WAP Security

Ric Howell, Concise Group Ltd

Security of applications and computer systems is an issue that, quite rightly, many IT professionals are concerned about. As corporations have utilized technologies, such as remote access, Java and component technologies, and infrastructural advances like the Internet, to facilitate new ways of working, new ways of doing business with clients, partners and suppliers, and even to create entirely new products, services and business models, the need for mechanisms to secure applications, networks and systems has become more and more important.

WAP is another technology that extends the reach of communication networks, provides new opportunities for innovative corporations, and adds to the complexity of the environment within which applications need to be designed, built and deployed. There is a set of concerns over how secure WAP is as a technology, and whether it is robust enough to implement mobile commerce applications, and other applications with stringent security requirements.

Before beginning this investigation of WAP security, it is worth noting that there is no such thing as a secure system. The phrase 'secure system' means one that cannot be compromised or accessed without authorization. Considering that hackers who set out to compromise or penetrate systems are resourceful and always target unexpected aspects of the systems, it would be a brave fool who declared a system to be immune to attack. What can be said is that a particular system meets certain predefined security criteria in that it can withstand attacks of a known type, and is therefore considered secure enough for its intended purpose.

If your interest in this paper is to come out with a definitive statement as to whether WAP is 'secure' or not, you will be disappointed. It is only feasible to make the assertion that WAP is or is not 'secure enough' for a particular application when you understand the security requirements of that application, the environment in which that application is to be deployed, the likelihood that the application will be subject to attempts to compromise its security, and the nature of the attempts that are likely to be made. Even then the statement is only valid until something changes in the environment, or someone discovers a new security exposure in the network, the environment, the technologies used or the platform on which the application is deployed.

This paper investigates the facilities and technologies that WAP has to offer for building and deploying secure applications. The presentation itself draws on the WAP Security chapter of the Wrox book "Professional WAP", and is intended to pick out some of the highlights from the book. The presentation, and this paper, do not necessarily provide a full treatment of the subject, or explain all of the concepts in detail; for that information you will need to read the book.

What Security is About

We are going to begin the investigation of the topic of security with a discussion of what security is about and why it matters. In this section we will investigate:

- The importance of security in mobile applications
- The role of security in protecting data and systems
- The basic issues which security solutions of all types need to address

The Importance of Security

Security has an obvious role to play with regard to m-commerce and the ability to secure transactions. Most people are aware of the need for securing information such as credit card numbers, but the need for security in both the wired and wireless environments is much broader than that.

At the moment, information often has a commercial value. Many dot-com organisations make money through the sale or re-sale of information. This is not a new phenomenon — newspapers have been doing it for centuries — but the new channels for this kind of commercial activity have lowered the barriers to entry and increased the amount (and hence the value) of the information available.

Information can also be sensitive. There are many reasons why this may be the case, ranging from a justifiable desire for privacy to information that is sensitive on a national security level. Sometimes the sensitivity comes from the content of the information, at other times the timing of the information. For example, it is unacceptable to allow some stock market investors to become aware of an impending profits warning from a company before others, so the information is regarded as sensitive until it is published formally to all investors.

The power associated with information must also not be underrated. Some organizations have legal obligations to safeguard certain items of information. In some cases divisions within organizations are subject to similar constraints. There are many examples of information that are intrinsically powerful, for example, information about military weapons.

Along with all of the sensitivity that naturally accompanies information, there is a growing need to communicate digitally, because of the speed and convenience of doing so. However, in certain ways these digital communications are more vulnerable to compromise. Two major weaknesses in digital communications arise from the fact that it is notoriously easy to intercept digital messages, and the fact that it is notoriously difficult to establish identity conclusively in an online environment.

All of this leads us to two inevitable conclusions that drive the need for robust security implementations: computer systems are critical to the operation of almost every society on earth; and computer systems are very vulnerable to abuse.

The Role of Security

Security is both an enabling and disabling technology. Its purpose is to enable communications and transactions to take place in a secure environment without fear of compromise, while at the same time disabling non-legitimate activities and access to information and facilities. Non-legitimate activities include eavesdropping, pretending to be another party (also known as impostering or spoofing), or tampering with data during transmission. In general these activities are either unacceptable or illegal outside of the digital environment, so security simply helps to enforce the status quo in that sense.

The Basic Issues

There are a number of basic issues around security that have to be addressed. Almost all of these have parallels in the real world, and often the solutions are based on, or similar to, real-world solutions.

These basic issues are:

- **Authentication** – being able to validate that the other party participating in a transaction is who the party claims to be, or a legitimate representative of that party
- **Confidentiality** – being able to ensure that the content and meaning of communications between two parties do not become known to third parties
- **Integrity** – being able to ensure that messages received are genuine and have not been tampered with or otherwise compromised
- **Authorization** – being able to ascertain that a party wanting to perform some action is entitled to perform that action within the given context
- **Non-repudiation** – being able to ensure that once a party has voluntarily committed to an action it is not possible to subsequently deny that the commitment was given by that party

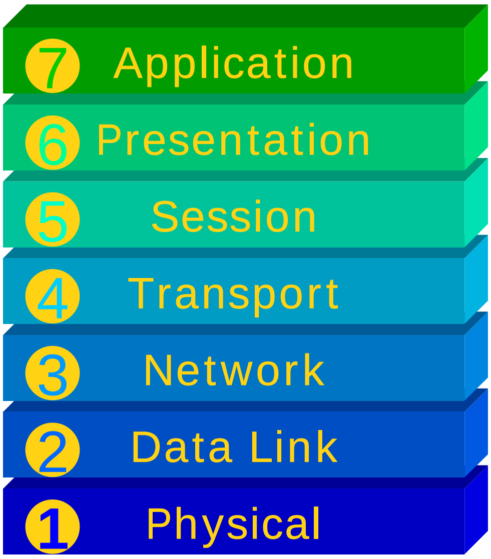
Concepts

Familiarity with some concepts relating to digital communications and to security are required in order to understand the points made later in this paper, and the place within the communications process of the existing security solutions.

Protocol Stacks

There is an industry standard theoretical protocol stack that was developed by the Open Systems Initiative (OSI) many years ago, in part to facilitate a common understanding of the functionality provided by a protocol stack and to facilitate comparisons between different vendor's implementations.

This stack is shown in the diagram below.



The 'bottom most' layer of the OSI stack, Layer 1 or the physical layer, defines the properties of the physical medium through which communications are transmitted and the characteristics of electrical transmission through that medium.

Above that is Layer 2, the data link layer. The data link layer is responsible for the transmission of data over the physical medium and also for the addressing of devices on the network.

