DOI:10.13878/j.cnki.jnuist.2021.05.017

许国栋[1]　刘光杰[1]　乔森[1]　陆赛杰[2]　赵华伟[3]

基于改进无证书公钥密码的轻量级 DTLS 协议设计

摘要

物联网在快速发展的同时,其数据交互容易遭受各种攻击.为了保证物联网传输层协议 UDP 传输数据的安全,在 TLS 协议架构基础上扩展形成了支持 UDP 数据报安全传输的 DTLS(DatagramTLS)协议.现行的 DTLS 协议基于公钥证书密码体制,证书管理复杂、网络通信开销大,难以满足物联网等资源受限型网络的安全通信需求.本文提出一种基于离散对数的改进无证书公钥密码方案,设计了适应资源受限网络的轻量级 DTLS 协议,并基于嵌入式 SSL 库 wolfSSL 进行了协议实现.从通信开销和握手连接时间两方面,将本文提出的基于改进无证书公钥密码的 DTLS 协议分别与基于传统公钥证书的 DTLS 协议及基于身份标识的 DTLS 协议进行了对比实验.实验结果表明,在保证安全性的前提下,基于无证书的 DTLS 协议在通信开销和握手连接时间方面均优于基于公钥证书的 DTLS 协议和基于身份标识的 DTLS 协议.

关键词

物联网;离散对数;无证书;轻量级;DTLS 协议

中图分类号 TP309.7

文献标志码 A

收稿日期 2021-04-18

资助项目 国家自然科学基金(U1836104,61801073,62072250)

作者简介

许国栋,男,硕士生,研究方向为网络安全与通信.20191218017@nuist.edu.cn

刘光杰(通信作者),男,博士,教授,研究方向为网络安全与通信.gjieliu@gmail.com

1 南京信息工程大学 电子与信息工程学院,南京,210044

2 南京地铁建设有限责任公司,南京,210000

3 北京城建设计发展集团股份有限公司,北京,100037

0 引言

随着万物互联时代的到来,物联网技术正深刻改变着世界.指数级增长的物联网设备,在给人们带来便利的同时,也面临着众多威胁和安全挑战[1].物联网安全逐渐引起了工程界和学术研究者的广泛关注.在众多物联网安全问题中,协议安全是实现点到点和端到端安全通信的基石,因此也成为了物联网安全研究的热点.网络协议研究者起初设计网络协议时主要考虑了可用性,而忽略了安全性问题.作为传输层协议典型代表的 TCP、UDP 协议本身也不具备安全性.SSL/TLS 协议在 TCP 协议之上提供数据的加密传输服务,而 DTLS 协议(DatagramTLS)扩展自 TLS 协议架构,为 UDP 协议提供端到端的安全通道.DTLS 协议也可与物联网相关应用层协议结合来实现安全通信.例如,CoAP 协议[2]经 DTLS 加密后形成的 CoAPs 协议,被广泛应用于物联网的端到端安全通信中[3-4].

握手是 DTLS 协议完成身份认证和秘钥协商的必要机制.然而,传统 DTLS 协议的握手过程更多是基于公钥证书,但基于证书的公钥密码体系在证书管理、身份认证与加密等方面存在不足.一方面,证书管理包括证书的撤销、发放和存储等环节,增加了系统的开销;另一方面,身份认证与加密包括通信终端的身份合法性认证、基于公钥的传输数据加密等过程,增加了计算开销.因此,本文主要研究基于无证书公钥密码的 DTLS 握手协议.针对该问题,研究者们也探索了诸多无证书[5]的解决方案.2003 年,Boneh 等[6]构建了第一个基于身份的加密方案(Identity Based Encryption,IBE).然而,IBE 方案使用双线性对运算,且用户的私钥完全由私钥生成器(Private Key Generator,PKG)生成,这不仅造成更高的计算代价,还引发出私钥托管问题.文献[7]实现了基于身份标识(Identify-Based Cryptography,IBC)的 DTLS 协议,但此方案中的用户私钥仍完全由 PKG 产生.若 PKG 节点遭受攻击,恶意节点将获取用户私钥,进而窃听合法用户间的通信内容.文献[8]提出了一种用户私钥生成方案来解决密钥托管问题,首先使用系统私钥生成初始秘钥值,再叠加上一个随机数,构成一部分用户私钥,但此方案在安全性证明方面还存在不足.文献[9]提出了一种双密钥对的方案,即 PKG 随机选取两个秘钥值作为系统私钥,并将其中一个秘钥值加上仅通过系统私钥产生的一部分用户私钥作为用户最终的部分私钥,但与文献[8]相比又增加了部分计算量,降低了方案的实

