Wired versus Wireless Security:
The Internet, WAP and iMode for E-Commerce

Paul Ashley, Heather Hinton, Mark Vandenwauver
IBM Software Group – Tivoli
{pashley, hhinton, mvanden} @ us.ibm.com

Abstract

The perceived lack of security in the wireless environment has delayed many initiatives in providing access to e-commerce applications from wireless devices. Many organizations are skeptical that the same kind of security protections that they are used to in the current Internet (wired) e-commerce environment are also available for wireless transactions. In this paper we will show that these perceptions are misplaced. We describe the security properties and mechanisms available for Internet (wired), WAP based and iMode e-commerce. We find that both WAP and iMode provide excellent security features and are geared to provide other security provisions over and above those commonly available in a wired environment.

1 Introduction

There is a common perception that wireless environments are inherently less secure than wired environments. Reports of phone masquerading and phone call tapping in mobile wireless environments have led many to believe that this is not an environment conducive for e-commerce [1]. While this was certainly true in the past, the wireless industry has been working hard at providing security protections strong enough for real mobile-device based e-commerce. In this paper, we focus on two “flavors” of wireless service, as provided by the WAP Forum and by NTT DoCoMo of Japan. WAP is the Wireless Application Protocol, “an open, global specification that empowers mobile users with wireless devices to easily access and interact with information and services instantly” [2]. iMode is a proprietary mobile ISP and portal service from NTT DoCoMo, Japan [19].

The WAP Forum is an industry association of over 500 members “that has developed the de-facto world standard for wireless information and telephony services on digital mobile phones and other wireless terminals”[2]. The primary goal of the WAP Forum “is to bring together companies from all segments of the wireless industry value chain to ensure product interoperability and growth of the wireless market” [2]. The focus of the current and recent past work by the WAP Forum has been to ensure mobile devices are sufficiently secure to allow e-commerce transactions of real value to occur. The iMode effort from NTT DoCoMo in Japan [13] focused first on market penetration with insecure, or vanilla type, services and handsets. Rather than adopt or develop a new approach to wireless transactions, DoCoMo adopted the Internet model and protocols. Security was added after the business case for wireless transactions had been conclusively demonstrated.

Both the WAP Forum and iMode are adopting security in a “staged” approach, although the WAP Forum has been more aggressive in their specification and adoption of security functionality and requirements. In the latest class of secure wireless protocols from the WAP Forum, client-side certificates are specified and used as part of client-side authentication and non-repudiation services. The current release of iMode allows for end-to-end Secure Sockets Layers (SSL) [7] with server-side authentication; client-side certificates are identified as future work by DoCoMo [16, 19].

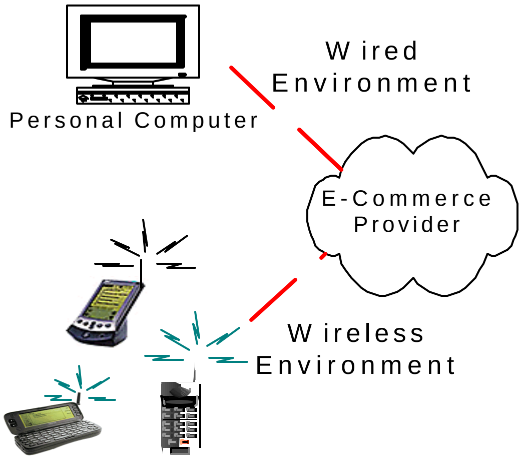

Figure 1: E-Commerce Security

Figure 1 shows the environment that this paper is concerned with. Users for some time have been able to access e-commerce (web sites) from traditional wired browsers (Netscape, Internet Explorer, and so on). Users are also able to access e-commerce sites from new wireless devices such as PDAs and mobile phones. This paper examines the security services that are provided at the user's browsing device, be it a traditional wired browser or a newer wireless device. The paper also focuses on security of the data in transmission from the user's browser to the e-commerce web site. Security issues related to the web site (such as storage of credit cards) are outside of the scope of this paper.

In order to provide security for e-commerce transactions in both the wired and wireless world, it is necessary to provide at least the following services [3]:

- *User authentication* – Provides the system proof that a user is who they claim to be.
- *Data authentication* – consists of two sub-services: data integrity and data origin authentication. With data integrity the receiver of data can be convinced that the data was not changed in transit. Data origin authentication proves to the receiver that data actually did come (originate) from the stated sender.
- *Data confidentiality* - Data confidentiality protects against disclosure of any data while in transit and is provided by encryption of data.
- *Authorization* - Authorization is the act of determining whether an (authenticated) entity has the right to execute an action. This is the responsibility of the system providing the e-commerce transactions/services.
- *Audit* - An auditing service provides a history of actions that can be used to determine what (if anything) went wrong, when it went wrong, and what caused it to go wrong. Audit services can also be used to pinpoint the last known "good" state of information.

In addition to these well-known services, it is increasingly common to expect that e-commerce transactions provide *non-repudiation*. Non-repudiation is proof that the user did in fact initiate a transaction. Non-repudiation is usually implemented by requiring a user to *digitally sign* a transaction. Digital signatures are unique to users and are used to provide proof that a user was involved in a given transaction. If the process of binding a user's name to the signing key (to create a digital certificate) meets certain security and legal requirements, a digital signature can be "strong enough" to provide non-repudiation of the user's actions at a later stage.

