文章编号:1001 - 9081(2005)04 - 0859 - 03

使用 ECMQV 密钥交换方案增强 WTLS 协议安全性

叶润国[1,2],冯彦君[1,2],虞淑瑶[1,2],吴 宇[1,2]
(1. 中国科学院 计算机网络信息中心,北京 100080; 2. 中国科学院 计算技术研究所,北京 100080)
(yerunguo@cnic.cn)

摘 要:ECMQV 协议是一种基于 ECDH 的认证和密钥交换方案,它具有高安全性和低计算开销等优点。通过将 ECMQV 协议集成到 WTLS 协议框架中实现了一种 WTLS 扩展协议,它在略微增加无线终端计算开销情况下明显提高现有 WTLS 协议安全性。在协议中采用了 Cookie 技术来防止可能的拒绝服务攻击。该 WTLS 扩展协议可在轻量级无线终端上实现,以满足无线终端在企业远程访问环境下的高安全性要求。

关键词:网络安全;WTLS;密钥交换方案;ECMQV
中图分类号: TP393.08 **文献标识码**:A

Employing ECMQV key exchange scheme to enhance WTLS security

YE Run-guo[1,2], FENG Yang-jun[1,2], YU Shu-yao[1,2], WU-Yu[1,2]
(1. *Computer Network Information Center, Chinese Academy of Sciences, Beijing* 100080, *China;*
2. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100080)

Abstract: ECMQV is an authenticated key exchange scheme based on conventional ECDH protocol, it possesses advantages on higher security and lower computation overhead. This paper implemented a WTLS protocol variant through the integration of ECMQV scheme into WTLS framework; the security of current WTLS protocol was greatly enhanced while only a little more computation overhead was incurred. The WTLS protocol variant can be deployed on lightweight wireless terminals and meet their high-security requirements under future enterprise remote access environments.

Key words: network security; WTLS; authenticated key exchange protocol; ECMQV

0 引言

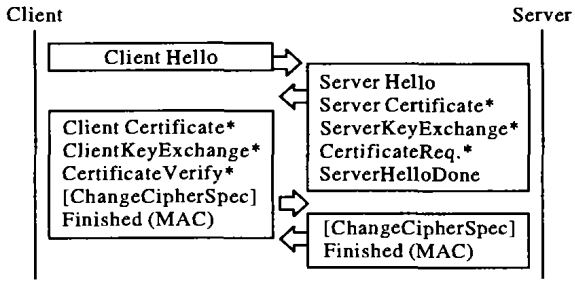
WTLS[1]是一种为无线应用协议(Wireless Application Protocol,WAP)设计的端到端安全协议,其目的在于为上层 WAP 应用提供加密、鉴别和数据完整性服务。WTLS 规范能够提供三类服务:1)类别 1,提供加密功能,而无认证功能;2)类别 2,提供加密功能,并提供服务端认证功能;3)类别 3,提供加密功能,并提供客户端和服务端双向认证功能。目前的 WTLS 规范只支持几种简单的类别 3 密钥交换方案,包括 RSA-512/768 和 ECDH-ECDSA,但是,这些密钥交换方案存在两个问题:1)ECDH-ECDSA 密钥交换方案采用静态 ECDH 公钥交换算法,它无法提供密钥完全前向保密功能,因而,无法满足无线终端在企业远程访问场景下的高安全性要求;2)RSA-512/768密钥交换方案采用显式签名方式来认证自己,对终端计算能力要求较高,因而,不适合轻量级无线终端。

ECMQV[4]是一种基于 ECDH 协议的密钥交换方案,它能够实现通信两端的身份认证以及会话密钥的协商。与传统密钥交换方案采用显式公钥签名认证方式不同,ECMQV 密钥交换方案采用隐式认证方式,因而有效地减少了通信两端公钥计算开销。

1 相关技术

1.1 WTLS 介绍

WTLS 由 TLS1.0[2]发展而来,针对无线网络环境中特有的无线终端计算能力、无线网络带宽限制等问题对 TLS 进行了优化,主要表现在:支持数据报传输协议、优化了传输分组大小以及允许动态密钥更新等。



