**1.0 Introduction**

**What is SSL?**

SSL is the ubiquitous security protocol used in almost 100% of secure Internet transactions. Essentially,

SSL transforms a typical reliable transport protocol (such as TCP) into a secure communications channel

suitable for conducting sensitive transactions.i The SSL protocol defines the methods by which a secure

communications channel can be established—it does not indicate which cryptographic algorithms need to

be used. SSL supports many different algorithms, and serves as a framework whereby cryptography can be

used in a convenient and distributed manner.

**Uses for SSL**

The uses for SSL are endless. Any application that needs to transmit data over an unsecured network such

as the Internet or a company intranet is a potential candidate for SSL. SSL provides security, and more

importantly, peace of mind. When using SSL, you can be fairly sure that your data are safe from eavesdroppers

and tampering.

SSL is relatively new to the embedded world because it has been too complex for traditional embeddedsystems

microprocessors to handle. However, starting with Rev. A of the Rabbit 3000 microprocessor,

hardware assistance has been added to speed up some of the more complex SSL cryptography operations,

making SSL a viable solution in a market where standard (usually complex) security protocols have not

traditionally been supported. The applications for embedded applications are as numerous as those for the

PC world. The following are just a few potential applications for embedded SSL.

**•** The Internet-enabled vending machine can now become a reality—SSL makes tampering with communications

almost impossible.

**•** Home automation systems can be Internet-enabled—forgot to turn off the oven? Just log into your

house from your computer at work and turn it off. SSL provides a secure means of protecting your

home from hackers.

**•** Readings from medical devices can be sent over a standard network—SSL protects your privacy.

**•** Make a telephone switch Web-configurable—SSL encrypts all data, so no one monitoring the network

can read your information. Since Web-based access means that your data will likely be travelling over a

competitor’s network, SSL makes a lot of sense.

**•** Remote-entry configuration—change the passcode on all the doors of a building simultaneously. SSL

protects the passcode, allowing the doors to be connected to a standard corporate network, no need for

expensive proprietary hardware!

i. At the time of this writing, HTTP file upload over an SSL-secured channel is not supported.

**2 www.rabbit.com** SSL

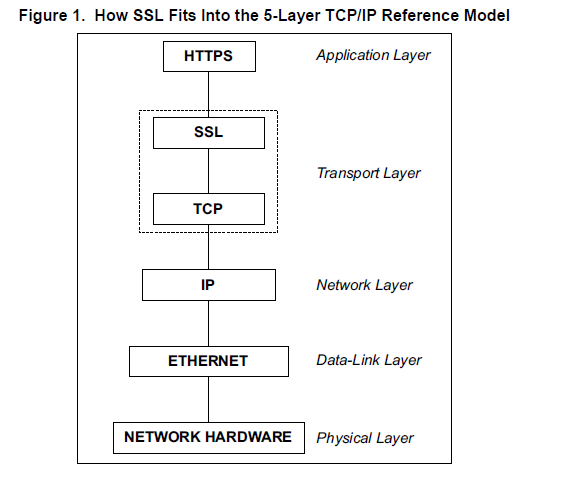
**•** Television cable box monitoring/billing—connect a cable box to the Internet to monitor use and provide

online billing.

**•** Utilities monitoring/billing (gas, electric, water)—connect gas and electric meters to the Internet without

the worry of users tampering with the information sent.

SSL is designed to run over TCP/IP. Figure 1 shows how the SSL protocol fits into the overall TCP/IP reference model.



**Figure 1. How SSL Fits Into the 5-Layer TCP/IP Reference Model**

**What can SSL do for my application?**

SSL protects the communications channel. It also provides authentication (on the client side, optionally on

the server side) of communicating parties. SSL can secure any connection between two points, and no one

monitoring the connection can do anything destructive or gain unauthorized access to any sensitive information.

SSL provides a secure channel without the need for either end to meet to exchange keys. SSL is to

secure communications as TCP is to normal communications—it provides a standard communications

infrastructure that compliant applications can use easily and nearly invisibly.

SSL provides a vitally important component of any secure system. Basic authentication mechanisms such

