# Securing wireless communication with the IBM Everyplace Wireless Gateway

Henry Welborn (hwelborn@us.ibm.com)
Software Engineer, IBM
July 2002

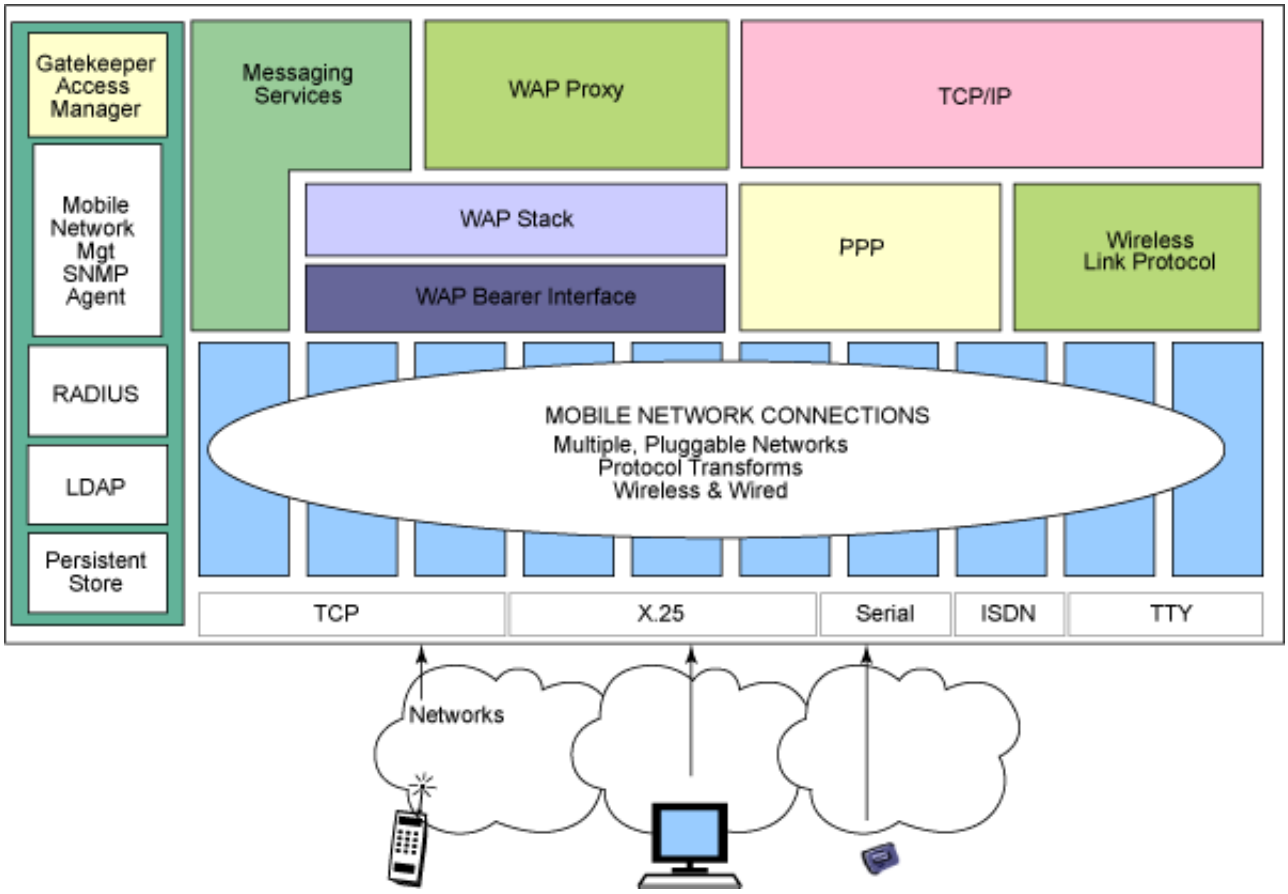
IBM Everyplace Wireless Gateway gives you a *lot* of control over the who, what, where, when, and how regarding access to your data. It offers a lot of ultra-secure features for standard Internet Protocol (IP), short messaging (SMS), and Wireless Application Protocol (WAP) clients, extending e-business and other line of business applications to a wide range of international wireless network technologies, as well as local area (LAN) and wide area (WAN) wire line networks. Recently, the product added an exciting new feature: secure cross-network roaming, which lets selected client platforms maintain their application sessions while moving among multiple wireless and wire line networks. This article focuses on the many security options in the IBM Everyplace Wireless Gateway components.