The third layer is the network layer, which is responsible for network addressing and for the routing of data between networks.

The transport layer is the fourth layer and is responsible for preparing data for transmission across the data-link. This includes such functions as segmentation and reassembly of packets of information, and also sequencing of packets and retransmission of packets that get lost or corrupted.

Layer 5 is the session layer, which is responsible for establishing and maintaining sessions between two devices across a network. What exactly this entails depends on the protocols involved.

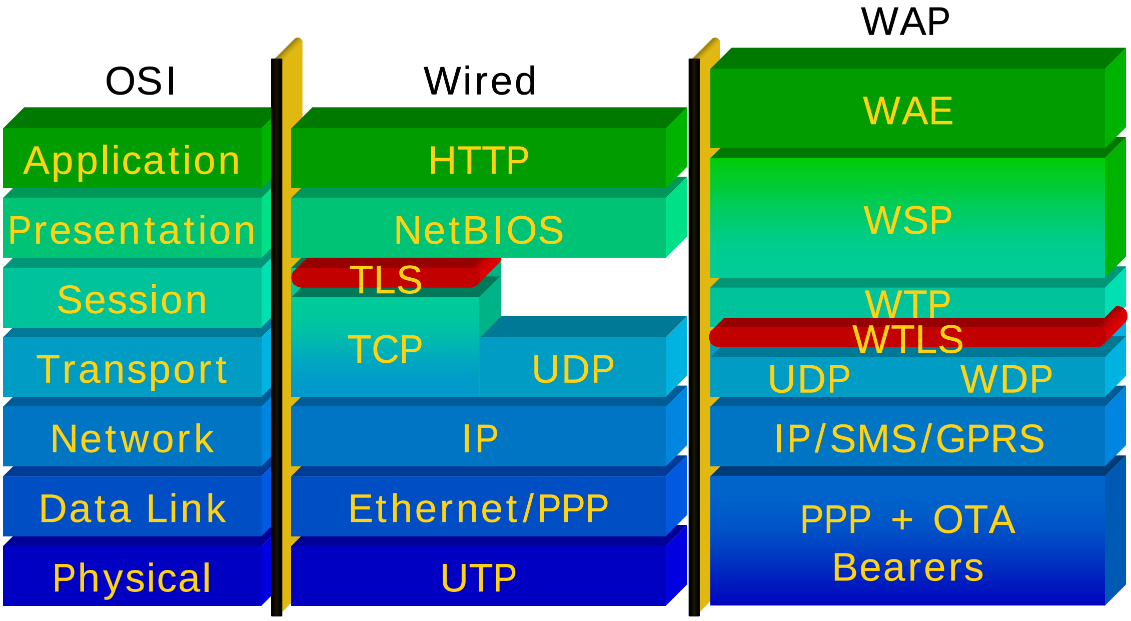
Above layer 5 is the presentation layer, which is responsible for translation and reformatting of data that is transmitted or received over the network. This helps to facilitate communication between computers that are based on different architectures and which utilize different information representation schemes.

The last layer, layer 7, is the application layer, which is responsible for identifying requests for remote resources and for the reformatting of those requests as remote requests. This allows applications to operate independently of the location of the services that they utilize.

Although the details of the actual protocol stack will vary depending on a whole host of factors (such as the type of network, where the client or server device resides) in general in the wired world certain participants in the stack are more common and more typical than others. The mapping of the wired protocol stack onto the OSI model is as follows:

- **Physical layer** — UTP or co-axial cable
- **Data link Layer** — Ethernet, Token-ring, FDDI or PPP
- **Network layer** — IP
- **Transport layer** – TCP or UDP
- **Session layer** — TCP
- **Presentation** — varies, but could be NetBIOS or XDR
- **Application layer** — depends on the service being invoked; a typical example is HTTP

In the wireless world a similar kind of mapping exists, although with different protocols at each layer. The mapping of both the fixed-wire and WAP protocol stacks is shown in the diagram below:



The WAP protocol stack contains the following elements:

- **Physical and Data link layers** — depends in part on the type of wireless network, but with WAP it will be PPP over one or more over-the-air bearer protocols.
- **Network layer** — IP is the network layer protocol of choice, although not all wireless networks are capable of transmitting IP, so SMS or some other non-packet network protocol may be used.

- **Transport layer** — the transport layer protocol of choice is UDP, but it may not be feasible over non-IP networks. For this and other reasons, WAP defines an additional transport layer protocol, WDP, which can be used where UDP cannot.
- **Session layer** — In the wireless world some of the functionality of the session layer is incorporated into WTP, while other aspects are included with WSP.
- **Presentation layer** — this functionality is included in WSP.
- **Application layer** — some aspects of application layer functionality are taken care of by WSP, whereas others are implemented in the Wireless Application Environment.

Encryption

Cryptography is the study of encryption, or the science of encoding data into another format that cannot easily be decoded or understood, using some sort of mathematical algorithm. The mathematical algorithms are based on an intractable (difficult to solve) problem. There are two of these problems that are commonly used for encryption: one is finding the prime factors of a very large integer; the other is finding the logarithm of a very large number to a known base.

Developing and proving the robustness of an encryption algorithm (called a **cipher**) is extremely difficult, so there are relatively few of these algorithms in existence. If everyone used the same few algorithms their effectiveness at concealing information would be severely limited, so the algorithms use keys, which are strings of bits, to 'customize' the behavior of the algorithm. What this means, in effect, is that the same algorithm can be used to encode the same original information twice using two different keys and produce two completely different encoded forms. This helps to make these algorithms useful for multiple people from the point of view that in order to decode the message both the algorithm and the key have to be known.

In general, the strength of the algorithm (usually defined in terms of how much effort is required to decode an encoded message) depends on the length of the key. Unfortunately the relationship is not actually that simple, because keys of equivalent lengths can provide different levels of protection when used with different algorithms. Therefore there is no general rule about how long a key should be, although some guidelines do exist for various algorithms. The problem with these guidelines is that as computer power increases the ease with which algorithms can be cracked also increases, so it is necessary to be constantly aware of advances in this area.

All cryptographic algorithms, because of their computationally intensive nature (remember they are dealing with intractable mathematical problems) are computationally expensive, which is a nice way of saying that they are slow on most computers. This has implications in most applications, where processing power is not unlimited and where response times count. However, it is also true that not all algorithms are equally computationally expensive.

In particular, there is a class of ciphers that are particularly expensive, but which provide some particularly useful features. These are called **asymmetric ciphers**. Their less computationally expensive counterparts are called **symmetric ciphers**. Symmetric ciphers make use of the same key to both encode and decode the data.