现效率,难以满足计算成本受限型节点的安全通信需求.

针对上述问题,本文改进了产生用户部分私钥的运算方法,提出了一种基于离散对数运算的无证书公钥加密(Discrete Logarithm based Certificate-Less Public Key Cryptography,DL-CL-PKC)方案,此方案主要有两个优势:一是规避了基于身份标识的方案中双线性映射带来的运算复杂度;二是此方案中私钥生成器只需要生成一对系统密钥值,节省了一对密钥值的生成过程和后续的存储空间.最后设计并实现了基于 DL-CL-PKC 的 DTLS 协议.改进的 DTLS 协议不仅简化了握手过程,并且可以在不使用证书情况下实现安全加密通信.

1 改进的无证书公钥密码体制

本文改进的无证书方案是基于离散对数运算而设计的,该体制主要包括密钥管理方案设计和密钥更新方案设计.下面将依次介绍离散对数数学难题的定义以及无证书方案中的密钥管理和密钥更新.

1.1 离散对数难题

本文主要是基于离散对数数学难题,设计了新的用户部分私钥生成算法.离散对数难题描述为:对于给定的 m,x 和 p,通过公式 $y = m^x \bmod q$ 很容易计算得出 y,但是若给定参数 y,a 和 p 想计算出 x 则极其困难. 目前来说离散对数的计算难度与 RSA 算法中大整数因数分解的难度相同,因此并没有有效的方法可以计算出模为素数的离散对数问题.本文基于离散对数难题重新设计 DTLS 握手协议主要基于两个原因:一方面 RSA 算法的加解密过程是非对称的且存在大量的模幂运算,本文采用对称加解密,提高了无证书方案的计算效率;另一方面 RSA 算法的加密公钥存储在证书中,不符合本文方案基于无证书的原则,无法达到降低系统开销的目的.基于离散对数难题,也衍生出一些相关密码体制,如 ElGmal 加密体制[10]、Diffe-Hellman 密钥交换方法[11]以及 Schnorr 数字签名方法[12].离散对数难题与大整数因数分解的安全性分析如下:

1)离散对数难题.假设随机选取大素数 p 和 q,在有限乘法群 Z_p^* 内,选取 g 作为 p 的生成元,并将 q 作为 g 的阶.本文的方案有两处涉及到离散对数运算:一是系统公钥的生成过程,首先选取系统私钥 $x \in Z_q^*$,计算系统公钥 $y = g^x \bmod q$,系统公钥 y、生成元 g 和群的阶 q 皆对外公开,在群的阶足够大的前提下,以现有的计算能力使攻击者利用已知的参数反求出系统私钥 x 极其困难,系统私钥可证安全;二是通信方完整公钥的生成,先通过计算出完整的私钥 S,再通过 $P = g^S \bmod q$ 计算出通信方完整的公钥,同样中间人攻击者可获取公开的公钥 P、生成元 g 以及 g 的阶 q,但是反求出通信方完整私钥 S 的概率极小,通信方完整私钥可证安全.

2)大整数因数分解.第一步选择两个大素数 p,q;第二步计算模数 $n = p \times q$ 和 $L = \varphi(n) = (p-1)(q-1)$,选择满足 $\gcd(e,L) = 1$ 的 e,并计算出满足 $d \cdot e \equiv 1 \bmod L$ 的 $d(1 < d < L)$;第三步是加密过程 $c \equiv m^e \bmod n$;最后一步是解密过程 $m \equiv c^d \bmod n$.只公开 n 和 e,因此攻击者想要获取解密钥 d,必须反求出大素数 p 和 q,再计算出 L, 其难度等同于大数分解.大数分解未能得到理论的证明,但是也未能从理论上证明已破译,其安全性还未可知;另外,RSA 算法的加解密是非对称的,速度较慢.综合以上原因,本文选取离散对数难题重新设计 DTLS 协议的握手过程.

