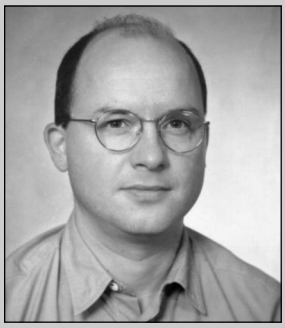
Is User "Sally Smith" Really Who She Claims to Be?! Lessons for Establishing Rock-Solid Authentication and Single Sign-On (SSO) Practices

Dr. Jürgen Schneider



Dr. Jürgen Schneider is currently the Development Manager for Security in SAP's Technology Development department. Before joining the SAP Security Basis development team in 1996, he led several research projects in the areas of network management and security at IBM's European Networking Center.

(complete bio appears on page 46)

Authenticating users is a security prerequisite for every SAP system. Before an application on your system grants Sally Smith the access privileges conferred to her in its access controls, that business application better be darn sure the user claiming to be Sally Smith really *is* Sally Smith. Even R/3 releases dating back to 1992 are replete with options, above and beyond the standard SAP user ID and password mechanism:

- **Password parameters:** When using the SAP user ID and password mechanism as the means for authentication, there are a slew of parameters that can be set to establish minimum password lengths, expiration dates, number of failed logins that may occur before a user is booted off the system, and so on.

- **SAP's secure network communications (SNC):** This option, which has been around since Release 3.1H, provides an interface through which you can integrate your SAP systems with an external authentication framework, and thereby delegate the authentication process of an SAP GUI for Windows user or an SAP Remote Function Call (RFC) client to a non-SAP infrastructure — perhaps one that is already up and running in your environment, like the Windows NT Domain Controller or a Kerberos system. So, this option does not use SAP passwords at all.

- **ITS-based support for X.509 digital certificates:** This ITS option, introduced in Release 4.5B, provides a way to authenticate a web client at the web server via the Secure Sockets Layer (SSL) protocol, which underlies secure web communications with HTTPS.

Here again, we provide you with a means to delegate the authentication process of your SAP users to an external entity — your web servers in this case — integrated with a public key infrastructure (PKI). Again, no SAP password is used here.

- **ITS-based Pluggable Authentication Services (PAS):** Recognizing that X.509 digital certificates come with a certain degree of overhead, Release 4.6D offers PAS, the means by which you can tap into lots of other types of external authentication products for browser users who require access to your SAP systems.

All of these options for authenticating users of your SAP systems will be discussed in this article. So will single sign-on (SSO) options, because authenticating users is by no means unique to SAP systems. Authenticating users is a security prerequisite for *every* important system in your IT landscape — SAP and non-SAP alike. Let's face it, as soon as users get numerous logon prompts from different systems, each requiring them to render different user IDs and passwords, you've got a usability problem. In my role as Development Manager for Security at SAP, I find customer after customer looking for a way to alleviate this situation. They are searching for flexible, practical, and secure options that can perform repeated authentication steps on behalf of the user, and at the same time preserve high levels of security across all systems. SAP's solution to this challenge is single sign-on (SSO). In the latter sections of this article, I will show you:

- The ways you can use SSO to authenticate users across multiple, standalone SAP systems (e.g., although Sally Smith is working in the Sales department and works with mySAP CRM applications most of the day, she also needs to use other SAP systems for vacation requests, to maintain her own address and bank account data, and to manage her own cost center).

- How the mySAP Workplace can provide single sign-on for integrated SAP systems and third-party applications across your company's intranet, and even across the Internet (e.g., access to Sally's daily working activities as described above are presented to her as a personalized menu in a single web page, regardless of how many SAP systems are involved, including links to non-SAP applications, external business partner applications, and public information sources).

For those of you who may be unfamiliar with SSO, let me offer a preview of what it's all about, and a quick explanation as to why such a simple concept has eluded the industry for such a long time. Application security typically begins with the perfunctory request for user ID and password information. We all dutifully follow suit, typing our user IDs and passwords into some kind of logon screen to gain access to our networks, e-mail, business applications, and so on. As the number of business applications grows, so does the burden of remembering different user IDs and passwords and for repetitive input of all this authentication information. At some point, perhaps it's the third or fourth logon prompt, I don't know about you, but I'm tempted to reply: "Hey! Enough already! It's okay. I'm still me!"

Single sign-on offers the remedy:

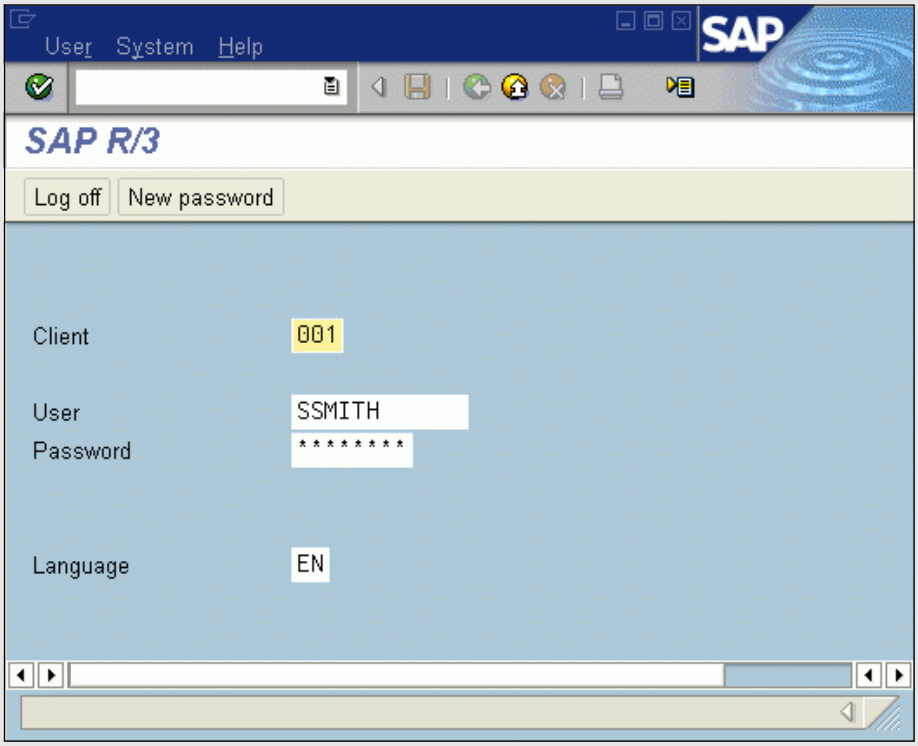
- An initial authentication step, which typically calls for the user to enter user ID and password information.

- A mechanism that, after initial authentication, allows for repetitive authentication of a user across one or more systems *without* requiring input from the user for some specified period of time.

Sounds simple enough, right? Wrong! Beneath the surface, *considerable* security challenges need to be addressed:

- Although the initial authentication takes place against one particular system or application, the mechanism should work against all the different systems and applications that will be subsequently accessed.

Figure 1 Logon with SAP User ID and Password Using the SAP GUI for Windows



- The set of applications supposed to accept the single sign-on usually includes a large number of intranet services, which come from various suppliers and may run on various distributed systems and platforms.

- Ideally, the single sign-on also extends to the Internet when users are accessing services provided by business partners or public web sites.

Since users are not directly involved in successive authentication steps after the initial one, it is very important that the mechanism is secure enough to prevent surreptitious misuse of the user's identity during the validity period so that single sign-on does not lead to subsequent breaches of security.

But I'm getting ahead of myself here! The SSO discussion will get its due in the second half of this article. Let me turn your attention now to some of the simplest, easiest options you can set up to significantly bolster the security of your environment's authentication activities.

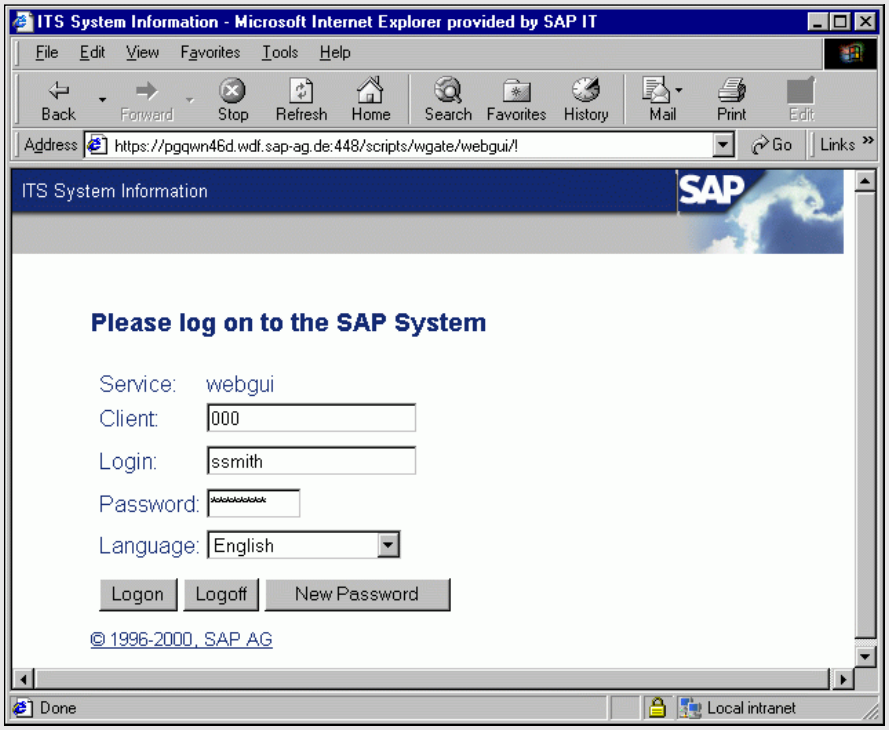
Parameters for Establishing SAP Password Policies

There are several ways a user like our friend Sally Smith can provide her SAP user ID and password to an SAP system:

- She can type her SAP user ID and password information into the tried-and-true SAP GUI for Windows, as shown in **Figure 1**.

- She can use an HMTL logon page that is made available to her, thanks to the SAP Internet

Figure 2 Logon with SAP User ID and Password Using an ITS HTML Logon Page



Transaction Server (ITS), in her web browser, as shown in **Figure 2**.

- And as of Basis Release 6.10, where we introduce the SAP Web Application Server, Sally can be prompted for her SAP user ID and password directly via the HTTP protocol, which is called *Basic Authentication* in web terminology. When a web server asks for Basic Authentication, the web browser displays its standard user ID and password pop-up, as shown in **Figure 3** for Internet Explorer. The user ID and password information that Sally types in is transferred directly to the SAP Web Application Server in the HTTP Basic Authentication protocol header.