The problem with these types of ciphers is that both the party encoding the message and the party decoding the message need to have a copy of the key, and finding a secure way to exchange the key is an intractable problem in its own right. Asymmetric ciphers make use of a complex mathematical property of the underlying algorithms that allows two different keys to be used — one for encryption, and one for decryption. The key that is used for encryption is known as the **public** key, and is derived from the **private** key, which is used for decryption. This arrangement means that there is no need to exchange keys, as the public key cannot be used for decryption, so it doesn't matter if it falls into the wrong hands. The private key has to be carefully guarded, but this is relatively easy to achieve, as there is no need for anyone other than the rightful owner to be given access to the key.

One way that we can address some of the performance issues associated with encryption, yet still make use of the most robust encryption methods available, is to make use of symmetric ciphers for most encryption and asymmetric ciphers to facilitate the exchange of the symmetric keys. In fact it is a little bit more complex than this, because these mechanisms of key exchange are often not used to exchange the symmetric key itself, but are instead used to exchange a piece of information called the **pre-master secret**, which is exchanged in encrypted format using asymmetric encryption. This pre-master secret can be used in conjunction with public and private keys to generate a secret key that is used for the symmetric encryption. The means by which this is achieved is quite clever, but I am not going to attempt to explain it here because there isn't enough space to go into all of the mathematics and the detail of how the ciphers work, which would be required to understand how it is done.

Certificates

Certificates are a convenient place for storing and managing public keys. They also form the basis of authentication in digital communications, being the digital equivalent of a passport. Like a passport, they have to be issued by a recognized authority and contain certain things that allow the subject's identity to be confirmed and the certificate's validity to be ascertained. The former is achieved by including some identifying information on the subject, along with the subject's public key. The latter is achieved by certificates being issued by a recognized Certification Authority, and being **digitally signed** by that authority. The Certification Authority's signature is widely and publicly available for use in validating the certificate.

Digital signatures are based on **hash algorithms** (also called **message digests**), which produce a 'digested' version, called the **hash code**, of the text that they take as input. The hash function is **deterministic**, which means that the hash value that it produces is dependent on the text that it takes as input in such a way that any alteration in the text produces a significant change in the hash code. A good hash function is also a **one-way function**, meaning that the function cannot be derived from the hash value and the input text, and it is also **collision resistant**, which means that no two input values should produce the same hash value. Digital signatures are based on a special type of hash function that takes a key as input, as well as the original text. This means that the hash value is dependent on both the input text and the key, and therefore if you and I both sign some text using our own keys, the hash value produced will be different. In this sense digital signatures are slightly unlike real-world signatures, in that they will vary depending on the content that is being signed, which also makes them almost impossible to forge.

Certificates are fairly complex documents, and are usually presented and validated on behalf of the user without any human intervention. This has two ramifications:

- The certificates end up stored on computer, floppy disks, etc.
- It is impossible to track down copies of certificates if it becomes necessary to change or replace one

The first of these issues causes some problems in the wireless-world, which we will investigate later on. The second is addressed by means of **Certificate Revocation Lists** (CRLs). These are lists that are maintained by the Certification Authorities of certificates that have been issued, but that have become invalid for some reason or another. CRLs should be consulted before simply accepting a certificate as being valid.

Because of the large universal need for certificates, it is not feasible for a single organization to be responsible for the administration of all certificates, so there is a facility whereby certification authority can be delegated to other organizations. Any organization, theoretically, can act as a certification authority, and many organizations fulfill that capacity for certificates used internally, for example by employees. However, certificates that are valid in the public domain have to be certified by a recognized authority. Certificate chains make this feasible; by chaining certificates to the certificates that certify their authenticity a trail is built back to some authority that can be deemed to be acceptable.

WTLS

WTLS is the **Wireless Transport Layer Security** protocol. As can be ascertained by the name, it operates at, or more correctly just above, the transport layer in the OSI protocol stack. It is based on transport layer security (TLS), which is the de facto security implementation on the Internet. It works by establishing a session between a client and a server (which in the case of WTLS is the WAP gateway), during which it negotiates security parameters to be used to protect the session. These include the encryption protocols to be used, signature algorithms, public keys, pre-master secrets, or the exchange of certificates, depending on the capabilities of both the client and the server and the required level of security. The process of establishing a session is called the **handshake**. Once a session has been established all communications between the mobile device and the WAP gateway are encrypted, and therefore should be unintelligible if they are intercepted.

WTLS includes support for both a full handshake, with negotiation of all security parameters, and for a 'lightweight' handshake in which the security parameters of another session are reused. Support is also provided for session suspend and resume, which is useful in a wireless environment where reception quality is not always that good and where connections can easily be lost. The sessions can continue to exist despite a terminated connection and can be resumed on reconnection. Using this facility, it is possible to have sessions that last for days at a time.

The advantages of sessions that can continue to exist for days at a time must be weighed against the security implications of this feature. The longer the session remains valid for, the longer the secret keys remain valid for, and, presumably, the greater the number of messages exchanged that are encrypted using this key. This all provides material to someone wanting to crack the security protecting the session and compromise the messages. To guard against this, WTLS allows keys to be renegotiated periodically during a session. Renegotiating keys is not as computationally expensive as establishing the keys in the first place, so this is still more efficient that tearing down and re-establishing the session.

Another advantage of WTLS over TLS is that it operates over UDP. TLS requires a reliable transport protocol, in particular TCP, so it cannot be used over UDP. WTLS addresses this shortcoming, and also functions over WDP in the absence of UDP.

Certificates, for all of their usefulness, were not really designed with mobile devices in mind. WAP defines a new format of certificate that is optimised for storage on mobile devices and for transmission over constrained networks. These certificates still provide all of the functionality and security of their more heavyweight counterparts, but rely on the server to perform more of the processing under some circumstances.

WTLS therefore provides a comprehensive, optimised solution for both client and server based authentication using certificates, secure exchange of symmetric keys, anonymous and authenticated encryption of data, and support for digital signing of data.