Non-repudiation is required for those transactions that are considered to be "out-of-economy" transactions, such as bill payments from a user's account to an account owned by a different entity. An "in-economy" transaction, such as a transfer from a given user's checking to their savings account, is fairly easy to unwind (if necessary). That is, recovering from a mistaken or fraudulent in-economy transaction is easily handled. Unwinding an out-of-economy transaction is a much more difficult process. Rather than provide a mechanism to unwind such transactions, most enterprises would choose to rely on non-repudiation proofs as evidence that a user did in fact agree to a transaction (so that unwinding is not necessary).

This paper examines the security of traditional Internet (wired) environments and new mobile wireless environments based on WAP and iMode in the context of their suitability for e-commerce. In Section 2 we provide an overview of the networking environment for wired and wireless protocols (WAP and iMode). Section 3 gives a high-level introduction to the WAP specification and iMode. The security functions implemented by WAP and iMode are discussed in Section 4. Sections 5 and 6 give an analysis of WAP and iMode security and relates this to wired security. The paper finishes with our conclusions.

2 Networking Environments

In this section we briefly describe the networking environments common to the wired and wireless worlds. We also describe the typical protocols and security requirements seen in the traditional wired world. This will provide a basis for comparison with the wireless world capabilities later in this paper.

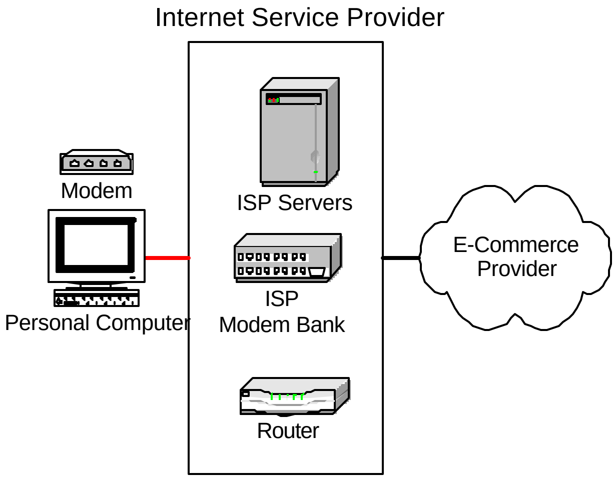

Figure 2: Traditional Wired Environment

2.1 Traditional Wired Environment

Figure 2 shows the traditional wired environment for a "home" user accessing an e-commerce provider (also known as the back-end or the Enterprise) from a browser. The user connects to their local Internet Service Provider (ISP). The connection can happen over telephone lines, a cable television network, …. The ISP provides the user access to the Internet and routes the user's requests. The networking protocol used from the user's browser to the e-commerce provider is TCP/IP [4].

2.2 WAP Networking Environment

WAP, or Wireless Application Protocol, is an industry initiated world standard [2] that allows the presentation and delivery of information and services to wireless devices such as mobile telephones or handheld computers. The major players in the WAP space are the Wireless Service Provider (WSP) and the Enterprise. The Wireless Service Provider is the wireless equivalent of an Internet Service Provider (ISP). The role of the WSP is to provide access to back-end resources for wireless users. The WSP provides additional services because wireless users must transition from the wireless to wired environments (unlike an Internet environment where the user is already "on" the Internet). The WSP's space contains a Modem Bank, Remote Access Service (RAS) server, Router, and potentially a WAP Gateway.

Figure 3 illustrates what may be considered the "traditional" WAP networking environment. This environment is analog to the wired environment, where all "connection-type" services are provided by the Wireless Service Provider. Much of this functionality overlaps with functionality currently provided by the telecommunications industry. We anticipate that the majority of this functionality will be implemented and managed by Telecommunication Companies such as Wireless Service Providers.

The WSP handles the processing associated with the incoming WAP communications, including the translation of the wireless communication from the WAP device through the transmission towers to a Modem Bank and Remote Access Server (RAS) and on to the WAP Gateway. The Modem Bank receives incoming phone calls from the user's mobile device, the RAS server translates the incoming calls from a wireless packet format to a wired packet format, and the Router routes these packets to the correct destination.

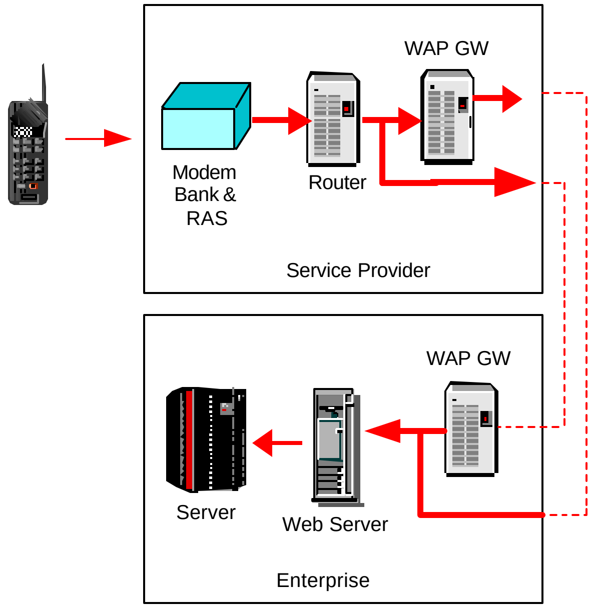

Figure 3: "Traditional" WAP Networking Environment

The WAP Gateway is used to translate the WAP protocols (protocols that have been optimized for low bandwidth, low power consumption, limited screen size, and low storage) into the traditional Internet protocols (TCP/IP). The WAP Gateway is based on proxy technology. Typical WAP Gateways provide the following functionality:

- Provide DNS services, for example to resolve domain names used in URLs.
- Provide a control point for management of fraud and service utilization.
- Act as a proxy, translating the WAP protocol stack to the Internet protocol stack.