In all three cases, Sally's SAP user ID and password information is sent to the SAP system and compared against the value stored in Sally's user master record in the SAP database (table USR02).[1] If the logon information is verified, the system grants access. If not, access is denied. (Note that when system administrators assign new passwords to users,

Figure 3 Logon with SAP User ID and Password Using HTTP Basic Authentication with the SAP Web Application Server

[1] In the SAP database, only hash values of the user ID and password information are stored using a slightly modified MD5 hash algorithm. The password itself is not stored and thus cannot be stolen from the database. As part of the logon process, the SAP user ID and password provided by the user is hashed and the hash value is compared against the value stored in the BCODE field of table USR02.

the new password is marked as *initial*. Users have to change their initial passwords at first logon.)

There are a number of password protection options (listed below) you can use to bolster the security of this initial authentication process. It is the manner in which you exercise these options that defines the password policies for the SAP systems used in your organization[2]:

- The `login/min_password_lng` parameter — To make passwords safe from guessing, you can enforce a minimum password length of three to eight characters. For a productive system, we recommend a minimum length of at least six characters.
- Prohibiting certain passwords in table USR40 — The SAP system will not permit the use of certain passwords, nor will it allow a password to contain three identical characters in sequence. Via table USR40, you can also prohibit the use of certain passwords.[3] You might, for example, want to prohibit the use of simple passwords that you know are apt to be widely used among your users. These types of passwords often include the name of your company, weekdays, months or seasons, person names, or simply things like "pass1," "pass2," "init," "initial," etc.
- The `login/fails_to_session_end` parameter — To prevent so-called "dictionary attacks," where thousands of words from a given dictionary and variations are tried as passwords, the SAP system aborts a user's logon session after a certain number of invalid logon attempts. The usual default is to allow three attempts to provide the correct password before the logon screen disappears and the user has to start a new session.
- The `login/fails_to_user_lock` parameter — In addition, a counter of consecutive invalid logon attempts is kept per user and the user account is locked in the SAP database after a certain limit is reached. For a productive SAP system, you can choose to reduce the default number of 12 invalid logon attempts in a row, before a user account is locked. The user lock can be automatically removed per default at midnight, or you can configure the system so that the account is locked until the system administrator removes it. In a productive SAP system, you may want to have account locks removed by your system administrator only.
- The `login/password_expiration_time` parameter — To reduce the risk that passwords get compromised or that compromised passwords can be used for a long period of time, passwords do expire. Once a password has expired, the user has to change it during the next logon. The SAP system stores the hash values of each user's last five passwords so that users cannot reuse those either. In a productive SAP system, a typical value for password expiration is four to eight weeks.

✓ *Tip*

To achieve top-grade security, behavioral practices must be in lock step with your technical password policies. Hashing, minimum password lengths, password expiration policies, and so on will all be in vain if users jot down their passwords and then paste them to their monitors. So don't put users in a position where they feel they have to resort to methods like these. Don't saddle them with overly complicated, constantly changing passwords and/or lots of them. Try to establish single sign-on instead, so that one password logon to a central system is sufficient to access applications in other systems as well (which is what this article is all about). Another thing to be mindful of is the all-too-common practice among users of disclosing their passwords to colleagues or sharing them with an assistant. Obviously, you want to discourage this and similar practices.

---

[2] You'll find the documentation and default values for all mentioned SAP profile parameters using SAP transaction RZ11 in your SAP system. See also the "SAP Security Guide," which provides additional information and recommendations, at **http://service.sap.com/securityguide**.

[3] Additional password rules enforcing combinations of letters, digits, and special symbols are currently planned for the SAP Web Application Server.

✓ Tip

Your users' SAP passwords should be treated as closely guarded secrets. Don't send this information over unprotected network connections. I see many customers do this by default. In a productive environment, you should use encrypted communications for sending passwords (as well as for all the important business data that is sent after the authentication was successful) to preempt network wiretap or network sniffer attempts. (It would not be difficult for people within your company to obtain a "network sniffer" program on the Internet.) For all communication protocols used by SAP systems, there exists an encryption option, and for the classical SAP protocols DIAG and RFC, you can use SNC, which exists since SAP Release 3.1H and 4.0, respectively. For the HTTP protocol, you can use HTTP over the Secure Sockets Layer (SSL) protocol (also called HTTPS) with all commercial web servers and web browsers.

SAP Secure Network Communications (SNC)

SNC is an integration layer that enables you to tie your SAP system into a third-party authentication product like *Secude for R/3* or *Entrust*, or even use the authentication services from your Windows NT or Windows 2000 infrastructure.[4] This is why, if you look at **Figure 4**, for example, the logon screen that is presented to user Sally Smith appears to be completely different from the standard SAP logon screen shown back in Figure 1. It *is* different! With SNC enabled to support Secude for R/3, Sally is not logging into the SAP system initially; she is logging into Secude for R/3 using the Secude client logon tool, which is called *PSE Management*.

When an SNC-protected network connection is initiated between two SAP system components, the SAP software retrieves the SNC name of the logged-in user via the SNC layer and maps it to an SAP user ID.

In Sally's case, with Secude for R/3 in the example, the SNC name is "CN=D022964, O=SAP-AG, C=DE." In our example, this is the "distinguished name" for Sally as provided in her X.509 digital certificate. Since the Secude product uses public key technology, it uses digital certificates as the digital identification information for users.

[4] For a complete list of available products from SAP Complementary Software Partners certified by SAP for the BC-SNC interface, see **www.sap.com/csp**, under the keyword "Network Security."

Figure 4 *Logon Using Secude for R/3*

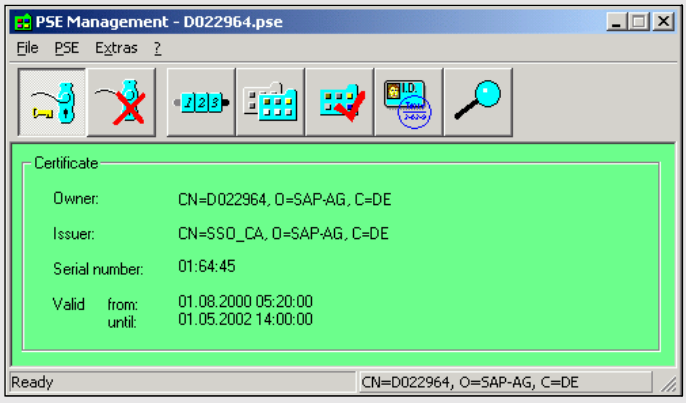
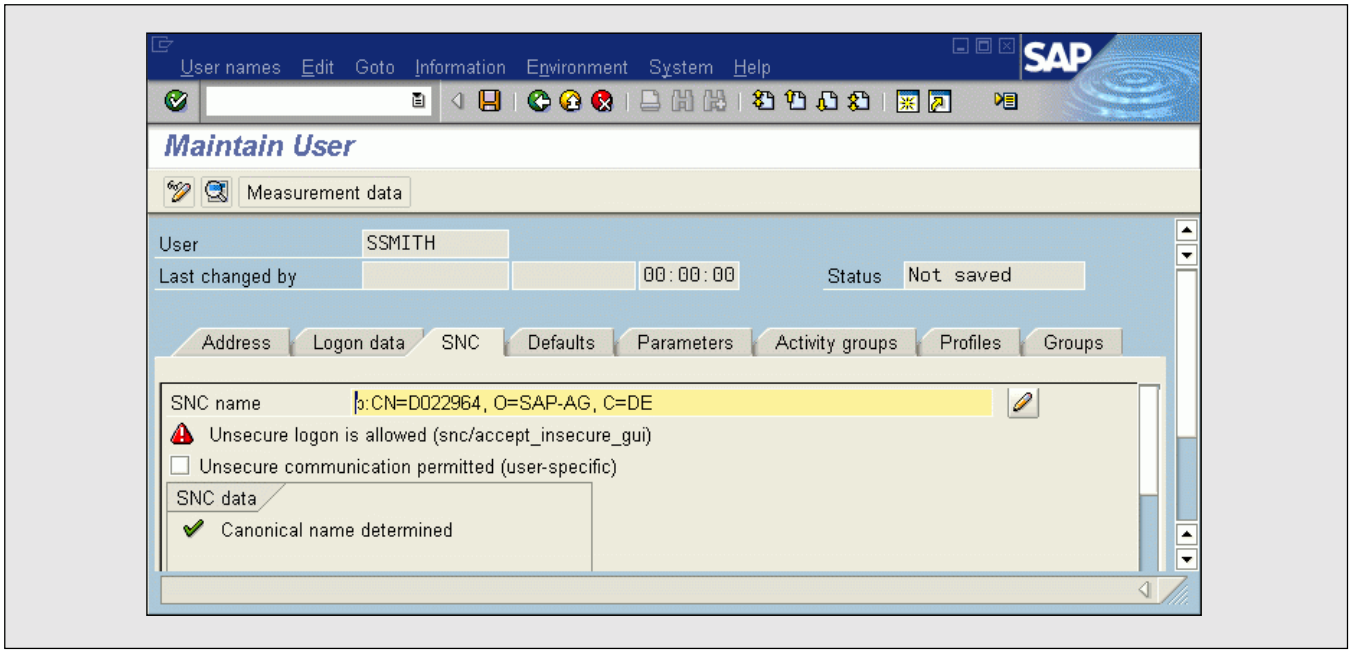


Figure 5 Configuring SNC Names for SAP Users

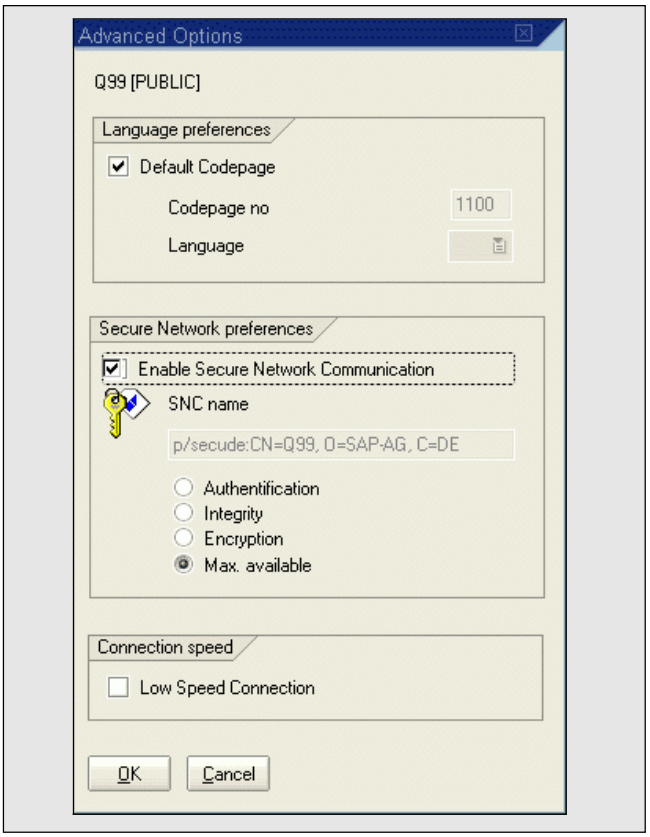


How is the mapping between the SNC name and the name Sally goes by in the SAP system affected? An SAP administrator has to facilitate this. On all SNC-enabled application servers, the standard user management transaction SU01 shows an additional card folder to configure the user's SNC name (see **Figure 5**). There are also reports for mass generation and maintenance of the mapping between SAP user IDs and their corresponding SNC names. At the SAP GUI for Windows, you can configure SNC with the SAP Logon tool by using the right mouse button on a logon group entry and maintaining advanced options, as shown in **Figure 6**.