There are three classes of WTLS implementation defined in the WAP specification. They are:

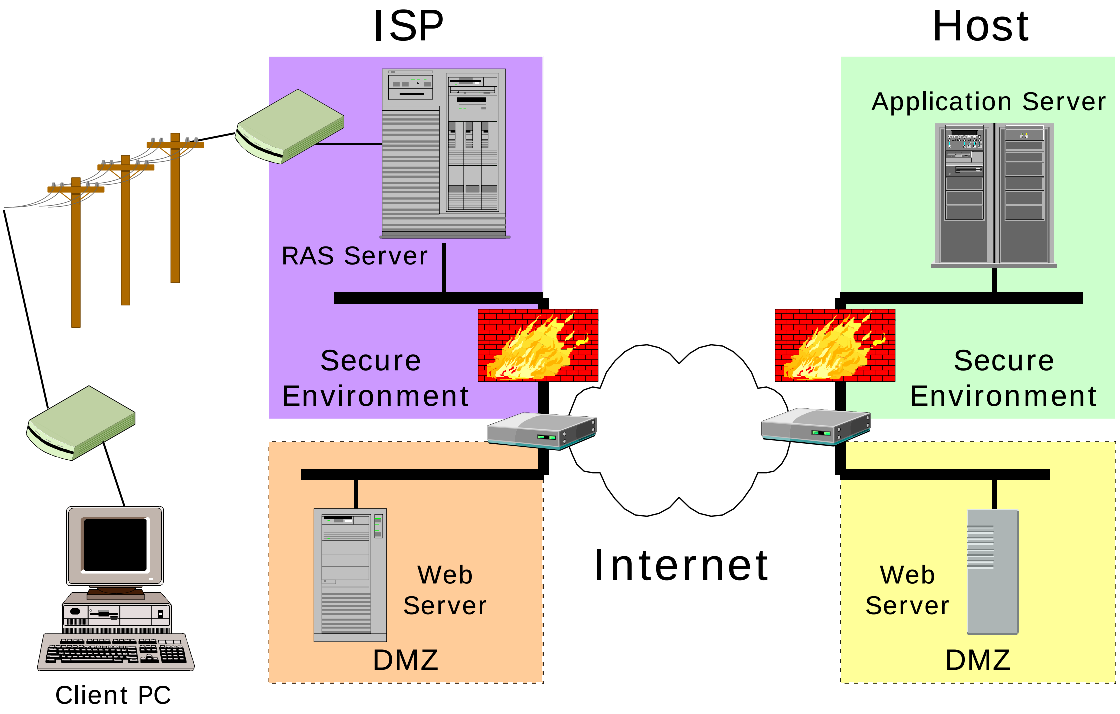
- ❑ Class 1: Anonymous key exchange with no authentication.
- ❑ Class 2: Certificate based server authentication. Server key is anonymous or authenticated, client key is anonymous.
- ❑ Class 3: Certificate based client and server authentication. Both client and server keys are anonymous or authenticated.

Communication Models

The best way to achieve an understanding of the merits of the implementation of security in the wireless environment is to compare it to the implementation of security in the fixed-wire world, that is, the Internet.

Internet Communication Model

A typical example of the Internet communication model is shown in the diagram below:



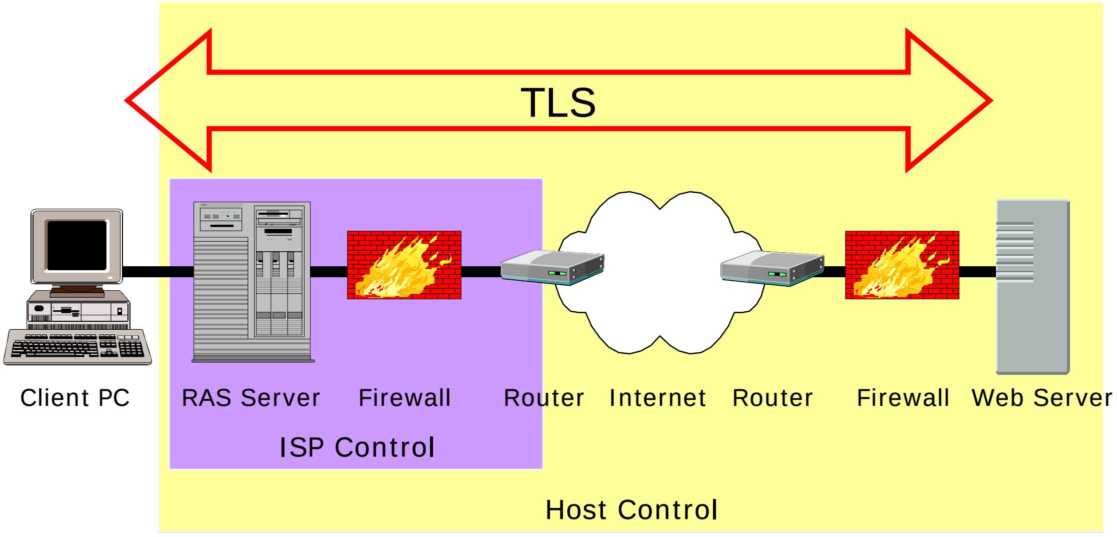
The Internet communication model assumes that a client PC connects to a server via an ISP dial-up connection. The client will be connected into the ISP systems over a PSTN or ISDN link, with PPP usually used as the bearer protocol.

The connection point on the ISP network is to a RAS server, which will perform certain functions on behalf of the remote client. In particular, the RAS server effectively acts as a proxy for the remote client, collecting network packets and forwarding them across the dial-up link. The RAS server is responsible for validating the client that is dialling in, and there are various means at its disposal to do that. The RAS server is typically on a secure part of the ISP network and thus provides the illusion to all other devices that the remote client is in fact also on the local network.

The ISP secure network environment is usually isolated from the Internet by means of a firewall of some sort. This firewall will attempt to regulate traffic that enters the local network, and protect the devices on the local network from malicious attacks over the Internet. The ISP may also choose to run one or more web servers and/or other facilities in a way that is more easily accessible to the public, and by extension also more vulnerable to attack. This area of the network is referred to as the **demilitarized zone** (DMZ), and is usually on a separate network segment from the secure area. Note that this is only one possible configuration for a network. Any particular implementation is likely to be far more complex and to be different in any number of ways.

