Symmetric Key Services Markup Language Use Cases

Document Version 1.1 - February 28, 2007

The OASIS Symmetric Key Services Markup Language (**SKSML**) is the proposed language/protocol that defines how a client on a network will request and receive services for symmetric encryption cryptographic keys from a server. This document describes some of the ways in which SKSML may be used to solve specific business problems.

Clients may consist of computerized devices such as Personal Digital Assistants (PDA), telephones, laptop, desktop and server-class computers, applications such as office productivity, database, e-commerce, healthcare, financial or other applications, and/or devices such as routers, printers, disks, tape-drives, etc. Symmetric encryption cryptographic keys may consist of Triple Data Encryption Standard (3DES) or the Advanced Encryption Standard (AES). Please note that SKSML is not restricted to these examples; these are used, merely, as an illustration.

Representative use cases for the following sectors are highlighted: **Retail, Healthcare, Government, Finance, E-commerce, Corporate** and **Education**. Since SKSML is a generalized protocol, it can be applied to any industry and any application that needs symmetric encryption key-management services.

Why SKSML?

In order to understand why SKSML is better suited to solving the problems described in this document, it will help to know why existing paradigms and solutions do not solve them.

Traditionally, any application that has needed to encrypt data for security purposes, has had to perform all the functions related to encryption and key-management by itself. These functions involved many of the following activities:

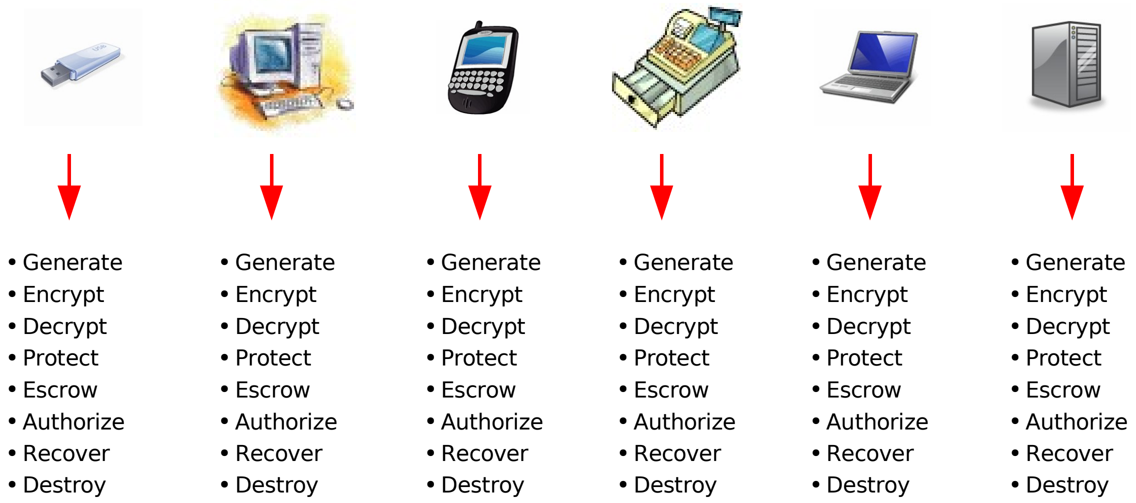
- Configuring one or more encryption policies in the application;
- Generating encryption keys;
- Using encryption keys for encryption and decryption;
- Protecting and managing access to the encryption keys;
- Rotating keys, when necessary;
- Destroying keys, when past their useful life;

Businesses accepted these responsibilities because they had little choice for a technical alternative. Besides, the number of applications required to perform these functions were few, so it seemed appropriate to limit the functions to just the applications that needed the encryption capability.

Currently, business are forced to contend with the encryption of sensitive data across many components of their distributed computing enterprise: Personal Digital Assistants (PDA), Laptops, Web servers, Application Servers, Databases, SANs, Backup devices, etc.