1.2 密钥管理

科研人员一直致力于密钥管理的简化研究.著名密码学专家 Shamir[13] 率先提出了基于身份的密码体制的概念,作为其核心组成的私钥生成器(Private Key Generator,PKG),也是本文设计的无证书公钥密码体制中的重要组成部分.PKG 产生无证书公钥加密方案中用户的部分私钥,省略证书参与用户身份验证的步骤.

无证书公钥加密方案中密钥管理如图 1 所示.由于 PKG 单独生成 IBE 加密方案中的用户私钥存在安全隐患,所以本文提出将用户私钥的产生过程分为两个阶段:首先用户向 PKG 节点请求生成部分私钥;其次则由用户根据 PKG 节点发送的系统参数随机生成另一部分私钥,并对 PKG 节点和其他用户不可见.然后用户公布自己的公钥.密钥管理包括以下几个方面:

1)PKG 节点初始化之后,监听两个通信终端的部分私钥生成请求;

2) PKG 节点将产生的系统参数和公私钥发送给监听到请求的两个通信终端;

3) PKG 节点将根据客户端或服务端的 ID 产生相应的部分私钥发送给客户端或服务端;

4)客户端或服务端通过系统参数再随机选取一个秘密值作为另一部分私钥,并通过这两部分私钥计算出完全的私钥.

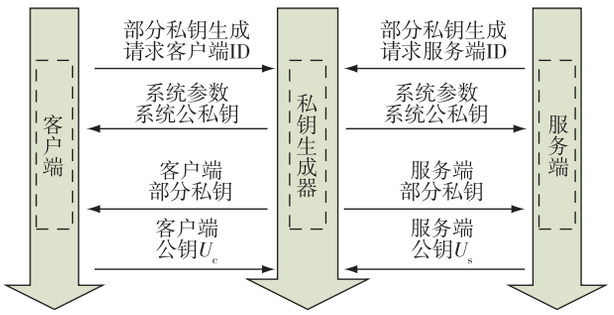

图 1　DL-CL-PKC 密钥管理方案

Fig. 1　Key management scheme for DL-CL-PKC

下面根据上述描述给出具体方案.

1) 系统参数生成

首先 PKG 节点任取大素数 a 和 b(要求满足 $b \mid a-1$),接着在有限乘法群 Z_a^* 内,选取 g 作为 a 的生成元,g 的阶为 b.PKG 节点随机选取 $x \in Z_b^*$ 作为系统的私钥,并计算 $y = g^x \bmod b$ 作为系统的公钥.选取 hash 函数 H_1、H_2 满足

$H_1:(0,1)^* \times Z_b^* = Z_b^*$, $H_2:(0,1)^{l_0} \times (0,1)^{l_1} = Z_b^*$.

公开系统参数 param:$\{a,b,g,x,y,H_1,H_2\}$.

2) 部分私钥提取

假设用户 A 的身份为 ID_A,PKG 节点计算用户 A 的部分私钥,步骤如下:

① 计算 $h_A = H_1(ID_A)$;

② 提取出用户 A 的部分私钥 $D_A = h_A^x \bmod b$.

3) 完整公私钥生成

用户通过系统参数随机选取 $\mu_A \in Z_b^*$ 作为完全私钥中部分密钥值不对外公开,通过部分私钥 D_A 和密钥值 μ_A 生成用户 A 完全的私钥 $S_A = \mu_A \times D_A$,并计算 $p_A = g^{S_A} \bmod b$ 作为公钥.