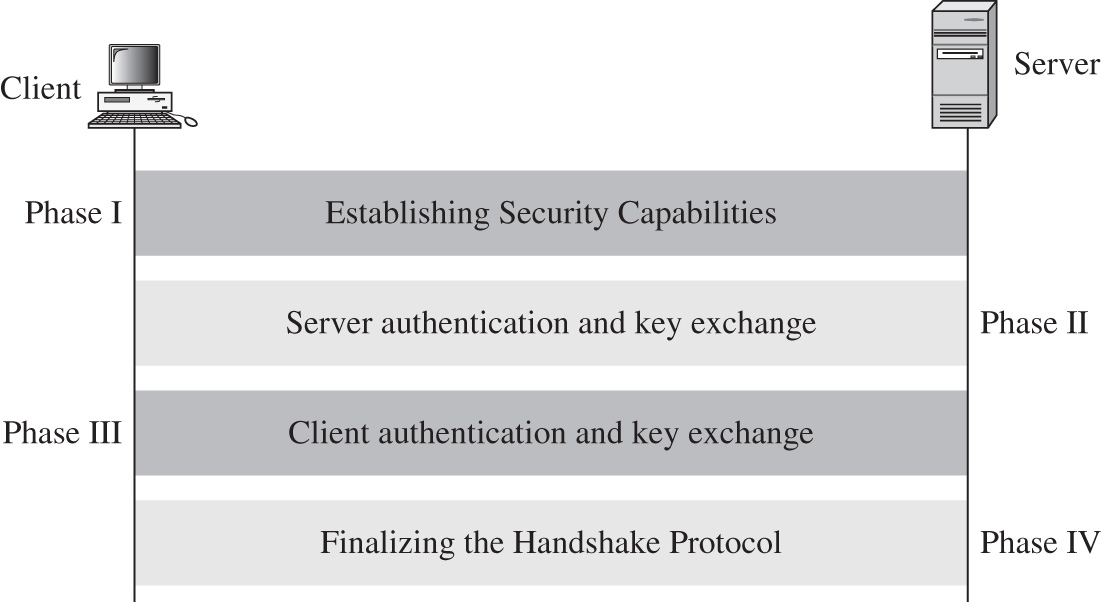
as the Telnet password and basic HTTP authentication become very powerful security options when executed

using SSL instead of plain TCP—passwords are no longer sent plain-text, making these methods

much more useful. SSL encrypts the *connection*, not the data at either end, and does not contain any mechanism

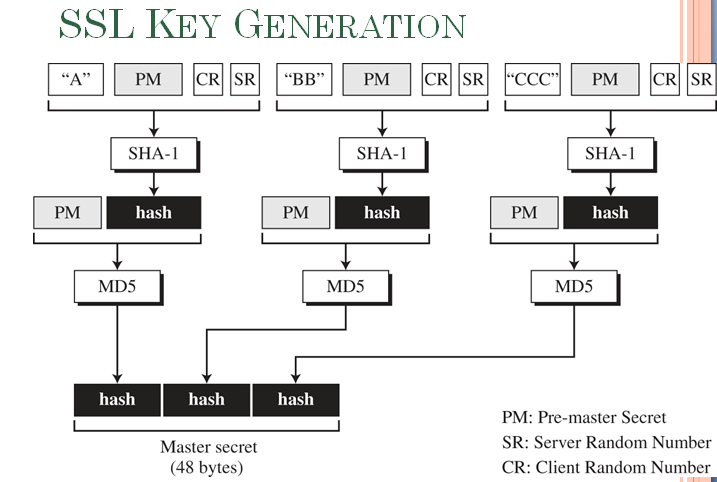
for user authentication or password protection (only the connection is authenticated—the security

fails if the machine at either end is compromised).Secure Socket Layer Protocol



Finalization handshakes protocol (Sender signs/encrypts “finished” message Receiver decrypts/verifies message to confirm keys)

The SSL handshake is a complicated process that involves significant cryptographic key exchanges. However, the handshake can be completed by calling SSL\_accept() on the SSL server and SSL\_connect() on the SSL client.



**Pre**-**Master Secret**: Key Exchange. The client and server exchange random numbers and a special number called the **Pre**-**Master Secret**. These numbers are combined with additional data permitting client and server to create their shared **secret**, called the **Master Secret**.

Client and server use to generate master key used to create cipher keys

In cryptography, **SHA**-**1** (Secure Hash Algorithm **1**) is a cryptographic hash function designed by the United States National Security Agency and is a U.S. Federal Information Processing Standard published by the United States NIST. **SHA**-**1** produces a 160-bit (20-byte) hash value known as a message digest.

**ALGORITHM USED:**

* **DES.** Data Encryption Standard, an encryption algorithm used by the U.S. Government.
* **DSA.** Digital Signature Algorithm, part of the digital authentication standard used by the U.S. Government.
* **KEA.** Key Exchange Algorithm, an algorithm used for key exchange by the U.S. Government.
* **MD5.** Message Digest algorithm developed by Rivest.
* **RC2 and RC4**. Rivest encryption ciphers developed for RSA Data Security.
* **RSA.** A public-key algorithm for both encryption and authentication. Developed by Rivest, Shamir, and Adleman.
* **RSA key exchange.** A key-exchange algorithm for SSL based on the RSA algorithm.
* **SHA-1.** Secure Hash Algorithm, a hash function used by the U.S. Government.
* **SKIPJACK.** A classified symmetric-key algorithm implemented in FORTEZZA-compliant hardware used by the U.S. Government. (For more information, see FORTEZZA Cipher Suites.)

**Triple-DES.** DES applied three times

