The dramatic growth of and easy access to public networks has spawned a new electronic economy. Businesses are exploiting public networks in order to be the most efficient at producing and bringing their products and services to market; they are connecting online with their suppliers, customers, banks, and associated government agencies. Because much of this activity involves the exchange of sensitive, personal, or confidential information, security concerns are paramount.

Cryptography is the foundation of security for public networks. It enables the confidentiality and integrity of data, and authenticates the identities of individuals and computers on networks. As part of the IBM SecureWay family of offerings, IBM KeyWorks - Secure Cryptography and Certificate Services (SCCS) Toolkit, the IBM Key Recovery Service Provider (KRSP), and the IBM Key Recovery Server (KR Server) bring a new level of functionality, integration, and standardization to critical cryptographic and other security services.

This paper provides a high level introduction to these products and the IBM key recovery technology.

Introduction to IBM KeyWorks and the IBM Key Recovery Technology

Introduction to IBM KeyWorks and the IBM Key Recovery Technology

IBM KeyWorks

As is the case with the introduction of most new technology, the huge installed base of existing products and procedures cannot be ignored. That is, new technology must be made available in such a way that minimizes the impact to a company's environment and operations.

IBM developed KeyWorks (5648-A52) to provide for the easy adoption of new and existing cryptography, certificate, and key recovery technologies, while not disturbing existing cryptographic (and other security) functions and operations.

IBM KeyWorks consists of a framework and several service provider plug-in modules. The IBM Key Recovery Service Provider (5697-C86) plug-in module works with the KeyWorks framework to enable applications to create key recovery fields. The IBM Key Recovery Server (5697-C84) is an application that, upon authorized request, reconstructs the key from key recovery fields in order to recover encrypted information.

Frameworks

A framework consists of software that provides a layer of isolation between applications on "top" (of the framework) and specific implementations of services or mechanisms on the "bottom" (See Figure 1 below). Applications can access the desired service using a single, standard application programming interface (API). Service provider implementations that support the service provider interface (SPI) can be plugged-in to the framework without any changes to the application. That is, an application using one service provider implementation today can decide to use a different provider's implementation tomorrow without any changes. So the primary benefit of the isolation "layer" provided by the framework is that software on one "side" of the framework can be modified, updated, or replaced without impacting the operation of software on the other side.