Many Gateways also include a "transcoding" function that will translate an HyperText Markup Language (HTML) page into a Wireless Markup Language (WML) page that is suited to the particular device type (such as a Nokia 6120 or Motorola Timeport mobile phone).

The Enterprise space contains the back-end Web and application servers that provide the Enterprise's transactions.

While it seems "natural" for the Wireless Service Provider to maintain and manage the WAP Gateway, there are circumstances under which this is not desirable. This is due to the presence of an encryption "gap", caused by the ending of the Wireless Transport Layer Security (WTLS) [6] session at the Gateway. The data is temporarily in clear text on the Gateway until it is re-encrypted under the SSL session established with the Enterprise's web server. This problem is discussed in detail in Section 5.3.

In such cases, the WAP Gateway should be maintained at the Enterprise, as shown in Figure 4. Maintaining a WAP Gateway does not require any telecommunications skills; the Gateway receives regular UDP packets. The problem with this solution remains the absence of the DNS client at the mobile device, which would require the storage of profiles for every target on the mobile device. This also requires that the Enterprise set up a relationship with the Service Provider whereby all incoming packets destined for the Enterprise (identified by IP address) are immediately routed by the WSP directly to the Enterprise and are never sent to the WSP's Gateway.

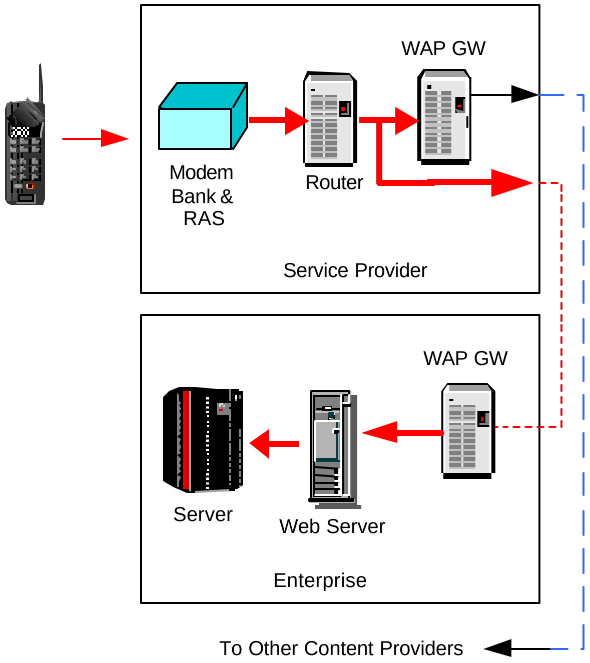

Figure 4: e-Commerce Specific WAP Networking Environment

We refer to this environment as an "e-commerce specific" environment as it is within the realm of e-commerce transactions that this type of architecture will be required.

2.3 iMode Networking Environment

iMode is the proprietary protocol of NTT DoCoMo of Japan. iMode provides Internet service using Personal Digital Cellular-Packet (PDC-P) and a subset of HTML 3.0 for content description [19]. iMode allows application/content providers to distribute software (Java applets) to cellular phones and also allows users to download applets (e.g., games). iMode uses packet-switched technology for the wireless part of the communication and is carried over TCP/IP for the wired part of the communication.

Packet switching systems send and receive information by dividing messages into small blocks called packets and adding headers containing address and control information to each packet. This allows multiple communications to be carried on a common channel. This allows for efficient channel usage with low cost.

"DoPa," which is DoCoMo's dedicated data communications service, offers connections to LAN and Internet service providers by applying this principle of packet switching to the wireless section as well. The mobile packet communications system has a network configuration in which the packet communications function is added and integrated into DoCoMo's Personal Digital Cellular (PDC) which is the digital system for portable and automobile telephones.

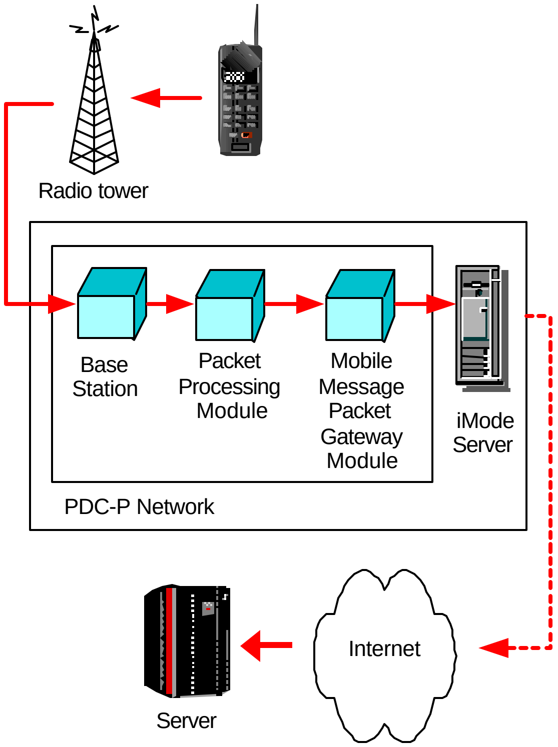

Figure 5: DoCoMo iMode Wireless Networking Environment

DoCoMo has developed a data transmission protocol specific to iMode. This protocol is used with DoCoMo's PDC-P system. Connections between the iMode server and the Internet use generic TCP/IP technology. The PDC-P network includes a mobile message packet gateway (M-PGW) to handle conversions between these two protocol formats.