## Overview

IBM Everyplace Wireless Gateway for multiplatforms, version 2.1, is a distributed, scalable, multipurpose UNIX communications platform that supports optimized, secure data access by Internet Protocol (IP), short messaging (SMS), and Wireless Application Protocol (WAP) clients over a wide range of international wireless network technologies, as well as local area and wide area wire line networks. The Wireless Gateway is made up of a gateway, a gatekeeper, and a client, which together provide mobile access services, messaging services, and a WAP version 1.2.1 proxy. The Wireless Gateway integrates the WAP version 1.2.1 standard, secure wireless-optimized IP communications, and WAP and non-WAP messaging interfaces for short messaging networks. Figure 1 below shows the Wireless Gateway component structure.

**Figure 1. The wireless gateway components**




Mobile access services

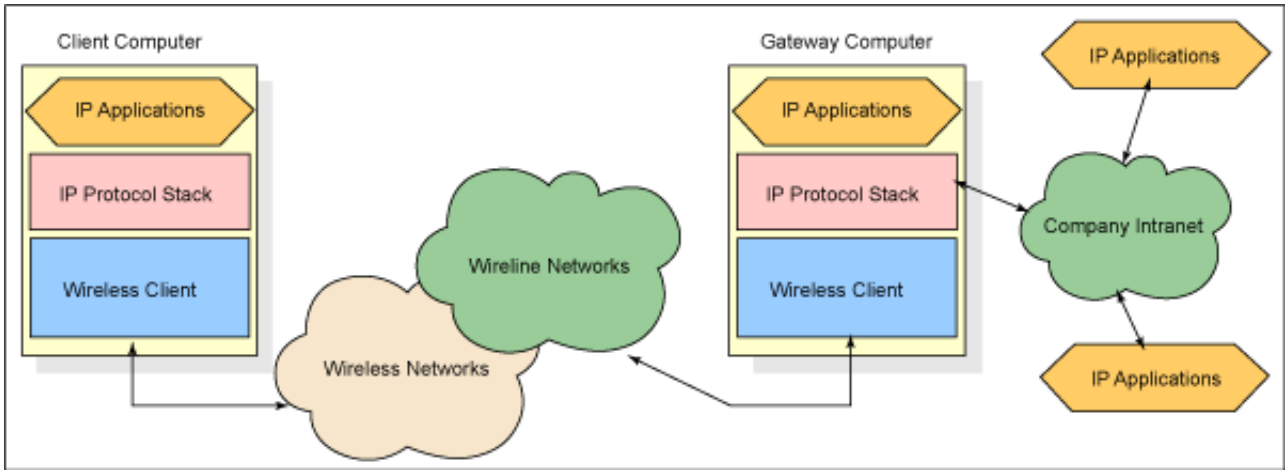
The mobile access services of the Wireless Gateway extends IP connectivity across a diverse set of wireless and wire line networks, giving e-business and other line-of-business applications seamless access to those networks. The Wireless Gateway provides a virtual private network tunnel, sensitive to the limited bandwidth capabilities and varying latencies found in wireless environments, so IP traffic can be routed efficiently and securely.

Connectivity is enabled over a wide variety of worldwide wireless networks, including 3G networks such as CDMA 2000, 2.5G networks such as GPRS, and existing circuit-switched and packet-switched networks, such as CDMA, 802.11, CDPD, GSM, DataTAC, Mobitex, Dataradio, AMPS, and Norcom Satellite. The Wireless Gateway and Wireless Client sit at opposite endpoints of the wireless network. Non-cryptographic functions of the solution, such as data compression, IP header reduction, selective packet filtering, and TCP retransmission optimizations make data communications over these networks more efficient and cost-effective. The cross-network roaming features of the product let client devices transition seamlessly from one network to the next, without interruption to end-user applications.

The secure, wireless-optimized communications channel resides as a layer 2 interface beneath the IP stack of the operating system, hiding wireless network-specific details from the application. Applications flow data through the Wireless Gateway using standard TCP/IP or User Datagram Protocol (UDP/IP) protocols, so there are no proprietary APIs, special interfaces, or toolkits required. Existing IP applications can be quickly and easily deployed in a wireless environment, while new applications can be written to standard interfaces, using your choice of development tools.

Figure 2 shows the layering model when using Everyplace Wireless Gateway.

**Figure 2. Everyplace Wireless Gateway layering model**




The product solution provides establishment, configuration, management, and authentication of the mobile link connection, along with optimization, encryption, and framing of the data flow. Authentication and encryption security features are built into its communications flow. Bidirectional authentication assures the identity of the mobile user and the Wireless Gateway to prevent unauthorized access. The Wireless Gateway authenticates all clients using a persistent user storage database, located in an LDAP-compliant directory server. To provide for data privacy and protection from eavesdropping, data is encrypted using Data Encryption Standard (DES), Triple DES, RC5, or Advanced Encryption Standard (AES) algorithms. A symmetric encryption key is used to encode or decode the data. The effective key strength varies by algorithm. The strongest allowed is a 256-bit key, which can be configured when using the AES algorithm.