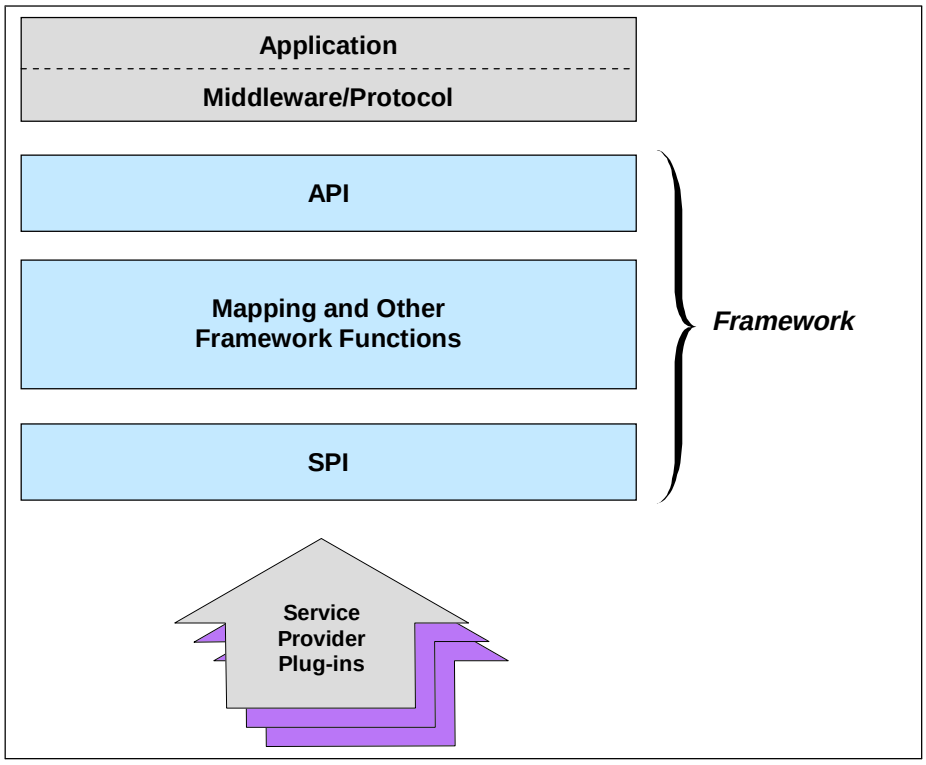

FIGURE 1: FRAMEWORKS

The framework functions include:

- identifying and registering the various service provider implementations
- maintaining and storing information about the current state of the connection between applications and plug-in modules
- verifying that the service provider implementations have not been tampered with.

IBM Key Works Description

As shown in Figure 2 below, IBM KeyWorks contains a framework and several service provider plug-in modules.

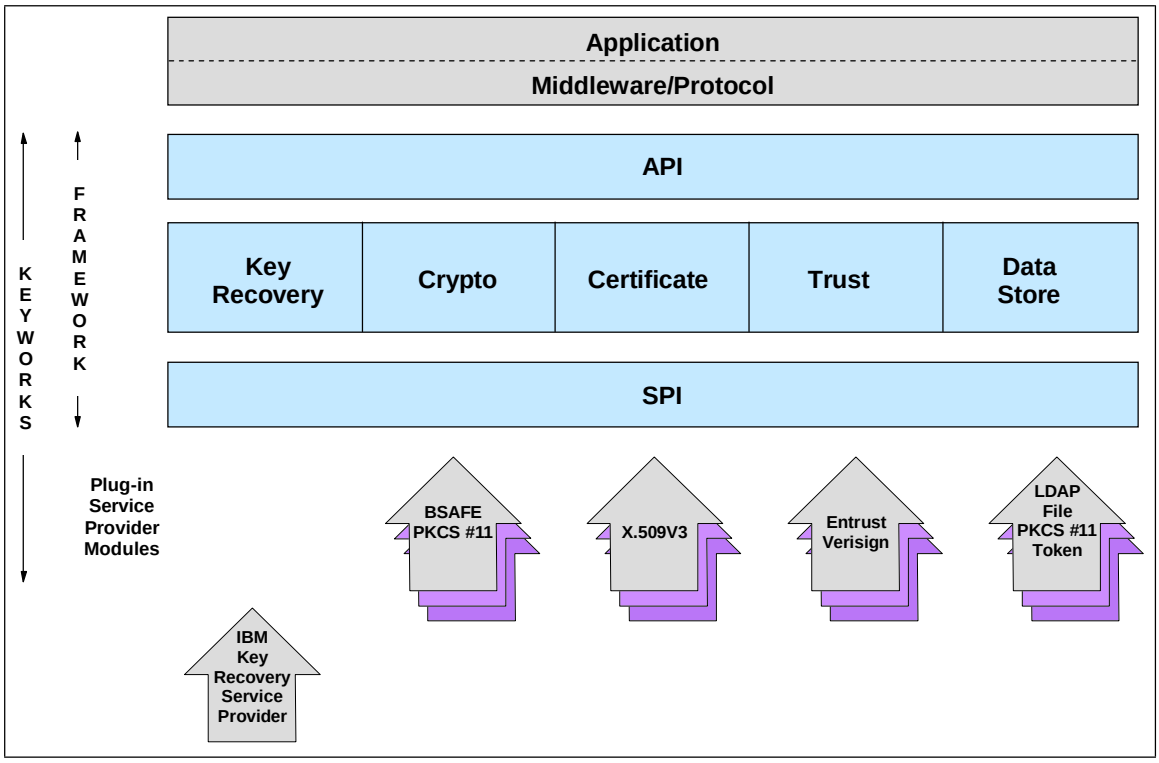

FIGURE 2: IBM KEYWORKS

The IBM KeyWorks framework is based upon Intel's Common Data Security Architecture (CDSA) that has been selected by The Open Group for fast path standardization. The API is the functionally rich Common Security Services Manager (CSSM) API; IBM has extended the API with the addition of key recovery functions. Through the use of the toolkit, an application can perform various security functions in a standard way. These functions include the following:

- encrypt and decrypt information
- verify a digital signature
- retrieve a certificate from a directory
- create key recovery fields
- decide if a certificate can be trusted

IBM KeyWorks also provides seamless support for hardware-based encryption.

The Key Works framework also provides critical "administrative" functions including:

- Providing protection against bypassing vital steps in a KeyWorks-supported policy enforcement
- Verifying that the service provider "plug-in" modules have not been tampered with
- Allowing use of the "plug-in" modules only through the framework
- Supporting country, enterprise, and user specific usage policies for key recovery.

The framework included in IBM KeyWorks is extendible; as additional functions are defined, the framework can be extended to include them. The framework can also accommodate multiple service provider implementations for each function.

IBM KeyWorks also includes service provider plug-in modules for the following functions: cryptography, certificate recognition, trust policy verification, and data (certificate and key) storage. These modules support open standards including PKCS#11 tokens, RSA cryptographic functions, DSA signature, X.509V3 certificates, the trust policies of Entrust and Verisign, and the Lightweight Directory Access Protocol (LDAP). The toolkit does not require the use of any specific certificate authority for the creation of certificates.

Cryptography and Key Recovery

Because of the meteoric rise in popularity of the Internet and its huge potential for communication and commerce, the topic of security has percolated to the top or near the top of the list of major concerns of businesses, individuals, and governments. The science of cryptography, once the sole province of the mathematically inclined, is now common conversation at parties and other gatherings. Cryptography usually involves numeric representations of information and mathematical manipulation of the numbers with the objective of hiding the original information. The basic cryptographic functions are encryption and decryption--using a mathematical algorithm and a secret value known as a **key** in the "the art of mangling information into apparent unintelligibility in a manner allowing *a secret method of unmangling* [italics added]."[1] This provides the ability to communicate between parties or store information in such a way that prevents other parties from accessing and understanding it.

In addition to reasons for hiding or encrypting information, there are legitimate reasons for recovering encrypted information such as :
- an individual has encrypted important information and has lost or forgotten the key
- a business needs access to employee-encrypted non-personal information and the employee is not available
- law enforcement procures a court order giving them the right to access information (for example, a search warrant).

The cryptographic **key** is the critical item required to recover encrypted information. Therefore, the availability or accessibility of the key is a pivotal issue. A number of key recovery technologies exist to make the key available to recover encrypted information. There are two basic types of key recovery techniques:

- those involving some form of escrow of the key or key parts with a trusted party and
- those non-escrow techniques that involve creating key recovery fields --that are mathematically related to but not actually the key or parts of the key--and associating these fields with the message. Later the key recovery fields can be used to recover the key. Such techniques are called encapsulation.

---

[1]Kaufman, Perlman, Spencer, *Network Security - PRIVATE Communication in a PUBLIC World*, Prentice-Hall, 1995.

The IBM key recovery technology, as implemented in the IBM Key Recovery Service Provider module plug-in to the KeyWorks framework, is based upon an encapsulation technique rather than an escrow technique. The following draws a comparison between the two.

We can draw an analogy to one's house key. We lock our houses to safeguard our possessions inside. We might choose to give a copy of the house key to one or more trusted relatives or friends in case we lose the key. In doing so, we have assumed a risk that our friend(s) or relative(s) are honest and not thieves who would steal our valuable possessions. This is the concept of **key escrow**.

