



**WHITE PAPER**

---

**TECHNOLOGY AND APPLICATIONS**

**Communication Security**  
**- available techniques**

## TABLE OF CONTENTS

1	Introduction .....	3
2	Communication Security Concepts .....	3
3	Avoiding Successful Eavesdropping .....	5
4	Preventing Malicious Modifications .....	8
5	Discovering any Forgery .....	9
6	Administrating and Distributing the Public Keys .....	10
7	Security over IP-based Networks .....	11
8	Conclusion .....	12
9	About Axis Communications AB .....	13

# 1 Introduction

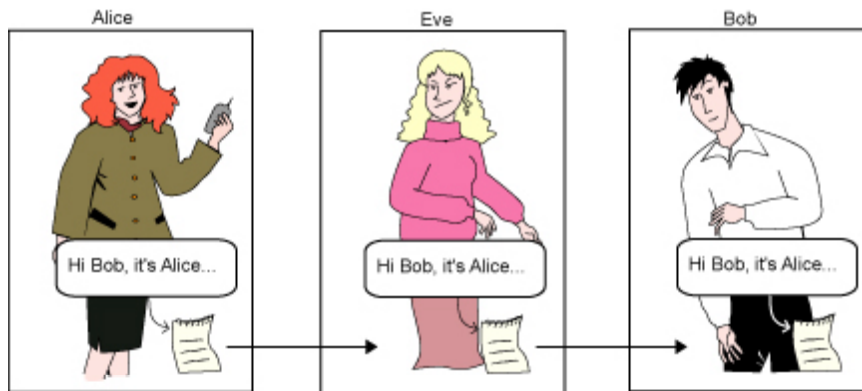
Today, more than ever, computer networks are utilized for sharing services and resources. Information travelling across a shared IP-based network, such as the Internet, could be exposed to many devious acts such as eavesdropping, forgery and manipulation. Fortunately, there are several mechanisms that can protect any information that needs to be sent over a network. This paper introduces security threats to today's IP-based networks and explains available security mechanisms to effectively prevent such threats from happening.

No one wants his or her confidential or classified information revealed. Confidential information that you do not want to share with others is the easiest to protect, but ever so often there is a need to share this type of information. Whenever this happens, you need to be able to send the information in a secure manner to your trusted receiver. This issue is particularly important when network communication is involved, since network communication has become the cornerstone for organizational effectiveness and today's digital communication often includes sensitive information such as control and corporate financial data. Consequently, we need security mechanisms whenever sensitive information is to be exchanged over the network.

## 2 Communication Security Concepts

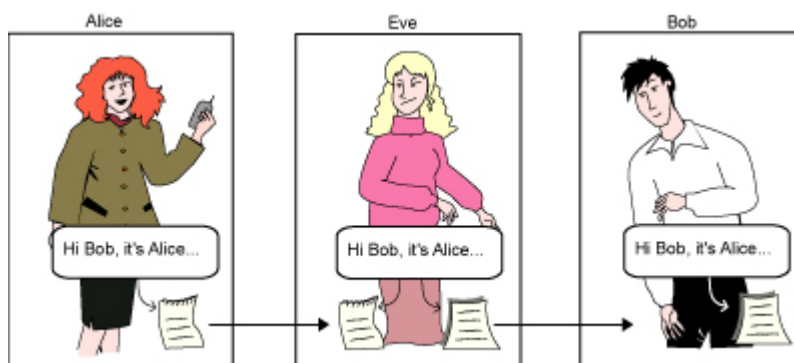
Fundamental to IP-based networks is the function of dividing data into packets and the independent routing of packets through a large network with no central control. Although each packet is marked with its sender and receiver, the packets are not invisible to other devices on the network. An intermediate network device can easily intercept and examine any passing packet. This property of IP-based networks creates several potential security problems that need to be dealt with.

First, we'll introduce three characters, Alice, Bob and Eve, who will help us to illustrate different concepts in communication security. Alice and Bob are two colleagues who want to exchange sensitive information over a shared network. However, Eve is connected to the same network and her intentions are less than good as she intercepts any message sent between Alice and Bob. This is of great concern since it allows Eve to eavesdrop on information sent between Alice and Bob (Figure 1). How can Alice and Bob prevent unauthorized users, such as Eve, from reading their message?