注意:图中带*为可选的消息发送
图 1 WTLS 协议完整握手过程

WTLS 协议体系结构分为上下两层:低层 WTLS 记录协议和高层 WTLS 管理协议。WTLS 记录协议负责封装 WTLS 高层管理协议和应用协议,为上层提供交互的传输服务;高层 WTLS 管理协议包括 WTLS 握手协议、Alert 协议和 ChangeCipherSpec 协议,其中最主要的是 WTLS 握手协议,它负责 WTLS 两端的身份认证并安全地协商一个会话密钥。图 1描述了一次完整的 WTLS 握手过程。

首先,客户端向服务端发送 ClientHello 消息,服务端响应一个 ServerHello 消息。这两个 Hello 消息交换完成本次 WTLS 会话参数的协商和一对随机数交换。

之后,客户端和服务端双方执行认证和密钥交换过程。

收稿日期:2004 - 09 - 20 基金项目:中科院知识创新工程下一代因特网综合环境(2001AA2130)子项目(2001AA112136)

作者简介:叶润国(1976 -),男,江西萍乡人,博士研究生,主要研究方向:下一代网络体系结构和计算机网络安全; 冯彦君(1969 -),男,河南洛阳人,博士研究生,主要研究方向:嵌入式系统和移动无线网路安全; 虞淑瑶(1975 -),女,浙江温州人,博士研究生,主要研究方向:网络安全; 吴宇(1973 -),男,广西桂林人,博士研究生,主要研究方向:分布式系统、网络安全.

比如,服务端可能发送 ServerCertificate 消息来认证自己,并可能发送 CertRequest 消息来请求客户端认证,客户端则必须回应 ClientCertificate 和 Certificateverify 消息来向服务端认证自己。ServerKeyExchange 和 ClientKeyExchange 消息用来实现服务端和客户端与 DH 或 RSA 交换方案相关的实时公钥信息交换。客户端和服务端双方根据交换的公钥信息计算本次 WTLS 握手过程协商的共享密钥。

最后,客户端和服务端双方相互确认该新生成的会话密钥,通过双方交换 Finished 消息实现。Finished 消息中包含了一个基于 WTLS 消息序列的 MAC 值,该 MAC 值受到新协商会话密钥的加密和完整性保护。一旦会话密钥确认完毕,双方可以进行下一步真正的数据通信。

1.2 ECMQV 介绍

基本 ECMQV[3,4]是一种对称的认证和密钥交换方案,它要求通信双方分别拥有一对静态公私钥,并要求双方预先知道对方公钥(也可在通信过程中通过交换证书获得)。在 ECMQV 执行过程中,双方还需要分别生成一对临时的公私钥。这些临时生成的公私钥和两端的静态公私钥都基于相同的椭圆曲线密码域参数 $\{p,a,b,G,n,h\}$[3]。分别用 $A^{(w_A,W_A)}$ 和 $B^{(w_B,W_B)}$ 表示 A 和 B 拥有的静态公私钥对,用 $A^{(r_A,R_A)}$ 和 $B^{(r_B,R_B)}$ 表示 A 和 B 临时生成的公私钥对,下面介绍基本的 ECMQV 交换过程:

首先,A 生成一个随机数 r_A,$1 \leqslant r_A \leqslant n-1$,计算 $R_A = r_A . G$,并将 R_A 以及经过认证的静态公钥 W_A 发送给 B;

之后,B 生成一个随机数 r_B,$1 \leqslant r_B \leqslant n-1$,计算 $R_B = r_B . G$,并将 R_B 以及经过认证的静态公钥 W_B 发送给 A;

A 检查公钥 W_B 和 R_B 有效性,通过公式(1),(2)计算共享密钥 K:

$$s_A = (r_A + \phi(R_A) . w_A) \bmod n \quad (1)$$

$$K = h . s_A (R_B + \phi(R_B) . W_B) \quad (2)$$

公式(1),(2)中,h 和 n 均为椭圆曲线域参数,$\phi(R)$ 为一函数,其值为椭圆曲线上点 R 的 x 坐标值后半部分,其比特位数正好为 n 的一半。因而,公式(2)中标量乘 $\phi(R_B) . W_B$ 的计算开销只是正常标量乘的一半。