In cryptography, key escrow can go further: parts of the numeric key could be escrowed with different parties. This would be analogous to the situation in which a house was locked with several locks; the key to each lock could be given to a different person. All would have to get together to unlock the locks and open the house. This practice spreads the risk-- all of the parties would have to conspire to get criminal access to the house.

While some countries appear to have accepted key escrow, US businesses have not been in favor of mandating the escrow of encryption keys. To address the shortcomings of key escrow, IBM has introduced its key recovery technology based upon an encapsulation technique.

With IBM key recovery, no party actually holds either the cryptographic key or parts of the key. Information from which the cryptographic key can be reconstructed is associated with an encrypted message or file for potential later use. Consider the scenario of a lock box containing the actual house key. Suppose this lock box had a combination lock on it, and each number in the combination is given to a different party. All of the parties would have to get together and then determine the order of the digits to open the lock box and obtain the key. None of the parties have a copy of the actual key. This substantially reduces the risk; all parties would have to be part of a criminal conspiracy to gain unlawful entrance to the house. Furthermore, the process could be made more difficult if the owner of the house did not specify the order of the digits in the combination; then further trial and error would be necessary to get access to the key and ultimately the house.

A variation of this scenario would be where all but one of the digits to the combination were given to different trusted parties. The parties would then get together to provide the combination digits they had. The last digit would have to be discovered by brute force guessing and trial and error.

And in any version of this scenario, the locked key **stays** with the house!

The IBM key recovery technology has a number of advantages over key escrow:

- no one has a copy of the key or any part(s) of the key
- there is no single point of vulnerability or compromise
- no communication with a third party (escrow agent) is required for each session/key initiation, leading to excellent scaleability
- there is no loss of control over key management to a third party

- there is no need for a key storage infrastructure

On October 1, 1996, the US government announced the relaxation of export restrictions on products that implement or use strong encryption (that is, encryption with a 56 bit key). This liberalization was predicated upon the computer industry developing key recovery technology that would balance the need for confidentiality with the legitimate security needs of the government.

IBM spearheaded the formation of an industry-led Key Recovery Alliance that currently has over 60 members worldwide. Providing a focal point for efforts involving commercially-acceptable key recovery solutions, its objective is to promote the pervasive, worldwide use of key recovery through the identification of inhibitors and the definition of guidelines and specifications as needed.

The IBM Key Recovery Service Provider

Along with the IBM Key Recovery Service Provider (KRSP), the KeyWorks framework (See Figure 2 above) enables applications that use cryptography to set up a key recovery environment that is not dependent upon any one provider's implementation of key recovery. The key recovery process is an add-on to existing encryption schemes and would be optionally invoked (through the framework) by cryptographic services in support of an application. The framework effectively isolates the application or middleware from the idiosyncrasies of a specific key recovery implementation. The framework enables the key recovery service to be used with any cryptographic service provider that supports the SPI. The framework also protects against the bypassing of key recovery (if that is the organization's or jurisdiction's policy) and sets up a complete key recovery environment before any encrypted communication takes place. This framework may be used to support interoperability between key recovery-enabled cryptographic applications and key recovery-unaware applications.

Key Recovery Information Flow

The IBM KeyWorks framework and the Key Recovery Service Provider are meant to be invoked by a protocol such as SSL, SMIME, IPSEC, and others, that have been modified to use the framework to invoke the security services. We will use the SSL environment as an example to describe the flow of information both with and without the use of key recovery. We will also use IBM KeyWorks and key recovery technology in these scenarios.