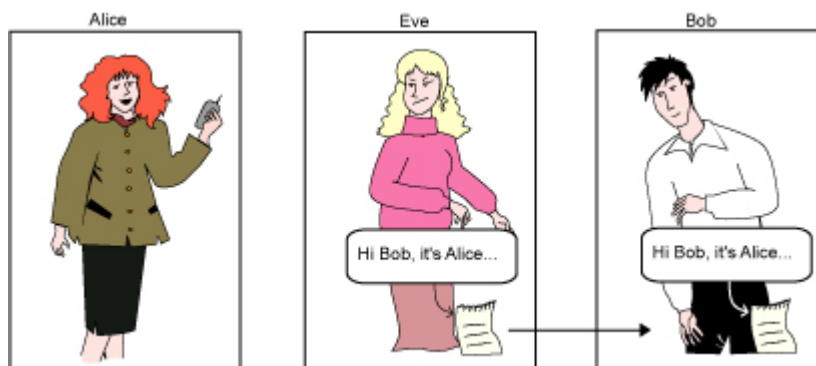
**Figure 1:** Eve can intercept and eavesdrop on communication between Alice and Bob.

Furthermore, suppose Bob receives a message from Alice. How can Bob verify that the message he received is really the exact message that Alice sent? Eve could have intercepted and modified the message (Figure 2).



**Figure 2:** Eve can intercept and modify a message in transit.

Additionally, suppose Bob receives a message that appears to be from Alice. How can Bob be confident that the message is actually from the source it appears to be from? Eve could have forged the message to deceive Bob (Figure 3).



**Figure 3:** Eve can forge a message to impersonate Alice and deceive Bob.

The events described might initially cause alarm. However, there are effective mechanisms to prevent each and every one of these incidents from happening. Effective security involves the combination of the mechanisms described below.

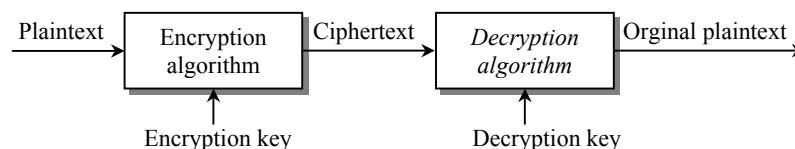
**Confidentiality** is the concealment of information from all but authorized parties. Suppose Alice has a message for Bob and she wants to keep the message secret. Only the sender and the intended receiver should be able to understand the content of the transmitted message. Because eavesdroppers like Eve may intercept the message, this essentially requires the message to be disguised somehow, so that an unauthorized party cannot understand an intercepted message.

**Data Integrity** is the assurance that unauthorized parties has not modified a message. When Alice and Bob are communicating, they want to ensure that the content of their communication is not altered, either maliciously by Eve, or by accident in transmission (as is the case in Figure 2).

**Authentication** is the assurance that the parties involved in a transaction are who they say they are. For example, Alice and Bob need to confirm the identity of each other when communicating. This will prevent Eve from deceiving Bob by impersonating Alice (as is the case in Figure 3).

### 3 Avoiding Successful Eavesdropping

Confidentiality is primarily accomplished with *cryptography*, which involves the design and implementation of systems that maintain secrecy. It is important to distinguish different kinds of messages (data) when cryptography is discussed. The messages that are to be transformed into a secret form are called *plaintexts* and, once transformed, the messages are called *ciphertexts*. A cryptosystem transforms plaintext into ciphertext, or vice versa, through the use of a set of *crypto algorithms*. Special pieces of variable data called *keys* determine how the crypto algorithms will transform the plaintext and ciphertext. The keys are chosen from a set of keys (keyspace). The process of transforming plaintext into ciphertext is called encryption, and the reverse process is called decryption (Figure 4).



**Figure 4:** The encryption and decryption process.

It is preferable that the security of a cryptosystem resides in the secrecy of the keys rather than with the supposed secrecy of the crypto algorithm. This means that it should be virtually impossible to decrypt a ciphertext to plaintext if the decryption key is unknown, even if the full details of the encryption and decryption algorithms are known.