B 检查公钥 W_A 和 R_A 有效性,通过公式(3),(4)计算共享密钥 K:

$$s_B = (r_B + \phi(R_B) . w_B) \bmod n \quad (3)$$

$$K = h . s_B (R_A + \phi(R_A) . W_A) \quad (4)$$

公式(2)和(4)中计算得到的 K 即为 A 和 B 的共享密钥,它为椭圆曲线上一个点。至此,基本的 ECMQV 过程结束。由于基本 ECMQV 过程采用隐式认证方式,所以,通信两端还需要选择一种合适的方案来确认该共享密钥 K 的一致性,从而间接地实现对参与密钥交换双方的认证。

2 一种基于 ECMQV 的 WTLS 扩展协议

2.1 WTLS 扩展协议介绍

通过将基本 ECQMV 协议集成到现有 WTLS 协议框架中,实现了一种 WTLS 扩展协议,它在略微增加无线终端计算开销的情况下明显提高现有 WTLS 协议安全性。图 2 说明了该 WTLS 扩展协议的 WTLS 握手过程。

首先,客户端向服务端发送 ClientHello 消息,它包含了客户端提议的 ECMQV 密钥交换方案以及客户标识 ID。

如果 WTLS 协议运行在数据报协议之上,并且服务端希望防止可能的拒绝服务攻击(详细分析见 2.2),它向客户端发送 MissingCookieAlert 警告消息,该消息中包含一个 Cookie 值;客户端必须重新发送 ClientHello 消息,并回送从服务端接收到的 Cookie 值(可选)。

如果服务端同意使用 ECMQV 密钥交换方案,它将发送包含 ECMQV 认证方案的 ServerHello 消息;随后服务端通过 ServerCertificate 和 ServerKeyExchange 消息发送其经过第三方认证的静态公钥 W_B 和其临时生成的公钥 R_B;然后服务端发送 CertificateReq 消息,请求客户端认证;最后,服务端发送 ServerHelloDone 消息,表示 Hello 过程结束,它将等待来自客户端的响应。

客户端首先通过 ClientCertificate 和 ClientKey-Exchange 消息发送其经过第三方认证的静态公钥 W_A 以及其临时生成的公钥 R_A;此时,客户端根据公式(1),(2)计算 ECMQV 共享密钥 K,并将 K 转换成一个二进制串 R,此 R 即为本次 WTLS 握手的 PreMasterKey;然后,客户端根据 WTLS 会话密钥推导公式导出加密和认证会话密钥;最后,客户端发送 ChangeCipherSpec 和 Finished 消息,向服务端确认该新生成的会话密钥。

服务端根据公式(3),(4)计算 ECMQV 共享密钥 K,并将 K 转换成一个二进制串 R,此 R 即为本次 WTLS 握手的 PreMasterKey;然后,服务端根据 WTLS 会话密钥推导公式导出加密和认证会话密钥,并使用这些会话密钥来确认客户端发送的 Finished 消息,如果确认失败,则向客户端发送 MAC 解码错误消息,否则;发送 Change-CipherSpec 和 Finished 消息,向客户端确认其生成的会话密钥。

最后,客户端对服务端发送的 Finished 消息进行确认,一旦确认成功,则本次 WTLS 握手过程成功完成。

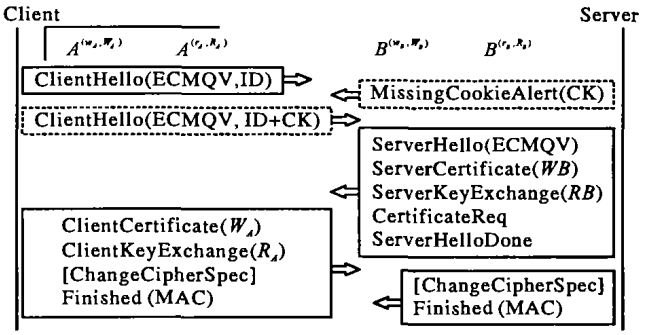


图 2 基于 ECMQV 认证方案的 WTLS 握手过程

由于 ECMQV 采用隐式认证方式,所以,服务端无需通过对 ServerKeyExchange 消息进行显式签名来认证自己;客户端也无需通过发送 CertificateVerify 消息来显式认证自己。该 WTLS 扩展协议对现有 WTLS 规范的消息格式扩展如下所示。