4) 加密(cl-encrypt)

用户 B 将消息发送给用户 A,步骤执行如下:

① 随机选取 $n \in Z_b^*$;

② 计算 $N = g^n \bmod b$ 和 $M = m \oplus H_2((p_A^{H_1(ID_A)})^n)$;

③ 用户 B 将 $C = (N,M)$ 发送给给用户 A.

5) 解密(cl-decrypt)

当用户 A 接收到用户 B 发送来的加密报文 C,用户 A 计算 $m = M \oplus H_2(N^{S_A})$ 恢复出明文信息.解密过程如下验证推导所示:

$$M \oplus H_2(N^{S_A}) = m \oplus H_2((p_A^{H_1(ID_A)})^n) \oplus H_2(N^{S_A}) =$$
$$m \oplus H_2((p_A^{H_1(ID_A)})^n) \oplus H_2((g^n)^{S_A}) =$$
$$m \oplus H_2((p_A^{H_1(ID_A)})^n) \oplus H_2((g^n)^{\mu_A H_1(ID_A)}) =$$
$$m \oplus H_2((p_A^{H_1(ID_A)})^n) \oplus H_2((g)^{n\mu_A H_1(ID_A)}) =$$
$$m \oplus H_2((p_A^{H_1(ID_A)})^n) \oplus H_2((p_A^{H_1(ID_A)})^n) = m.$$

1.3　密钥更新

密钥是加密通信的关键,因此为了保证系统的安全性,必须在一定周期内进行更新(主要是对用户私钥的更新).基于无证书公钥加密方案中的密钥更新过程主要包括以下两个步骤:

1) 预先设定密钥更新周期,时间周期设置得越短则系统安全性越高.为了让密钥更新的周期更加合理,可根据网络动态和需求,实时恰当地调整更新周期.

2) 当密钥更新周期到达时,用户重新生成部分私钥 $\mu_A \in Z_b^*$,且公式 $p_A = g^{S_A} \bmod b$ 和 $S_A = \mu_A \times D_A$ 表明用户完全的私钥和公钥也得到更新.由于部分私钥 μ_A 的更新,不影响 PKG 节点依公式 $D_A = h_A^x \bmod b$ 生成的另一半私钥,因此 μ_A 的更新不会增加 PKG 节点的负担.

1.4　安全性分析

在本文设计的加密体制中,主要考虑以下两种攻击类型:

第一种攻击.假设攻击者获取了系统私钥,就会带来安全威胁,即攻击者利用获取的系统私钥去生成用户的部分私钥,这里攻击者作为不可信的 PKG 节点.但是由于此方案中用户完全私钥的部分是由用户自己生成的,并且不对外公开,因此不可信的 PKG 节点无法获取用户完全的私钥,也就无法去监听与其他用户之间的通信内容.

第二种攻击.假设攻击者作为一个不合法的用户,获取的是某个合法用户完整的私钥,但是由于此方案采用了离散对数运算,不合法的用户无法通过合法用户的私钥去倒推出系统私钥,也不可能去生成其他合法用户的私钥,因此它只能与其他用户之间进行通信,是无法监听到甚至解密其他用户之间的通信内容的.

这样在系统更新用户私钥时,不合法的用户在向 PKG 节点申请部分私钥时就无法通过 PKG 节点的身份认证,原先获取的私钥也就失效了.

2　基于改进无证书公钥密码的 DTLS 协议设计

本文设计的基于无证书公钥加密的 DTLS 协议主要针对握手协议进行改进.由于目前 DTLS 握手协议双方在进行通信认证时仍然需要加载证书,增加了握手过程的复杂性和成功建立通信连接所占用的

时间.因此,在 DTLS 协议中引入改进的无证书公钥密码体制,设计新的握手协议,可有效避免证书的交换和验证的过程,减少通信双方需要传递的信息量和交互的回合数,从而缩短通信连接建立的时间.

