Universidade de Brasília – UnB
Escola de Extensão
Curso Criptografia e Segurança na Informática

Segurança do Wireless Aplication Protocol (WAP)

Aluno: Orlando Batista da Silva Neto
Prof: Pedro Antônio Dourado de Rezende

Brasília, 13 de Agosto de 2000

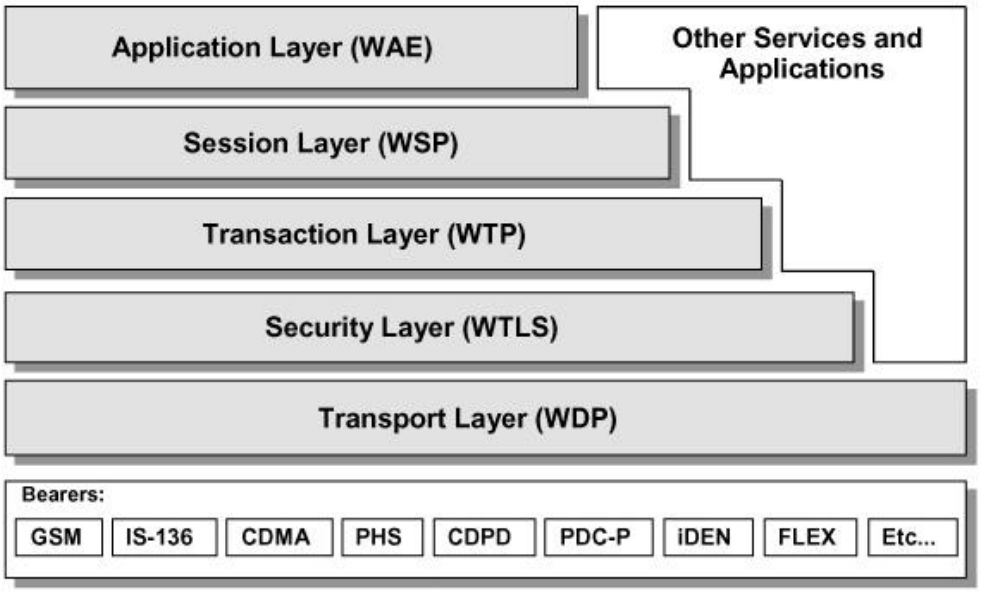
Índice

Introdução

O segmento de equipamentos de comunicação sem fio, tem sido alvo das atenções tanto do mercado de tecnologia, quanto da mídia, principalmente esta última. Todas as revistas especializadas têm reservado espaços generosos de suas capas para o segmento cujas pesquisas indicam ser a maior onda tecnológica depois da WEB, ou seja, o segmento da comunicação sem fio. Junto com este novo paradigma, o do ambiente *Wireless*, se destaca o tema desta monografia, o *Wireless Aplication Protocol* ou simplesmente, o *WAP*, um padrão de protocolo emergente para apresentação e entrega de informações e serviços para equipamentos de comunicação sem fio. As especificações do *WAP* estão sendo desenvolvidas pelo *WAP Forum*, um consórcio de empresas interessadas no segmento de comunicação sem fio, principalmente as fabricantes de aparelhos celulares, as quais parecem vislumbrar um futuro promissor no WAP.

Embora a última versão das especificações do *WAP* estejam em sua versão 1.2, a maioria das implementações distribuídas ainda são baseadas na verão 1.1.

O *WAP* adota um padrão de camadas similar ao ISO/OSI no qual uma camada oferece e requisita serviços das camadas inferiores e superiores (Figura 1). O *WAP* apresenta muitas semelhanças com o protocolo *TCP/IP*, de modo a ser apresentado como o protocolo que vai permitir que os dispositivos *wireless* se conectem à Internet tendo acesso a conteúdos web, e-mails dentre outros.