消息格式定义语言请参考文献[1],这里只书写了与 ECMQV 密钥交换方案有关的消息扩展部分,因而它必须和文献[1]中定义的相关消息格式合并在一起。这里定义了一种认证方案,并对现有 WTLS 规范中 ServerKeyExchange 和 ClientKeyExchange 消息格式进行了扩展。

```
enum { ECMQV (19), (255)} KeyExchangeAlgorithm
enum {Missing_Cookie_Alert (101) } AlertDescription
enum {AEAS_ID (253) } IdentifierType
Struct
  { IdentifierType identifier_type;
    Select ( identifier_type)
      { Case AEAS_ID: Opaque id_cookie <1..2^8 - 1 >;}
  } Identifier;
Struct
  { ParameterSpecifier Parameter_specifier;
    select ( KeyExchangeAlgorithm)
```

```
    { case ECMQV: ECPublicKey params; }
                                    /*服务端临时公钥 RB */
  }ServerKeyExchange;
Struct
  { select (KeyExchangeAlgorithm)
    { case ECMQV: ECPublicKey param;
                                    /*客户端临时公钥 RA */
    } exchange_keys;
  }ClientKeyExchange;
Struct                              /*其余参数没有列出来*/
  { uint8 Resume_Counter;           /*当前的恢复次数*/
    uint8 Max_Resume_Counter;       /*允许恢复的最大次数*/
  }SecurityParameters;
```

2.2 采用 Cookie 技术抵制拒绝服务攻击

WTLS 协议支持数据报传输协议,因此,它可能遭受两种拒绝服务攻击:1)攻击者通过发送 ClientHello 消息来消耗服务端计算资源和存储资源;2)通过伪造 ClientHello 消息的源 IP 欺骗服务端向无辜节点发起反射放大拒绝服务攻击。为了抵制这两种拒绝服务攻击,为 WTLS 扩展协议设计了一种服务端 Cookie 方案(见图 2),其工作原理为:当服务端接收到客户端 ClientHello 消息后并不急于处理该消息,而是向客户端发送一个 MissingCookieAlert 警告消息,该消息中包含了一个服务端生成的 Cookie;客户端必须再次发送 ClientHello 消息,并回送这个 Cookie;服务端通过检查该 Cookie 值来验证客户端 IP 的真实性;为了避免对 ClientHello 消息格式进行改动,这里采用捎带方式来回送 Cookie 值,即客户端将用户 ID 与 Cookie 级联在一起,服务端可以直接从 ID 中抽取 Cookie 值。

对于 WTLS 简化握手过程,同样可以采用服务端 Cookie 方法抵制拒绝服务攻击。但是,我们设计了一种客户端 Cookie 方法,目的在于减少 WTLS 简化握手过程的消息交换次数。该客户端 Cookie 基于客户端欲恢复的 WTLS 会话信息,计算公式为 $Cookie = HMAC(PMK, SID, Counter, SIP, DIP)$,其中:*SID* 和 *PMK* 分别为该预恢复的 WTLS 会话的标识 ID 和预主密钥 PreMasterKey; *Counter* 为一计数器,它记录了该 WTLS 会话的恢复次数,初始值为 1,最大值为该 WTLS 会话允许恢复的最大次数,每成功恢复一次 WTLS 会话该 *Counter* 加 1; *SIP* 和 *DIP* 分别为源和目的 IP 地址。服务端接收到 ClientHello 消息后,根据其缓存的 WTLS 会话数据验证该 Cookie。由于 PremasterKey 是秘密的,因此攻击者无法伪造该 Cookie。

3 WTLS 扩展协议安全性分析

由文献[4]可知,基本 ECMQV 协议拥有会话密钥完全前向安全属性,即在一个或多个静态密钥泄密情况下不会导致先前协商的会话密钥泄密,并且,当临时私钥 r_A 和 r_B 泄密时仍然能够保障静态密钥以及协商的会话密钥安全性;此外,基本 ECMQV 协议能够抵制 Subgroup 攻击,即它能够防止协商的会话密钥局限在一个极小的点集合中,以至于攻击者能够很容易地猜测到协商的共享密钥。由于基本 ECMQV 协议不使用显式方式来确认共享密钥 K 的一致性,因而,它并不具备 Unkown-Share-Key 安全属性[5]。但是,在我们的 WTLS 扩展协议中,WTLS 握手过程中的 Finished 消息交换能够实现对 ECMQV 共享密钥 K 的确认,因而,该 WTLS 扩展协议具备 Unknown-Shared-Key 安全属性。