2.1 DTLS 握手协议

DTLS 协议建立在传输层协议之上、应用层协议之下,在完成加密算法、密钥协商、服务器端认证后,即可建立应用层协议的通信.因此,后续应用层协议发送的数据都会进行加密处理,以保证网络通信中数据的安全.图 2a 给出的是一次完整的 DTLS 握手流程.DTLS v1.2 中 ClientHello 和 ServerHello 消息序列中包含一个预共享密钥(pre_shared_key),DTLS v1.3 的 ClientHello 和 ServerHello 在 DTLS v1.2 基础上又增加了一个密钥共享(key_share)的扩展部分.关于握手消息序列,文献[14]进行了较为详尽的介绍,因此本文不在此赘述.

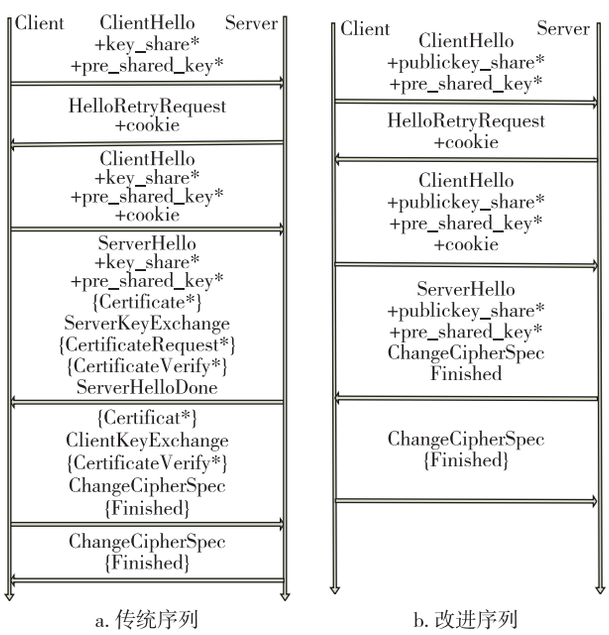

图 2 传统 DTLS 握手消息序列与改进 DTLS 握手消息序列

Fig. 2 Traditional DTLS handshake message sequences (a) and improved DTLS handshake message sequences (b)

2.2 改进 DTLS 握手协议设计

本文提出的改进 DTLS 握手协议通过+key_share*扩展在 Client 端与 Server 端间交换公钥信息,并将其重命名为+publickey_share*.而 Client 端和 Server 端的私钥,一部分由私钥生成器根据两者的 ID 生成,另一部分由自身根据系统参数随机生成.本文所设计的握手协议如图 2b 所示,握手过程如下:

1) Client 端向 Server 端发送一个没有携带 cookie 值的 ClientHello 报文,以确认 Server 端已经启用.

2) Server 端收到此报文后,回复携带 cookie 值的 HelloRetryRequest 给 Client 端.

3) Client 端从 HelloRetryRequest 报文中取出 cookie 值,放入新的 ClientHello 报文中发送,扩展+publickey_share*中附带自身的公钥信息.

4) Server 端验证 Client 端发送的 ClientHello 报文中携带的 cookie 值,选择合适的加密算法(本文设计为 DTLS_PKC_WITH_AES_128_CBC_SHA512)和压缩算法,并生成随机数,通过 ServerHello 报文发送到 Client 端,扩展中携带自身的公钥信息.

5) Server 端发送 ChangeCipherSpec 报文告知 Client 端密码规范已发生改变,接下来将利用协商相同的安全密钥来保障数据的安全传输.Server 端通过密钥交换算法生成预主密钥,预主密钥通过与随机数进行伪随机运算生成主密钥以及会话密钥.

预主密钥生成过程如下:

假设 Client 端和 Server 端通过 ClientHello 和 ServerHello 报文互换各自的公钥信息 p_c 和 p_s,Client 端利用自身私钥 S_C 与 Server 端公钥 p_s 进行离散对数运算得到共享密钥 $p_s^{S_C} \bmod q$.同样地,Server 端利用自身私钥 S_S 与 Client 端公钥 p_c 进行离散对数运算得到共享密钥 $p_c^{S_S} \bmod q$.根据取模运算规则可验证其正确性:

$$\begin{aligned} K &= p_s^{S_C} \bmod q = (g^{S_S} \bmod q)^{S_C} \bmod q = \\ &(g^{S_S})^{S_C} \bmod q = g^{S_S S_C} \bmod q = \\ &(g^{S_C})^{S_S} \bmod q = (g^{S_C} \bmod q)^{S_S} \bmod q = \\ &p_c^{S_S} \bmod q = K. \end{aligned}$$

6) Server 端利用协商好的算法和密钥,将 Finished 报文加密后发送给 Client 端.此报文用于验证密钥交换是否成功.