Message confidentiality is primarily accomplished with *symmetric algorithms* (secret-key algorithms). A symmetric algorithm utilizes the same *secret key* for encryption and decryption. The historical Caesar’s cipher can serve to illustrate the use of a symmetrical algorithm. The method is simple: shift a plaintext alphabet three letters over to transform it into a ciphertext alphabet. (Figure 5).

Plaintext	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	...
Ciphertext	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	...

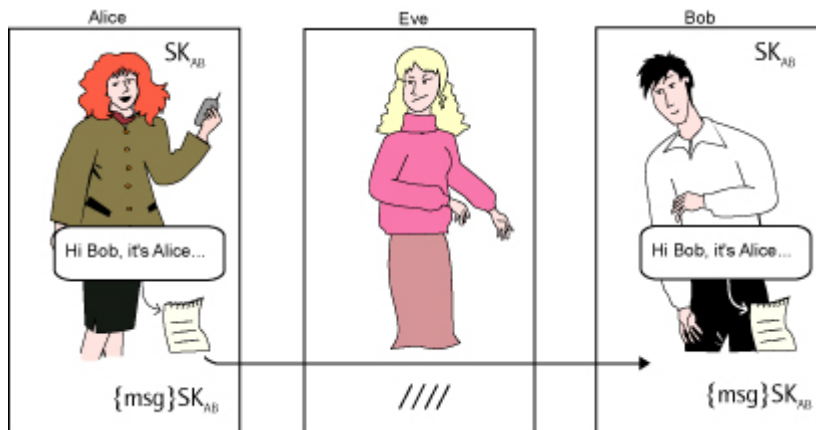
**Figure 5:** The two alphabets in Caesar’s cipher using three as the key.

The key in this particular case is three and the algorithm simply changes the plaintext letter with the corresponding ciphertext letter based on the key. Instead of exchanging the full alphabets, Alice and Bob need only exchange the cryptographic key, three. In our example, the plaintext “ALICE” becomes the ciphertext “DOLFH”. Decrypting the ciphertext is the reverse process; the ciphertext letter is changed to the corresponding plaintext letter based on the same key.

An obvious method for trying to break a cryptosystem that utilizes a public knowledge algorithm is to try all possible keys in the keyspace until the right one is found. This method is commonly referred to as *brute force*. The time required for this method depends on the size of the keyspace and the amount of computer processing power available. The Caesar’s cipher has 25 possible keys, and a person like Eve could easily find the key that was used by simply trying out all the possibilities. Fortunately, modern symmetric algorithms have replaced simple substitution methods, like Caesar’s cipher, with far more sophisticated mathematical methods, and these keyspaces make the available symmetrical algorithms virtually unbreakable with the brute force method.

In the following example, Alice and Bob use a symmetrical algorithm to provide confidentiality to a message (Figure 6).

1. First, the secret key needs to be securely exchanged between Alice and Bob. Let’s call the secret key, known only to the two of them, “SK<sub>AB</sub>”.
2. When Alice wants to send a message to Bob, she encrypts the message, “msg,” using the symmetrical encryption algorithm and the shared secret key “SK<sub>AB</sub>” producing the encrypted message, “{msg}SK<sub>AB</sub>”.
3. Alice can send the encrypted message to Bob, using the shared network, with the assurance that Eve will not be able to decrypt and interpret the message since she doesn’t possess the secret key. Eve can only intercept an indecipherable message, “/////”.
4. When Bob receives the message, he decrypts it by using the secret key to reveal Alice’s message.