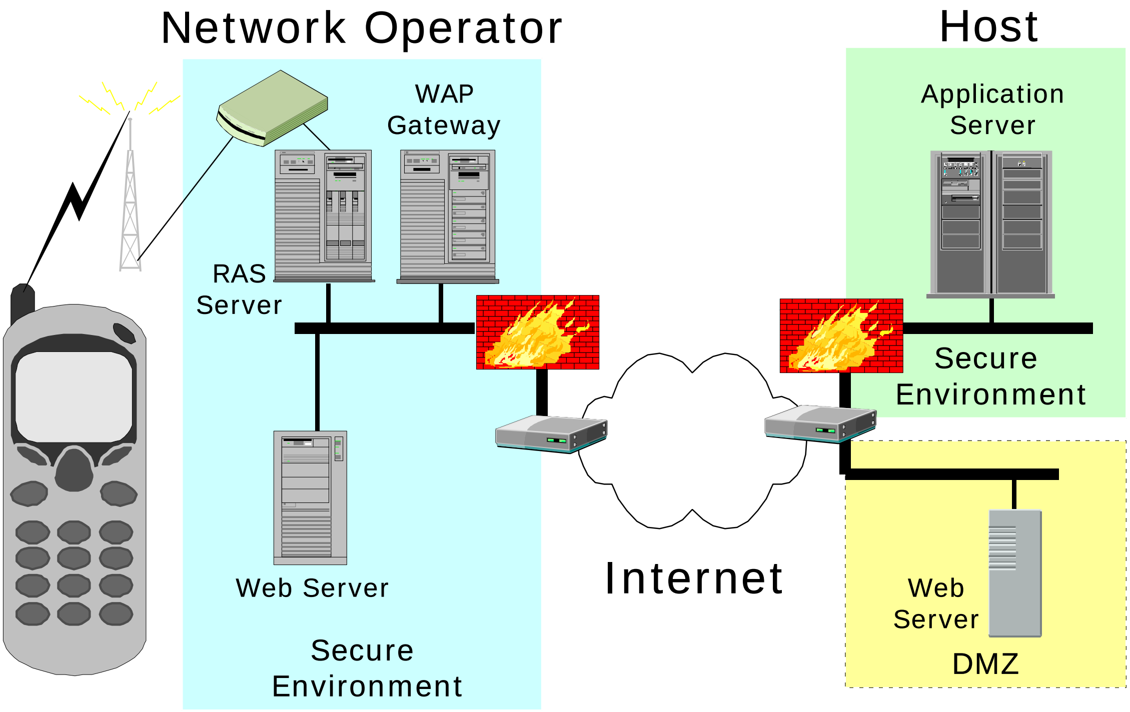
Access to the Internet is typically facilitated by one or more gateway devices, which are connected both to the ISP network and to some other network, possibly one run by one of the global Internet backbone providers. Any message entering the network across the gateway will be forwarded from gateway to gateway across the Internet, until it arrives at the destination network. It will then cross the gateway and enter the local network of the target host. In a way similar to the ISP, the host may also have a DMZ which houses the web server, with traffic entering the secure network filtered through a firewall. The firewall may only permit traffic originating from the web server to enter the secure network. On the network behind the firewall will reside any additional applications required to fulfil the request, and these will be used by the web server as required.

In examining the Internet model from the perspective of who controls or has the ability to influence the connection from a security point of view, it is apparent that the TLS connection exists between the client device and the web server. In effect this forms a tunnel between the client and server, and anyone penetrating this tunnel would not be able to decipher any messages intercepted. The ISP retains responsibility for the devices on its own network and for validating that the client is permitted to connect to the network in the first place, but has no ability to influence the TLS session. The extent of each parties influence is illustrated below.



Wireless Communication Model

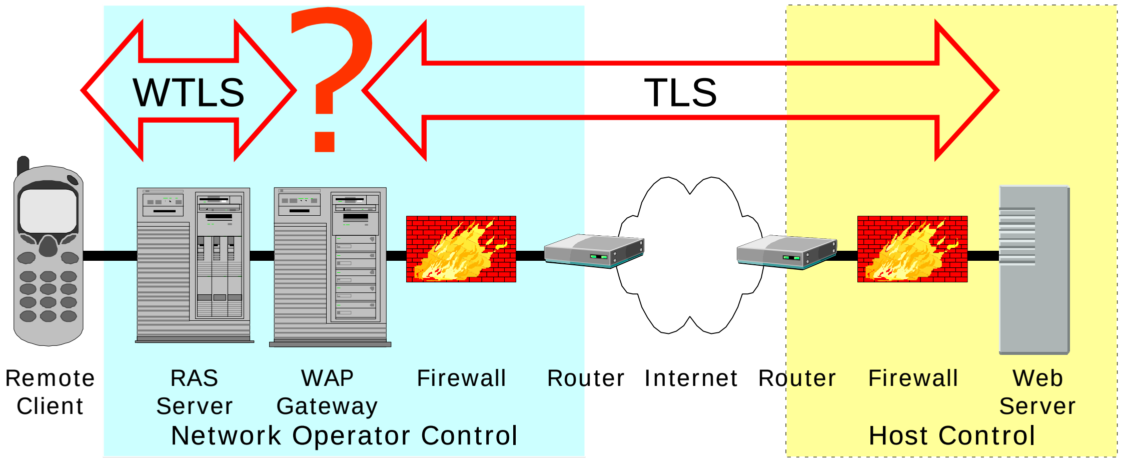
The wireless communication model is more complex because there are more ways in which the connection could be achieved. The model that we will examine at this point in time is one which many, possibly the majority of, connections that take place between the person-in-the-street and some WAP enabled web site will take place over. That is not to say that this is necessarily the best model from any particular point of view, just that many connections will be effected in this manner. This model is illustrated below:



In this model the remote client is a mobile device, but still dials into an RAS server on some network somewhere. This is likely to be an RAS server hosted and owned by the network provider, and is therefore likely to be on the network provider's own local network. The network provider will typically also host the WAP gateway, and a web server to provide access to the premium rate services that the network provider offers to their members. If access is required to services hosted on another server somewhere across the network, then the WAP gateway will act as a proxy for the client mobile device in establishing the required sessions with the remote host.

From the point of view of security, this scenario has various implications. WTLS is the security protocol that will be used to secure communications to and from the mobile device, but the mobile device's session is necessarily with the WAP gateway rather than the remote host's web server. At the gateway, the secure session terminates and all encrypted material is decrypted. Should there be a requirement for a secure session for communication with the web server, it will be established by the WAP gateway on behalf of the mobile device. The WAP gateway will use TLS to establish such a secure session. While TLS is obviously a robust security protocol, it remains a fact that the secure session is not between the mobile device and the web server. There are actually two secure sessions in play: one between the mobile device and the WAP gateway and the other between the WAP gateway and the web server. This means that there is a security gap, in which the data is not encrypted, at the WAP gateway.

This gap, and the span of control of the host server and network operator are illustrated below.



The host server's span of control is severely compromised in comparison to the Internet model. In fact, the host has absolutely no control over the security that exists between the mobile device and the WAP gateway. The host also has limited control over the TLS session between the WAP gateway and the web server, and will be limited to providing security that does not exceed a level determined by the network operator. This may or may not be adequate for the host.

WAP Security Issues

There are two issues with regard to security in the WAP environment. There are ways of addressing both of these issues, but they both remain issues that need to be addressed.

The Gateway

We have established that there is a security gap in the WAP model in the form of the WAP gateway. Because of the way that WAP works it is not feasible to do away with the gateway, so we need to establish to what extent it actually is a risk and what the alternative ways of addressing the risk are.

