello applicativo. Ad esempio, ho portato qui con me una delle smart-card utilizzate nel progetto DISTINCT. È un progetto della Comunità Economica Europea che coinvolge varie città attraverso l'Europa, in particolare partecipa anche la città di Torino, con il comune, gli uffici anagrafici e il politecnico di Torino stesso. Come vedete le carte possono essere personalizzate. Questo è il bozzetto della carta che si vorrebbe dare ai cittadini di Torino per permettere di usufruire delle funzionalità anagrafiche, di trasporti e così via.

Sistemi biometrici

Per evitare il problema che una smart-card, o uno di questi autenticatori hardware, venga rubato al

legittimo titolare e poi utilizzato per accedere ad un sistema, è stato proposto di utilizzare questi dispositivi hardware in congiunzione con sistemi di identificazione personale di tipo biometrico. In pratica il sistema biometrico serve a sostituire la password, o il PIN, che normalmente deve essere utilizzata per abilitare le funzioni della carta. Si utilizzano basilarmente tre tipi di tecniche. Esistono dei dispositivi in grado di abilitare l'utilizzo della carta solo in base al riconoscimento di una determinata impronta digitale, oppure del tono vocale dell'utente o, se dotato di una piccola telecamera oculare, effettuando lo scanner dei vasi sanguigni presenti sul fondo della nostra retina. Tutte queste tecniche sono ugualmente valide in teoria, in realtà l'implementazione pratica non ha ancora raggiungo un grado di affidabilità sufficiente in qualunque ambiente. Il problema che ci troviamo ancora a fronteggiare è quello delle cosiddette accettazioni o rifiuti di collegamento falsi. Un falso rifiuto si ha quando l'utente viene rifiutato, cioè non gli viene dato il permesso di accedere, nonostante lui sia veramente chi dice di essere: basta ad esempio aver posto male il dito sul lettore di impronte digitali e questo potrebbe capitare. Più pericoloso ancora il caso contrario, in cui nonostante un utente non autorizzato sia dotato di un impronta digitale diversa da quella autorizzata, mettendo il dito sul vetrino gli viene concesso l'accesso. Ovviamente più il sistema è sofisticato più crescerà il suo prezzo, ma questo ovviamente tenderà anche a diminuirne l'accettazione in vari ambiti applicativi. Sistemi biometrici in unione con dispositivi di accesso hardware sono però sicuramente una delle strade maestre che si diffonderanno largamente nell'utilizzo dei sistemi informativi nei prossimi anni.