Figure 3 (below) shows the protocol flow without key recovery:

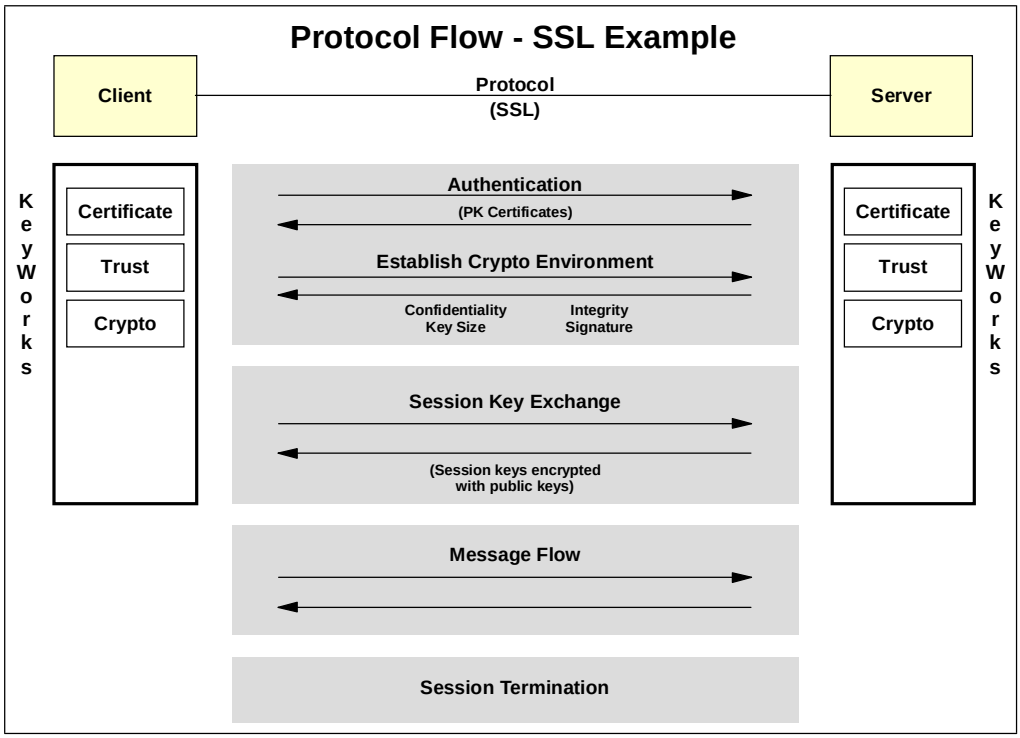

FIGURE 3: PROTOCOL FLOW SSL EXAMPLE

The Web Browser client and the server first authenticate themselves to each other, that is, they each prove their identities to each other. In our Internet example, the most common credentials for authentication are public key certificates, containing information about the certificate holder and the digital signature of a trusted certificate authority. Public key certificates are exchanged to authenticate each party in the communication.

Next, the cryptographic environment is established through a negotiation between client and server as to which mechanisms each can support and what is the stated priority sequence for their use. Items such as key lengths, encryption algorithms, and hash algorithms are determined.

The session key is then generated (there are a number of ways to accomplish this) which the client and server then exchange (not in the clear, of course). They verify that they each have the key and can encrypt and decrypt information.[2] Once this is done, the encrypted communication takes place, until it is complete and the session is terminated.

Figure 4 (below) shows how that flow is altered by the inclusion of IBM key recovery technology:

---

[2]The session key environment in this example is a symmetric key environment -- the same key is used for encryption and decryption. Key recovery is possible for various types of key environments.

.