As a consequence, instead of a few applications performed encryption and key-management tasks, companies must contend with performing all these activities on every platform and application that must encrypt data (as depicted in the representative diagram below):

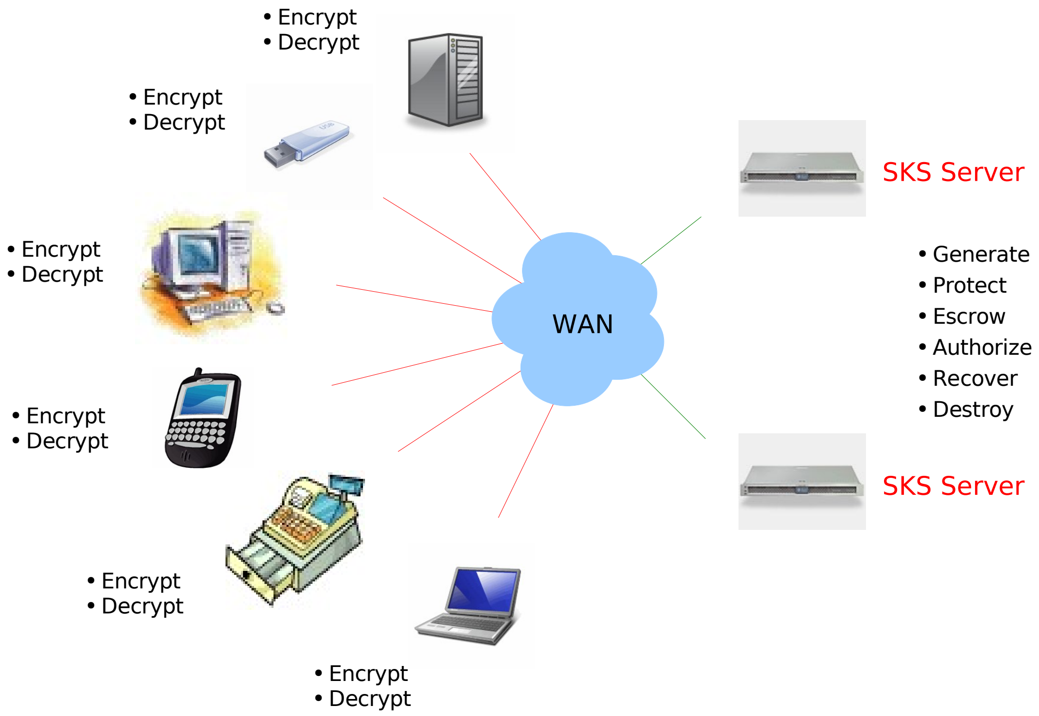
Additionally, because these key-management functions were embedded in applications that were spread out physically across the network, companies were forced to perform most key-management functions manually – i.e. while they had tools that performed the actual task, more often, companies had to dispatch Administrators to the physical machine to perform key-management functions.



This represents a significant operations challenge to any organization. It may also represent a serious security challenge, given the size and complexity of today's globally distributed businesses.

A **Symmetric Key Management System (SKMS)** addresses this problem by abstracting the functions related to encryption key-management, and placing them in a server on the network. Clients requiring key-management services for encryption keys use the SKSML to communicate, securely, with one or more **Symmetric Key Services (SKS)** server over any public network and avail services.

The benefits of centralization becomes readily apparent from the following diagram. As with Domain Name Service (DNS), an SKMS permits immense scaling of key-management functions, while permitting companies to perform all aspects of key-management, centrally.