The iMode server is a regular Web server. It can reside at NTT DoCoMo or at the Enterprise. DoCoMo has been acting as a portal and so "normally" maintains the iMode server. For future implementations with advanced security requirements, it is possible to host the iMode server at the Enterprise.

The web site http://ww.kyoto-bauc.or.jp/i (requires a Kanji-enabled browser) is a standard iMode site, while the site https://kabu.com is an SSL protected iMode site.

3 The Wired and Wireless Protocols

This section provides a brief introduction to the two wireless protocols, WAP and iMode. We also discuss the similarities and differences in these protocols. We do not include a discussion of the wired protocols, SSL/TLS [7], TCP/IP [4] and so on, for which there are plenty of excellent references available.

3.1 The WAP Specification

The WAP specification defines an open, standard architecture and a set of protocols for the implementation of wireless access to the Internet.

The WAP specification includes among others [1]:

- *An XML-type markup language, Wireless Markup Language (WML):* WML and WMLScript provide a set of markup tags appropriate for wireless devices. WML content is accessed on a (traditional) web server over the Internet using standard Hypertext Transfer Protocol (HTTP 1.1) requests.
- *A "microbrowser" specification:* This defines how WML and WMLScript are interpreted at the wireless handset.
- *A lightweight protocol stack:* Designed to minimize bandwidth requirements, this allows different wireless networks to run WAP applications. Wireless Session Protocol is the equivalent to HTTP in a compressed format.
- *Framework for Wireless Telephony Applications (WTA):* This provides access to traditional telephony services (such as Call Forwarding) through WMLScript.
- *Provisioning:* This allows Service Providers to reconfigure mobile telephones from a distance using Short Messaging System (SMS) (note that SMS is a GSM standard) [5].

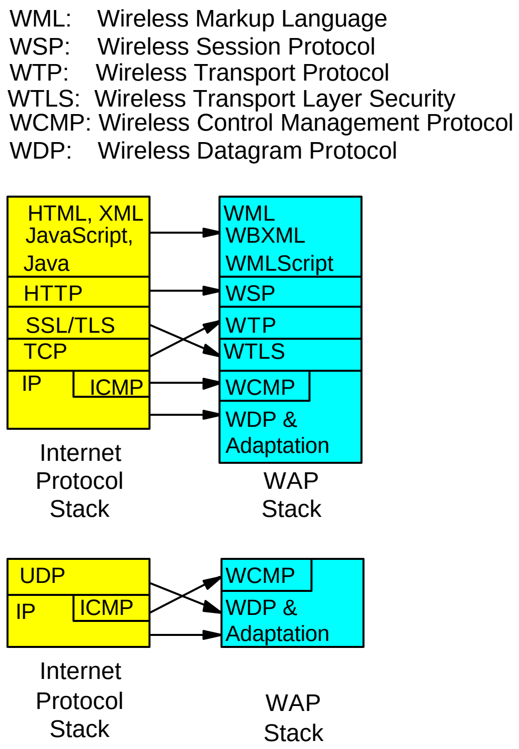
WML: Wireless Markup Language
WSP: Wireless Session Protocol
WTP: Wireless Transport Protocol
WTLS: Wireless Transport Layer Security
WCMP: Wireless Control Management Protocol
WDP: Wireless Datagram Protocol
HTML, XML
JavaScript,
Java
HTTP
SSL/TLS
TCP
IP
ICMP
WML
WBXML
WMLScript
WSP
WTP
WTLS
WCMP
WDP &
Adaptation
Internet
Protocol
Stack
WAP
Stack
UDP
IP
ICMP
WCMP
WDP &
Adaptation
Internet
Protocol
Stack
WAP
Stack

Figure 6: WAP Protocol Stack Specification

Figure 6 shows the relationship between the traditional Internet protocols and the WAP protocols.

3.2 The iMode Specification

iMode is a proprietary service currently only offered in Japan and can not be made readily available on any other service carrier's network [16]. The iMode specification is a proprietary protocol of NTT DoCoMo of Japan. As such, the details of the protocol and specification are not publicly available. The information that we are reporting in this paper is based on personal experience with iMode and the resources sited in this paper.

The protocol stacks used by iMode have been reported in public forums (at RSA 2001 [15] and at IETF 47 [19]). Figure 7 shows the iMode protocol stacks. iMode Security is provided at the transport layer using SSL/TLS and is based on the security provided by these Internet protocols. The TL and LAPD-M protocols are standards of the Association of Radio Industries and Business (ARIB) [20].

iMode uses "compact HTML", or "cHTML" for representing on-line (on-air) content. The structure of cHTML means that a user can also view "traditional" HTML pages (although cHTML pages look better). This is in contrast to WAP, where HTML pages must be translated to WML.

