Common Credential
A Sevan White Paper

sevan networks

This Sevan White Paper discusses the issues concerning user authentication across multiple applications in multiple enterprises. It contrasts Sevan's “Common Credential” approach with “Single Sign-On” and “PKI” solutions. We show that today's authentication solutions create a fragmented world-view. Users are either buried under piles of credentials, or they must depend on complex infrastructures for authentication. Sevan's approach allows a user to employ a single SSL certificate as his common credential. Users can conveniently and confidently use their common credentials for all of their authentication needs.

The scenario we are addressing is shown in Illustration 1. This hypothetical user must access six different applications, which are in three independent enterprises. His employer's applications might include an e-mail system, an human resources self-service application, and a corporate financial system. The user must also access two applications in a partner company as well as his personal banking information.

Today, users have an authentication credential for each application. These credentials are information or items that allow the application to authenticate the users. Most often, each credential is a unique user name and password. So, our hypothetical user has six passwords to manage.

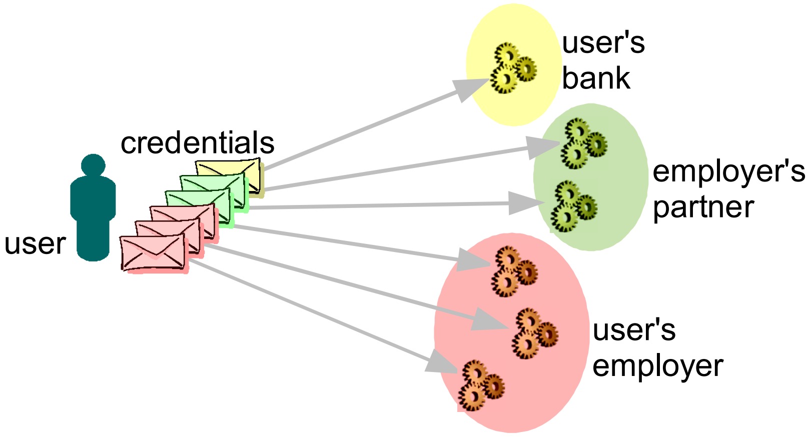


Illustration 1-- credential per applications: typical use of passwords

The user's burden increases with the number of passwords. Users spend more effort managing their passwords, and this increased burden translates into frustration and weaker security. Overwhelmed users tend to write down passwords; making them easier to steal. Users with many passwords are more likely to lose a password, thereby requiring password recoveries or resets, which are expensive and exploitable.

There is strong motivation on the user's and provider's behalf to reduce the number of passwords. Consider the following password consolidation schemes:

- Use the same user name and password for multiple applications and even across multiple enterprises. The benefit to the user is obvious: fewer passwords. However, this is devastating to security. A dishonest administrator can obtain the user's password and use it to access another applications in other enterprises. Reusing passwords is prohibited by all reasonable security policies.

- Securely store the passwords so they are available to only the appropriate user. The simplest example of these schemes is the browser remembering and automatically providing passwords. More sophisticated solutions involve specialized client applications that store and manage passwords. The user unlocks the password store with yet another password. The justification of these solutions is that the user is required to remember only one password, since the application manages all others.

- Securely store the passwords in an enterprise-wide solution. Many times these solutions are integrated with the password management facilities. This class of solution is widely known as *single sign-on,* or SSO.

Single Sign-On

There are many forms of single sign-on. Most store the application passwords in some sort of *vault*. The user opens his vault with his password and the single sign-on solution automatically provides the application passwords. In this manner the user might have a large number of application passwords, but must personally manage only one: the password that unlocks his vault.

Illustration 2 shows how single sign-on reduces the number of user passwords. In this example, the user has one password to unlock his employer's vault and another password to activates his partner's vault. The user must remember and protect only two passwords instead of six. This greatly increases the convenience to the user, lowers the cost of ownership, and strengthens the security.