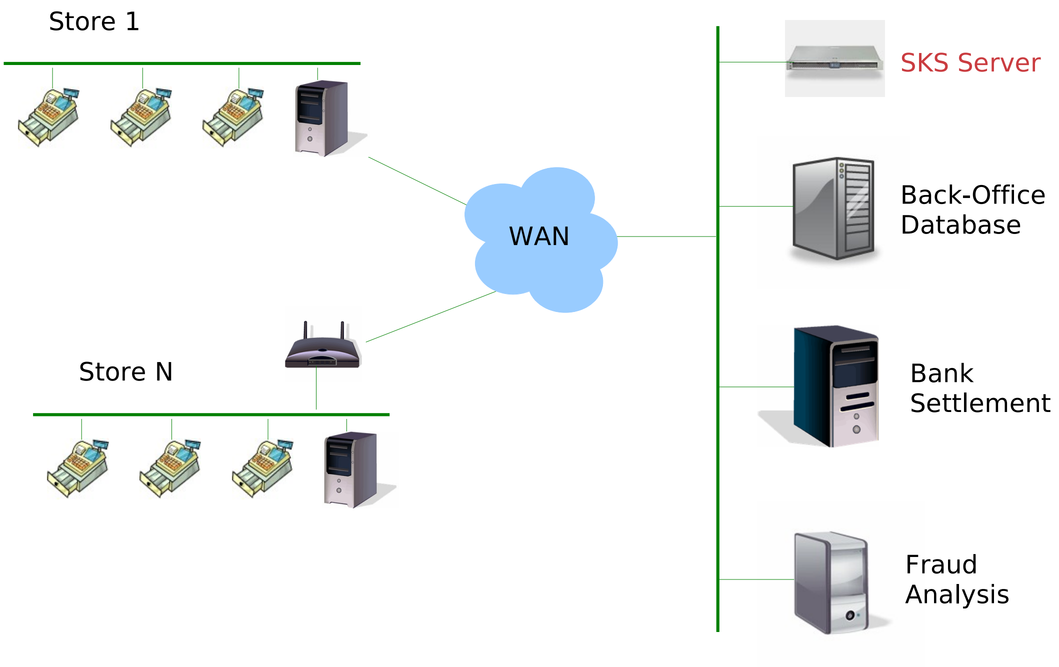
Why not PKI?

Public Key Infrastructure (PKI) was designed to manage asymmetric cryptographic keys and digital certificates, and has well defined protocols associated with the request/response service. However, asymmetric keys are limited in their capability to encrypt bulk-data, and consequently, technologists have combined symmetric and asymmetric keys for bulk-data encryption.

SKSML does depend on the Web Services Security (WSS) protocol layers (see SKSML Specification) for security, which in turn permits the use of asymmetric-key cryptography.

Retail

A retail company with hundreds of stores and thousands of **Point-of-Sale (POS)** terminals, accepts credit cards, whose **Credit Card Numbers (CCN)** must be encrypted to comply with the **Payment Card Industry Data Security Standard (PCI-DSS)**. The encrypted CCN and the decryption key must be available to Store employees and Customer Support Representatives (CSR), the back-office processing database and application environment, the bank-settlement application and the fraud analysis application.