It can be argued that the WAP gateway is not actually a security risk because the gateway vendors are aware of the issue and therefore take steps to ensure that the process of decrypting from WTLS and re-encrypting into TLS cannot easily be compromised. Typical of the steps taken will be to ensure that the decryption and re-encryption takes place in memory, that keys and unencrypted data are never saved to disk, and that all memory used as part of the encryption and decryption process is cleared before being handed back to the operating system.

The first problem with all of this is that there are no standards of guarantees about these precautions. You have no way of ascertaining how robust your vendor's implementation actually is, and in the case of a gateway that is hosted by a network operator you may not even be able to tell whose implementation it is. One can also questions of the vendor's promises: in a heavily loaded server, how exactly does the gateway prevent the operating system from swapping memory pages out to swap space?

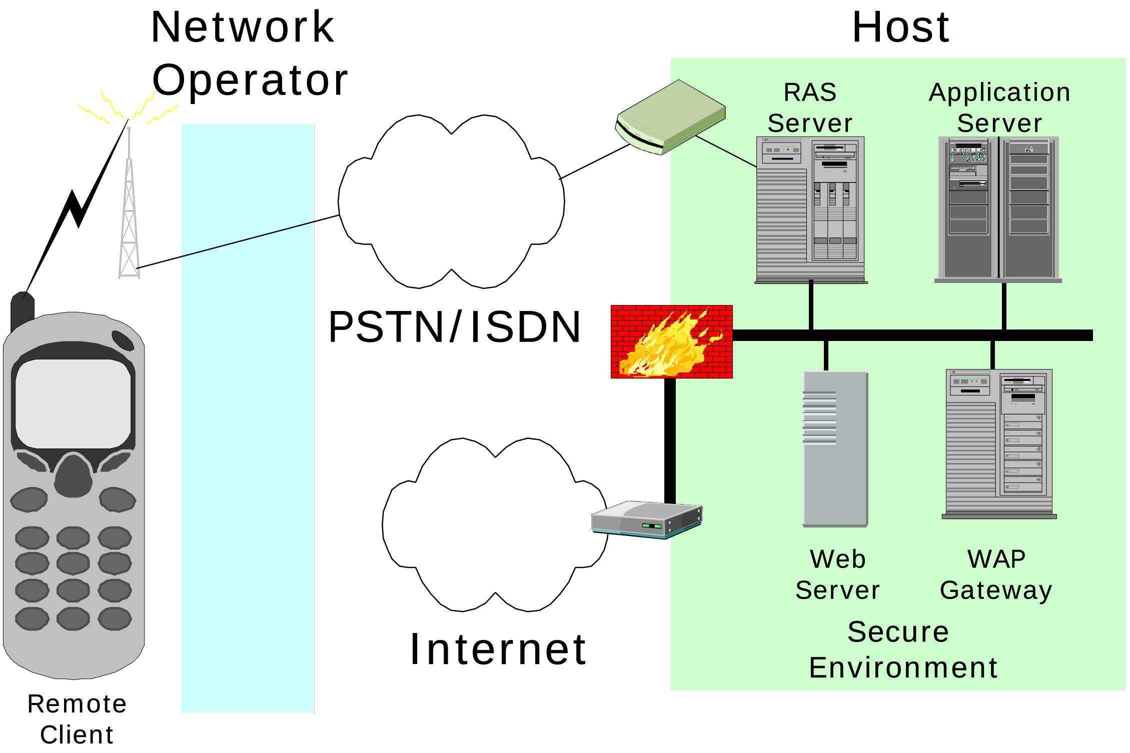
In a sense, saying that the vendors are aware of the issue and taking steps to address it is comforting. However, it must also be remembered that Microsoft are aware of the security exposures of the Internet Explorer browser and the Outlook mail client, but they continually take flack as a result of a seemingly unending stream of security loopholes and exposures that are constantly being discovered and exploited. I am sure that a vendor that has extremely competent programming staff and designers, and which implements their product only on a very secure operating system in a thoroughly secured environment under the control of extremely competent administrators, could provide a reasonably secure implementation. Still, I am equally sure that there is still an exposure around the gateway and that sooner or later it will become a target for hackers.

What you need to consider is how much of an exposure it is for the kinds of applications that you are developing. For some applications the risk-reward ratio, when compared to the cost of implementing a more secure solution, may be small enough that the vendor decides to take the risk. For others, where the risk by far outweighs any possible reward, there is no question that it is a complete show stopper.

If we accept that there is an exposure at the gateway, no matter how small or how hard the vendors work to protect the unencrypted data, the real question then becomes: who hosts the gateway? Whoever hosts the gateway has the responsibility for protecting it and the data that goes through it, and also has access (potentially, at least) to all of the data that goes through the gateway in unencrypted form.

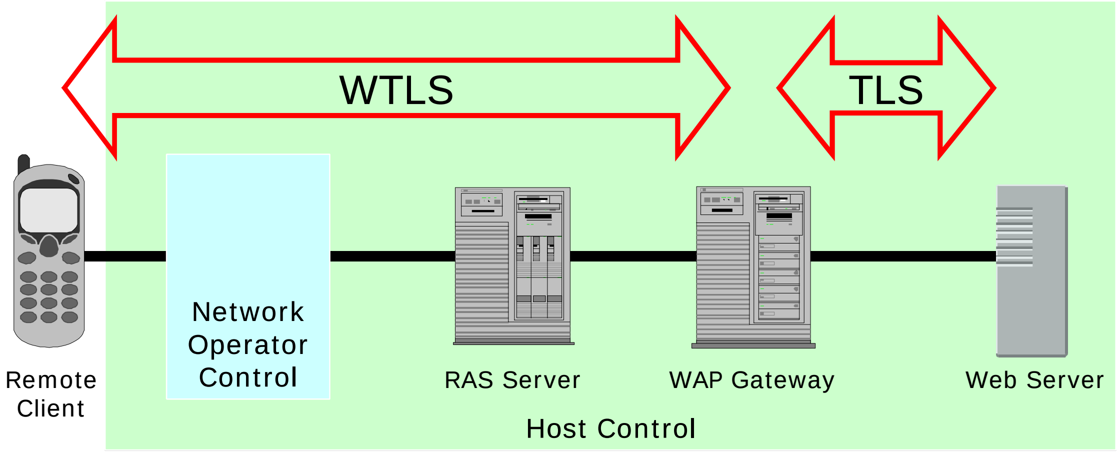
The good news is that it is entirely possible for you to host your own gateway, although before doing so you should consider the implications, in terms of cost and otherwise, of doing so. There are also two different architectures that can be implemented to facilitate hosting your own gateway, and each has different characteristics in terms of security and cost overheads.

