Ensuring Security in WAP and Usability in WAP Applications

Al-Mukaddim Khan Pathan and Minhaz Fahim Zibran
Department of Computer Science and Information Technology
Islamic University of Technology
{mukaddim@iut-dhaka.edu, zibranm}@iut-dhaka.edu

Abstract: Keeping pace with the growing communication need of modern life today’s technology also is focusing towards pervasive computing and ubiquitous computing. Computing devices are becoming smaller and multi-purpose wireless devices like palm-top PCs, PDAs, mobile phones are becoming more popular. Access to the World Wide Web from such devices is usually done through Wireless Application Protocol (WAP). There is a set of concerns over how secure and usable WAP is as a technology, and whether it is robust enough to implement mobile commerce applications, and other applications with stringent security requirements. In this paper we have performed a review of WAP applications and investigated out the facilities and the technologies that WAP has to offer for building and deploying secure and usable WAP applications.

Keywords: Wireless Application Protocol (WAP), security

1. INTRODUCTION

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 [1].

Wireless Application Protocol (WAP) is a protocol for accessing information and services from wireless devices. WAP is defined and coordinated by the WAP Forum, a consortium of industry players, whose objective is to define a standard application framework that will be universal, and that will allow seamless interoperability of all of the components required for mobile access to network applications [2]. There is a set of concerns over how secure and usable WAP is as a technology, and whether it is robust enough to implement m-commerce (mobile commerce) applications, and other applications with stringent security requirements. Being concerned about the security issues in WAP technology, in this paper, we have performed a review of the WAP applications and implementations and worked out the enhancements that are to be made and the facilities to be provided by WAP as a technology, to ensure security.

The rest of the paper is organized as follows: Section 2 provides an overview of WAP, the security and usability, Section 3 performs a review of the existing WAP applications and implementations, Section 4 enlightens the requirements that WAP has to offer for building and deploying usable and secured applications and hence provides a guideline and Section 5 concludes the paper.

2. WAP, SECURITY AND USABILITY

Before starting our review on WAP applications and the concerned security issues, let us have an overview of WAP and the security.


Most people would agree that there are more protocols associated with computing than you could shake a stick at, and it can be challenging trying to keep up with them all. Even though, we need another protocol like Wireless Application Protocol (WAP), because phones are not PCs. To be specific, most of the protocols in use today make a set of assumptions about the environment, such as the type of network that will be available (particularly from the point of view of bandwidth and reliability), the types of devices that will be accessing the services, and the types of services that will be accessed. These
assumptions do not necessarily hold true in the wireless world.

There are a number of differences in terms of the device itself:

- **Form Factor:** A mobile device needs to be small enough to move around and ideally to be able to fit in the palm of one’s hand or carry in a shirt pocket.

- **CPU:** In a mobile device, the CPU is not nearly as powerful as a desktop PC, and is almost certainly of a different architecture.

- **Memory and storage:** This is a lot more constrained than on a PC, because handset manufacturers are cost-sensitive, and thus reluctant to add any additional components unless it is really necessary. Also some mobile devices do not have a persistent storage of their own.

- **Battery:** Mobile devices are battery powered, and the need to have the device available for long periods of time means that the processing CPU cannot make significant demands on the battery.

- **Display:** This is typically limited in size and resolution, and often cannot cope with color.

- **Input:** Mobile devices typically do not have keyboards, or if they do they are limited in size. Therefore, input is more challenging than on a typical PC.

A wireless network is considerably different to a fixed-wire network in a number of ways. The types of applications that are suitable for use on mobile devices are not the same as those that are popular on fixed-wire environments. Typical users of mobile applications are likely to be a broader subset of the population than PC users. Even the context in which the applications are going to be used will be different. This highlights the most important aspect of mobile application design, which is to make the application easy to use in the context, and on the device that it will be accessed from. And all these issues gave rise of the Wireless Application Protocol (WAP).

2.2. Security [1]

The phrase 'secure system' means one that cannot be compromised or accessed without authorization. Security has an obvious role to play with regard to m-commerce and the ability to secure transactions. 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. There are a number of basic issues around security that have to be addressed.

These basic issues that are to be addressed for secured applications development as well as WAP applications are as follows:

- **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 third parties do not know the content and meaning of communications between two communicating 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.