Fonte: Especificação WAP Architecture Version 30-Apr-1998 (WapForum)

Figura 1

Como os recursos de equipamentos como aparelhos celulares são muito limitados: pouca memória, velocidades de transmissão de dados baixas, o acesso à Internet via *WAP* possui toda uma formatação especial de modo a permitir que usuário, através de um *browser* instalado em um celular venham a ter acesso aos mesmos conteúdos que está acostumado a ter através de seu computador pessoal.

Da perspectiva da segurança, que será a abordagem desta monografia, o *WAP* especifica uma camada responsável por garantir a segurança nas conexões, definida no protocolo *Wireless Transport Layer Security,* ou seja, o *WTLS*, um protocolo desenvolvido baseado no *TLS*, o qual, por sua vez, é baseado no *SSL* 3.0.

O Protocolo Wireless Transport Layer Security - WTLS

O *WTLS* foi desenvolvido de modo a atuar junto aos protocolos de transporte do *WAP*, com a característica de ser otimizado para canais banda limitada, como por exemplo, a telefonia celular. A especificação do *WTLS* tenta assegurar: integridade, privacidade, autenticação e proteção a ataques de negação de serviço.

O *WTLS* contém facilidades para detecção e rejeição de informações repetidas ou não verificadas com sucesso, dificultando desta forma, ataques *DoS* (*Denial of Service*) e protegendo as camadas superiores.

O *WTLS* é, no ambiente sem fio, responsável pelas mesmas funcionalidades oferecidas pelo SSL no ambiente tradicional da internet com fio, ou seja, a implementação de *Private Key Infrastructure (PKI),* permitindo desta forma que usuários de uma rede pública sem segurança, como é o caso da comunicação sem fio, possam trocar informações de forma segura através da utilização de pares de chaves públicas e privadas obtidas de "autoridades confiáveis". Uma estrutura de autoridades certificadoras e certificados digitais X.509, como a da Internet, com as adaptações necessárias para o mundo de recursos limitados dos dispositivos sem fio, é utilizada no *WTLS*.

O *WTLS* utiliza certificados digitais para garantir a autenticidade dos servidores, da seguinte forma: Quando um cliente *WAP* visita um *site* seguro, seu *microbrowser* envia um mensagem "*Client Hello*" para o servidor *WAP* indicando que uma sessão segura está sendo iniciada. O *WAP* server responde à requisição enviando seu certificado acompanhado de sua chave pública. o *microbrowser* vai verificar se o certificado é válido e se for, o *microbrowser* vai gerar uma chave de sessão, encriptar com a chave pública do servidor (envelope digital) e enviar para o servidor, para que a sessão possa ser aberta.

O *WTLS* utiliza dois tipos de certificado:

- Certificados *WTLS* de servidor, definidos no *WAP* 1.1, utilizados para autenticar um servidor *WTLS* perante um cliente *WTLS* e prover uma base para o estabelecimento de uma chave para encriptar uma sessão Cliente-Servidor. São similares a certificados *SSL*, exceto pelo fato de que dois formatos de certificado são definidos: Certificados X.509, como no *SSL*, e mini-certificados *WTLS*, com função similar à do X.509, com o detalhe de serem menores e mais simples, de modo a facilitar seu processamento pelos clientes, os quais no paradigma *wireless*, possuem recursos muito limitados.

- Certificados *WTLS* de clientes, definidos no *WAP* 1.2, utilizados para, agora, autenticar o cliente *WTLS* perante o servidor *WTLS*. Este, também é definido em dois formatos: X.509 e mini-certificados.

Três tipos de *handshakes*, ou seja, negociação entre cliente e servidor para o início de uma sessão, são definidos:

- *Full Handshakes*: São negociados todos os parâmetros da conexão e são trocados os certificados entre cliente e servidor;
- *Optimised Handshakes*: Neste caso, os certificados do cliente estão ao alcance do servidor sem que seja necessário uma conexão direta com o cliente
- *Abbreviated Handshakes*: Neste caso a conexão é realizada em cima de uma sessão feita anteriormente, via *Optimised Handshake* ou *Full Handshake*.