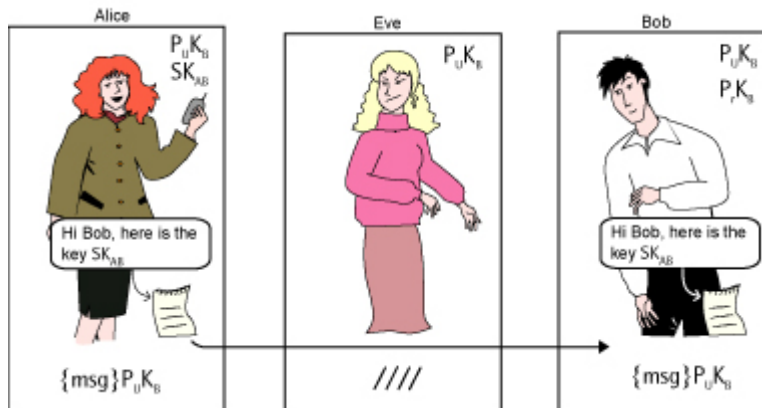


**Figure 6:** Alice and Bob use a symmetrical algorithm to achieve confidentiality.

Utilizing symmetrical algorithms provides the confidentiality mechanism for the message sent between Alice and Bob. However, in this case, confidentiality is completely dependent on the secret key “ $SK_{AB}$ ”. Confidentiality would be jeopardized if successful transfer of the secret key in the first step cannot be assured. Fortunately, there exist mechanisms that allow us to accomplish secure key exchanges, namely *asymmetrical algorithms* (public-key algorithms).

An asymmetric algorithm utilizes different keys for encryption and decryption, and the *decryption key* (private key) cannot be calculated, or found out, from the *encryption key* (public key). These asymmetrical algorithms are completely different from the symmetric ones because the encryption key can be made public. Anyone with the public key can encrypt a message but only someone with the corresponding private key can decrypt the message. The key-owner keeps the private key secret while the public key is distributed through available mechanisms such as databases. This system solves the problem inherent in distributing secret keys. Asymmetric algorithms are ideal for encrypting and distributing secret keys but they are too slow to use for encryption of large quantities of data. For this reason, symmetric algorithms are used for this purpose. The two techniques are best used together: an asymmetrical algorithm distributes the secret key used by a symmetric algorithm for encryption of the data.

Now Alice can utilize Bob’s public key “ $P_uK_B$ ” for confidential distribution of her secret symmetrical key “ $SK_{AB}$ ”. Bob, the owner of the private key “ $P_rK_B$ ”, will consequently be the only one that can decrypt the message and interpret Alice’s secret key (Figure 7).



**Figure 7:** Alice utilizes an asymmetrical algorithm for confidential distribution of her key.

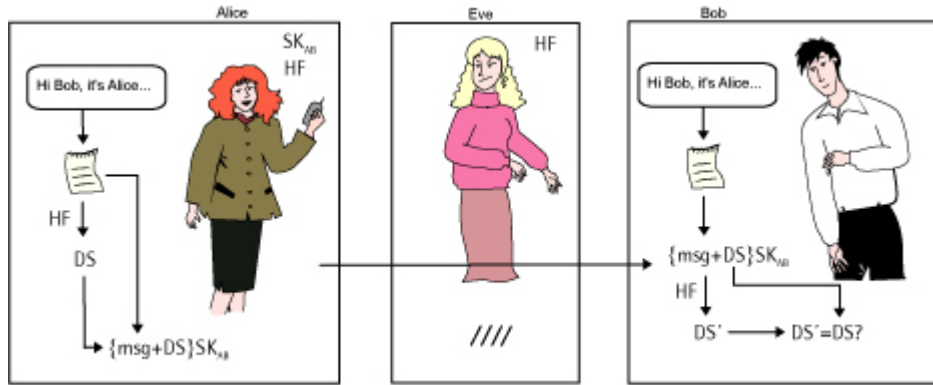
## 4 Preventing Malicious Modifications

We have seen how crypto algorithms can be effectively utilized for protecting messages from any eavesdropping by third parties. However, there are still some important security issues that need to be considered. As previously mentioned, we need some data integrity mechanisms that prevent messages from being maliciously modified during transit. Data integrity can be accomplished with a *one-way hash function*, which is used for calculating a *digital summary* (message digest) of a message. The digital summary can be seen as a fingerprint of the message and can be effectively utilized to provide data integrity.

To illustrate, consider our friends Alice and Bob again. They use a symmetrical algorithm together with a one-way hash function to provide both confidentiality and data integrity protection for a message (Figure 8).