Only authenticated client traffic passes through the Wireless Gateway. The Wireless Client gains IP network access by talking to the mobile access services of the Wireless Gateway. The Wireless Gateway mobile access services work with either the Wireless Client or a PPP client at the other end of the secured point-to-point connection. The Wireless Client only communicates with a single Wireless Gateway per session.

See the Mobile access services cryptography section for more information on the security of these services.

Messaging services

The messaging services feature supports push technology (so you can transmit information to client devices without requiring the end user to take an action), and supports processing messages originating from a client device. You can use the APIs in the IBM Everyplace Wireless Gateway Messaging Services and Push Toolkit to create applications that use the messaging gateway to send and receive short messages to and from a variety of client devices.

The messaging API provides a unified interface to many different types of networks and client devices. An application using the messaging API can push a message to an e-mail client or to a Global System for Mobile Communications Short Messaging System (GSM-SMS) phone by specifying different address types. The complex details of how that message is encoded, the specific protocols used by each network, maintaining connections to the network providers, and other device and network-specific activity, is hidden by the API.

Messaging services support a variety of client devices, including:

- WAP client devices (for example, WAP-enabled mobile phones)
- GSM-SMS mobile phones
- Simple Mail Transfer Protocol (SMTP) e-mail clients
- Simple Network Pager Protocol (SNPP) supported pagers and phones
- Mobitex client devices

The messaging services of the Wireless Gateway can use secured connections when communicating with push initiators (PI) using the secure socket layer (SSL). The SSL environment must be configured at the application server and at the Wireless Gateway. Result notifications and mobile-originated messages transferred between the messaging service and the message-processing server are sent over a secured link

when the application selects to use the secure HTTP (HTTPS) protocol.

The WebSphere Developer Domain article "IBM Everyplace Wireless Gateway: Short-messaging support in the WebSphere Everyplace Suite" discusses the messaging features of the product in greater detail (see Resourcesfor a link). For specific information on messaging security, refer to the Messaging services cryptography section.

WAP proxy

The Wireless Gateway's WAP proxy feature is the means of access from a WAP client device to a range of Web based services and content in the information network of a service provider or an enterprise. The WAP proxy behaves much like a Web proxy in conventional Internet-based networks. The end user's WAP device is configured to communicate with the Wireless Gateway using the appropriate network address (MSISDN, IP, and so on). The WAP proxy terminates the WAP session and provides the appropriate conventional HTTP or HTTPS session to the remote content (Web) server. The WAP session can be secured by the Wireless Transport Security Layer (WTLS) protocol, as defined in the WAP specification. The HTTPS session is secured by the SSL protocol.

The WAP proxy can include more value-add functions such as security, integration with an enterprise authentication server, WAP content caching for improved performance, cookie storage on behalf of WAP clients, and so forth. The content caching can also be performed at a separate intermediate proxy. If the WAP proxy establishes an SSL connection to the content server, an intermediate proxy cannot access, modify, or cache the content. The WAP proxy cryptography section has more information on WAP security.

Wireless Gateway administration

The Wireless Gateway is administered from the Wireless Gatekeeper, a Java application that's installed locally, on the computer where the Wireless Gateway resides, or remotely, on a computer that has TCP/IP access to the Wireless Gateway.

All administrators are required to log in to the Wireless Gateway, which then authenticates all administrators against a persistent user storage database in an LDAP-compliant directory server. Administrators have an access control list (ACL) associated with their user ID that governs the level of administration that can be performed.