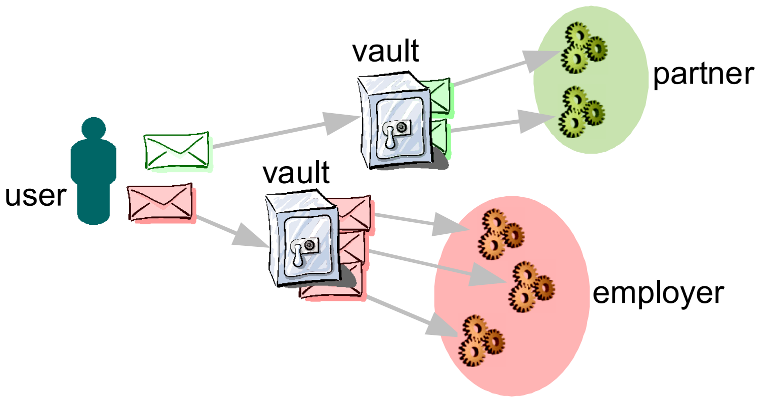


Illustration 2-- single sign-on: password caching

This simple application of single sign-on requires each enterprise maintains its own password. The user must maintain a different identity for each enterprise.

Another interesting variation on the single sign-on is the attempt to use a single vault across multiple enterprises. This requires that the enterprises somehow cooperate to trust and maintain the vault. The simplest form is shown in Illustration 3, where both enterprises rely on a common vault. The user has a single credential to unlock his *common identity store*, which contains the names and passwords used across all enterprises. This usually requires a *trusted third party* to maintain the vault. This third party is essentially an agent for the user and holds all of his passwords or identities.

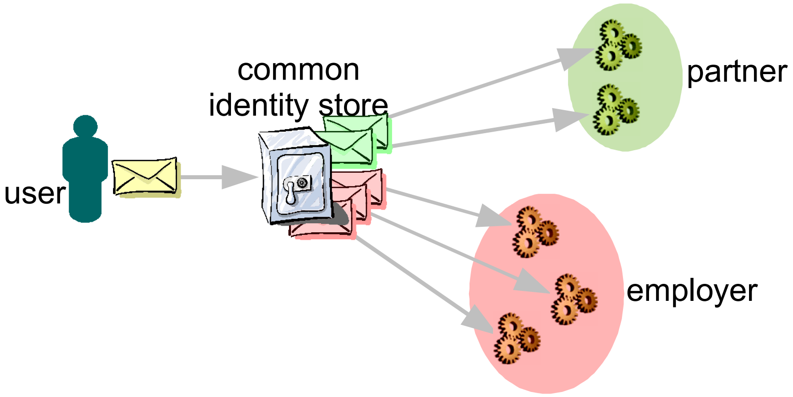


Illustration 3-- common identity store

There are a number of initiatives to support a common identity store across multiple enterprises. All are struggling with the fundamental issue of how can independent enterprises trust a common identity store? The utility of a common identity store appears to be limited to applications requiring modest security or among enterprises where a trusted third party naturally exists. It is unlikely that common identity stores can be widely deployed.

A recent variation is shown in Illustration 4, which suggests cross-enterprise single sign-on through a federated organization. In this scenario the user has a single identity, which he maintains with his employer. When the partner enterprise requires user authentication, it requests that the employer enterprise provide it. Technology to support federated identity management has recently been standardized, and a few pioneer systems have been deployed. The early attempts have shown that federated identity management is a difficult problem, and that broad deployment is far from a sure thing.