The following SAP product components all have an additional SNC layer:

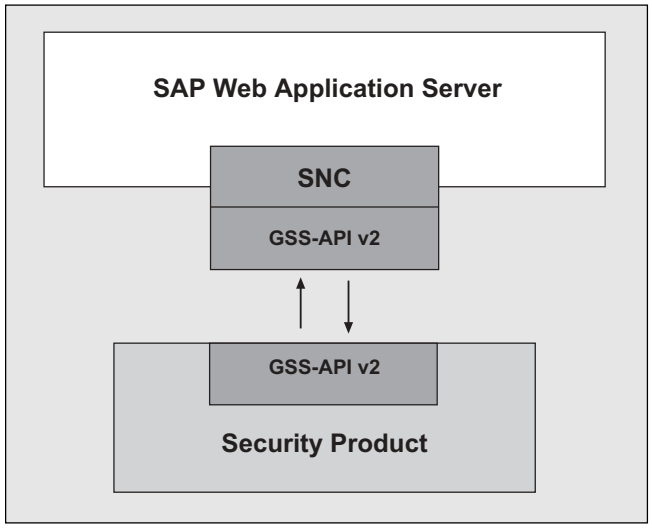
- SAP Web Application Server
- SAP Graphical User Interface (SAP GUI)
- Remote Function Call (RFC) library
- SAP Gateway, SAProuter
- SAP Internet Transaction Server (ITS) WGate (Web Gate) and AGate (Application Gate) components

Figure 6 Configuring SNC with the SAP GUI for Windows



You elect whether or not to enable this layer, which exposes a "C" function call interface according to the "Generic Security Services API Version 2," an IETF standard that was defined with SAP participation between 1995 and 1998 (see **Figure 7**).[5] When the optional SNC layer is turned on, it loads an SNC product library that provides the GSS API Version 2 into the SAP runtime system.[6]

Figure 7 The Optional SNC Layer



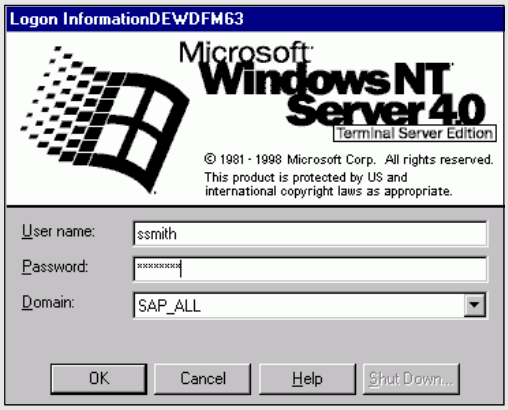
✓ Tip

Once a third-party security product from an SAP partner like Secude or Entrust assumes responsibility for user (and system) authentication, users actually log into that external security environment. The main advantages of this setup include: stronger authentication via challenge-response protocols and cryptography support as provided by the SNC product library of choice; single login into one security environment (we get back to this in the single sign-on section of this article); and the possibility of using smartcards to store user identity information.

✓ Tip

*As another alternative for choosing an SNC product, SAP does provide mapping libraries for Windows NT (called GSSNTLM.DLL) and Windows 2000 (called GSSKRB5.DLL), which do not contain cryptographic functions for authentication and encryption themselves, but call into the Microsoft Security Provider APIs on these platforms. If either of these mapping libraries is used as the SNC product library, our friend Sally's initial logon screen is the Windows NT or Windows 2000 logon (see **Figure 8**).*

Figure 8 Using Windows NT/2000 Domain Logon for SNC

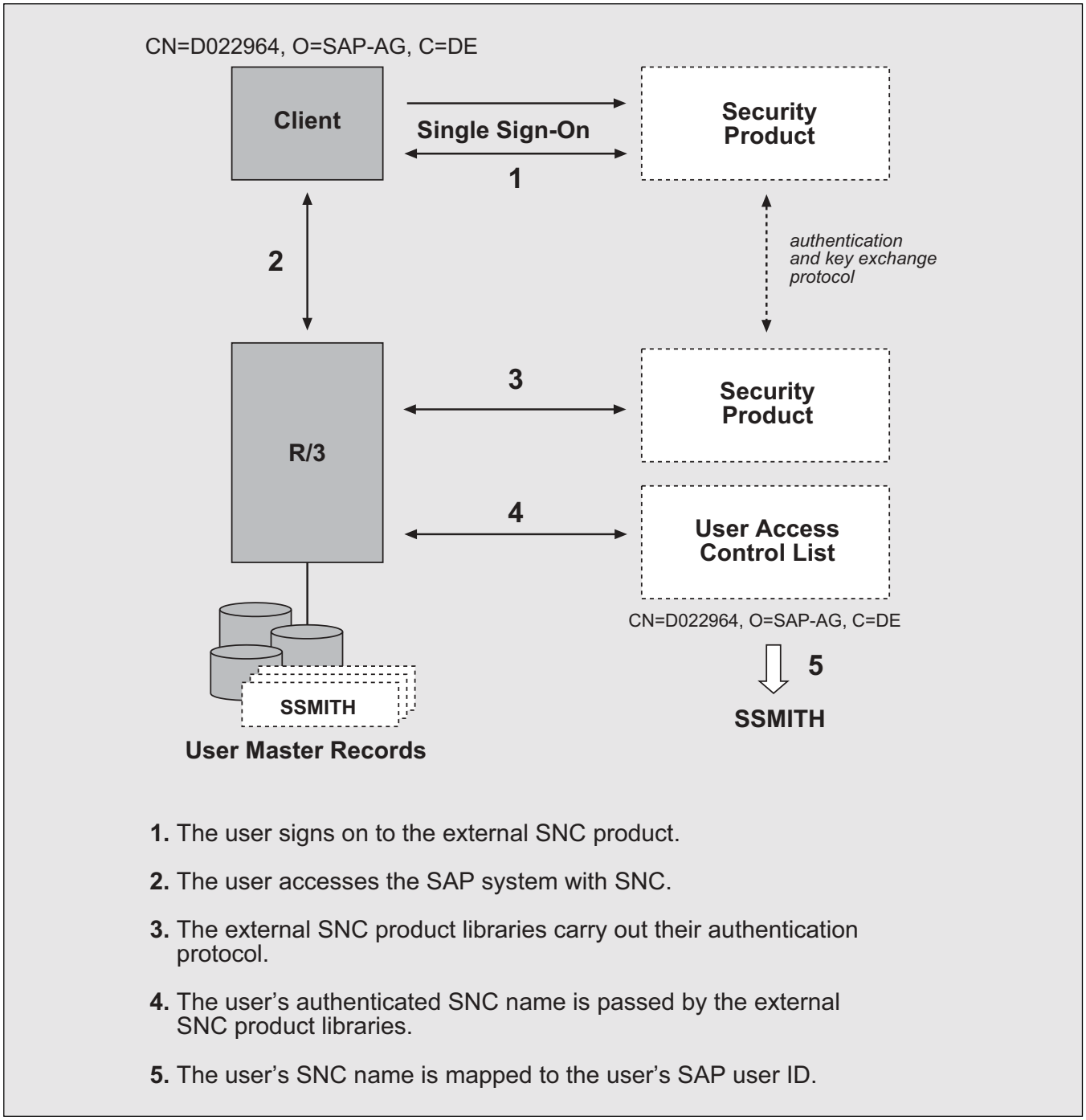


The mapping library provided by SAP has to be configured as the SNC product in your SNC-enabled SAP systems. When using the mapping library for Windows NT, for example, the SNC names for your SAP users are of the form <NT Domain>\<NT Userid>. For Sally, this could be something like "CompanyXDomain\SSMITH" or "CompanyXDomain\D022964," depending on whether person names or employee numbers are used for the Windows NT user IDs. In this case, your SAP administrator needs to maintain the mappings from these SNC names to SAP user IDs. (See SAP note 352295 for more details.)

[5] Detailed information about how to configure SNC with the various product options described in this section, and how to enable SNC across different SAP system components, can be found in the "SNC User's Guide" handbook available at **http://service.sap.com/security** under "Media Center."

[6] You have to obtain this library from the vendor whose product you have selected to perform the external authentication. For SNC protection of SAP server-to-server communications only, SAP also provides a default SNC product library (Secude OEM). See **http://service.sap.com/security** and look for "SAPCRYPTOLIB."

Figure 9 *Logon Process with SNC*



Once loaded into the SAP runtime process, the SNC product libraries that reside on the systems of the connection initiator and connection acceptor carry out their own proprietary authentication protocol (see **Figure 9**). Note that the SAP password of the user is not used at any time during SNC user authentication. And here's a highly desirable added bonus — data that is sent and received upon completion of the authentication process can be encrypted using the exchanged cryptographic key information!

One of the major limitations of SNC in its current form is that it cannot be used from a web browser. You can protect the SAP protocols (DIAG, RFC, ITS AGate/WGate) with SNC, but not the Internet standard protocol HTTP. So any access to SAP systems from a web browser (via SAP Internet Transaction Server or SAP Web Application Server) cannot be authenticated using SNC. As such, SNC can still be the mechanism of choice for SAP access via the SAP GUI for Windows or the SAP GUI for Java, but it needs to be complemented by another logon mechanism applicable for web access, such as HTTP Basic Authentication or X.509 digital certificates.