1. As before, the secret key is securely exchanged between Alice and Bob by utilizing an asymmetrical algorithm. They then publicly determine what one-way hash function “HF” they are going to use.
2. Alice composes her message and puts the message through the one-way hash function to produce the digital summary, “DS”.
3. She then encrypts the concatenation (i.e., the linking together) of the message “msg” and the digital summary “DS” using her secret key, producing “{msg+DS}SK<sub>AB</sub>”.
4. Alice sends the encrypted concatenation to Bob over the shared network. Although Eve can intercept the message, she will not be able to decipher it because she does not possess the secret key “SK<sub>AB</sub>”. The only thing that Eve can try is to make some random changes in the encrypted concatenation, which will modify the message or the digital summary.

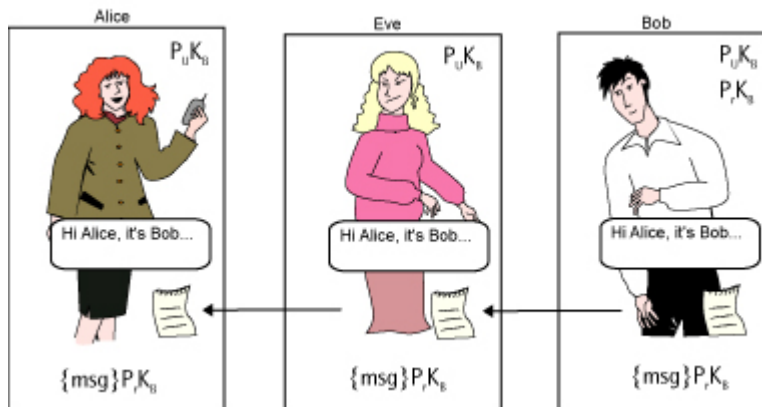
- Bob receives and decrypts the concatenation. He then puts the message through the one-way hash function to produce a digital summary, which he can then use to compare it to the one received. If both digital summaries are identical, Bob can be sure that no third party has modified the message in transit.