The first model, which is shown in the following diagram, is probably only suitable if you want to provide access to a limited number of people who are not the general public, possibly employees:



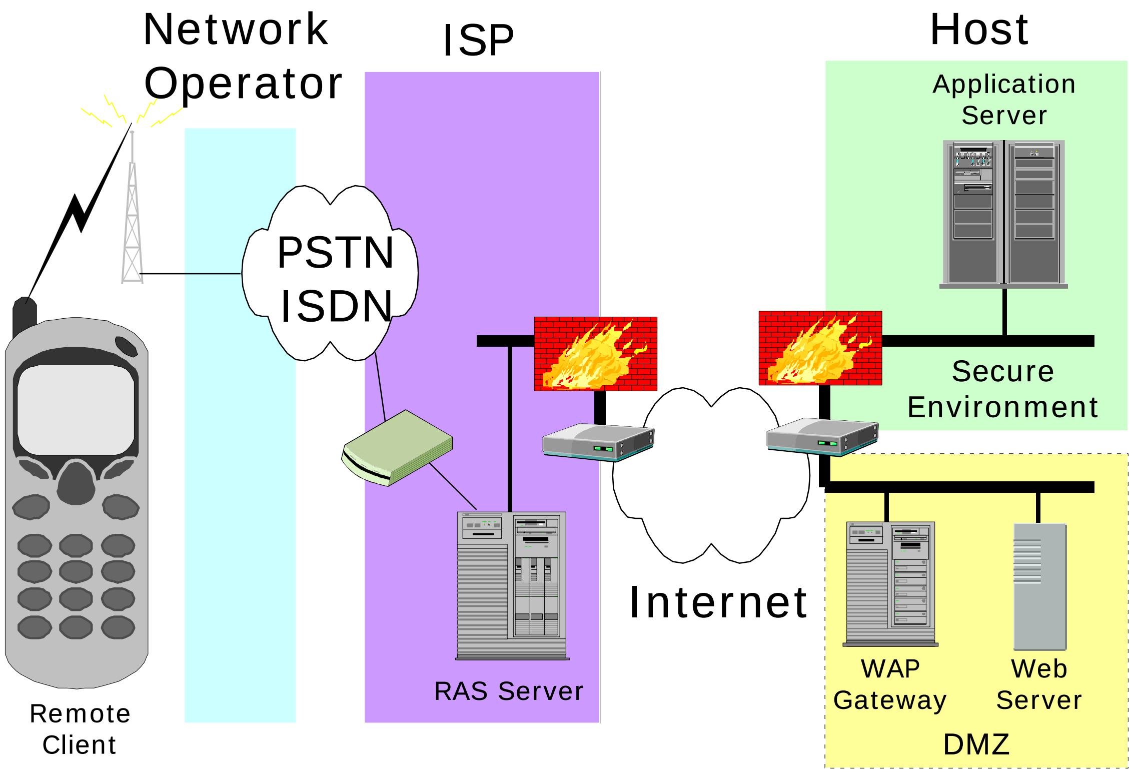
Here, security is absolutely paramount. In this scenario you would choose to establish an environment similar to any other highly secure dial-up environment. You would establish a bank of dial-up modems connected to one or more RAS servers on your local network. You would be responsible for establishing, maintaining and administering the environment, including details such as dial-up security (possibly through RADIUS or similar). You would then be able to strictly control who has access to the gateway, when this access is possible, and via what telephone numbers. You could implement dial-back to a limited set of numbers, control the IP addresses available, issue and use your own certificates for authentication, and anything else that would contribute to your secure environment. All of the relevant servers would be a secure segment of your local network, and access to and from the Internet may or may not be available. If it is available it will almost certainly be protected by one or more firewalls.

In this environment, as illustrated below, the network operator's sphere of influence is almost non-existent:



The network operator is restricted to connecting the call and has no influence over any of the communications between the client and server. The network operator does not have a gateway that participates in the communication process, and has no role to play with regard to security. The mobile device establishes a WTLS session that tunnels through the RAS server to the gateway, and a TLS session from the gateway to the web server, all on your own secure network.

The second model eliminates the need for the modems and RAS server by making use of the services provided by an ISP. The diagram below shows this model:

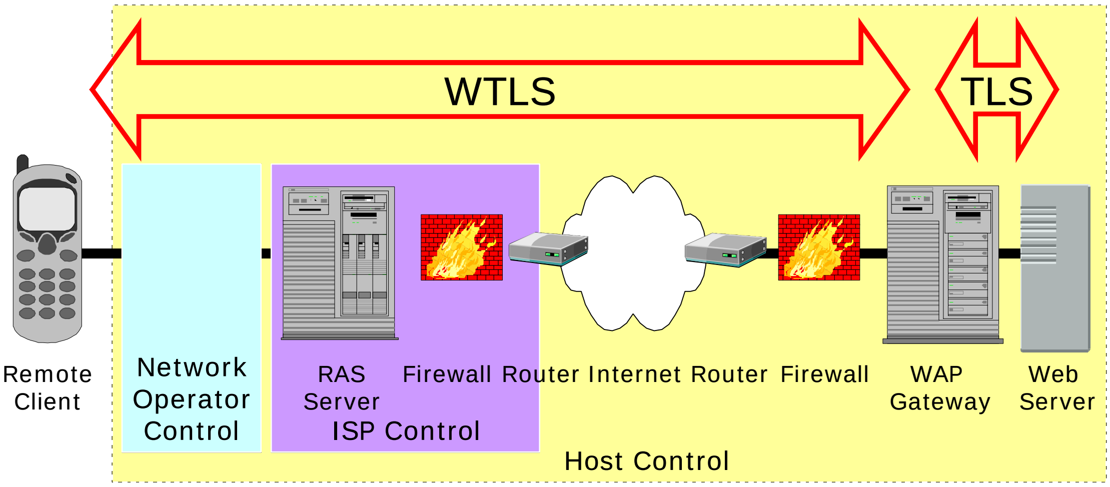


This model is in fact very similar to the Internet model, although there are some differences. The remote mobile device will establish a dial-up connection with the ISP's RAS server through a modem hosted by the ISP. The network operator is restricted to connecting the call and has no further influence on the session or the security environment. The RAS server at the ISP acts as a proxy for the mobile device on the ISPs network, and provides all the services that it would to a fixed-wire dial-up client. The ISP network is connected to the Internet via a gateway and is protected by a firewall.