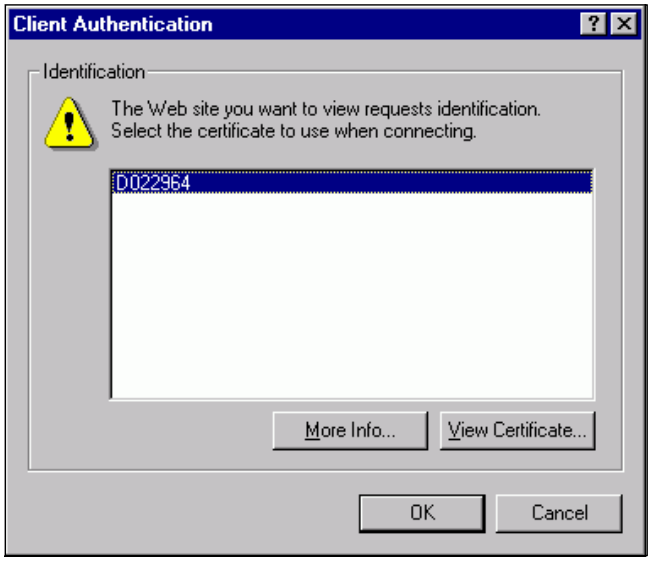
X.509 Digital Certificates

When accessing an SAP system from a web browser, X.509 digital certificates can be used for user authentication. This option requires the existence of a public key infrastructure (PKI), provided by a trust center internal to your company or from an external trust center provider. Through the PKI, all your users get private cryptographic keys and corresponding X.509 digital certificates, which become their digital identity cards. The investment in such a PKI is high, but then again, so are the benefits associated with this authentication approach:

- Very strong user (client) and system (server) authentication as part of the Secure Sockets Layer (SSL) protocol.
- Secure authentication protocol that cannot be attacked by listening to the network communications (eavesdropping).
- Use of an Internet standard protocol available to SAP and non-SAP applications.

The SSL protocol, which underlies web communications with HTTPS, features an optional client authentication that can be requested by the web server when HTTPS URLs are used (see **Figure 10** for the corresponding digital certificate selection pop-up in Internet Explorer). This is a standard authentication protocol supported by all of the commercial web servers today where you would run the SAP Internet Transaction Server (ITS) WGate. This feature is also supported by the SAP Web Application Server.

Figure 10 X.509 Digital Certificate Selection Prompt for SSL Client Authentication

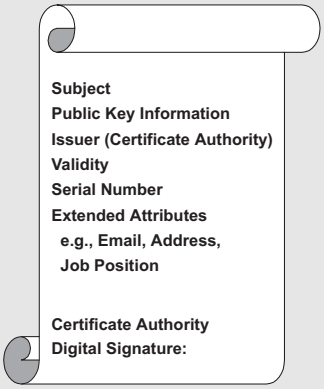


✓ Tip

User logon to SAP systems with X.509 digital certificates via the ITS became available with Release 4.5B. This is a secure, standard protocol that can be used transparently in your intranet and on the Internet at the same time. One of the benefits of this authentication method is that you can dispense with passwords. Users would not need to render their passwords to access the SAP system. Supplying a password would only take place when accessing his or her PIN-protected certificate. The storage and protection of the private key belonging to each user's certificate is an important issue. If PIN-protected smartcards are used, the private key is well protected and can be removed from the system by the user. If the private key is imported into the web browser and stored on the PC workstation, it can be protected by a password, but also depends on the operating system security of the PC workstation.

What Is a Digital Certificate?

A user's digital certificate, according to the X.509 standard, basically defines a unique binding between that person and his or her public key. This binding is valid for a certain period of time (the certificate validity period) and confirmed by the trust center that issued the certificate by digitally signing the certificate, as shown in the diagram on the left.

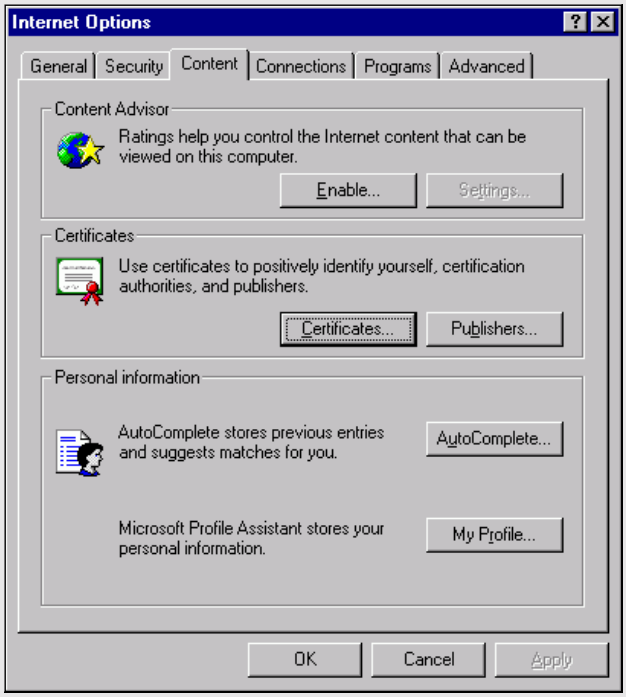


The trust center's digital signature protects the authenticity and integrity of the certificate information, so that a certificate can be sent over unprotected communication paths without problems.

Based on the server authentication, a web browser can securely exchange a fresh session key with a web server and use it for encrypting the whole communication that belongs to that web request.

The digital certificate itself is not a highly guarded secret. It is the matching private key that the user keeps secret using a PIN-protected smartcard, for example, or an encrypted file managed by a web browser product. The private key is required to create authentication data in the course of a challenge-response protocol, but the knowledge of the public key in the user's certificate is sufficient for others to check this authentication data.

A user's digital certificate can be imported into web browsers, or accessed from web browsers using a smartcard reader. For Microsoft Internet Explorer, you can look up the certificates available in your browser installation by selecting **Tools** → **Internet Options** → **Content** → **Certificates**, as shown in the screen shots on the right.



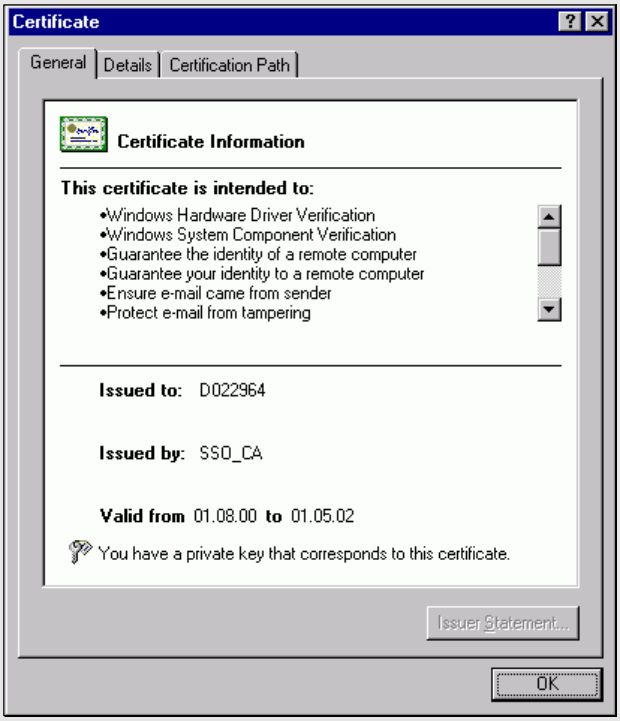


Figure 11 SAP Logon with X.509 Digital Certificates

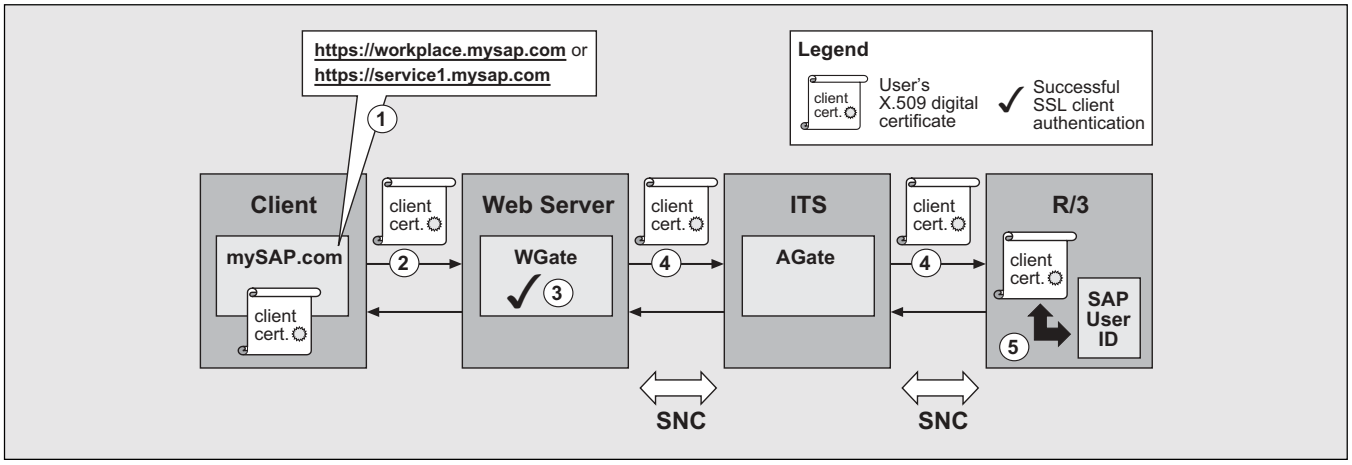
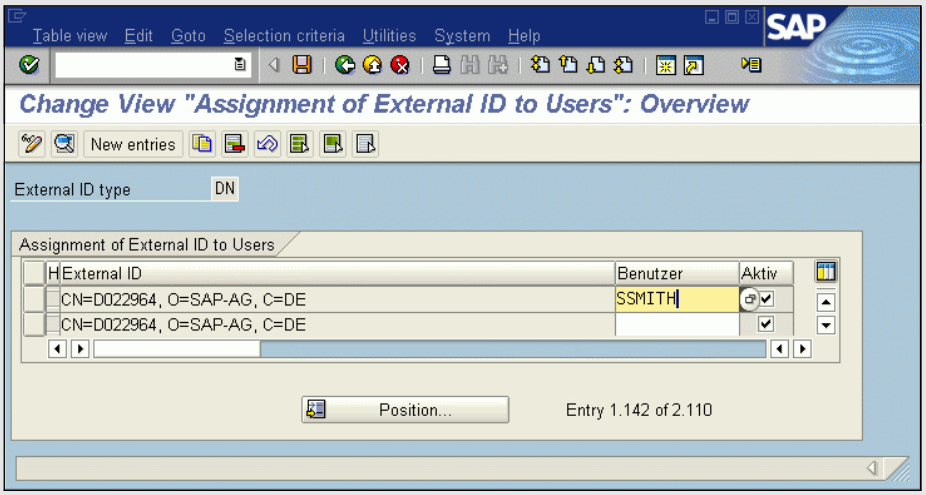


Figure 12 Maintaining the User Name Mapping for X.509 Digital Certificates



When using the ITS, the user's digital certificate is passed from the web server after successful SSL client authentication to the SAP Web Application Server via the ITS WGate and AGate components (see **Figure 11**). An SNC-protected communication path needs to be established for that purpose.