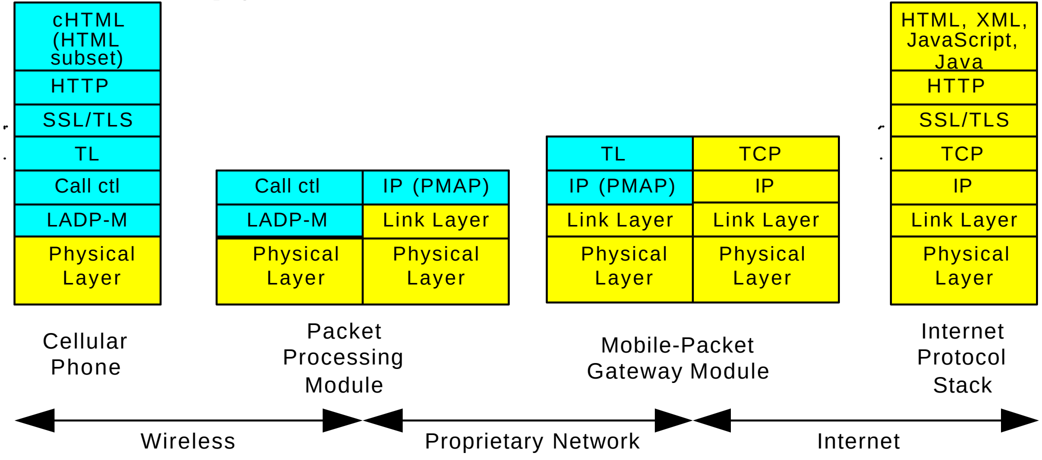

Figure 7: iMode Protocol Stacks

4 Wired and Wireless Security Features

In this section we provide a brief overview of the security features of each of the wired and wireless protocols.

4.1 Wired Security

The majority of e-commerce applications run over TCP/IP protocols. Typically, resources accessed by a user are HTML pages or Java-based resources. These resources are accessed using the HTTP. If the user requires access to a sensitive resource, they will be required to establish an SSL session with the back-end. This session establishment will be prompted by the back-end and may be "server-side" or "mutually authenticated" SSL.

A "server-side" SSL session means that the server presents a digital certificate to the user's browser as proof of the server's identity (providing a binding of the public key presented with the name contained in the certificate). A "mutually authenticated" SSL session means that both the user's browser and the server present to each other a digital certificate to prove the authenticity of their identities (as bound to the public keys presented) to the other party. Most back-end systems will (at a minimum) require a server-side SSL session. This is the "easiest" type of SSL session to establish as it does not require the user to obtain a certificate (thereby passing the "ease-of-use" test). It is foreseen that in the future it will become more common for a user to be required to have and present a certificate for mutually authenticated SSL session establishment.

A user will be required to obtain a certificate from a Certificate Authority (such as VeriSign). This certificate will be stored in a certificate store at the user's browser. Certificate management is handled by the user's browser. A user will be required to enter a password to "unlock" the certificate store and provide access to a certificate as part of the establishment of a mutually-authenticated SSL session. This type of certificate can be referred to as an "authentication certificate" as it contains the key-name binding for the keys used for authentication purposes (Certificates may also be used to provide a binding of user name to signing key. It is recommended that a user have separate signing and authentication keys.).

A discussion of how certificates are managed and distributed in the Internet world is beyond the scope of this paper. For more information on this topic, the interested reader should refer to [3].

4.2 WAP Security

There are several components to the security features available with the WAP specification. These include the WTLS protocol for securing communications, WAP Identity Module (WIM) smartcards for storing user certificates, and functions such as `Crypto.signText()` to allow for signing of WAP transactions. In this section we briefly describe these features and the implications of their use.

The WTLS Protocol. WAP communications are protected using the WTLS protocol [6]. WTLS provides entity authentication, data confidentiality and data integrity. It is based on the IETF SSL/TLS [7] protocols. WTLS is used to secure communications between the WAP device and the WAP Gateway. There are three different classes of WTLS:

- *Class 1*: This type implements unauthenticated Diffie-Hellman key exchange to establish the session key.
- *Class 2*: This enforces server side authentication using public key certificates similar to the SSL/TLS protocol. The WAP Gateway uses a WTLS certificate (a particular form of X.509 certificate compressed to save on bandwidth).
- Class 3: Clients implementing this level are able to authenticate using client side certificates. These certificates are regular X.509 format and can be stored either on the client or on a publicly accessible server (in this case a pointer to the certificate will be stored on the mobile device).

Early WAP devices only implement WTLS Class 1. This level of security is insufficient [8] and should not be used for e-commerce transactions. Devices supporting WTLS Class 2 are currently available. These devices are being used in several read-only access and in-economy banking applications in Europe and the UK [14]. WTLS Class 3 devices are not yet generally available, although there have been announcements of trials of phones that support Class 3 [9].

The WAP Identity Module. To facilitate client side authentication, new generation WAP phones will provide a WIM [10]. The WIM will implement WTLS Class 3 functionality. The WIM has embedded support for public key cryptography – RSA [8] is mandatory and Elliptic Curve Cryptography [8] is optional. An example of a WIM implementation is a smart card. In a wireless phone, it could be part of the Subscriber Identity Module (SIM) card (in the case of GSM [5]) or an external smart card (referred to as a WIM card). In the case of a combined SIM-WIM card this is typically called a SWIM card.