O *handshake* do *WTLS* envolve os seguintes passos:

- Troca de mensagens de "*Hello"* para a escolha de uma algoritmo e troca de valores randômicos
- Trocas dos parâmetros criptográficos para permitir que cliente e servidor concordem com uma pré-chave de sessão;
- Troca de certificados e informações criptográficas para a autenticação entre cliente e servidor;
- Geração de uma chave de sessão, a partir da pré-chave de sessão gerada e dos valores randômicos trocados;
- Permitir que cliente e servidor verifiquem que seus pares calcularam os mesmo parâmetros de segurança e que o *handshake* ocorreu sem a interferência de nenhum intruso.

No *WTLS* as chaves de sessão têm uma vida relativamente longa, que pode ser de vários dias. Esta longa vida se justifica para evitar *handshakes* completos os quais são relativamente pesados devido à grande quantidade de informações envolvidas. Estas chaves de sessão são utilizadas como fontes de entropia, calcular *MAC*s (*Message Autentication Code*) e chaves de *encriptação* que serão utilizadas para um número limitado de mensagens, por segurança.

O *WAP* 1.2 também define uma biblioteca baseada em *PKI* que não faz farte do *WTLS*. Esta biblioteca, a qual permite um cliente *WAP* assinar digitalmente uma transação, é conhecida

como *WML*[1] *Script Crypto Library*, e é indicado para aplicações que requeiram assinaturas não repudiáveis dos clientes. A função *WML Script Sign Text*, mais especificamente, provê assinaturas digitais para clientes *WAP*.

Muitas aplicações requerem provas persistentes de que alguém autorizou um transação. Embora o *WTLS* proporcione a autenticação do cliente durante a conexão ele não proporciona uma autenticação persistente para as transações que venham a ocorrer durante a conexão. Para suprir esta deficiência, o *microbrowser* deve suportar funções *WMLScript*, "*Crypto.signText"*, a qual pergunta ao usuário se ele fazer uma assinatura eletrônica. A chamada ao método "*signText*" exibe o texto a ser assinado e pede uma confirmação para a assinatura. Depois de a assinatura ter sido feita, ambas, assinatura e informações são enviadas pela rede, de modo que o servidor possa extrair a assinatura digital , validá-la e opcionalmente armazená-la.

No ambiente internet tradicional, a tecnologia de chave pública empregada comercialmente e quase sempre baseada em *RSA*. No ambiente sem fio existem discussões sobre a utilização de uma tecnologia alternativa, denominada *Elliptic Curve Cryptography* (*ECC*), a qual tem as mesmas funções básicas do RSA, mas com uma demanda menor de CPU. Mas esta questão é mais pertinente, quando se considera a criptografia para autenticação de clientes ou assinaturas digitais, mas a utilização do *ECC*, cujas funções são exigem menos recursos que as equivalentes no *RSA*, ainda não mostra sinais de crescimento diante da enorme infra-estrutura já pronta utilizando o *RSA*.

[1] O Wireless Marckup Language (WML) é o equivalente WAP ao HTML. É uma linguagem orientada a hipertexto adaptada aos poucos recursos das interfaces dos clientes, tais como aparelhos celulares.

O WAP Identity Module – WIN

Para uma melhor segurança, algumas funcionalidades de segurança podem ser efetuadas por um dispositivo externo ao equipamento móvel, evitando assim o acesso de possíveis atacantes a dados sensíveis, tais como, as chaves privadas utilizadas no *handshake* com autenticação do cliente e nas assinaturas digitais.

Também pelo fato deste processo de manipulação de chaves simétricas e assimétricas consumir muitos recursos de processador, é especificada a utilização de dispositivos auxiliares na realização destes processos, estes dispositivos são definidos como *WIN*, ou seja *WAP Identity* Modules.

O *WIM* (*Wap Identity Module*) pode ser utilizado para armazenamento das chaves privadas e cálculo da assinatura. Para a verificação da assinatura digital, o servidor deve ter acesso ao certificado do cliente, que deve ser validado por uma autoridade certificadora conhecida pelo servidor. Existem várias forma de o servidor ter acesso ao certificado do cliente: O certificado é aposto à assinatura, O *hash* da chave pública é aposto à assinatura e o servidor é capaz de comparar com o certificado do cliente que está armazenado na autoridade certificadora e etc.....