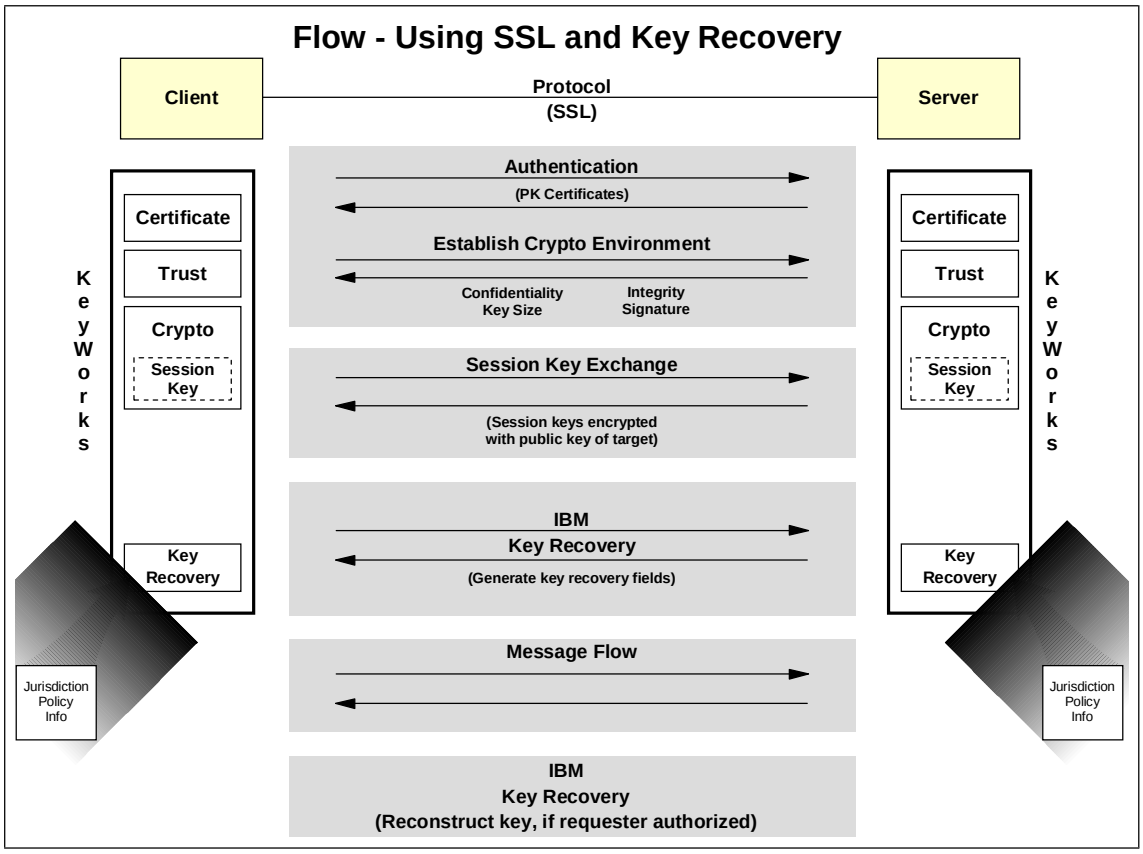

FIGURE 4: SSL PROTOCOL EXAMPLE--WITH KEY RECOVERY

The protocol flow will now be:

- The session key is made available to the KRSP.
- The KRSP generates the key recovery fields for the session key. This information is NOT the key or any portion of the key, but is information that can be used to recover the key. The KRSP has access to location-unique jurisdiction policy information that controls and modifies some of the steps in the generation of the key recovery fields. Only once the key recovery fields are generated, and both the client and server sides have access to them, can the encrypted message flow begin. This creates a complete environment so that once the encrypted message flow begins, all the necessary information is accessible in case key recovery needs to be performed.[3]
- Then the encrypted messages can flow back and forth between the two parties.
- If required or desired, and authorized, the encrypted messages could be decrypted by recovering the session key through the use of the IBM Key Recovery Server.

[3] This flow assumes that the framework set up functions (such as identification and registration of the key recovery agents) have already been done.

The above scenario described the flow using the IBM KeyWorks framework and IBM KRSP. The flow would be the same for any other key recovery technology that is written to support the SPI of the framework.

Key Recovery Operation

IBM key recovery differs from key escrow in that no key or parts of the key are actually held by any party. Instead, data fields that will enable the recovery of the key are created and associated with the data message. There are still outside parties involved in the key recovery process; they are called key recovery agents and are analogous to the neighbors who each hold one digit of the combination of the lock box containing the key, as described above.