A WIM will be configured (at the manufacturer) with two sets of private-public keypairs (one for signing and one for authentication) and two *manufacturer's* certificates. Note that the manufacturer's certificates bind the manufacturer's name with the public keys configured on the WIM. Thus *all* WTLS sessions established through a WIM and a WAP Gateway will use the same public keys for the initial session negotiations. Each session will (potentially) include a different certificate for this key. Manufacturer's certificates are used by default until a user has registered additional certificates with back-end Enterprises.

The WIM is also able to store some number of user certificates or user certificate references, such as a URL-based reference (Certificate references, such as a URL used to access a back-end certificate store are the preferred means of storing certificate information. Because the user certificates are X.509 based, storing full X.509 certificates on the WIM will very quickly use up all available storage space). A user will be required to "enroll" or otherwise register a certificate at each Enterprise (such as Bank A and Insurance Company Z). These certificates will bind the user's public key (hardcoded on the WIM) with their local Enterprise name (hence the requirement of one certificate per Enterprise). The process of enrollment may require "proof-of-identity" information from the user; a discussion of the enrollment process is beyond the scope of this paper.

A basic requirement for WIMs is that they are tamper-resistant. This means that certain physical hardware protection is used, which makes it not feasible to extract or modify information in the module (volatile, non-volatile memory and other parts). This is a strong requirement due to the presence of the user's private keys on the WIM. Note that these private keys *never* leave the WIM.

4.3 iMode Security

Information about the security of iMode is hard to obtain as iMode is a proprietary protocol and service. As quoted on the SANS forum, "One message from a mailing list said that since NTT would not publish any information about iMode security, it should not be considered secure. Without more details, what choice do we have?" [17]

We do know that the iMode protocols are based on Internet protocols. The HTTP and SSL/TLS protocols are used end-to-end by iMode; this was introduced in March of 2001. Lower level protocols are proprietary NTT DoCoMo protocols.

The Japan-based consulting firm Eurotechnology has a list of frequently asked questions (FAQ) on iMode security. In this list, they identify five security issues with iMode, and state that these issues must be addressed separately (although they say nothing about how they should be addressed) [18]:

1. Security of the radio link between iMode handset and the cellular base station (this link uses proprietary protocols and encoding controlled by NTT DoCoMo).
2. Security of the transparent public Internet connection between iMode sites and the handset in the cHTML layer.
3. Security of private networks on iMode.
4. Security of private network links between the iMode center and special service providers such as banks.
5. Password security.

Based on recent presentations on iMode, we believe that some of these points are no longer valid, given iMode's adoption of SSL. As such, points 1, 2, and 3, above, are arguably handled by the use of SSL (although it is still not possible to make any statements about the strength of the lower level protocols and their use).

Security Protocols. iMode security relies on "standard" Internet security as provided by SSL. The security protocols used within the DoCoMo network and over the air are proprietary protocols that run over SSL (using the packet switched capabilities of the DoCoMo network).

iMode only recently adopted SSL (in March, 2001). Before this, iMode provided 'only air interface security' [15]. This required that the PDC-P network be secure and trusted. With the adoption of SSL, however, iMode now provides end-to-end security "in the entire mobile network (security within a carriers network and security between a carrier's network)" [15].

Certificate Management. iMode does have the ability to handle server-side authenticated SSL sessions. iMode phones are pre-configured with root CA keys from PKI vendors Baltimore and VeriSign [15]. This will allow for the establishment of a server-side authenticated SSL session between the iMode device and the Enterprise. This SSL session is established between the iMode device and the iMode server. For most e-commerce applications, the iMode server will be hosted by the Enterprise

iMode does not yet have the capability of handling client-side certificates and as such there are no requirements for management of client-side certificates. This also implies that non-repudiation is not possible with current implementations of iMode.

Technical presentations on iMode have listed smart cards as a new, future direction. These smart cards will be similar to the WAP WIM smart cards, with all of their advantages.

Downloadable Applications. iMode allows users to download Java applications. iMode devices have a Java Application Manager (JAM) that controls download and management of Java applications. Java applets cannot control the JAM, nor can they launch new applets nor access "traditional handset resources" (such as user's telephone book). Thus iMode has provided a version of the Java sandbox on the iMode device to contain these downloadable applications.

5 Wired and Wireless Security Services

In this section we provide a brief discussion of security services such as user identification, authentication, and non-repudiation. We also discuss the so-called "security gap" as it exists in the WAP environment.

5.1 User Identification and Authentication

User identification and authentication can take several forms in the wired world, the most common being username and password. Username and password authentication can be accomplished using "Basic Authentication" in HTTP or through a "forms-based" authentication process (a user fills in their username and password in an HTML form). Additional forms of authentication are provided through the use of token (SecurID, for example), or digital certificates.

Depending on the class of WTLS service, the type of user identification and authentication possible with WAP differs. In all classes of WTLS, it is possible to do a username/password identification and authentication using WML forms sent between the server and the mobile device.

With WTLS Class 3 service, it is possible to have client-side identification and authentication based on the public/private key pair that is hardcoded on to the WIM card and bound with the user's name in their certificate.