基本 ECMQV 协议能够提供对客户端和服务端的基于证书的双向认证,并能够在认证完毕后协商一个安全的共享密钥用于保护后续的数据交换,因而,集成了基本 ECMQV 协议的 WTLS 扩展协议能够提供类型 3 的安全服务。

4 各方案性能评估与比较

本文从用户角度比较 WTLS 扩展协议密钥交换方案与 ECDH-ECDSA、ECDHE-ECDSA 密钥交换方案[6]的性能。选择 ECDH-ECDSA 和 ECDHE-ECDSA 密钥交换方案的原因是:ECDH-ECDSA 代表一种最简单的密钥交换方案,而 ECDHE-ECDSA 代表目前安全性最高的密钥交换方案。

一个密钥交换方案性能主要取决于它所需要的公钥计算开销,因而,可以从公钥操作次数(这里公钥操作包括公钥签名、验证、加密和解密)来评估一个密钥交换方案的性能。上述三种密钥交换方案都基于椭圆曲线密码系统,所有公钥操作都表现为标量乘,因此,只需要计数各方案所需要的标量乘数。这里假设对服务端证书的验证只需执行一个 ECDSA 签名验证操作(一个 ECDSA 签名验证操作需要 2 个标量乘)。

WTLS-ECMQV 方案性能:基本 ECMQV 协议包含三个标量乘运算,即 $r_A \cdot G$、$\phi(R_B) \cdot W_B$ 和 $h \cdot s_A(R_B + \phi(R_B) \cdot W_B)$;由于 $\phi(R_B) \cdot W_B$ 只算半个标量乘,因而,基本 ECMQV 协议只需要 2.5 个标量乘;由于对服务端证书签名有效性验证需要 2 个标量乘运算,因此,WTLS-ECMQV 方案总共需要 4.5 个标量乘运算。如果采用离线方式预先计算 $r_A \cdot G$, 以及验证服务端证书有效性,则 WTLS 扩展协议密钥交换方案在线标量乘为 1.5 个。

ECDH-ECDSA 性能:在该方案中,客户端只需执行 1 个与 ECDH 算法的标量乘(即 $r_A \cdot W_B$) 以及对服务端证书签名的有效性验证,因而,其总共需要的标量乘数为 3 个。如果采用离线方式预先验证服务端证书的有效性,则其在线标量乘数为 1 个。

ECDHE-ECDSA 性能:在该方案中,客户端需要执行 7 个标量乘,其中 2 个用于 ECDH 交换(它们是 $x \cdot G, xy \cdot G$),1 个用于 CertVerify 消息签名操作,2 个用于对服务器端 ServerKeyExchange 消息中 ECDSA 签名的验证,另 2 个用于对服务端证书有效性验证。如果采用离线方式预先计算 $x \cdot G$、验证服务端证书有效性以及签名算法中的临时公私钥,则在线标量乘为 3 个。

可以看出,WTLS-ECMQV 方案具有比 ECDHE-ECDSA 更好的性能,它所能提供的安全性却与 ECDHE-ECDSA 相当;WTLS-ECMQV 认证方案的标量乘开销只略高于 ECDH-ECDSA 方案,但是,其安全性却比 ECDH-ECDSA 方案高很多。

参考文献:

[1] WAP Forum. Wireless Application protocol-Wireless Transport Layer Security Specification, Ver-06 [EB/OL]. http://www.wapforum.org, 2001.

[2] DIERKS T, ALLEN C. The TLS Protocol Version 1.0, IETF RFC 2246[S]. 1999.

[3] Certicom, Corp. SEC 1: Elliptic Curve Cryptography Version 1.0 [EB/OL]. http://www.secg.org/collateral/sec1.pdf, 2000-09.

[4] LAW L, MENEZES A, QU M, *et al*. An Efficient Protocol for Authenticated Key Agreement [R]. Technical Report CORR 98-05, Dept. of C&O, University of Waterloo, Canada, 1998.

[5] BURTON S, JR KALISKI. An unknown key-share attack on the MQV key agreement protocol [J]. ACM Transactions on Information and System Security, 2001, 4(3).

[6] GUPTAV, BLAKE-WILSONS, MOELLERB, *etal*. ECCCipherSuitesfor TLS[Z]. IETF Draft(Work in progress), 2004.