The host's environment would usually be similar to an environment for access by fixed-wire clients over the network. The major difference would be that the host would have a WAP gateway available on the network, typically in the DMZ. Any secure connection from the mobile device would establish a secure session that tunnels through the ISP's RAS server to the WAP gateway. The WAP gateway would then establish a secure TLS session through to the web server, which would make use of services on the application servers hosted on the secure network behind the firewall.

In this scenario we are making use of WTLS in a similar way to a Virtual Private Network (VPN), in that the mobile device establishes a secure tunnel through to the target network. In the case of a VPN, the tunnel is typically to the router on the network, although it doesn't have to be, whereas in this model the 'VPN' tunnels to the WAP gateway. You will need to examine the security requirements of your application to determine whether WTLS provides a secure enough 'VPN' for your application.

The other thing to be aware of in this model is that the WAP gateway is typically on the DMZ, which means that it is not as heavily protected as it would be if it were behind the firewall on the secure network segment. This makes it more vulnerable to attack by hackers. If there is little chance of the WAP gateway being targeted then this is probably not an issue, but for a large retail bank, for example, where the gains to be had from cracking the gateway may be significant, it may present a temptation. On the other hand, if you have to provide public access to your WAP gateway then there is little in the way of feasible alternatives, unless you want to become a network provider in your own right. The span of control of the network operator, ISP and host are shown in the diagram below:



For almost all applications that have security requirements that prohibit the use of a network provider's gateway, one of these two models will almost certainly be sufficient. The trick is to match the trade-offs, in terms of cost and overheads, against your security requirements and risk to achieve an optimal solution.

User versus Device

The second issue that is worth considering with mobile devices, and which is not really a consideration for fixed-wire devices, is the issue of who or what is being authenticated by the certificate. I mentioned previously that a certificate is a reasonably large and complex thing, certainly too complex to type in each time it is required. The result is that the certificate usually ends up being held on your computer, often without you even being aware that it is there, and the system will take card of presenting and validating certificates as and when required.

While this is very convenient, it does have some security implications, in that anyone who gains access to your computer can make use of your certificates. The prerequisite is for the person to gain access to your computer. In many cases this is not that easy to achieve, requiring breaking and entering or something similar.

Mobile devices are different in that they are mobile and are therefore carried around. This also leads to them being lost, left on trains, and so on. In 1999, Railtrack, the company responsible for the rail network in the UK, announced that for the first time this century the umbrella has been overtaken as the most popular item to leave on a train — by mobile phones. This gives us a feeling for the magnitude of the problem. Clearly, where access to data or services has to be strictly controlled it will not be an acceptable solution to store certificates on the phone if those certificates provide access to data and services.

The most immediate way of tackling this problem is to accept that the certificate is going to be stored on the phone, and the phone may be lost. The certificate is still made use of to validate that the mobile device is entitled to access the network, which at least serves to eliminate all of those mobile devices that do not have the required certificate. Once the mobile device is reported missing the certificate can be placed on a certificate revocation list to ensure that it does not provide access in the future.

To further validate that the current user of the authenticated device is the rightful user you can make use of a variety of systems, which vary in their complexity and robustness from a simple PIN number through to a SecureID token. While it is easy to dismiss a PIN as being inadequate, pause to remember that almost all of us make use of automated teller machines, and in doing so daily rely on simple PIN numbers to protect our financial assets. Of course when asking users to enter PIN numbers on a mobile device the necessary precautions must be taken, such as masking the numbers with asterisks, and so on.

Future

It is always difficult and risky to gaze into the future and predict what is coming down the line, but it is also necessary to make educated assessments of the current technology and what is likely to be addressed in the near future. I will now attempt to do just that.

WTLS

WTLS, being based on an established and stable standard, is unlikely to change significantly or fundamentally for the foreseeable future. I expect that most of the changes in the next few releases will be oriented towards clarifying some issues in the specification and general 'housekeeping'. The 1.2 specifications did this, and added some advice about guarding against certain types of attacks. All of this information is only of relevance to people who are developing their own WTLS implementation, and also much of the information dates very quickly, so I would expect it to be refreshed in just about every release. I do not, however, anticipate any major changes unless a major security exposure in one of the ciphers is identified.

End-to-End Security

The WAP Forum has made it clear that they are aware of the issues around the security gap at the WAP gateway. They have also make it clear that they intend to plug the gap by providing an end-to-end security standard in a subsequent release. There have been hints that they would attempt to address this through changes to WTLS, but I think that this is marketing rather than technology speaking. The issue does not arise because of any weakness in WTLS and is caused solely by the position that the gateway fulfils in the WAP communications chain. In order to address the issue, either the gateway has to be eliminated or some other solution has to be implemented, probably at a higher level in the protocol stack. The WAP Forum has also indicated that the WMLScript Crypto library may be extended in the future to include cryptographic functions. At this point in time there is only a function that supports signing data. To my mind, it seems logical that the way to implement end-to-end security is by means of encryption functionality at the application level. A necessary prerequisite for this, however, will be the capability of mobile devices to deal with the processing loads associated with encryption functions. Part of the solution to this problem may actually lie in the WIM.

WIM

The Wireless Identity Module specification is new in the WAP 1.2 specification. It provides a means to offload the storage of keys and of cryptographic functions onto what is described as a tamper proof device. This is basically a smart card, although it could also be a SIM. The specification covers only the low level capabilities of a WIM in the current specification, and doesn't present an API for making use of a WIM when present, although I expect that an API and framework will be provided in a future release. The introduction of the WIM could help to address the issues around authenticating the device as opposed to the user.

Conclusions

There has been a lot of fuss about security in the WAP world, some of it justified, but most of it being misinformation and misunderstanding. I have often heard it observed that WAP 1.1 does not include security. This is an example of the misinformation that has been around in the industry: WTLS was part of WAP 1.1 and is almost unchanged in WAP 1.2. Security has been there all the time. What is true is that not all vendors have implemented all parts of the specification, and WTLS has often not been implemented at all or has only been implemented at class 1. This will be resolved in time, as vendor's products become more mature and robust, and as the public need for robust security implementation forces vendors to include security in their product offerings.

Even if your WAP gateway does not include WTLS, a WTLS gateway can be obtained from some reliable security solution vendors, like Baltimore Technologies, which will sit on your network between the mobile device and your WAP gateway to provide a WTLS implementation. This type of solution is only feasible if you are hosting your own WAP gateway.

WAP can and does provide a robust, secure environment in which an organisation can conduct m-commerce or communicate securely. Attention does need to be paid at this stage to the specifics of the implementations, so I would advise a thorough evaluation before committing to a particular vendor's implementation. However, there are robust products our there that you can use to implement a secure environment.