User authentication in iMode is the same process as in the Internet. HTTP Basic Authentication is supported by iMode. It is not clear if forms-based authentication is supported.

5.2 Non-Repudiation

Non-repudiation is one of the security services required by "advanced" e-commerce transactions. Non-repudiation requires client-side certificates that bind the user's signing key with their name. The process of generating the signing keys and of generating the user's signing certificate must be "secure" (requirements are often legislated and vary from country to country), so that the user cannot repudiate a transaction based on the keys, certificate, or signature used.

To support the requirement for non-repudiation, the WAP browser (on the WAP device) provides a WMLScript function, `Crypto.signText()` [11]. It is somewhat similar to JavaScript/Java in the Internet environment. This function will require a user to sign a string of text. A call to the `signText()` method displays the text to be signed and asks the user for confirmation. Some implementations may require the user to enter data and simultaneously sign it while other implementations will require users to enter information, send it to a server, where the server will return information with a `signText` call. After the data has been signed and both the signature and the data have been sent across the network, the server can extract the digital signature and validate it, and possibly store it for accountability purposes.

Current implementations of iMode do not facilitate non-repudiation. Once DoCoMo and iMode allow for client-side certificates and client-side authentication, non-repudiation should follow.

5.3 The Security GAP

Security "gaps" appear when a secure session is terminated "prematurely". For example, consider an intended end-to-end secure application using SSL between components A and B, passing through component Z (see Figure 8). A security gap would result if the SSL session (supposedly between A and B via Z) was broken at component Z and re-established between Z and B. This would result in two secure sessions, between A and Z and between Z and B. The message would be secure while on the network, but would be in an insecure state on component Z.

The risks associated with this situation can be mitigated if components Z and B are placed within a secure system (so that there is no security gap while on the Internet, for example). Another means of mitigating the risks associated with a security gap is to provide integrity protection on information (for example using a digital signatures). This will allow the back-end system (component B) to determine if the information has been modified by component Z while in the security gap.

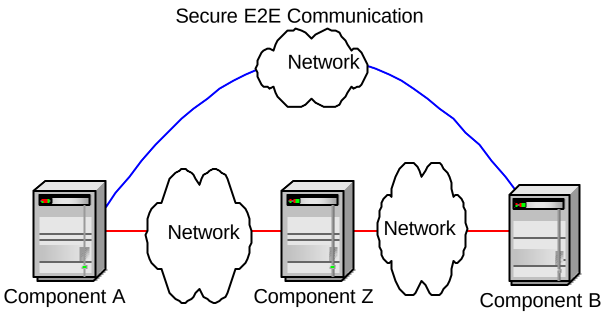

Figure 8: Security "Gap"

A WAP Gateway introduces an encryption "gap" since the WTLS session exists only between the WAP device and the Gateway (only between components A and Z as shown in Figure 8). In a secure architecture, an SSL session will then be established between the Gateway and the back-end. One way to mitigate this "problem" is to place a WAP Gateway within the premises of the Enterprise (see Section 2.2 for more details).

The WAP Forum has adopted the Transport Layer E2E (end-to-end) Security Specification [12] to address this issue. The Transport Layer E2E specification works as follows (see Figure 9):

- The WAP client tries to send a request through its default Gateway to a secure domain.
- The secure content server determines that the request must arrive through the WAP

Gateway in its domain and returns a HTTP redirect message.

- The default Gateway validates the redirect and transmits it to the client.
- The client then caches the new connection and transmits transactions destined for the secure domain to the subordinate WAP Gateway.
- After the connection is terminated the default Gateway is re-selected.

This specification enforces the type of architecture described in Figure 4.

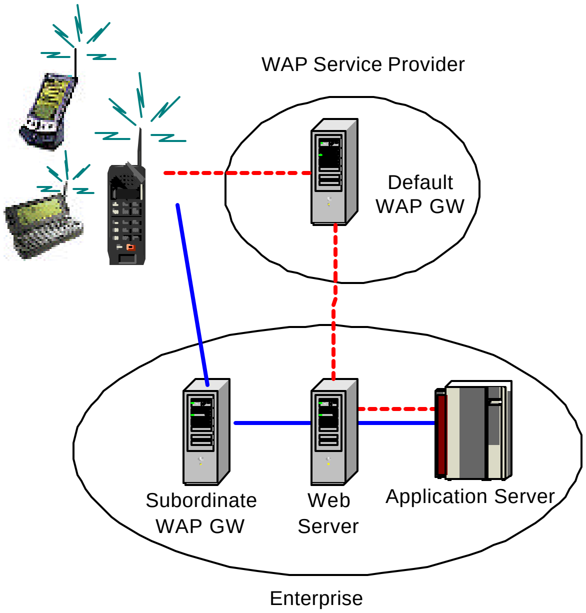

Figure 9: WAP End-to-End Security

6 Analysis

The WTLS specification is sound from a security perspective. Unfortunately WTLS has been specified to be optional to be conformant to the WAP standard. As a result many current WAP phones do not support WTLS, thereby relying on underlying communication services (e.g. GSM security, which stops at the RAS) to provide protection including entity authentication, data integrity and data confidentiality.

Most of the currently available implementations of WTLS in mobile devices only provide Class 1 security. This is insufficient for secure e-commerce transactions and is not used by those parties who provide e-commerce transactions.