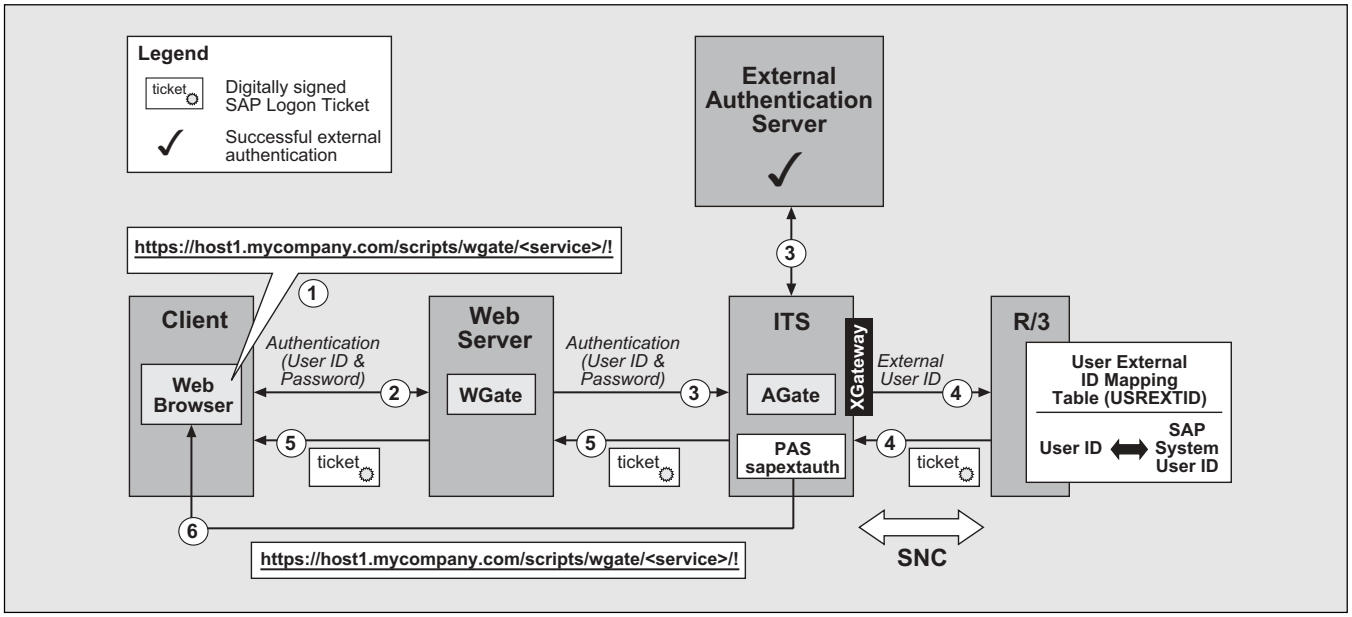
At the SAP Web Application Server, no further authentication takes place, but the server looks up whether a name mapping from the certificate owner's distinguished name to an SAP user ID exists in table USREXTID. This mapping can be maintained using transaction SM30 with view VUSREXTID (see **Figure 12**), or can also be generated using associated reports.[7] If no mapping entry exists for the user, or if the user is locked, the logon is rejected.

To use X.509 certificate logon, you have to configure the web server that is running the ITS WGate with HTTPS and then switch on SSL client authentication. At the ITS AGate, you have to set the ITS service file parameter[8] for X.509 certificate logon, `~clientcert`. This parameter specifies whether you want to deny, accept, or require

[7] See the detailed documentation "X.509 Certificate Logon via the ITS" provided at **http://service.sap.com/security** under "Media Center."

[8] Refer to the SAP ITS documentation for details.

Figure 13 Pluggable Authentication Services (PAS) over the SAP Internet Transaction Server (ITS)



client certificates according to the following three settings:

0: Client certificates are not accepted

1: Client certificates are accepted but not required

2: Client certificates are required

At the SAP Web Application Server, you have to enable X.509 certificate logon via two profile parameters, `snc/extid_login_diag` and `snc/extid_login_rfc`. Each of these parameters can have a value of 0 or 1. A value of 0 means that logon via X.509 digital certificates is disabled for this protocol. A value of 1 indicates that logon via X.509 digital certificates is enabled for this protocol.

These administrative settings represent only a fraction of the work needed to set up X.509 digital certificates in an SAP environment (or any other environment for that matter). The lion's share of administrative effort comes with establishing a public key infrastructure (PKI). You need a trust center to dole out the public and private keys, and you need a way to constantly administer key-related activities for all users. Many customers turn to an established trust center service like VeriSign in the US or Deutsche Telekom in Germany, just to provide two examples, to perform the activities of user registration and certificate enrollment, revocation, update, and renewal. Some customers also want to leverage internal trust centers they have built on their own, to support secure e-mail and other office applications with their SAP systems, for example. SAP has also started a trust center service for mySAP.com customers (see **http://service.sap.com/tcs** for details).

Pluggable Authentication Services (PAS)

With Release 4.6D, the SAP Internet Transaction Server (ITS) offers Pluggable Authentication Services (PAS). The idea here is that you can utilize an external authentication method by instantiating a specific ITS service using the ITS XGateway mechanism. A special XGateway library module, `sapextauth`, contained in the ITS Package `ntauth.car`, is provided by SAP with the ITS and loaded onto the ITS AGate. Upon successful external authentication, users receive a "logon ticket" that is recognized by the SAP system.

The process (shown in **Figure 13**) operates as follows:

1. After opening the web browser, a user accesses the SAP system via the ITS-configured PAS URL.

2. The user receives a logon page, which is customized for the external authentication method and provides the user's authentication information. This could be a user ID and password pair that is valid within the external authentication environment — for example, a distributed directory accessed via LDAP (Lightweight Directory Access Protocol). Authentication information might alternatively be in the form of a short-term password that is generated by hardware authentication tokens, such as SecureId cards from RSA Security.

3. When the logon page with the authentication data is posted back to the web server, the authentication data is sent from the ITS WGate to the ITS AGate. On the ITS AGate, the XGateway library forwards the data to the external authentication mechanism via a C programming language API defined for this purpose.

4. If the external authentication server returns a status of "ok," the XGateway on the ITS sends the external user name to the SAP system and requests a logon ticket for the user. The use of SNC is required to protect this communication path.

5. In your SAP system's user database, a mapping entry from the external user name to an SAP user ID needs to exist (table USREXTID) with the corresponding external identity type. If the mapping exists and the user isn't locked, an SAP Logon Ticket is created and returned in a non-persistent cookie stored in the user's web browser. What you need to understand is that the PAS maps the external authentication mechanism configured on the ITS AGate to the SAP Logon Ticket mechanism.

6. With the HTTP response from PAS, the user is redirected to the application he or she wants to access.

The ITS parameters you need to configure to support a PAS are listed in **Figure 14**.

Besides the PAS file, several ITS templates are used that need to be created or modified from the ITS standard delivery (`login.html`, `redirect.html`, `error.html`).[9]

✓ *Tip*

When user authentication data is passed to PAS, it should be protected by HTTPS communications as it travels from the web browser to the web server, and then to the ITS WGate. For the connection from the ITS WGate to the ITS AGate, the use of SNC is strongly recommended. From the ITS AGate to the SAP system, SNC is required. This is necessary to protect the SAP system from AGate spoofing attacks. With PAS, such attacks would enable attackers to get SAP Logon Tickets for any user, which is why strong authentication of the ITS AGate that is requesting the tickets is a must. The SAP Logon Ticket created for the authenticated user also benefits from the communication path protection with SNC and HTTPS on its way back into the user's web browser.

When does it make sense to use the PAS approach? This is your best bet when you want to integrate your SAP system web access authentication into an existing authentication infrastructure other than the SAP user ID and password and without using X.509 digital certificates and the PKI. Such an existing authentication infrastructure could be directory-based authentication via the Lightweight Directory Access Protocol (LDAP) or your Windows NT/Windows 2000 infrastructure (if you are also using Microsoft's Internet Information Server as a web server and Internet Explorer as a web browser). Through the PAS configured on your ITS, your users' authentication data is just passed through from the

[9] For further information, refer to the detailed PAS documentation "SAP Pluggable Authentication Services" provided at **http://service.sap.com/security** under "Media Center."

Figure 14 ITS Service File Parameters for Pluggable Authentication Services (PAS)

Parameter	Allowed Values	Description
~xgateway	sapextauth	Specifies that the XGateway sapextauth should be used.
~extauthtype	exe, dll, ntlm, ntpasswd, x509	Type of external authentication. The following types are allowed[10]: • External executable program or shared library (exe, dll) • Windows NTLM authentication (ntlm) • Password authentication on the Windows NT Domain Controller (ntpasswd) • The use of X.509 client certificates (x509)
~extauthmodule	<name of authenticating module>	Specifies the external module that performs the authentication. This parameter is only necessary if ~extauthtype = exe or dll.
~extid_type	<user-defined>	The type of external identification used for the mapping in table USREXTID. This parameter does not need to be specified if ~extauthtype = ntlm, ntpasswd, or x509.
~ntdomain	<Windows NT domain>	If ~extauthtype = ntpasswd and your users exist in a single Windows NT domain, then you can use this parameter to define the domain in the service file. Otherwise, you need to include the domain in the login template.
~redirectHost ~redirectPath ~redirectQS ~redirectHttps	<host_name> <path> <query_string> 0: Use HTTP 1: Use HTTPS	Data that is used for the redirect URL. The defaults for each of the parameters is the value of the current request.
~extrequesttimeout	<user-defined>	If ~extauthtype = exe, then use this parameter to define the number of seconds after which the external process should terminate by force.
~dont_recreate_ticket	0: Create ticket with each request 1: Create ticket once only	Determines whether a ticket should be created with each request or only created if no ticket is present.
~login_to_upcase	0: Do not convert 1: Convert	Convert the ~login string (user ID) to uppercase before submitting the ticket request to the backend. This may be necessary if the user ID entries in the mapping table (USREXTID) are maintained in capital letters. (The entries in USREXTID are case-sensitive.)