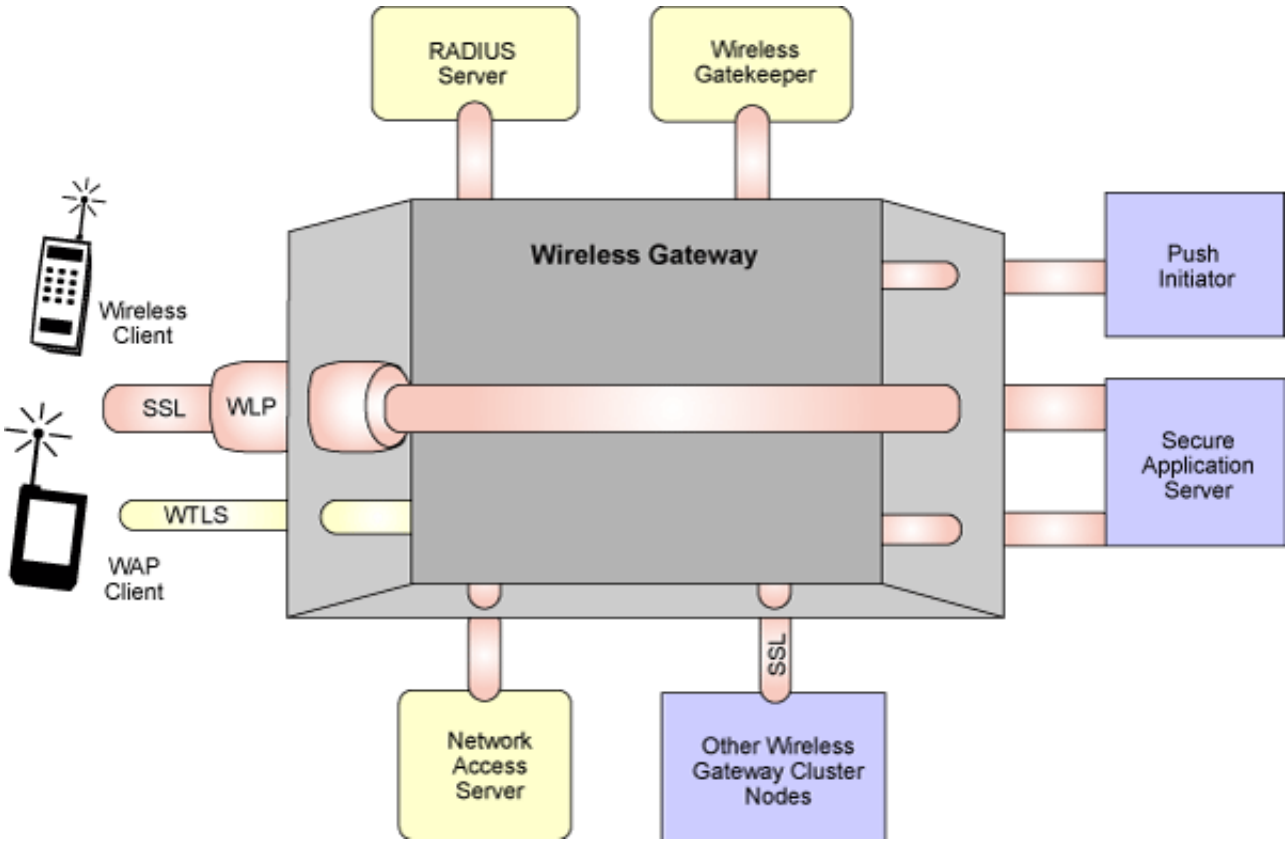
The connection between the Wireless Gatekeeper and the Wireless Gateway may be secured over a secure sockets layer (SSL) tunnel.

More information on Wireless Gatekeeper security is in the Wireless gatekeeper cryptography section.

Wireless gateway security

The components that comprise the security of Wireless Gateway are shown in Figure 3.

**Figure 3. Wireless Gateway security options**




The following sections discuss security for these options in detail:

- WAP proxy cryptography
- Mobile access services cryptography
- Messaging services cryptography
- Wireless Gatekeeper cryptography
- Inter-gateway cryptography

WAP proxy cryptography

The WAP proxy security features are:

- Authentication between Wireless Gateway and the WAP client
- Encryption between Wireless Gateway and the WAP client
- Authentication and encryption between Wireless Gateway and a secure Web server

Authentication between Wireless Gateway and the WAP client

Several methods can be configured for Wireless Gateway to validate a WAP client, as described below.

**Native PPP connections**

The Wireless Gateway accepts connections from dial-in devices that support the Point-to-Point Protocol (PPP) protocol. When users are first configured by the administrator, they are assigned a password that's stored securely by the gateway and is communicated to the person using the ID (by phone, secure e-mail, or postal mail) so both parties know the password.

Using either the PAP or CHAP authentication protocols, users negotiate authentication with the Wireless Gateway. Wireless Gateway validates the user's credentials using a persistent user storage database in an LDAP-compliant directory server, or using an external authentication server that uses the RADIUS protocol.

The authentication for establishing a PPP session to Wireless Gateway serves as the authentication mechanism for WAP proxy transactions.

**Device resolver**

Though not a method of authentication, device resolver serves to uniquely identify a WAP client.

Authentication of the WAP client takes place elsewhere in the environment, usually at the server that provides network access.