WTLS Class 2 conformant WAP devices are currently available on the market. Wireless e-commerce transactions rely on the security provided by server-side authentication and the ability to sign a transaction. Because WTLS Class 2 does not allow for client-side certificates, non-repudiation based on signing is not possible. Currently, those e-commerce services provided with WTLS Class 2 service do not allow “out-of-economy” transactions, such as bill paying or transfers to another account.

The “best” security provided by WAP requires WTLS Class 3. It is anticipated that WAP devices supporting WTLS Class 3 will be generally available around the end-of-year 2001. Several phone manufacturers, such as Motorola and Nokia have announced trial implementations based on limited-run, special production, WTLS Class 3 conformant devices. Limited production runs of WTLS Class 3 phones have been introduced; they are not yet generally available to the mass-market. When this class of WAP device is available, it is anticipated that many banking institutions will enhance their current wireless e-commerce service offerings to allow for “out-of-economy” services.

iMode security is based on the same “principles” as Internet security (provided by SSL/TLS). Current iMode implementation provide a server-side authenticated SSL session. Current iMode security is equivalent to security offered in the current wired environment.

Although DoCoMo has claimed that the introduction of client-side certificates is part of future work for iMode, there have been no indications of when this capability will be available.

WAP suffers from the “security gap” problem. The WTLS protocol for WAP is broken at the WAP Gateway. As the WAP Gateway can be placed within the control of the Enterprise, the risks introduced by this gap can be mitigated.

Within the Internet world, certificates and keys can have different “strengths”, depending on how they were generated. Non-repudiation and digital signatures require users to download and install a browser plug-in or Java applet. Key and certificate strength equivalent to that implemented with the WAP specification requires a smart card reader and smart card.

7 Conclusion

The security that can be achieved for WAP or iMode enabled transactions is in our opinion at least equal to the one that can be achieved in current Internet browser based transactions.

Furthermore WAP seems to be a step ahead with regards to specifications, using the public-private key pairs hardcoded on the WIM, the ease-of-use of client-side certificates and the built-in `Crypto.signText` function. All client-side public-private key pairs are generated in a known, secure manner. Through the adoption of the WIM module it is possible to achieve a more reliable form of authentication (public key and smart card). Because the process of key and certificate generation is controlled and known, WAP can claim non-repudiation at a level not generally possible in the Internet world.

Whether WAP or iMode will be a great success or not therefore should not depend on security issues. The remaining issues such as usability and openness should determine in how far WAP and/or iMode can be globally adopted [16].

8 References

1. D. Denning, *Information Warfare and Security*, P163-183, Addison-Wesley Publishers, 1999.
2. The WAP Forum, http://www.wapforum.org
3. P. Ashley and M. Vandenwauver, *Practical Intranet Security – An Overview of the State of the Art and Available Technologies*, Kluwer Academic Publishers, 1999.
4. W. Stevens, TCP/IP Illustrated Volume 1: The Protocols, Addison-Wesley Professional Computing Series, 1994.
5. GSM Association, http://www.gsmworld.com
6. Wireless Application Protocol Wireless Transport Layer Security Specification (WTLS), http://www1.wapforum.org/tech/documents/WAP-199-WTLS-20000218-a.pdf.
7. T. Dierks, C. Allen, The TLS Protocol – Version 1.0, RFC 2246, 1999.
8. J. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography*, CRC Press, 1997.
9. Nokia, KPN Mobile and Interpay Test Mobile Commerce Solution, http://press.nokia.com/PR/200104/816440_5.html
10. Wireless Application Protocol Identity Module Specification, http://www1.wapforum.org/tech/documents/WAP-198-WIM-20000218-a.pdf.
11. Wireless Application Protocol WMLScript Crypto Library Specification, http://www1.wapforum.org/tech/documents/WAP-161-WMLScriptCrypto-19991105-a.pdf.
12. WAP TM Transport Layer E2E Security Specification, http://www1.wapforum.org/tech/documents/WAP-187-TransportE2ESec-20000711-a.pdf
13. All About iMode Index, NTT DoCoMo, http://www.nttdocomo.com/i/index.html
14. Dankse Bank, Denmark, www.danskebank.dk
15. Satomi Okazaki, Atsushi Takeshita, Yiqun Lisa Yin, New Trends in Mobile Phone Security, presentation given at RSA 2001
16. Batista, Elisa, “WAP or I-Mode: Which Is Better?”, at Wired.com http://www.wired.com/news/wireless/0,1382,38333,00.html
17. John Schramm, Security Issues in WAP and I-Mode, SANS Institute, SANS Information Security Reading Room, December 2, 2000 http://www.sans.org/infosecFAQ/wireless/WAP4.htm
18. “The unofficial independent iMode FAQ”, Eurotechnology.com, http://www.eurotechnology.com/imode/faq.html , Eurotechnology Consulting Web Site
19. “DoCoMo’s iMode: Toward Mobile Multimedia in 3G”, presentation given at March 2000 IETF Meeting, IETF 47, Plenary Session http://www.ietf.org/proceedings/00mar/slides/plenary-imode-00mar/sld002.htm
20. NTT DoCoMo Technology Site, Mobile Computing and Imode, http://www.nttdocomo.co.jp/corporate/rd/tech_e/mobi01_e.html
21. Association of Radio Industries and Business, http://www.arib.or.jp/index_English.html, (English index)