Figure 5 below depicts the general environment in which encrypted messages are being transmitted between Alice and Bob, each in a different country or jurisdiction. Also, in each jurisdiction are key recovery agents. Alice and Bob each have the freedom to choose the number of and the specific agents that they wish to use. In our example, we assume that they have each chosen two agents.[4] Each jurisdiction also has law enforcement agents that may have a legal need (and legal instrument of authorization -- like a "search warrant") to know the contents of the messages between Alice and Bob.

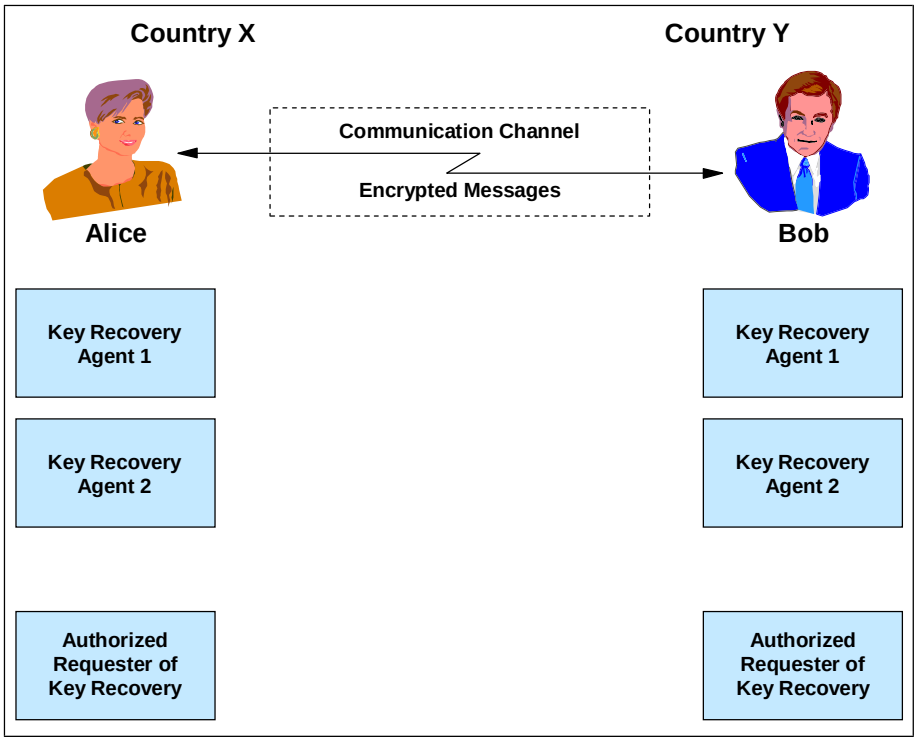

FIGURE 5: GENERAL ENVIRONMENT FOR KEY RECOVERY

[4] Note that the number of key recovery agents on each end of the communication does not have to be equal. That is, Alice could have chosen three agents while Bob chose two.

As described above, the KRSP is invoked from the KeyWorks framework to create the key recovery fields. There are two major pieces of the key recovery fields: block 1 contains information that is unrelated to the session key of the transmitted message and encrypted with the public keys of the selected key recovery agents; block 2 contains information that is related to the session key of the transmission. Figure 6 below is a simplified view of the format of key recovery fields.