7) Client 端在收到 Server 端发送的 ServerHello、ChangeCipherSpec 和 Finished 报文后计算出相应的会话密钥,解密 Finished 报文并对其数据进行验证.验证通过发送 ChangeCipherSpec 报文告知 Server 端接下来使用协商相同的安全密钥来保障数据的安全传输.

8) Client 端利用协商的算法和密钥,加密 Finished 报文后发送给 Server 端,Server 端解密此报文并验证,验证通过后,两者即可正式建立连接.

对比图 2a 与 2b 可以发现,与传统的 DTLS 握手

过程相比,本文在握手消息 ClientHello 和 ServerHello 中使用扩展+publickey_share * 交换公钥信息,并省略了握手过程中与证书相关的交互报文.

2.3 无证书公钥加密算法及握手协议实现

握手协议中的无证书公钥加密算法(Discrete Logarithm based Certificate-Less Public Key Cryptography,DL-CL-PKC)可以通过 GMP 库[15]来实现.GMP 库是一种高精度的算术运算库,可支持对浮点数、有符号整数等数据类型进行相关算术运算.它旨在为所有需要更高精度的应用程序提供最快的算法.DL-CL-PKC 算法主要分为 PKG 节点初始化函数、部分私钥生成函数、完全私钥生成函数、用户公钥生成函数以及加密和解密函数.为了程序后续的调用,将主要算法进行了封装.主要函数接口如表 1 所示.

表 1 DL-CL-PKC 算法函数接口

Table 1 Function interfaces for the DL-CL-PKC algorithm

函数接口名	函数功能
setup_pkg( )	PKG 节点初始化
set_partkey( )	根据用户身份生成部分用户私钥
set_privatekey( )	生成用户完全的私钥
set_publickey( )	生成用户公钥
pkg_encrypt( )	Client 端利用 Server 端的公钥对消息 m 加密
pkg_decrypt( )	Server 端利用自己的私钥对加密消息 c 解密

通过上述封装的算法,可以在 wolfSSL[16]库中实现基于该算法的 DTLS 握手协议,具体可分为以下两个步骤:

1)定义新的加密套件

DTLS 握手过程所涉及的多种算法(密钥交换、认证、对称加密及哈希算法),都是通过加密套件进行定义的.测试程序会根据选择的加密套件去执行相应算法,本文所定义的加密套件是 DTLS_PKC_WITH_AES_128_CBC_SHA512,此加密套件表明密钥协商算法是 DL-CL-PKC,对称加密算法采用 128 位的 AES[17]算法,加密模式采用 CBC 模式,哈希算法采用 SHA512.

2)密钥协商

Client 端和 Server 端互换公钥信息后,Client 端使用密钥交换算法计算预主密钥,并加密发送给 Server 端,Server 端解密得到预主密钥,接着 Server 端通过预主密钥结合随机数做伪随机算法后得到主密钥和后续会话的密钥.

通过以上两个步骤,双方即可在省略证书加载和认证过程的情况下正式建立连接.

3 实验分析

本文分别在 Ubuntu 18.04.3、CentOS 3.10.0 和 Deepin 5.4.50 系统上实现基于无证书公钥密码的 DTLS 协议.Client 端将本文定义的加密套件通过 ClientHello 报文发送给 Server 端,Server 端选择此加密套件,实施本文提出的握手方法.如果 Server 端没有选择此加密套件也可以跳转到其他加密套件,本文以 PSK 和 ECDHE 为两个备用套件.

实验环境为戴尔台式机,处理器是 Core(TM)i7-9700,拥有内存 32 GB,虚拟机类型为 VMware Workstation Pro,虚拟机分配虚拟内存为 4 GB,虚拟机系统为 Ubuntu 18.04.3、CentOS 3.10.0 和 Deepin 5.4.50.