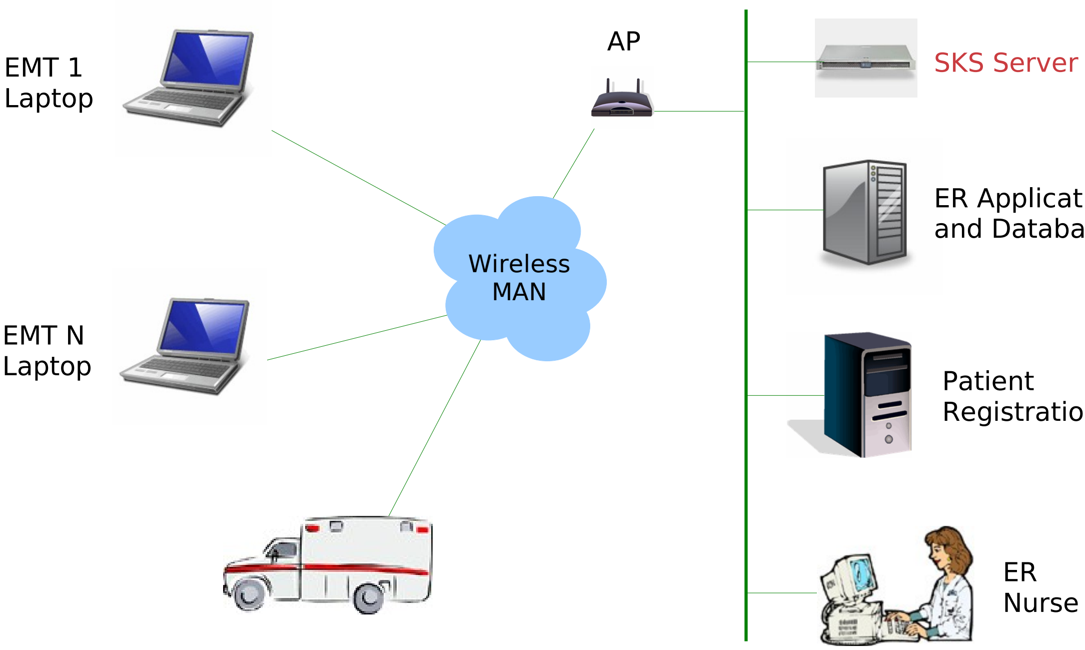
With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the retailer's network, the POS terminal application can securely request and receive a symmetric encryption cryptographic key from the SKS server using SKSML, encrypt the CCN and store the encrypted “ciphertext” with the symmetric key's **Global Key ID (GKID)** in the application's database. The symmetric key itself may now be discarded by the application or securely cached for future use on the POS terminal, based on the company's policy.

Any authorized application on the retailer's network – the back-office database, the bank-settlement application, any other POS terminal, the CSR application – may request the same symmetric key from the SKS server using the GKID, and upon receiving it decrypt the ciphertext to recover the “plaintext” CCN.

Because the symmetric key and its meta-data are securely generated and stored on the SKS server even before a client uses the key, the company is assured of recovering the decryption key by any authorized entity using the SKSML protocol at any time.

Healthcare

Emergency Medical Technicians (EMT) respond to medical emergencies and collect sensitive medical data on laptop computers. The **Health Insurance Portability and Accountability Act (HIPAA)** requires this information to be maintained confidentially, while doctors and nurses in the **Emergency Room (ER)** must be able to access this it without any hindrance.



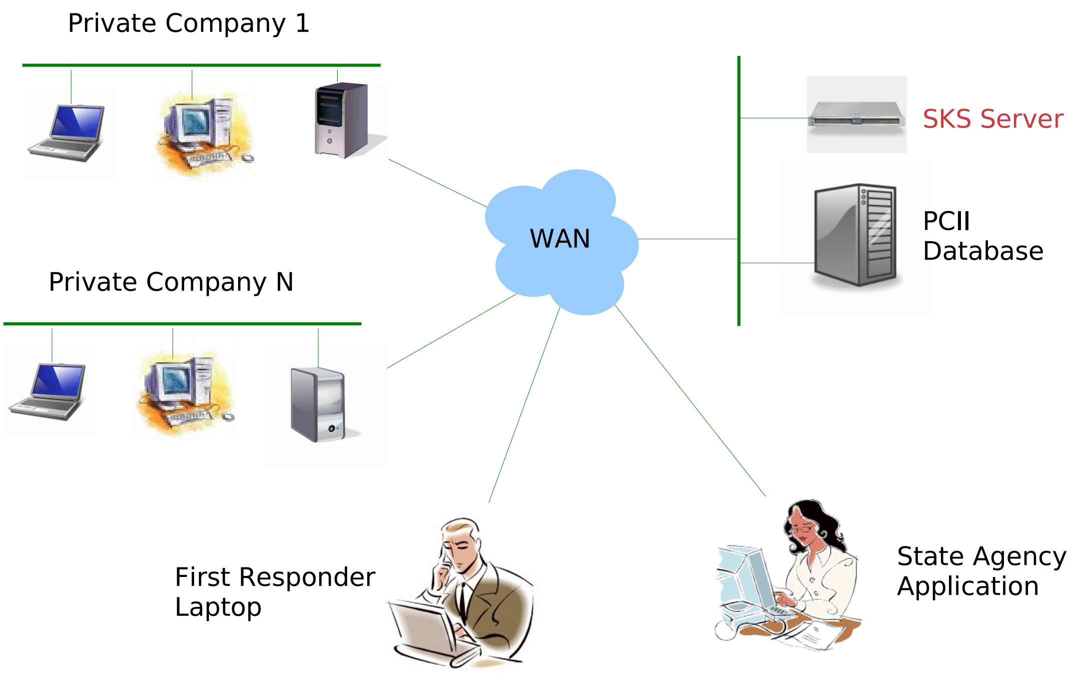
With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the healthcare provider's network, the laptop application can securely request, receive and cache multiple symmetric encryption cryptographic keys from the SKS server using SKSML, even before using the keys to encrypt any medical data. This ensures that the EMT can encrypt/decrypt data even if they do not have access to the network.