O *WIM* é utilizado para melhorar a segurança de implementações da camada de segurança (*WTLS*) e de certas funções da camada de aplicação. Suas estrutura é baseada no *PKCS15*. Utiliza um modelo de objetos que faz possível o acesso a chaves, certificados, objetos de autenticação e objetos de dado proprietários em dispositivos simples. Suas funcionalidades podem ser implementadas em *smart cards*. O *WIM* é definido como uma aplicação *smart card* independente.

Para o *WTLS*, o *WIM* é usado para os seguintes propósitos:

- executar operações criptográficas durante o *handshake*, especialmente quando estas exigem autenticação dos clientes
- sessões seguras *WTLS* de longa-duração.
- cálculo (*ECDH*) ou geração (*RSA*) de segredos para a geração de chaves.
- cálculo e armazenamento da chave de sessão para cada sessão

O *WIM* é utilizado para proteger chaves privadas, normalmente certificadas. O *WIM* armazena as chaves e executa operações do tipo:

- operações de assinatura para autenticação do cliente quando necessário
- operações de Troca de chaves utilizando uma chave de cliente fixa.

O *WIM* pode armazenar certificados tanto de Autoridades certificadoras quanto de usuários. O armazenamento de certificados das Autoridades Certificadoras tem um grande importância do ponto de vista da segurança: ela pode ser exposta sem o perigo de ser alterada. Se existem muitos certificados para se armazenar será necessário armazená-las no cliente, um telefone celular por exemplo, mas como seus recursos tendem a ser escassos, os certificados armazenados podem ser transferidos para o *WIN*.

Do ponto de vista da segurança ,o certificados não precisam necessariamente ficar armazenados no dispositivo *WIM*, No *WTLS* o servidor pode pedir o certificado do cliente de várias formas inclusive pedindo apenas um *URL* do certificado, ao invés do próprio certificado.

O *WIM* mantém informações sobre os algoritmos que suporta. Quanto dois equipamentos se comunicando utilizando o protocolo *WAP* com comunicação segura, por exemplo, existe uma comunicação prévia para que possam ser trocadas informações sobre os protocolos suportados pelos dois equipamentos.

Um telefone celular, por exemplo, armazenaria *MACs* e chaves de encriptação de mensagens à medida que fosse necessário. Estas chaves têm um tempo de vida que pode ser negociado no *handshake WTLS*, em casos extremos elas são utilizadas para uma mensagem apenas. O telefone pode apagá-las da sua memória quando sair de uma aplicação *WAP*. As chaves poderão ser geradas novamente a partir do segredo armazenado no *WIM*. Um atacante que obtenha uma chave de sessão ou uma chave *MAC* poderá utilizá-las somente pelo tempo determinado no *handshake* da sessão *WAP*.

Operações em nível de aplicação que utilizam o *WIM* incluem assinatura e decriptação de chave. Ambas operações utilizam uma chave privada que nunca deixa o dispositivo WIM. No caso de chaves públicas cifradas, o equipamento móvel envia uma chave pública encriptada para o WIM que a decifra com a sua chave privada e devolve a chave pública para o equipamento, que por sua vez utiliza a chave pública para decriptar mensagens que venham a chegar. Para assinar, o equipamento móvel calcula o *hash* dos dados, formata de acordo com os requerimentos da aplicação

envia o *hash* formatado para o dispositivo *WIM* que por sua vez calcula a assinatura digital utilizando a chave privada e retorna para o equipamento móvel a assinatura.

O *WIM* é utilizado para prover a identidade a autenticação do cliente. Isto envolve pegar a chave pública ou certificado do dispositivo *WIM* e executar operações de assinatura identificando o usuário. O *WIM* também é utilizado para gerar números randômicos a serem utilizados na geração das chaves.