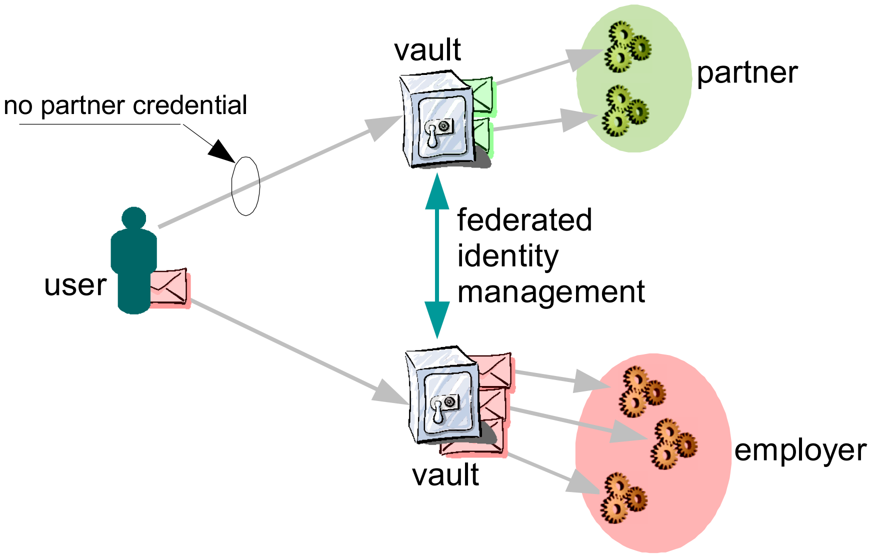


Illustration 4-- single sign-on through federation

Common Credential

Our desire is for the user to maintain a single identity and be able to use it across many applications and many enterprises. The concept of a common credential is shown in Illustration 5. The user has one credential for all applications in all enterprises. We've already discussed why passwords cannot be used among different enterprises as a common credential. Fortunately *certificates* based on public key cryptography don't have these limitations. This is because public key authentication demands two components: the user's certificate and the user's *secret key*. Although the certificate is widely known, the user and only the user knows his secret key. Since administrators don't have access to user's secret key, a dishonest administrator cannot masquerade as a user. The user's identity is safe as long as he protects his secret key. Details can be found in the Sevan White Paper, *User Authentication through Public Key Technology*.

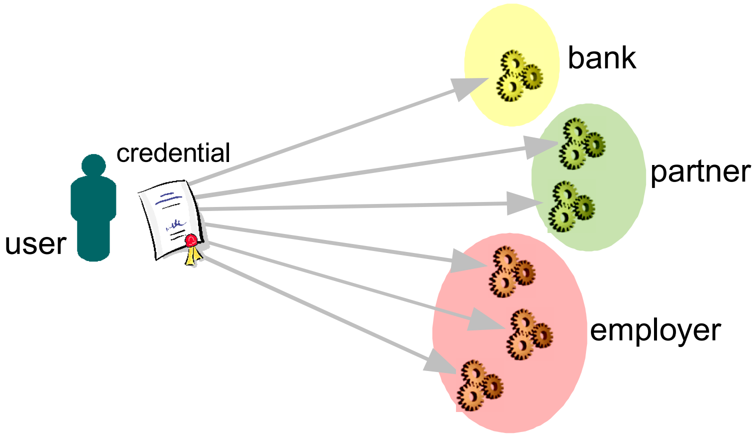
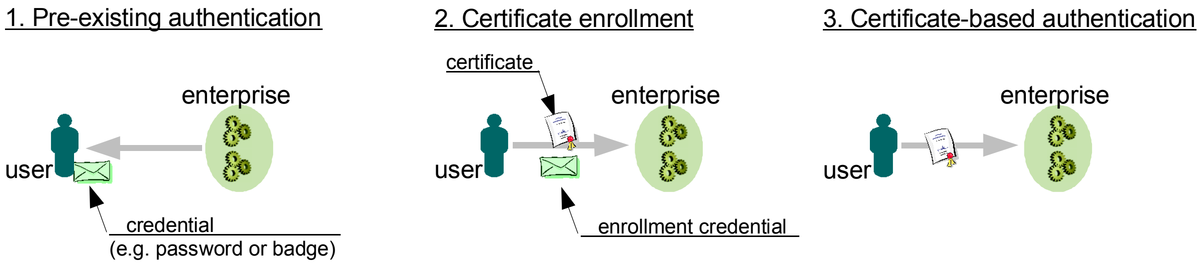


Illustration 5-- common credential

There is one additional requirement before a certificate can be used as a common credential: each enterprise must recognize the certificate and trust that it belongs to one of its authorized users. The X.509 model (sometimes referred to as the PKI) requires that each enterprise trusts the Certificate Authority that created the user's certificate. When the enterprises are truly independent, it is difficult to find a trusted third party who can serve as the certificate authority. Therefore, a common credential requires a mechanism for each enterprise to independently generate and maintain its trust in the user's certificate. The simplest and most general way of generating trust is through a certificate enrollment scheme, where the user presents his certificate (which is not previously known nor trusted by the enterprise) along with information that is trusted (e.g. a pre-existing user name and password). The enrollment scheme essentially transfers the trust from the trusted information to the certificate. Other enrollment schemes are possible. For example, the user appears in person, presents his certificate, along with other pre-existing trusted physical information (e.g. a thumb print, driver's license, or employee badge).