Additionally, all EMTs can be consolidated into a “**KeyGroup**” within the SKS server, and ER doctors and nurses can be authorized to retrieve any symmetric key created at the request of any member of the EMT KeyGroup. This ensures that when the EMT encrypts medical data and transmits it to the hospital, perhaps over wireless network even before getting the patient there, doctors and nurses can receive the ciphertext and decrypt the information as soon as it arrives. They would be able to access the same symmetric key used by the EMT in the field, and escrowed in the SKS server, through ACLs established in advance – even for keys that have not yet been generated.

Government

The US Department of Homeland Security (DHS) operates a program known as the **Protected Critical Infrastructure Information (PCII)** Program. This program is designed to encourage private industry to share its sensitive security-related business information with the US Federal government.

However, a problem arises in that the private industry would like to see its data maintained confidentially on DHS systems, while the DHS needs to share it with appropriate First Responders and other security personnel at the Federal, State and City level.

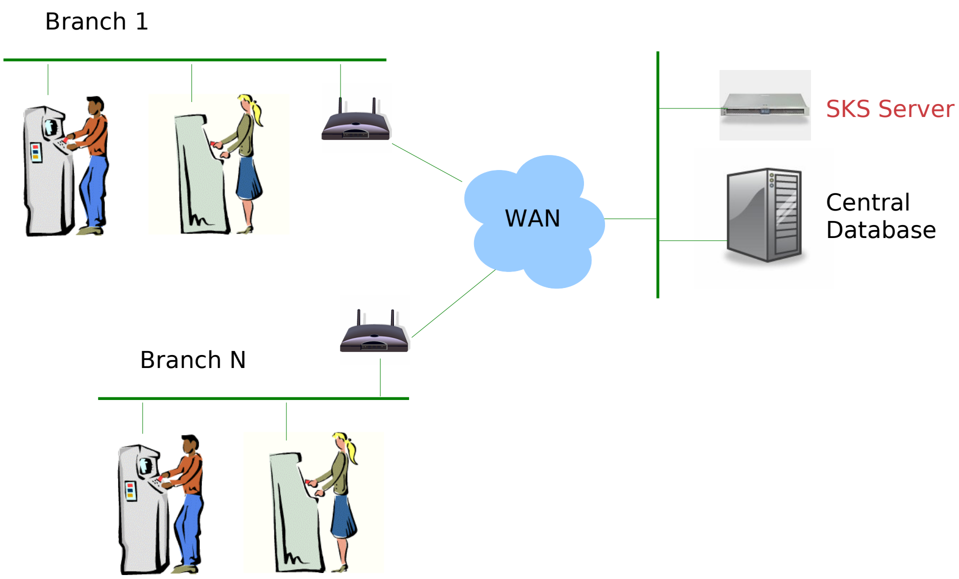


With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the DHS network, private companies working with the DHS on this program could request and receive symmetric encryption keys using SKSML and transport the encrypted ciphertext to DHS in any means suitable to them. The symmetric key itself may now be discarded by the private company as the DHS may establish a policy of using such keys for a single encryption only.