**Figure 8:** Alice can create and encrypt a digital summary to protect the integrity of the data.

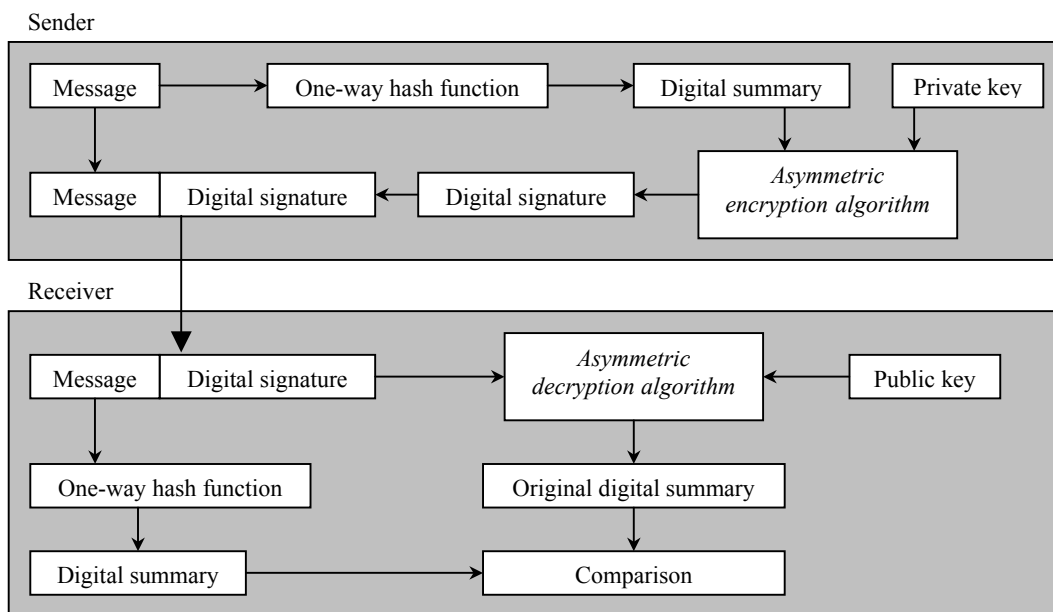
## 5 Discovering any Forgery

The previous sections described how Alice and Bob could communicate without a third party being able to eavesdrop or modify messages in transit. There is one more security issue that needs to be taken care of. How can Bob be confident that the previous messages are actually from Alice? Since everyone knows Bob's public key, anyone can claim to be Alice and send a secret key in the first encrypted message. For this reason, a mechanism is needed to enable Bob to identify the sender of the message. Fortunately, this can be done using previously described methods. First, consider what happens if Bob uses an asymmetrical algorithm and the private key " $P_rK_B$ " to encrypt a message. Everyone with access to the public key " $P_uK_B$ " can consequently decrypt and be sure that Bob sent the information (Figure 9).



**Figure 9:** Bob can utilize an asymmetrical algorithm together with his private key to provide assurance that he actually sent the message.

In doing so, a digital signature can be created. A digital signature is a cryptographic conversion made in a manner that only the valid sender can perform. A one-way hash function is utilized to calculate a digital summary of a message. This digital summary can then be encrypted with an asymmetric algorithm into a digital signature that is then concatenated - linked together - to the message before it is sent. The receiver of the message then calculates a temporary digital summary of the message, decrypts the digital signature with the public key and compares the two digital summaries (Figure 10).



**Figure 10:** The process of creating and verifying digital signatures.

If the digital summaries are equal, the receiver can rely on the fact that the message has not been tampered with and that the sender’s private key has signed the message, i.e. data integrity and authentication is provided. Additionally, to provide confidentiality, the message can be encrypted using the receiver’s public key before it is sent.

## 6 Administrating and Distributing the Public Keys

Secure distribution and management of public keys is of great concern when asymmetric cryptology is to be utilized for secure communication. The secure distribution of public keys is done utilizing specific certificates. Secure public key management is accomplished with a *Public Key Infrastructure* (PKI), which contains catalogs with public keys, as well as such information as the validity period of the keys.

A certificate for public keys is a document that confirms the connection between the public key and the key-owner. Each certificate includes the name of the authority that issued it, the name of the entity to which the certificate was issued, the entity's public

key, and time stamps that indicate the certificate's expiration date. The organization that issues a certificate is commonly called a *Certificate Authority (CA)*. By signing the certificate, the CA guarantees the reliability of the public key. In order for this system to work, it is important to trust the certificate and the signature of the CA.

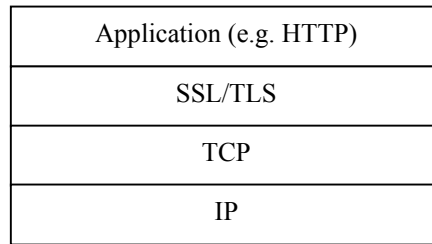
## 7 Security over IP-based Networks

We have now seen how Alice and Bob can communicate securely by utilizing different security mechanisms. These mechanisms include concepts like cryptographic algorithms, hash functions and digital signatures.

The Internet is the world's largest interconnected network. All communication over the Internet is made using the Internet protocol suite. The Internet protocol suite consists primarily of the Internet Protocol (IP) and the Transport Control Protocol (TCP). It has become common to use the term TCP/IP to refer to the whole protocol family. The TCP/IP architecture divides data into packets and then independently routes each packet through the network.

The Internet protocol suite provides no security at all. However, many applications using the Internet protocol suite require or could benefit from a mechanism that provides a higher-level of security involving such aspects as confidentiality, data integrity and authentication. Security protocols can be utilized on all layers in the protocol suite to protect data in different ways and to the extent needed. The IPsec protocol suite operates on the IP layer. Security systems that operate at the Application layer include protocols like Pretty Good Privacy (PGP). In between these are protocols that operate on the transport layer, i.e. the IETF's Transport Layer Security (TLS) standard and the older protocol from which it is derived, Secure Socket Layer (SSL).