In a typical service-provider environment, the network access server (NAS) has direct access to the provider's wireless network infrastructure, where unique information about a device, like its phone number, is available. The NAS uses the RADIUS protocol to inform the WAP proxy of that unique identity. The WAP proxy uses device resolver to match this information to WAP requests and validate that the device has previously been authenticated.

**Account resolver**

Account resolver is similar to the device resolver function, but instead of using the RADIUS protocol to get information about the WAP client, the WAP proxy uses HTTP to query the NAS for information about the device.

**HTTP challenge**

The Wireless Gateway can require that user credentials exist in each WAP transaction header.

When user IDs are first configured in the system by the administrator, they are assigned a password that's stored securely on the Wireless Gateway and is securely communicated to the person using the ID (by phone, secure e-mail, or postal mail) so both parties know the password.

Any transaction that does not contain a user ID and password is discarded, and the WAP client browser is sent a reply with a status code of either Proxy-Authenticate or WWW-Authenticate, depending on the environment. The WAP client browser prompts users to enter their user ID and password, and then resubmit the transaction with the credentials included. These credentials, along with the network address of the WAP client, are used to authenticate the user. The Wireless Gateway can validate the user's credentials using a persistent user storage database in an LDAP-compliant directory server, or using an external authentication server, using the RADIUS protocol.

Encryption between Wireless Gateway and the WAP client
The WAP Forum has defined the WTLS layer as the means for securing communications between a WAP client and a WAP proxy. Wireless Gateway's WAP proxy supports the WTLS Class 2 options. The following WTLS protocols are supported:

- Public key algorithms
  - RSA (1024-, 768-, or 512-bit keys)
  - Diffie-Hellman (1024-, 768-, or 512-bit keys)
  - Elliptic curve Diffie-Hellman (113-, 131-, or 163-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - Triple DES (168-bit key)
  - RC5 (40-, 56-, or 128-bit keys)
- Message authentication codes
  - SHA-1

The WAP proxy can be configured to use either a WTLS certificate as defined in the WAP WTLS specification, or a randomly generated key-pair for public key communications with the WAP clients. X.509 and X9.68 certificate formats, although supported by the WAP WTLS specification, are not supported by the WAP proxy. The Wireless Gateway administrator can configure the desired protocols allowed (including disabling encryption altogether), and the desired strength of the protocols.

Encryption and hashing of transactions flowing between the WAP client and the WAP proxy are used within the confines of the WTLS protocol. The encryption algorithms are implemented using the RSA BSAFE Crypto-C encryption toolkit, version 5.2.0. The Cryptographic library utilization section has information on the usage guidelines for the encryption toolkit.

Authentication and encryption between Wireless Gateway and a secure Web server
When a WAP client requests secure Web content using the HTTPS protocol, the WAP proxy forwards the communications to the content server. The WAP proxy decodes the WAP request, encrypted with the

WTLS protocol, and sends the request to the secure Web server using an SSL connection. Both version 2 and version 3 of the SSL protocol are supported. The following SSL protocols are supported:

- Public key algorithms
  - RSA (1024-, 768-, or 512-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - Triple DES (168-bit key)
  - RC4 (40-, 56-, or 128-bit keys)
- Message authentication codes
  - SHA-1
  - MD5

You can use an X.509 certificate to provide client-side authentication for the WAP proxy's SSL communications. These certificates, along with root certificates to validate the Web server's certificate, are stored in a key database that's installed with Wireless Gateway. The Wireless Gateway administrator can configure the source of this database using the Wireless Gatekeeper administration console. The administrator can configure the desired root certificates and client side certificates using the administration interface of the SSL toolkit, IBM Key Management.

Encryption and hashing of transactions flowing between the WAP proxy and the secure Web server are used within the confines of the SSL protocol. SSL is provided by the IBM Tivoli Secure Sockets for C toolkit, Version 5. The Cryptographic library utilization section has information on the usage guidelines for this toolkit.

Mobile access services cryptography
Authentication and encryption services between the mobile access services of the Wireless Gateway and the Wireless Client components are provided within the secure wireless-optimized communications layer.

For authentication, the Two-Party Key Distribution protocol (2PKDP) is used. As an independent vendor, IBM developed 2PKDP as an open authentication protocol in 1994, and has been available in a published research paper since that time. In a networked environment, it is often desirable for both parties to be authenticated before sensitive data is exchanged. 2PKDP combines bi-directional authentication with key distribution using minimal messages.