With the ciphertext in the DHS databases, and the symmetric key in the DHS SKMS, any authorized entity working with the DHS that has access to the ciphertext, may be granted access to retrieve the symmetric key. The authorized entity – be it a DHS Agent, or personnel at the Federal, State, County or City level - can use SKSML to request the same symmetric key to decrypt the necessary information to perform their tasks.

Finance

Banks, Credit Unions and other financial institutions with **Automated Teller Machines (ATM)** have a need for protecting the **Personal Identification Numbers (PIN)** of their customers during ATM transactions and in communications with the financial institution's network.



Traditional symmetric key management requires physical distribution mechanisms to deliver key packages (either in the clear or encrypted with a key-encryption key that is identical across their network) to the devices in the network. In a highly distributed global network, this creates a resource-intensive environment to create, package, deliver and supersede the key material in use. In many circumstances, symmetric keys may be used for a period longer than recommended, due to the expense of physically replacing the symmetric key-material.

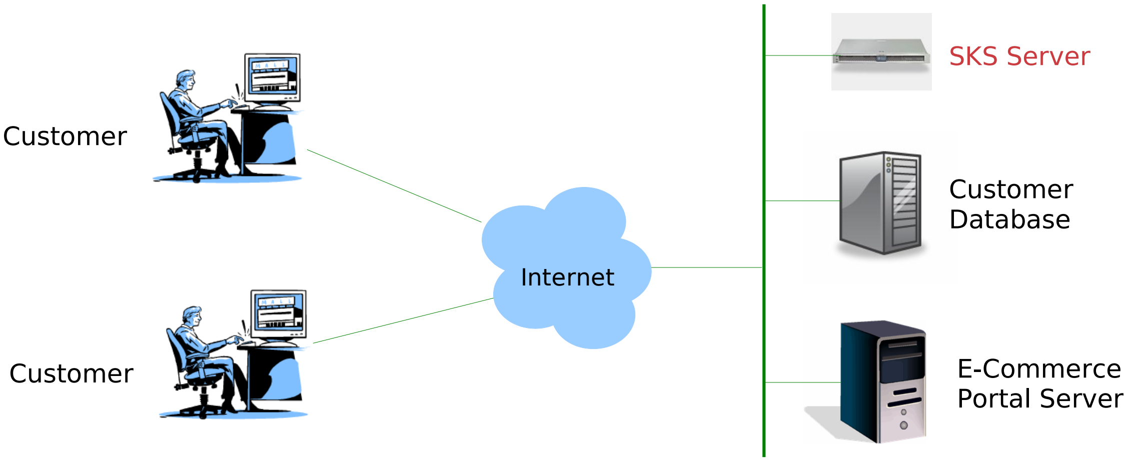
With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the financial institution's network, all ATMs can use SKSML to request and receive symmetric encryption keys securely, **over the network** without having to send personnel to the ATM. Additionally, because of the message-level security inherent in SKSML, they do not need a private or secured network to communicate these keys – they can be transported over the internet while maintaining the same level of security in the message as **Secure Socket Layer (SSL)**, **Transport Layer Security (TLS)** or **Internet Protocol Security (IPSec)**.

Secondly, the symmetric keys delivered to the ATMs over the network are encrypted with unique transport keys, thus ensuring that a compromise on any one ATM would not affect any other ATM on the network.

Financial institutions can even establish policies on the SKS server to **rotate symmetric encryption keys daily, or even hourly** on the ATM, if desired since the process of delivering the symmetric encryption key to the ATM is now completely automated.

E-Commerce

E-commerce portals that accept credit cards for transactions must not only comply with PCI-DSS, but must ensure that their 24-hour internet-facing portals do not reveal sensitive CCN to attackers that may have taken control of their web, portal or application server.