[10] Two variants of Windows NT domain authentication are also possible as Pluggable Authentication Services: using the result of NTLM authentication from your Microsoft Internet Information Server and verifying user ID and password against the Windows NT Domain Controller.

web request to the external authentication environment of your choice (LDAP Directory, Windows NT Domain Controller, Remote Access Server, and more). The benefits of this approach include:

- No additional authentication with SAP user ID and password is needed.
- Since an SAP Logon Ticket is created if PAS reports successful external authentication, it can serve as the initial authentication for single sign-on to your SAP system landscape.

Single Sign-On for a "Standalone" SAP System

In this section we look at ways to alleviate situations where users must repeatedly enter their authentication information in order to gain access to the same SAP system over and over again. This is very common in environments where you use a stateless protocol, such as HTTP, or users frequently start new sessions over the SAP DIAG or RFC protocols, instead of staying within one session all the time (for example, if they need to quickly access many different SAP transactions and work with them in parallel).

For SAP GUI for Windows users, SNC actually delivers the SSO mechanism. For browser access, there are two options. The first is the SAP Logon Ticket used in tandem with the standard SAP user ID/password mechanism, PAS, or X.509 digital certificates. The second is using X.509 digital certificates straight up — that is to say, without the SAP Logon Ticket for each and every required authentication. From a security perspective, using digital certificates without the SAP Logon Ticket is my strong preference, but recognize that performance with this approach might not be optimal for some customers.

SNC Equals SSO!

If you are already using the secure network communications (SNC) option, you've already got your SSO solution in place for SAP GUI for Windows users. SNC is an SSO mechanism! The SNC layer in SAP system components assumes that users have already completed an initial authentication step in the external security environment as provided by the SNC product, and each SNC logon is a repeated user authentication. When the SAP logon is carried out with SNC, authentication is done by the two SNC product libraries loaded into the corresponding SAP system components. The process assumes that the SNC product library on the client side has already acquired credentials for repeated user authentication and uses these credentials when the SAP system is accessed.

So, if SNC has been configured by your SAP system administrator, how does a user like Sally Smith benefit from it? For example, if Secude for R/3 is used as the SNC product, Sally arrives bright and early to work and the first thing she does is log on to the PSE Management tool, as shown back in Figure 4. She has to provide one password (or PIN) to grant access to her private key information and digital certificate, which is managed by the Secude product in an encrypted file or on a smartcard. As long as Sally is logged on to the Secude product, she can start SAP GUI for Windows sessions and RFC connections to SAP systems using SNC, and the logon process to the SAP system is carried out by the Secude libraries loaded into the SAP GUI and SAP Web Application Server processes. Sally doesn't have to type in any SAP user ID and password; with her initial menu, she will be welcomed by the SAP system. As another example, if the Windows NT domain logon is configured as the SNC product, Sally's initial logon to Windows NT, as shown in Figure 8, is sufficient to start SAP GUI and RFC sessions.

Using SAP Logon Tickets

Since SNC cannot be used from a web browser, a different single sign-on mechanism had to be provided based on available web technology[11] with the

[11] The Generic Security Services API Version 2 used for SNC is not available in standard web browsers.

SAP Internet Transaction Server (ITS) and the SAP Web Application Server. This was the motivation for introducing the SAP Logon Ticket mechanism. But it was never our intent to create a logon ticket that would act as an initial authentication mechanism in and of itself. That is why, to receive an SAP Logon Ticket, users must first log on once using one of what I like to refer to as the "initial authentication" options — the standard SAP user ID and password, X.509 digital certificates, or a PAS such as the Windows NT domain or Windows 2000 logon carried over the ITS.

For the user (Sally, in our example), the initial logon process when using a web browser usually means that she has to type her user ID and password information (either an SAP user ID and password or an external user ID and password when using PAS) into an HTML page or Basic Authentication pop-up as displayed by the web browser upon first access to the SAP system via a corresponding URL (see the examples provided back in Figures 2 and 3). With Sally's initial logon into Secude for R/3 (see Figure 4), for example, or with her Windows NT/Windows 2000 logon (see Figure 8), she doesn't have to provide any further user ID and password information because these products can make her authentication information available in her web browser.

Regardless of the initial authentication method you elect to use, you configure an SAP Basis Release 4.6D system to create an SAP Logon Ticket for the user and return that ticket to the ITS AGate if the initial authentication is successful (see Figure 13, which shows the use of PAS, but you can also run your SAP user ID and password directly against the SAP database over the same communication path). The ITS then forwards the SAP Logon Ticket to the user's web browser in the HTTP response using the HTTP cookie mechanism. With the SAP Web Application Server, the HTTP response, including the cookie containing the SAP Logon Ticket, is sent directly back to the web browser.

The cookie containing the user's SAP Logon Ticket is called `MYSAPSSO2`.[12] If such a cookie exists in the user's web browser, the cookie is sent with each HTTP request to the web server that created the cookie or to web servers belonging to the same DNS domain, depending on the domain specification set with the cookie. With the HTTP requests that follow after the initial logon, the ITS server or SAP Web Application Server receives the `MYSAPSSO2` cookie, retrieves the user's SAP Logon Ticket, and uses it for logon to the SAP system.

To configure the use of the SAP Logon Ticket mechanism, you have to set the corresponding service file parameters on the ITS AGate (see **Figure 15**).

[12] The predecessor to the SAP Logon Ticket was a mechanism that was also based on cookies. In this early cookie mechanism, included with the first versions of the mySAP Workplace, the SAP user ID and password information was stored, in an encrypted fashion, in a cookie called `MYSAPSSO`. This mechanism had several drawbacks, the most severe one being the requirement to use identical passwords for all SAP systems for which access is required. Use of the `MYSAPSSO` cookie is still possible, but not recommended, and there is an ITS service file parameter to switch it off (`~mysapcomnosso1cookie`, see Figure 15). Since the SAP Logon Ticket is available now, SAP plans to remove the first cookie version as soon as possible.

Figure 15 ITS Service File Parameters for the SAP Logon Ticket

global.srvc: Parameter	Value	Comment
~login ~password	(space) (space)	When these parameters are empty in both global.srvc and the individual service file, the user is prompted for his or her user ID and password when initially logging on to the ITS service.

(continued on next page)

Figure 15 (continued)

global.srvc: Parameter	Value	Comment
~cookies	1	Enables the creation of cookies on the ITS.
~mysapcomusesso2cookie	1	Enables the user to log on to the system using an existing SAP Logon Ticket.
~mysapcomnosso1cookie	0: SSO cookies and SAP Logon Tickets 1: SAP Logon Tickets only	Disables the creation of SSO cookies in the mySAP Workplace.
<service>.srvc: Parameter	**Value**	**Comment**
~login ~password	(space) (space)	See above.
~mysapcomssonoits	1	Ensures that the SAP system client is not included in the SAP Logon Ticket, allowing for single sign-on across multiple SAP system clients.
~mysapcomgetsso2cookie	1	Enables the creation of the SAP Logon Ticket after successful logon.
~ssorequiressl	0: If you want to allow transfer of SAP Logon Tickets over unencrypted HTTP connections 1: If you want SAP Logon Tickets to be sent only over encrypted HTTPS connections	Enforces transfer of SAP Logon Tickets only over encrypted HTTPS connections.

Figure 16 SAP Web Application Server Profile Parameters for the SAP Logon Ticket

Set the Parameter	Value	Comment
login/accept_sso2_ticket	1	Allows the SAP Web Application Server to accept an existing SAP Logon Ticket for logon.
login/create_sso2_ticket	1: If the SAP Web Application Server's certificate is to be included in the logon ticket 2: If the SAP Web Application Server's certificate is not to be included in the logon ticket	For best results, set this parameter to the value 1 if the SAP Web Application Server possesses a certificate signed by the SAP Trust Center. Set it to the value 2 if the certificate is self-signed.
login/ticket_expiration_time	Desired value	Default = 60 hours. See SAP note 337794 for information about how to set the expiration time in minutes.

You can set these parameters in the `global.srvc` file for all ITS services, or in the individual service files. If you use the SAP Web Application Server to support SSO with logon tickets, it would need to be configured for creation/acceptance of SAP Logon Tickets via the profile parameters you see listed in **Figure 16**.

The SAP Logon Ticket contains the SAP user ID of the user for which it enables logon. It also contains a creation time-stamp and is only valid for a short period of time, which can be configured from a few minutes to several hours. To protect the authenticity and integrity of the SAP Logon Ticket data, it is digitally signed by the SAP system that created it.

✓ *Tip*

When implementing SSO for standalone SAP systems with SAP Logon Tickets, be mindful of two critical security facts:

- *The HTTP cookie mechanism has certain well-known risks, which can be reduced by additional measures.*
- *The SAP Logon Ticket information contained in the MYSAPSSO2 cookie needs to be closely guarded. You don't want this information to fall into the wrong hands. It will compromise the security of your SAP systems.*

To protect the MYSAPSSO2 cookie at the client side, it is created as a non-persistent cookie in the user's web browser. Non-persistent cookies only exist in the browser's main memory and are deleted when the user logs off or when the browser is closed. To protect the MYSAPSSO2 cookie from eavesdropping attempts on communication lines, I strongly recommend that you only use it with HTTPS. This can be enforced by setting the ITS service file parameter ~ssorequiressl (see Figure 15). For protection at the server side, it is also possible to restrict the MYSAPSSO2 cookie to be only sent back to the web server that created it. This is recommended for single sign-on to a standalone SAP system.