2PKDP is a shared-secret authentication scheme, where the shared secret is derived from a password stored securely on the gateway in a persistent user storage database in an LDAP-compliant directory server. That password has to be initially communicated to the person using the ID by a secure media, so both parties know the password. By using 2PKDP, both the client and the gateway authenticate each other, yet never transmit the password during the authentication process. Passwords can be set to expire at configured intervals, requiring the user to change the password, and accounts may be locked once a threshold of invalid logins is reached.

The shared secret derivation depends on the gateway's configuration, using either an MD5 or SHA-1 based methodology. To ensure the proper strength of the shared-secret key, rules can be established on Wireless Gateway to enforce the unpredictability of the password. A seed may also be used to strengthen the key space of the secret key.

If you want additional authentication, you can configure the mobile access services of Wireless Gateway to require a second level of authentication. The user will be prompted to enter another user ID and password, which will then be validated against a RADIUS compliant authentication server, such as the RSA ACE/Server, used for the RSA SecurID system. The gateway can also use this scheme to verify possession of a valid X.509 certificate on a smart card.

The session key exchanged as part of 2PKDP is used to establish data encryption of the communications channel. Mobile access services supports the following encryption algorithms:

- Symmetric key algorithms
  - CDMF (40-bit key)

- DES (56-bit key)
- RC5 (56-bit key)
- Triple DES (168-bit key)
- AES (128-, 192-, 256-bit keys)

The encryption algorithms supported by the Wireless Client vary based on platform, as shown in the table below.

| | |
|---|---|
| Wireless Client for Windows | DES, Triple DES, AES |
| Wireless Client for Windows CE | DES, Triple DES, AES |
| Wireless Client for Palm OS | RC5 |

Encryption is used to secure data packets flowing between the Wireless Client and the Wireless Gateway within the confines of the secure network tunnel. The encrypted data is data originated from any number of applications that are configured to route traffic through the IP tunnel created by the Wireless Client and Wireless Gateway. Only data flowing though this IP tunnel is encrypted. It is possible for the Wireless Gateway administrator to disable encryption for specific clients, if it is not desired.

For the Wireless Client for Palm OS, the encryption algorithms are implemented using the RSA BSAFE Crypto-C encryption toolkit, version 4.0. For the mobile access services of the Wireless Gateway and all other Wireless Client platforms, the encryption algorithms are in the Everyplace Wireless Gateway cryptographic library. Cryptographic library utilization has more information on the usage guidelines for the encryption toolkit and library.

Messaging services cryptography
You can configure the messaging services of Wireless Gateway to use the SSL protocol when communicating with its message-processing servers or push initiators. Both version 2 and version 3 of the SSL protocol are supported. The following SSL protocols are supported:

- Public key algorithms
  - RSA (1024-, 768-, or 512-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - Triple DES (168-bit key)
  - RC4 (40-, 56-, or 128-bit keys)
- Message authentication codes
  - SHA-1
  - MD5

You can use an X.509 certificate to provide authentication for the SSL communications. These certificates, along with root certificates to validate the other party's certificate, are in a key database that is installed with Wireless Gateway. The Wireless Gateway administrator can configure the source of this database using the Wireless Gatekeeper administration console. The administrator can configure the desired root certificates and client side certificates using the administration interface of the SSL toolkit, IBM Key Management.

Encryption and hashing algorithms of transactions flowing between the messaging services and the message processing server or push initiator are used within the confines of the SSL protocol. SSL is provided in the messaging gateway by the IBM Tivoli Secure Sockets for C toolkit, version 5. The Cryptographic library utilization section has more information on the usage guidelines for this toolkit.

Messages flowing between the messaging gateway and the client devices are not encrypted.

By default, the IBM Everyplace Wireless Gateway Messaging Services and Push Toolkit does not provide for SSL communication to the messaging gateway. Customers who bought Wireless Gateway and want to write an SSL enabled message-processing server or push initiator application can contact IBM to get

access to an SSL enabled toolkit. The SSL interface boundary is contained within the toolkit, and is neither exposed nor controlled by applications written using the toolkit.

The IBM Everyplace Wireless Gateway Messaging Services and Push Toolkit uses either the IBM Tivoli Secure Sockets for C toolkit, version 5, or the IBM Tivoli SSLight for Java, version 5, depending on whether the C or Java version of the toolkit is used. More information on usage guidelines for this toolkit is in Cryptographic library utilization.