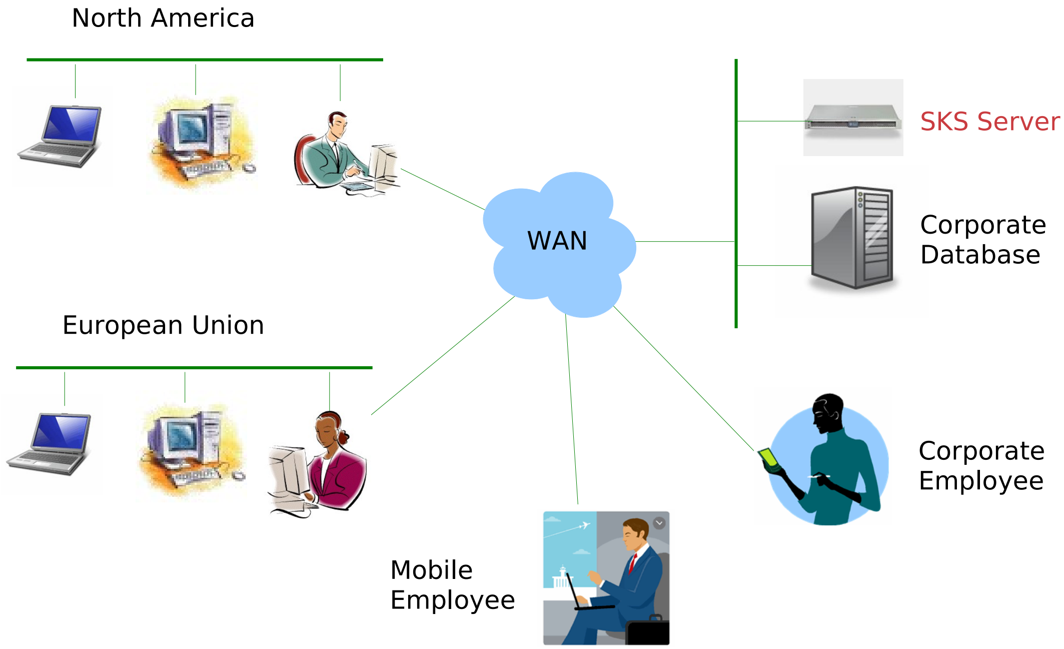


With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the e-commerce vendor's network, all servers and databases within their e-commerce applications can use SKSML to request and receive symmetric encryption keys to protect CCN and/or other sensitive data from unauthorized access.

Because all applications within the e-commerce infrastructure are part of the same SKMS, they do not need to transport the decryption key to other servers on the network, even as they transport the ciphertext to them. The server down the fulfillment-chain will have access to the same symmetric key from the SKS server using SKSML.

Corporate

Large corporations need to protect sensitive data on tens of thousands of laptops, Personal Digital Assistants (PDA), databases, and servers across their global infrastructure. **Chief Security Officers (CSO)** at these companies would like to define policies centrally, on how symmetric encryption keys can be used and managed across the enterprise.