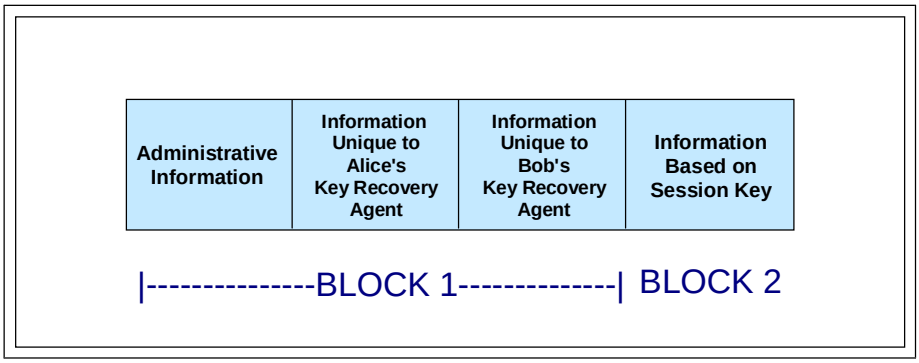

FIGURE 6: KEY RECOVERY FIELDS

Figure 7 below presents a simplified picture of key recovery operation. Under proper authorization, the authorized party requests that each key recovery agent decrypt its section of the key recovery fields that are associated with the transmission. Then those pieces of information are used in the process that derives the session key.[5]

There are many benefits to the IBM key recovery technology, especially when compared with key escrow:

- The technology is inherently scalable for global use in network computing environments
- The technology works with all encryption environments and key lengths
- No communication with a third party is required during preparation of the key recovery information, leading to performance optimization
- There is no single point of attack
- Key recovery in multiple jurisdictions can be handled.
- The key recovery operation is transparent to end users.
- No keys or key parts are exposed outside of the cryptographic facility

[5]As part of the technology, there is an option to hold back a portion of the key from the entire process. That means that in addition to the process shown in Figure 7, brute force or trial and error needs to be utilized to determine the held back portion of the key. This makes recovery more difficult.

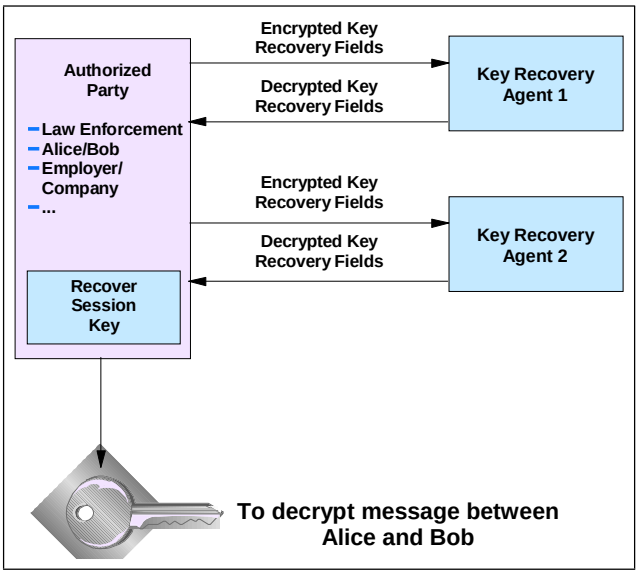

FIGURE 7: KEY RECOVERY

Summary

Industry worldwide is committed to strengthening the global enterprise and continues to believe that key recovery technology is the solution that will make this a reality. IBM is committed to key recovery and has implemented it in its products -- IBM Key Recovery Service Provider (KRSP), and IBM Key Recovery Server (KRS). Because there are so many existing cryptographic environments that will want to incorporate key recovery functions, IBM is providing IBM KeyWorks, Secure Cryptography and Certificate Services Toolkit, that provides a framework to isolate those cryptographic environments from the differences in implementation of the various key recovery schemes. Frameworks are the integration enablers of the open, heterogeneous, multivendor world in which we live today. IBM KeyWorks and its open standards-based framework provide the right vehicle to protect your investment in your current security products, operations and processes, while allowing them to take advantage of new technologies such as key recovery.

IBM KeyWorks, IBM Key Recovery Service Provider, and IBM Key Recovery Server products are part of the broad IBM SecureWay portfolio of security hardware and software products, solutions, services, consulting, and research activities. The SecureWay brand provides end-to-end security solutions such as cryptographic facilities, single-sign-on, distributed security administration, access control, firewalls, secure web servers and browsers, secure electronic commerce, smart cards, anti-virus, and overall Internet security.

For additional information visit the IBM SecureWay home page at http://www.ibm.com/security.