Wireless Gatekeeper cryptography

The Wireless Gateway uses its access manager subsystem to communicate with the Wireless Gatekeeper administration console. You can configure the access manager to use the SSL protocol when communicating with Wireless Gatekeepers. Both version 2 and version 3 of the SSL protocol are supported. The following SSL protocols are supported for access manager:

- Public key algorithms
  - RSA (1024-, 768-, or 512-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - Triple DES (168-bit key)
  - RC4 (40-, 56-, or 128-bit keys)
- Message authentication codes
  - SHA-1
  - MD5

You can use an X.509 certificate to provide authentication for the SSL communications. This certificate, and root certificates to validate the Wireless Gatekeeper's certificate, are stored in a key database that is installed with Wireless Gateway. The Wireless Gateway administrator can configure the source of this database using the Wireless Gatekeeper administration console. The administrator can configure the desired root certificates and client side certificates using the administration interface of the SSL toolkit, IBM Key Management.

The following SSL protocols are supported for Wireless Gatekeeper:

- Public key algorithms
  - RSA (512-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - RC4 (40-bit key)
- Message authentication codes
  - SHA-1
  - MD5

Again, you can use an X.509 certificate to provide authentication for the SSL communications. These certificates, along with root certificates to validate the access manager's certificate, are stored in a key database that is installed with Wireless Gatekeeper. The Wireless Gatekeeper user configures the source of this database when defining a connection to a Wireless Gateway. The administrator can configure the desired root certificates and client-side certificates using the administration interface of the SSL toolkit, IBM Key Management.

In addition to X.509 certificate validation, each administrator ID has an associated password. The Wireless Gateway can validate the administrator's credentials against a persistent user storage database in an LDAP-compliant directory server. That password has to be initially communicated to the end user administrator by a secure media, so both parties know the password. Passwords can be set to expire at configured intervals, requiring the administrator to change the password. Rules can be established on Wireless Gateway to enforce the unpredictability of the password.

The various resources of Wireless Gateway can be organized into separate organizational units (OU), with separate administrators having different access rights to these resources. And, access manager can restrict

the source IP address from which a Wireless Gatekeeper may connect from, and can force all Wireless Gatekeepers to use SSL communications.

Encryption and hashing of transactions processed in access manager are used within the confines of the SSL protocol. SSL is provided by the IBM Tivoli Secure Sockets for C toolkit, version 5. Encryption and hashing of transactions processed in Wireless Gatekeeper are also used within the confines of the SSL protocol. SSL is provided by the IBM Tivoli SSLight for Java, version 5. Cryptographic library utilization has more information on the usage guidelines for these toolkits.

Inter-gateway cryptography
For load balancing and high availability performance, several Wireless Gateway installations can be configured to work together in a clustered configuration. The cluster manager subsystem controls the communications among the nodes. You can configure the cluster of Wireless Gateway nodes to communicate using the SSL protocol. Both version 2 and version 3 of the SSL protocol are supported. The following SSL protocols are supported:

- Public key algorithms
  - RSA (1024-, 768-, or 512-bit keys)
- Symmetric key algorithms
  - DES (56-bit key)
  - Triple DES (168-bit key)
  - RC4 (40-, 56-, or 128-bit keys)
- Message authentication codes
  - SHA-1
  - MD5

You can use an X.509 certificate to provide authentication for the SSL communications. These certificates, and root certificates to validate another node's certificate, are stored in a key database that is installed with each Wireless Gateway. The Wireless Gateway administrator can configure the source of this database using the Wireless Gatekeeper administration console. The administrator can configure the desired root certificates and client side certificates using the administration interface of the SSL toolkit, IBM Key Management.

Encryption and hashing algorithms of transactions flowing between cluster manager nodes are used within the confines of the SSL protocol. SSL is provided by the IBM's Tivoli Secure Sockets for C toolkit, version 5. More information on the usage guidelines for this toolkit is in Cryptographic library utilization.

Cryptographic library utilization
Wireless Gateway uses the following toolkits and libraries to secure the various features and components:

- RSA BSAFE Crypto-C
- IBM Tivoli Secure Sockets for C
- IBM Tivoli SSLight for Java
- Everyplace Wireless Gateway cryptographic library

The encryption algorithms implemented using the RSA BSAFE Crypto-C encryption toolkit, version 5, are only in object code made available by RSA. No source code is available to end users of Wireless Gateway. No modifications are made to the binaries provided by RSA. The toolkit is statically linked into the product code, and no cryptographic APIs are exposed outside of the product.

The encryption algorithms implemented using the RSA BSAFE Crypto-C encryption toolkit, version 4, are in source code made available by RSA to IBM, and compiled by IBM into object code for the Palm OS platform. No source code is available to end users of Wireless Gateway. No modifications were made to the code provided by RSA. The library is statically linked into the product code, and no cryptographic APIs are exposed outside of the product.