The SAP Logon Ticket, in combination with the HTTP cookie mechanism, offers a convenient single sign-on mechanism for accessing SAP systems from a web browser. Its main advantage over using X.509 digital certificates all the time (i.e., not only for initial authentication but also for each authentication process) is that it doesn't require a PKI, or even if a PKI exists, it provides better performance since the SSL client authentication protocol part, which is rather expensive due to the cryptography involved, is only used for the initial authentication. However, the risks involved with the HTTP cookie mechanism are still a matter of debate. If cookies are sent back to several servers in the same domain, care has to be taken that there are no malicious applications running on these web servers, which could gather `MYSAPSSO2` cookies and use them without permission. In the past, there were also a number of browser implementation bugs that could be exploited by malicious web applications to steal cookies from the client. In terms of security, using X.509 digital certificates with the SSL protocol for each authentication required is clearly the superior technique on the web.

Using X.509 Digital Certificates Exclusively

The X.509 digital certificate logon approach, which takes place over the SSL protocol with HTTPS and SSL client authentication, is a very nice and very secure solution for single sign-on, and, of course, can also be used without the creation of SAP Logon Tickets. Using SSL client authentication each time means that no sensitive information, such as a password or SAP Logon Ticket, needs to be transferred over network connections (so you also don't need cookies). After the user (Sally, in our example) grants access to her certificate by providing her password to the web browser, the SSL authentication protocol is carried out without additional input required from Sally. (The access to the certificate can be protected by a PIN, in the case of smartcards, or by a password when the certificate is imported into the web browser and stored somewhere in the file system.)

The security and beauty of using X.509 digital

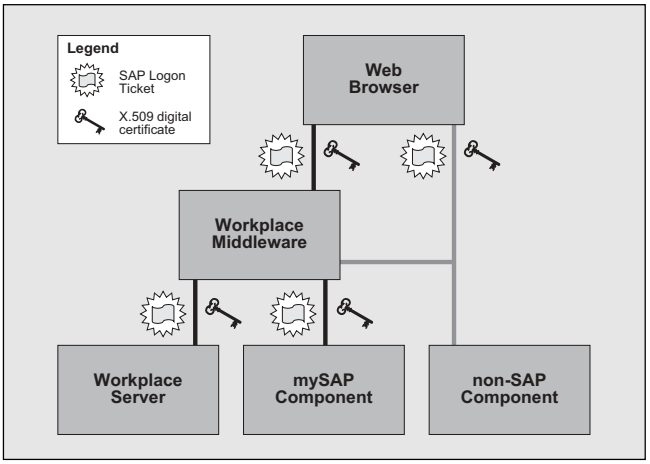
certificates for accessing SAP systems (and other systems that support this Internet standard) has its price, however. Your users have to receive their individual certificates via a secure certificate enrollment process, and you have to manage certificate revocation and renewal in a PKI. It also requires more processing time during each authentication process, which can be alleviated using SSL session caching, but is still slower than using cached SAP Logon Tickets (we are coming up against the well-known tradeoff between security and performance here).

> ✓ ***Tip***
>
> *The convenience of the single sign-on achieved this way depends on how password protection for your certificate is handled by your web browser. With Internet Explorer, for example, you either have the choice of no password or having to provide the password each time the certificate is used for authentication. The latter case is not a single sign-on, whereas the former case actually means that the certificate in your browser is unprotected from users who have administrator access to your PC. If you are the only user of your PC and you trust your administrators, this can still be a viable option. If multiple users share your workstation, it is not recommended.*

Single Sign-On with the mySAP Workplace

I've saved the best for last — single sign-on to a full landscape of SAP and non-SAP systems. This feature is provided with the mySAP Workplace. Imagine Sally now seated at her mySAP Workplace client. She comes in bright and early, completes her initial logon with one of the three "initial authentication" mechanisms described earlier (standard SAP user ID/password, X.509 digital certificates, or PAS), and proceeds with her personal menu displayed in her web browser. For the rest of the day, no further application running in the Workplace system itself or in one of the Workplace component systems will present her with a logon screen. Wouldn't that be nice? The mySAP Workplace is aiming at that goal.

Figure 17 mySAP Workplace Single Sign-On



Behind the scenes, SAP Logon Tickets, X.509 digital certificates, or both come into play (see **Figure 17**). You have a choice of one or the other, or even using both, for implementing SSO with the mySAP Workplace:

✓ I recommend you use the SAP Logon Ticket in situations where you don't have a PKI in place and you do not want to invest in setting up a PKI, so your users don't use X.509 digital certificates. If your SAP system landscape is your main concern, your users use their SAP user ID and password for initial authentication against the Workplace system and the SAP Logon Ticket for SSO. If you want to leverage your Windows NT/Windows 2000 infrastructure, or your corporate LDAP Directory service, set up the corresponding PAS for initial authentication against your Workplace system and use the SAP Logon Ticket for SSO. The benefits of this approach include implementation of SSO with minimal effort and the ability to integrate your SAP system landscape with your existing authentication infrastructure.

✓ I recommend you use X.509 digital certificates in situations where you already have a PKI in place, or your security requirements demand strong

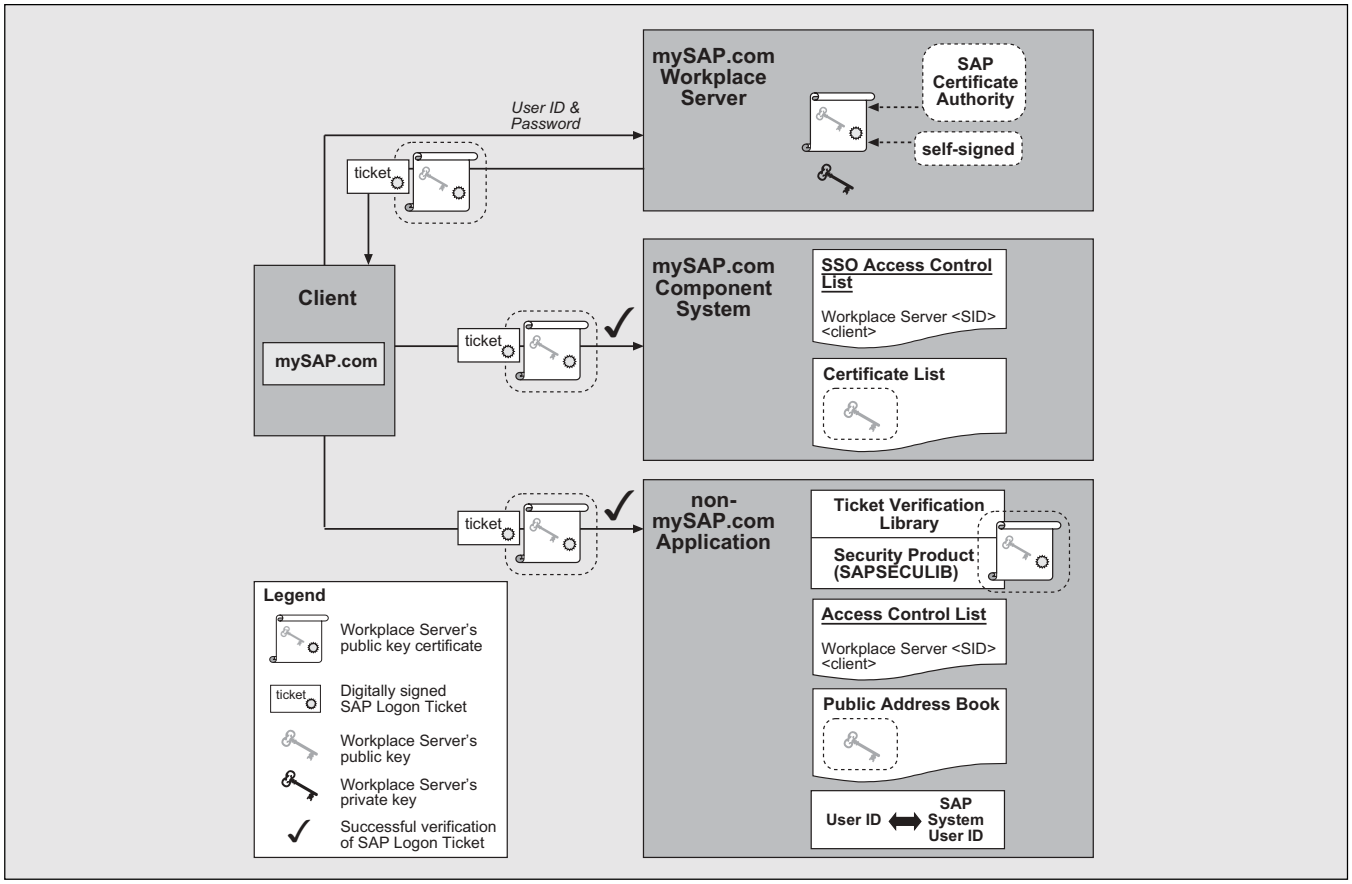
authentication procedures for all applications integrated under your mySAP Workplace system, so you are willing to set up or connect to a PKI service. The benefits of this approach include strong authentication procedure using the SSL protocol (which has passed public security reviews for years now), the possibility of using smartcards to store your users' identity information, usage of an Internet standard protocol, and the possibility of extending your SSO solution over the Internet to business partners and public services.

✓ I recommend you use *both* X.509 digital certificates and SAP Logon Tickets in situations where you want to achieve a secure initial logon (with X.509 digital certificates using the SSL protocol) against your Workplace system, but then switch to the simpler SAP Logon Ticket mechanism for accessing Workplace component systems. This is because performance becomes an issue, or simply because your older release SAP systems or third-party applications do not support X.509 certificate logon. The benefits of this approach include secure initial logon, easy integration of older releases of SAP systems and third-party applications supporting the SAP Logon Ticket mechanism, and better authentication performance for subsequent authentication processes after the initial logon.

Using the SAP Logon Ticket

Users carry out their initial logon against the Workplace system using one of the "initial authentication" mechanisms — the standard SAP user ID/password, X.509 digital certificates, or PAS — and receive an SAP Logon Ticket that is accepted by all Workplace component systems (see **Figure 18**).

Figure 18 SAP Logon Ticket Mechanism with the mySAP Workplace



The SAP Logon Ticket mechanism became available with the mySAP Workplace 2.10 release, the first Workplace release built on SAP Basis Release 4.6D.[13] The integration of non-SAP systems is facilitated by an external ticket verification library provided for all SAP operating system platforms. The ticket verification library is available from SAP (contact SAP via e-mail at security@sap.com) as a C runtime library, COM object, or Java class. It can be integrated into non-SAP applications, which need to be modified, however, so that they call into the ticket verification library for verifying the validity of an SAP Logon Ticket received by the application.