O Dispositivo *WIM* precisa armazenar as seguintes informações:

- informações relativas às propriedades do dispositivo, tais como algoritmos suportados e etc;
- Pares de chaves para autenticação e assinatura digital;
- próprios certificados;
- certificados confiados;
- informações relativas às sessões *WTLS*.

Todas estas informações são armazenadas de acordo com o *PKCS*#15. Na especificação atual os dispositivos *WIM* devem ser compatíveis com o *RSA* e o *ECDH*

Certificados de usuários podem ser armazenados no dispositivo *WIM*, neste caso os certificados devem obedecer certos padrões, entretanto, devido ao tamanho destes certificados passa a não ser mais tão interessante o armazenamento, neste caso, o dispositivo *WIM* pode armazenar *URLs* para os certificados ou os certificados podem ser acessados de um diretório utilizando um índice de chave, como por exemplo o *hash* da chave pública, como critério de busca. Note que do ponto de vista de segurança não ser faz necessário que os certificados fiquem em algum tipo de dispositivo especial.

Pelo fato de o formado de armazenamento dentro do dispositivo *WIM* seguir o padrão *PKCS*#15, aplicações não *WAP* também podem ter acesso às informações nele contidas.

Exemplos de implementações de dispositivos *WIM* são os *smart cards* e cartões *SIM*(*Subscriver Identity Module*)

O papel dos *Gateways* WAP na segurança

Um dos componentes essenciais da infra estrutura do *WAP* é o *gateway*, responsável pela passagem dos pacotes do mundo internet tradicional, para o mundo da internet sem fio. Este *gateway*, normalmente disponibilizado pela operadora do ambiente sem fio (Ex. Operadora de Celular), provê uma ligação entre o ambiente sem fio (Ex. *WAP*), e o ambiente internet tradicional (Ex. *HTTP*/*HTML*). Do ponto de vista da segurança o gateway wireless pode executar funções intermediárias de segurança, tanto como manter a proteção *HTTP*/*SSL* do lado sem fio, quanto manter a proteção do *WAP*/*WTLS* do lado da internet tradicional.

Para os certificados de servidores ou *gateways*, o formato de mini-certificado é muito importante. Estes certificados precisam ser transmitidos através do ar para os clientes sem fio e processados pelos seus recursos limitados.

No ambiente *WAP*, a autenticação do cliente perante o *gateway* ou servidor, é proporcionada pelo *WTLS* na versão 1.2 do *WAP*. Já autenticações do cliente perante aplicações requerem funções de camadas mais altas que o *WTLS*

Conclusão

O comércio eletrônico no ambiente *wireless* tem despontado como a próxima grande onda tecnológica, isto por que, ele combina as duas tecnologias mais faladas da economia atua: a *WEB* e a comunicação sem fio. A segurança, essencial em qualquer transação comercial, amplia o potencial desta nova forma de comércio. Os conceitos de confidencialidade, autenticidade, integridade e não repúdio, se fazem mais do que necessários em um ambiente completamente aberto como o sem fio.

Para aplicações que utilizam o *SSL*, os *gateways WAP* gerenciam automaticamente e transparentemente manutenção da segurança do *SSL* para o *WTLS*. É neste ponto que pode ser observado o ponto de maior venerabilidade durante uma conexão, pois neste momento, tudo que vinha sendo protegido pelo *WTLS* precisa ser manipulado para entrar no ambiente internet protegido pelo *SSL*.

O modelo de segurança do *WTLS* se estende do cliente até o *Gateway WAP*. O Proverdor do serviço WAP é responsável pelo *gateway* de modo a garantir

Os aplicativos são capazes de habilitar ou desabilitar as funções do *WTLS* dependendo da segurança que requeiram.

A utilização de certificados é obrigatória no *WTLS*. No *WTA* (*Wireless Telephony Application*), por exemplo , o uso de sessões *WTLS* é obrigatório.

Bibliografia

Especificações do Wireless Aplication Protocol, *http://www.wapforum.org/;*

Understanding Digital Certificates and WTLS, http://www.entrust.net/products/learnwap/index.htm