2.3. Usability

Usability is a term that indicates the degree of user-friendliness of a system. That is, Usability refers to the ease with which users of an application can perform the operations and complete the tasks that the application is supposed to help them achieve. If users perceive an application to be easy-to-use, straightforward, and forgiving, it is a usable application.

Building usable WAP applications is not simple. Wireless devices have many limitations, and the average user of a WAP application is not technically oriented (and possibly not even used to the Internet). Again, the interpretation of WML varies greatly differs among devices from different vendors. So, this poses an extra challenge to good usability.

3. EXISTING WAP IMPLEMENTATIONS AND APPLICATIONS

3.1. The WAP protocol stack

The WAP protocol stack is shown in Figure 1.
The WAP protocol stack contains the following elements: physical and data link layer, network layer, transport layer, session layer, presentation layer and the application layer.

### 3.2. Wireless communication model

In the wireless communication model that we have considered, communication takes place between the person-in-the-street and some WAP enabled web site. The model is illustrated in Figure 2.

In this model the remote client is a mobile device, but still dials into a Remote Authentication Server (RAS) on some network somewhere. The network provider will typically also host the WAP gateway, and a web server. 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.

#### 3.2.1. Security problems in Wireless Communication model

- From the point of view of security, this scenario has various implications. Wireless Transport Layer Security (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. If there is 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 Transport Layer Security (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 in Figure 3.

#### 3.3. WAP v1.1

In WAP: v1.1, WTLS provides the capability to generate Master secrets, which are used as a source of entropy, to calculate MAC (Message Authentication Code) keys and message encryption keys that are used to secure a limited number of messages, depending on usage of WTLS. It also allows for server and client authentication. But the security defined by the WAP: v1.1 is not strong enough.

---

**Figure 1:** The WAP protocol stack [1]

**Figure 2:** The wireless communication model [1].

**Figure 3:** Security gap in wireless communication model [1].
3.3.1. Loopholes in WAP v1.1

- Banking and commercial organizations are generally not satisfied with the security offered by WAP v1.1 [5]. Some of the reasons for this are due to the patchy levels of implementation that are characteristic of an emerging technology — some of the first WAP phones available on the market didn't support WTLS encryption at all. Of those that do, at the time of presenting only one (the Nokia 7110) does any WAP gateway authentication — that is, the ability to request a WAP gateway's certificate, authenticate it, and ask the consumer whether it should be accepted. All of the other phones that support WTLS use it only for encryption, not for the additional safeguard of authentication.
- Many of the WAP gateways deployed early on didn't support WTLS either, and some didn't even support TLS/SSL (Secured Socket Layer) [7] interaction with the content servers.
- The WAP gateway has the job of linking the WAP protocols on the wireless side to the web protocols of the wired world. For a secure data transfer, this requires that the WTLS-encrypted data sent from the WAP client must be momentarily decrypted to clear text, then immediately encrypted for onward transmission to the content servers over TLS/SSL. Since this only happens for an instant, and WAP gateways tend to be operated by mobile phone network operators (which we would like to think are trusted organizations), it could be argued that to all intents and purposes, this is a secure connection. However, that can't be absolutely guaranteed, and consequently the gateway represents the most vulnerable link in an end-to-end transaction.
- Applications such as e-commerce require the ability to provide persistent proof that someone has authorized a transaction. Although WTLS provides transient client authentication for the duration of a WTLS connection, it does not provide persistent authentication for transactions that may occur during that connection.
- The Wireless Telephony Application Interface in v1.1 suffers from ambiguities and incorrect specifications.

3.4. WAP: v1.2

WAP: v1.2 provides for additional, robust, application-level security, over and above that specified in version 1.1. In v1.2, the Wireless Identity Module (WIM) allows secure storage and use of permanent private keys. It provides secure storage of the Master secrets defined for WAP 1.1, and the client keys used for client authentication. In addition, using the new WMLScript crypto library, it allows for digital signatures to be generated for use in end-to-end security at the application level. Crucial to this is a tamper-proof module in the mobile device called the WIM.

3.4.1. Problems with WAP v1.2 [3]

- The Wireless Identity Module (WIM) in v1.2 has been defined with smart cards in mind, although the physical packaging of the WIM has been left to the discretion of manufacturers.
- Often it will be combined with the existing mobile phone SIM (which is, after all, a kind of smart card), giving rise to the rather unfortunately named SWIM.
- For optimum security, some parts of the security functionality need to be performed by a tamper-resistant device, so that an attacker cannot retrieve sensitive data. It again burdens the overall security process.