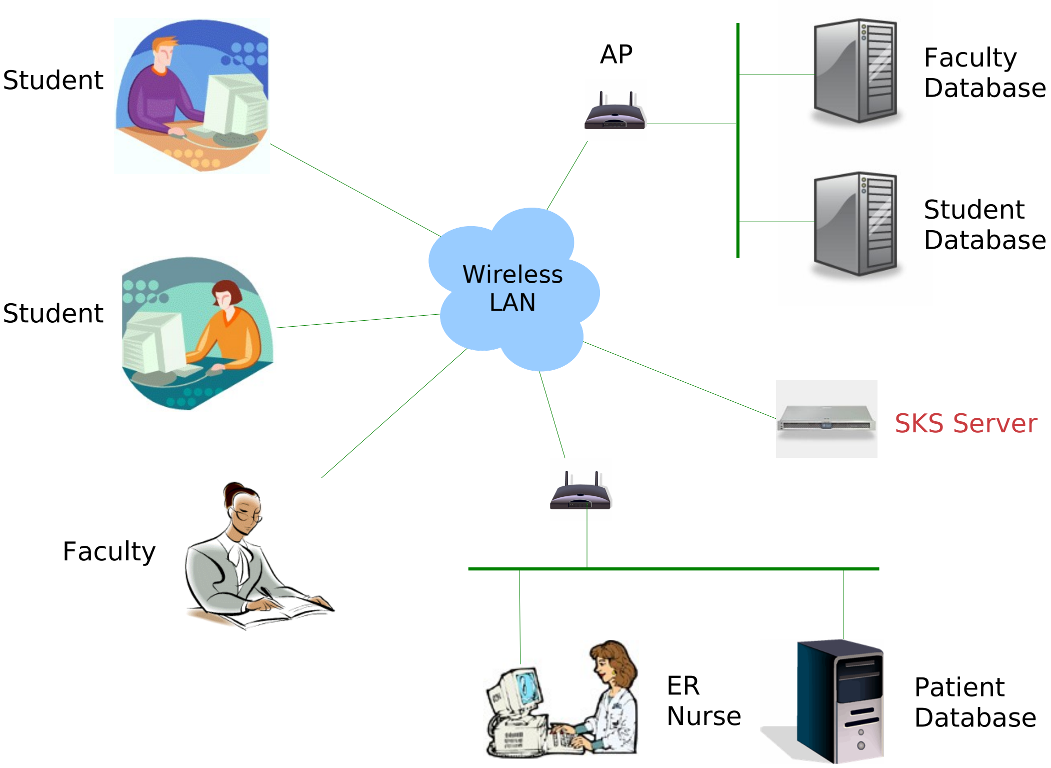


With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the corporation's network, all laptops, PDAs, databases, desktops and servers can use SKSML to request and receive symmetric encryption keys to protect sensitive data from unauthorized access.

Even if laptops and PDAs are stolen, and databases breached, the CSO will not have to worry about disclosing such breaches to the press because the SKSML protocol allows for secure caching of symmetric keys on the PDAs and laptops. The mobile devices can have the same level of security in protecting data and symmetric keys as desktops and servers on the corporate network.

Education

Schools and Universities have multiple challenging requirements. On the one hand, they receive and must protect personal identification information (PII) of students, their parents and guardians from financial aid applications, and if they have a medical school and hospital associated with the university, the medical information of their patients, while on the other hand, they need to create an open network environment that fosters creativity and collaboration to enable their students to learn from their teachers, peers and resources on the internet.



With a **Symmetric Key Management System (SKMS)** and a **Symmetric Key Services (SKS)** server on the university's network, they can address these diverse interests by having the applications dealing with sensitive data use SKSML to request and receive symmetric encryption keys to protect this information. The Student Database in the Registrar/Bursar's office, the Faculty Database in Human Resources, the Patient Database in the medical school (if any), etc. can all share the services of an SKS server on the network to protect sensitive information.

Even if the student network was directly connected to the university's “business” network, sensitive data would still be secure since the students would not have the appropriate credentials to make requests to the SKS server. The SKSML protocol carries digital signatures in its header; therefore the SKS server has strong assurance of an authentic request from a client, and every client has strong assurance that it is not being attacked by someone masquerading as an SKS server.

Summary

The Symmetric Key Services Markup Language (SKSML) is a generalized, XML-based protocol that can be used by applications to request symmetric encryption key-management services from a Symmetric Key Services (SKS) server within a Symmetric Key Management System (SKMS) on the network.

The SKSML is encapsulated within a Simple Object Access Protocol (SOAP) object, secured with the Web Services Security (WSS) protocol, through the use of Digital Signatures and Encryption. This enables SKSML messages to be secure enough that companies can deploy an SKMS without the use of Secure Socket Layer (SSL), Transport Layer Security (TLS) or Internet Protocol Security (IPSec). SKSML uses plain-old HTTP for sending and receiving messages within an SKMS.

SKSML can be implemented in any programming language or platform that supports SOAP enabled with WSS. An open-source Java implementation is available from www.strongkey.org, while the OASIS standardization process allows anyone to implement the protocol on a royalty-free basis.