The encryption algorithms implemented using the IBM Tivoli Secure Sockets for C toolkit, version 5, are only in object code made available by Tivoli. No source code is available to end users of Wireless Gateway. No modifications are made to the binaries provided by Tivoli. The SSL toolkit, in turn,

incorporates the RSA BSAFE Crypto-C encryption toolkit, version 3, for cryptographic function. BSAFE is statically linked into the SSL toolkit.

The SSL library is dynamically linked into the product code. The level of cryptography used in SSL is determined at compile time of the Wireless Gateway product, and is not configurable by the customer. No source code of cryptographic functions or access to cryptographic features is supplied as part of the product deliverable.

The encryption algorithms implemented using the IBM Tivoli SSLight for Java, version 5, are only in bytecode made available by Tivoli. The cryptographic function is provided by internal Java classes. The toolkit's cryptographic methods are declared as "static final." The lack of access modifier on the definition implicitly tells the Java compiler that this method is additionally package-friendly, meaning that a class outside of `com.ibm.sslight` package may not access the cryptographic methods. The bulk encryption capabilities of the cryptography in the toolkit is limited to 56-bits at compile time in the export version of the package. RC4 and DES are the only bulk data ciphers built into the toolkit. Code obfuscation is provided by the Java compiler optimization (-O option). This level of optimized code makes it difficult to decompile and re-use the derived source code.

The tests performed by IBM with popular decompilers (Jasmine, for example) showed that decompiled toolkit code cannot be compiled and used without extensive alteration. Further obfuscation is done in a precompile step to hide the names and entry points of the encryption methods and classes by mangling the internal function names.

The encryption algorithms implemented using the Wireless Gateway cryptographic library are in source code developed by the IBM Zurich Research Lab "BlueZ" security team as part of the CryptoLite in C toolkit (CliC), version 2, and compiled by IBM into object code for the AIX, Solaris, Windows, and Windows CE operating systems. The cryptographic library is currently obtaining FIPS 140-2 validation.

The Wireless Gateway cryptographic library is dynamically linked into the product code; no source code is made available to end users. The library requires role-based authentication for access to the cryptographic APIs, as defined in the FIPS 140-2 publication. The authentication credentials are determined at compile time of the Wireless Gateway product, and are not configurable by the customer, thus preventing external applications from gaining access to the cryptographic APIs.

Resources

- Participate in the discussion forum on this article by clicking **Discuss** at the top or bottom of the article.
- Find out more about the IBM Everyplace Wireless Gateway on the IBM Pervasive Computing site.
- Learn about short-messaging support provided by the IBM Everyplace Wireless Gateway in "IBM Everyplace Wireless Gateway: Short-messaging support in the WebSphere Everyplace Suite" (*WebSphere Developer Domain*, April 2001). IBM Everyplace Wireless Gateway: Short-messaging support in the WebSphere Everyplace Suite
- Explore the pervasive computing enablement resources available from the IBM Solution Partnership Centers.
- Take IBM pervasive computing classes to learn how to implement wireless solutions.
- Visit the IBM e-business Wireless University.
- Tour the developerWorks Wireless zone for more resources.
- Check our references:
  - Simpson, W., et al. July 1994. *The Point-to-Point Protocol (PPP)* . The Internet Engineering Task Force. Internet. 22 April 2002.
  - Lloyd, B., et. al. October 1992. *PPP Authentication Protocols* . The Internet Engineering Task Force. Internet. 22 April 2002.
  - Simpson, W., et al. August 1996. *PPP Challenge Handshake Authentication Protocol (CHAP)* . The Internet Engineering Task Force. Internet. 22 April 2002.
  - Rigney, C., et. al. April 1997. *Remote Authentication Dial In User Service (RADIUS)* . The

Internet Engineering Task Force. Internet. 22 April 2002.

- WAP Forum. February 1999. *Wireless Application Protocol Wireless Transport Layer Security Specification11-February-1999* . WAP Forum. Internet. 22 April 2002.
- Janson, P. and Tsudik, G. 1994. *Secure and Minimal Protocols for Authenticated Key Distribution* . Computer Communications.
- National Institute of Standards and Technology. May 2001. *Security Requirements for Cryptographic Modules* . Federal Information Processing Standards Publication 140-2.
- Rivest, R. April 1992. *The MD5 Message-Digest Algorithm* . The Internet Engineering Task Force. Internet. 22 April 2002.

About the author

Henry Welborn is a Software Engineer for IBM in Research Triangle Park, North Carolina. He started with IBM as a co-op student from Virginia Tech in 1994, has been developing wireless middleware since he joined IBM full time in 1998, and has served as lead developer of the Everyplace Wireless Gateway security team since 2000. You can reach Henry at hwelborn@us.ibm.com.