3.5. Push and Pull applications

Push is a major new piece of functionality that is a significant addition to the WAP developer's toolkit. The WAP push framework introduces a means within the WAP effort to transmit information to a device without a previous user action. In the normal client-server model, a client requests a service or information from a server, which then responds by transmitting information to the client. This is known as "pull" technology — the client "pulls" information from the server.

3.5.1. Push Security

Needless to say, feeding content to a device without the specific request of the client requires careful consideration of security. The WAP specifications provide a lot of ideas about how this is to be achieved, but leaves it to the implementers to decide exactly what approach should be followed, which make it more difficult to implement the push applications with the security incorporated in it.
3.6 WML Interoperability Issues
WML delivers content and user interfaces across very different kinds of devices. The various browser implementations render WML in different ways, and this will affect the usability of WAP applications. Though, a usable WAP application should never confuse users; just one click away. Unfortunately, if an application is tweaked to be more usable on a particular device, the chances are that usability will suffer on other devices.

4. GUIDELINE FOR DEVELOPING SECURED AND USABLE WAP APPLICATIONS
The guideline for developing secured and usable WAP applications are listed below:

- The Wireless Telephony Application Interface should be enhanced to remove ambiguities and to correct the incorrect specifications.
- For push applications, clients can delegate trust to the Push Proxy Gateway (PPG) [3]. The client can maintain a list of trusted PPGs, and as long as the push content is being submitted by one of those, the client can accept that the PPG has in turn authenticated the push initiator.
- A special "signature" button can be added to mobile device keypads. This will help to create in the mind of the end user the notion that only mobile devices with such a button provide access to secure electronic/mobile commerce. [3]
- For mobile devices, there is the issue of who or what is being authenticated by the certificate. So, in order to validate that the current user of the authenticated device is the rightful user a variety of systems can be used, which vary in their complexity and robustness from a simple PIN number through to a SecureID token.
- Since there is a security gap at the WAP gateway, it can be resolved by providing an end-to-end security standard. The WMLScript Crypto library may be extended in the future to include cryptographic functions.
- Part of the security issue in WAP can be addressed by the use of Wireless Identity Module (WIM). Though it is defined in WAP: v1.2, it is needed to work out the physical packaging of the WIM.
- Some general rules for building usable applications can be:
  a. Top 20% of functionality: When porting an existing application to WAP (an HTML page, for example), the main activities that users will be interested in using while on the road should be identified.
  b. Rate user activities: The main activities that the majority of the users perform should be identified, and applications should be built in a way that will let users perform these activities in the fastest way possible.
  c. Design it as a tree structure: Lay out a hierarchical tree of activities. Users should enter the application at the root and be able to perform any of the available activities through some path starting at the root. Each level of the tree should be laid out according to the likely popularity of the activities it contains.
  d. Minimize data entry: The users should not be made overwhelmed to remember codes or other information when visiting the application. In addition, the input mode of the terminal should be set to support the expected format for the data that users will enter. This can be achieved through input masks.
  e. Text should be terse: Short, polished, and informative text is vital to guide users.
  f. Always implement 'back' functionality: All users like to explore when confronted
with a new application, and a 'back' function should be available to them at all times.

g. **Consistency is very important:** Applications can often require users to perform the same activity (or very similar activities) in different parts of the application. It is important that a consistent set of metaphors is deployed that will help users find their way around easily.

h. **Push:** Real-time information is a key piece of functionality that will give extra value to WAP. Unfortunately, push is not part of the WAP standard yet, but we should see it implemented in WAP 1.2. In spite of this, several proprietary possibilities to do push through WAP already exist, and it should be exploited if possible.

i. **Be prepared to test:** If WAP application is deployed that is even moderately complex, prototypes should be built as early as possible in the development process [8] [9] [10].

5. **CONCLUSION**

Though WAP applications available today are not free from usability and security problems, they are essential to deliver the web-based content to the mobile devices. If the security and the usability problems, which exist in those applications, can be beaten, it will be possible to ensure that WAP browsers are going to reach a good many consumers. We hope that, the guideline that we have provided in this paper to build secure and usable WAP applications will be very helpful for the researchers to overcome the above stated problems of WAP applications and it will be possible to build robust applications, which can be used to implement secured environment.

**REFERENCES**