We will now discuss the Secure Socket Layer (SSL) protocol to illustrate how network communication security can be accomplished. SSL is today the most widely used means of securing transactional data over the Internet. SSL creates a secure connection between a client and a server over which any amount of encrypted data can be sent. SSL, which formed the basis for TLS, is a general-purpose protocol that sits between the application layer and the transport layer. From the applications' perspective, this protocol looks like any other transport protocol, except for the fact that it provides security. That is, the sender can open a connection and deliver information for transmission, and SSL/TLS will provide confidentiality, data integrity and authentication mechanisms. Since SSL/TLS is running on top of the Internet protocol suite, all normal communication features are available (Figure 11).



**Figure 11:** SSL/TLS is inserted between the application and TCP layer.

When HTTP (protocol used in a Web browser) is used in this way, it is known as HTTPS (Secure HTTP). One of the great benefits that this setup provides is that HTTP does not need to be changed; it can simply utilize SSL/TLS instead of the default TCP.

The SSL/TLS protocol includes several sub-protocols. The *SSL record protocol* and the *SSL handshake protocol* are the best-known sub-protocols. The SSL record protocol is used for actual data transfers, while the SSL handshake protocol is used to negotiate parameters for a secure connection and is designed to facilitate the following:

- **SSL server authentication** allows a client to confirm a server's identity. SSL-enabled client software can use standard techniques of public key cryptography to check that a server's certificate and public key are valid and have been issued by a CA that is listed in the client's list of trusted CAs.
- **SSL client authentication** allows a server to confirm a user's identity, using the same techniques as those used for server authentication.

An encrypted SSL connection requires all information sent between a client and a server to be encrypted, i.e. protected by a confidentiality mechanism. In addition, all data sent over an encrypted SSL connection is protected with a data integrity mechanism.

## 8 Conclusion

In this white paper, we looked at several communication security problems and described solutions for each of these problems in order to achieve a secure communication.

Confidentiality is achieved using a cryptosystem that prevents any third party from eavesdropping. Modification of messages by a third party can be avoided by using a combination of a one-way hash function, which produces a digital summary, and a cryptosystem. Forgery can be prevented by using one-way hash functions and asymmetrical algorithms to produce digital signatures used for authentication purposes.

The best applications combine different cryptosystems. For encryption, the best solution is to utilize both asymmetrical and symmetrical cryptosystems together in order to get both the management advantages of asymmetrical systems and the speed advantages of symmetrical systems.

The Internet protocol suite provides no security. Thus, additional protocols, e.g. SSL/TLS, need to be utilized if an IP-based communication is to benefit from the previously described security mechanisms. Security mechanisms can be implemented in any Internet protocol suite layer to provide different levels of protection for sensitive information. In summary, as we've seen here, communication security can be accomplished for almost any system and user requirement.

## 9 About Axis Communications AB

Axis increases the value of network solutions. The company is an innovative market leader in network video and print servers. Axis' products and solutions are focused on applications such as security surveillance, remote monitoring and document management. The products are based on in-house developed chip technology, which is also sold to third parties.

Axis was founded in 1984 and is listed on the Stockholmsbörsen (XSSE:AXIS). Axis operates globally with offices in 14 countries and in cooperation with distributors, system integrators and OEM partners in 70 countries. Markets outside Sweden account for more than 95 % of sales. Information about Axis can be found at: [www.axis.com](http://www.axis.com)

**Contact Axis**  
info@axis.com

**Head office, Lund**  
Axis Communications AB  
Emdalavägen 14  
SE-223 69 Lund  
Tel: +46 46 272 18 00  
Fax: +46 46 13 61 30

### Subsidiaries

BOSTON: Phone: +1 978 614 20 00	SHANGHAI: Phone: +86 21 6431 1690
LONDON: Phone: +44 870 162 0047	SINGAPORE: Phone: +65 6 836 2777
MIAMI: Phone: +1 305-860-8556	SYDNEY: Phone: +612 9957 6700
MADRID: Phone: +34 91 803 46 43	TORINO Phone: +39 011 841 321
MUNICH: Phone: +49 811 555 08 0	TAIPEI: Phone: +886 2 2546 9668
PARIS: Phone: +33 1 49 69 15 50	TOKYO: Phone: +81 3 5531 8041
ROTTERDAM: Phone: +31 10 444 34 34	SEOUL: Phone: +82 2 780 9636