To achieve single sign-on, the Workplace component systems must be configured to accept SAP Logon Tickets from the issuing Workplace system(s). Therefore, each component system has an access control list, which contains the names of all Workplace systems from which the component system accepts SAP Logon Tickets. For security reasons, this access control list needs to be maintained in each component system. In addition, the SAP Logon Ticket is digitally signed by the issuing SAP system using public key cryptography. The ticket-issuing system uses its private key to sign tickets. To be able to verify the digital signature, accepting systems need access to the public key of the ticket-issuing SAP system. There are two possible ways to pass this public key to the component system:

- The first possibility is to retrieve the public key from the ticket-issuing system during maintenance of the access control list for ticket acceptance via RFC. This method is required when your ticket-issuing system uses a so-called "self-signed" signing certificate — i.e., a certificate that was not signed by a trust center.

- The second possibility is to pass the public key of the ticket-issuing system with the SAP Logon Ticket as part of the digital signature. This method is possible if the ticket-issuing system uses a certificate that was signed by a trust center. In this case, the validity of the ticket signature and the signer certificate sent with the signature can be verified by the accepting system if the accepting system has access to the public root key of the trust center. If the ticket-issuing system uses a signing certificate that was issued by the SAP Trust Center, the root key is embedded within the SAP Logon Ticket verification library. Root keys from other trust centers can be configured in SAP systems using transaction STRUST — for the external ticket verification library there is a command line executable.

This may all sound difficult, especially if you are not familiar with public key technology. Therefore, SAP provides the SSO administration wizard transaction SSO2 (see **Figure 19**) to be used with SAP component systems for configuring single sign-on with SAP Logon Tickets issued from a central system, such as the mySAP Workplace. Using this wizard, you immediately get a status overview of public key certificates configured in your system and which ticket-issuing systems are contained in your SSO access control list. You can use SSO2 for adding and deleting ticket-issuing systems to and from the list. In case you need the public key of a new ticket-issuing system, SSO2 gets it for you via RFC and puts the key into your local keystore.[14]

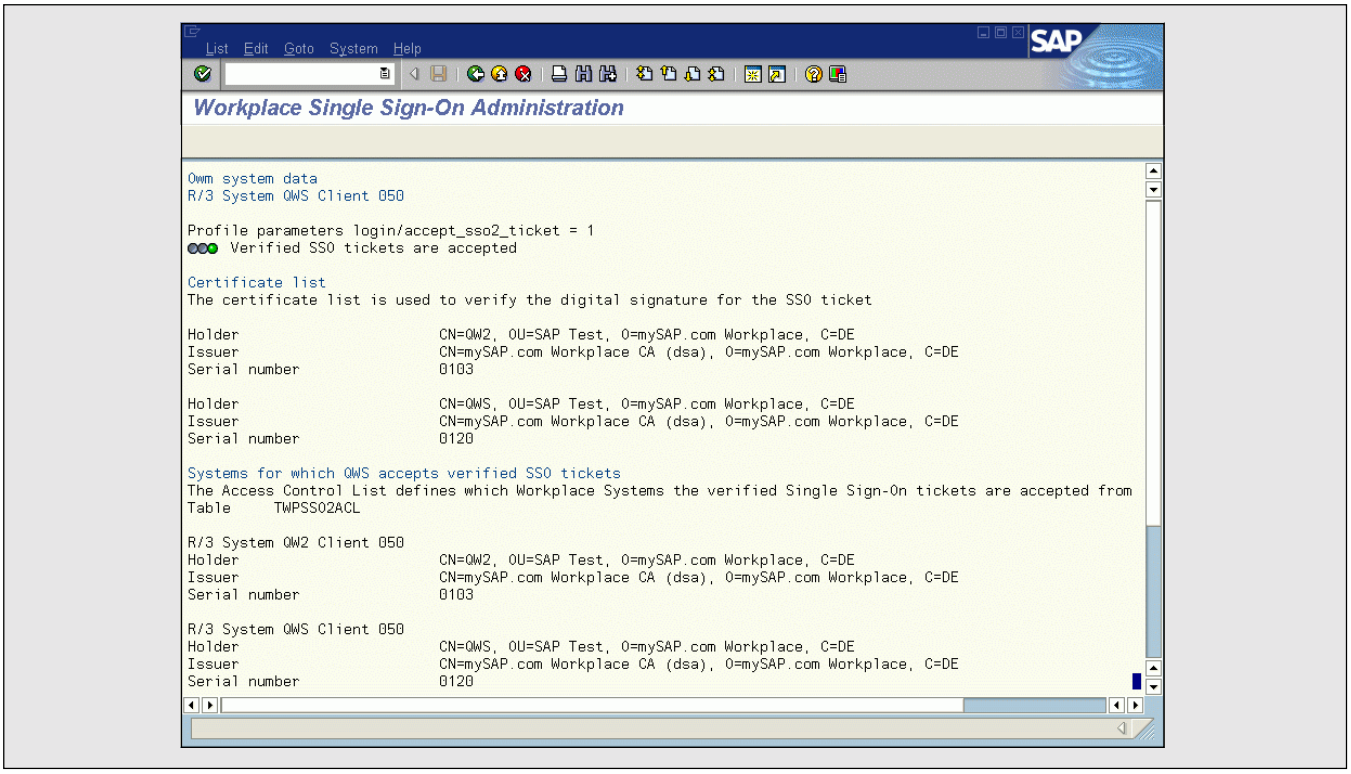
Using X.509 Digital Certificates

The second option for single sign-on with the mySAP Workplace is to use X.509 digital certificates. Your users access the Workplace and its component systems using HTTPS URLs with SSL client authentication. The benefits, you might recall, include very strong user and system authentication via the SSL protocol, resistance to "network

[13] mySAP Workplace component systems accept an SAP Logon Ticket back to Release 4.0B (SAP kernel patches are required for releases lower than 4.6D).

[14] For further details, including how to get your Workplace server's certificate signed by a trust center, such as the SAP Trust Center, refer to the detailed documentation provided in the document "Single Sign-On Solutions with the mySAP Workplace" available at **http://service.sap.com/security** under "Media Center."

Figure 19 Configuring Single Sign-On with SAP Transaction SSO2 in an SAP Component System



listening" attacks, and compatibility with both SAP and non-SAP systems.[15]

Using X.509 digital certificates requires you to set up and administer a PKI. This is not unique to SSO across a varied system landscape (the topic of this section). This type of administrative investment holds true whether you are setting up X.509 digital certificates for use in initial authentication activities, as well as for setting up SSO for repeated access to just one standalone system. So if you already have a PKI in place — set up previously for your SAP systems or for other systems, like secure e-mail or Lotus Notes — you can leverage that (as long as the PKI is based on the X.509 certificate standard). If you do not yet have a PKI in place, SAP does offer some relief by way of the SAP Trust Center Service. Each mySAP Workplace contains a "Registration Authority" function that can be used to automate the certificate enrollment process for your users. The SAP Trust Center Service provides an interesting option for entering into the world of PKI without the burden of running your own PKI, and is free-of-charge for mySAP Workplace customers.[16]

Customers can also use a trust center service from an SAP partner to enable certificate-based user logon in the Workplace, or they can follow an internal trust center approach.

Combined Solutions

You have just seen that the mySAP Workplace offers the two principal SSO mechanisms — SAP Logon Ticket and X.509 digital certificates. There are times when you might want to combine these two approaches and use X.509 digital certificates to log on to the Workplace and then receive an SAP Logon Ticket. This approach is useful in

[15] The configuration steps required for your web servers, ITS, Workplace, and component systems are all described in detail in the document "Single Sign-On Solutions with the mySAP Workplace" available at **http://service.sap.com/security** under "Media Center."

[16] More details about the SAP Trust Center Service are available at **http://service.sap.com/tcs**.

environments where not all of your Workplace component systems support SSL client authentication and certificate logon.

The authentication and SSO method used can then be selected by the individual component system. It has to be noted, however, that once present, the `MYSAPSSO2` cookie containing the SAP Logon Ticket is always sent with HTTP requests, even if a component system uses certificate logon. So, the combination is useful for a transition period toward certificate-based logon, but it requires additional measures to protect the `MYSAPSSO2` cookie during network communications, as mentioned earlier.

✓ Tip

So what about SNC in the mySAP Workplace? Since you access the mySAP Workplace from a web browser, you cannot use SNC to reach your personalized Workplace browser menu page. However, if you have a Workplace component system configured to use SNC, these systems can still be accessed with SNC when an SAP GUI for Windows is started from the Workplace menu.

If this is done, keep in mind that the initial SNC user logon took place in the security environment of the SNC product used. This can be a different logon step than your Workplace logon, and the SAP software assumes that this was done deliberately. So don't be puzzled when your Workplace menu welcomes you with a different name than one of your component systems that uses your SNC logon. This can happen if the SNC user name mappings are different, or if you just used different accounts for initial authentication for SNC and for Workplace.

Conclusion

Authentication forms the foundation of your security infrastructure. Access privileges granted to users (or systems) are predicated on the notion that users are who they claim to be. Rock-solid authentication practices are therefore key to safeguarding your SAP systems. At the same time, your users are urgently requesting to reduce the number of logons they have to complete every day and are looking for a single sign-on solution. You have a number of options available to help you do this with SAP systems, and they are not all created equal!

Remember, each particular solution needs to be well designed and carefully deployed. Make your choice, but don't mix too much. For example, try to avoid using SAP user ID and password for the mySAP Workplace logon when using SNC in parallel with another authentication infrastructure and, for example, LDAP Directory-based pluggable authentication for a few additional applications. Although this is facilitated by the flexible options provided by SAP systems, you should strive for solutions that use the option that fits best with your existing IT infrastructure instead of using all of them. A large mixture of authentication options and single sign-on support can be confusing and difficult to oversee. With too many options used in parallel, or without adequate additional protection of communication paths for some of these options, your "single sign-on" may become a "multiple break-in" if even one of the options can be exploited.

Dr. Jürgen Schneider received his Diploma and his Ph.D. in computer science from the University of Kaiserslautern, Germany, in 1987 and 1991, respectively. Between 1991 and 1996 he led several research projects in the areas of network management and security at IBM's European Networking Center in Heidelberg. In 1996, Jürgen joined the SAP Security Basis development team in Walldorf, and since 1998 he has been the Development Manager for Security in SAP's Technology Development department. He can be reached at j.schneider@sap.com.