The enrollment process is shown in Illustration 6. Step 1 shows a pre-existing relationship between the user and the enterprise: the user has a credential that allows the enterprise to authenticate him. Step 2 shows the enrollment process: the user presents his credential along with his certificate. The enterprise simply transfers the user's authentication from the credential to the certificate. In step 3, the user is authenticated on the basis of his certificate – he no longer needs his enrollment credential.

Notice that the user can independently enroll his certificate with multiple enterprises. Usually the user has a unique credential for each enterprise, which reflects his unique trust relationship with each enterprise. After the user enrolls his certificate, he has a single credential (the certificate) for accessing all of the applications in all of the enterprises.

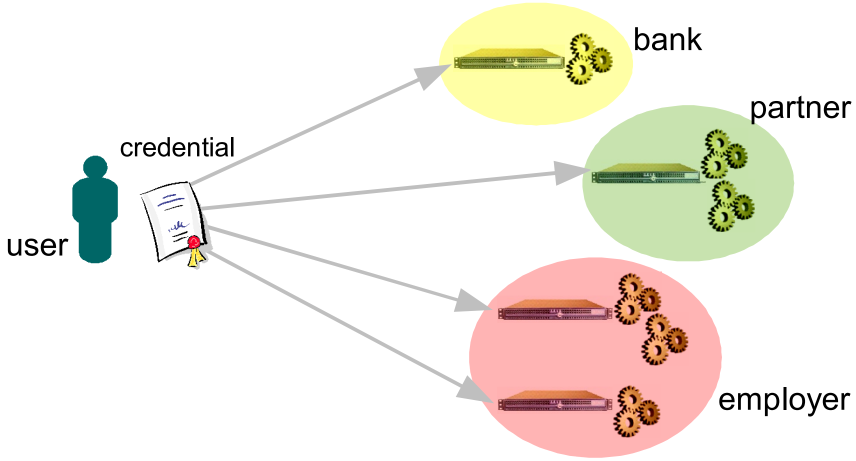


The use of a common credential might raise privacy concerns with some users. They might reason that a common credential provides a mechanism to track their activity throughout the internet. Fortunately the concepts presented here can be extended to multiple user identities through the use of multiple certificates. If the user is concerned about his privacy, each certificate can represent an independent user identity. A user can enroll one certificate with his employer and a different certificate with his bank. Since the certificates do not contain personally identifying information, the employer and bank cannot use the certificates to track the user's activities. By using multiple certificates, users can independently manage their identity and privacy. We expect that most users will opt for convenience of a common certificate for all applications, but it is up to the user to make the trade-off between convenience (common credential) and privacy (many credentials).

The Sevan WSA: Practical Approach to Common Credential

Illustration 5 suggests that each application accepts a common credential. It is unrealistic to assume that applications should or even can be modified. A more practical approach is to provide the common credential authentication through add-ons: either hardware or software that provide authentication services for the applications.

Illustration 7 shows how the Sevan WSA provides common credential authentication. The WSA authenticates the user on the basis of his certificate. If the user is authorized, the WSA allows him access to the application. The WSA has an integrated certificate enrollment, so new users can enroll their certificates if they have a valid enrollment name and password.



The WSA includes features that support migration of legacy applications to common credential. In situations where the existing application requires a user name and password, the WSA maps the certificate authentication to the user's legacy name and password. In situations where the application requires other forms of user identity, the WSA maps the certificate authentication to identity-specific cookies or URLs. When the WSA is optionally integrated with an LDAP directory server, certificates can be enrolled through existing names and passwords – no need to create and distribute new passwords.

In summary, the Sevan WSA provides user authentication through a common credential, the SSL certificate. The user can leverage his common credential to access all of his enterprises – a truly unified view of his identity. Since the user controls his certificate or certificates and how and when they are enrolled, he has complete control over his identity. The Sevan WSA can be transparently added to any web site, so the benefits of common credential are available without touching the servers and applications.

sevan
networks

Sevan Networks, Inc
1310 Hollenbeck Ave, Ste F
Sunnyvale, CA 94087
Tel: 408.830.1000
Fax: 408.830.1001
www.sevannetworks.com