通过将本文设计的基于 DL-CL-PKC 算法的 DTLS 握手协议与 DTLS 两个备用套件 PSK[18]、ECDHE[19]和基于身份标识的 DTLS 握手协议方案[7]进行对比,通过比较它们的消息传输开销和连接时延来检验本文所设计方案的性能.

3.1 消息传输开销

通过 wireshark 分别抓取 4 种方案交互传输的报文,对比 4 种握手过程中产生的信息字节数,结果如表 2 所示.首先可以看出 DTLS 备用的 2 个套件的交互次数都是 6 次,而本文提出的方案与基于身份标识的 DTLS 握手协议方案都将交互次数减少了 1 次;其次在这些交互过程中由于剔除证书发送和验证过程,使本文设计的方案和基于身份标识的 DTLS 握手协议方案所需传输的握手消息数量也有所减少,相比 PSK,发送的消息数减少了 3 条,而相比 ECDHE,减少了 8 条,大大降低了握手过程中的通信流量,且本文所提方案的握手消息字节略小于基于身份标识的 DTLS 握手协议方案.

表 2 4 种方案消息传输开销

Table 2 Messaging overhead for the four schemes

方案	交互次数	握手消息数	握手消息/B
PSK[18]	6	11	1 821
ECDHE[19]	6	16	5 787
PBC[7]	5	8	2 349
DL-CL-PKC	5	8	2 315

3.2 连接时间

分别测出 4 种方案在以 linux 为内核的 3 种操作系统中的连接时间,测量方法是记录发送第一个报文开始到建立连接所用的时间,并且对这 4 种方

案的连接时延进行多次测量,最后求出平均值,结果如图 3 所示.由于 PSK 是直接以双方事先就已经约定好的密钥为基础来进行加密通信的,所以此方案节省了在握手过程中计算密钥所花费的时间,并且 PSK 方案[18]也不需要进行证书的认证,同样节省了连接时间,最终都可以在较小时延下完成连接,但是 PSK 方案安全性较低.ECDHE 方案[19]需要解析证书以作认证,所以花费时间较多,完成连接的时延最大.PBC 方案[7]同样实现了基于无证书的 DTLS 握手过程,节省了连接时间,但此方案没有考虑到 PKG 节点遭受攻击造成系统私钥和合法用户私钥泄露的问题.而本文所提方案在考虑安全性的前提下,省去证书验证环节,也可以在相对较小的时延下建立连接.和 ECDHE 方案[19]相比,本文所提的方案将连接时延降低了 40%以上;和 PBC 方案[7]采用双线性对运算获取共享密钥和后续会话密钥相比,本文采用运算速度更快的离散对数运算,因此握手连接时间更短.

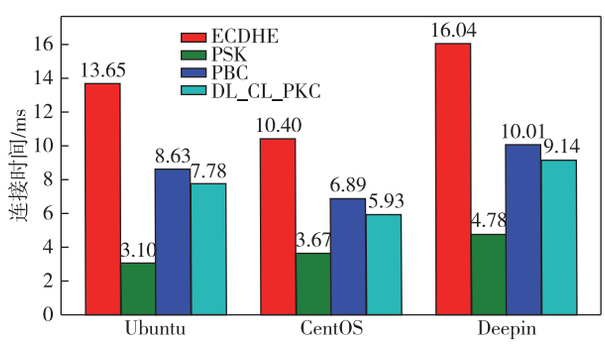

图 3　3 种操作系统下的 4 种方案连接时间对比

Fig. 3　Comparison of connection time between four schemes under three operating systems

4　结束语

本文利用离散对数运算,实现了 DL-CL-PKC 算法,设计并实现基于改进离散对数运算的无证书公钥密码的 DTLS 握手协议,此方案可将 Client 端和 Server 端产生的公钥通过扩展+publickey_share 分别发送到对端,参与后续会话密钥的生成,简化了握手过程,减少了交互次数,节省了通信开销.实验结果表明,本文在不降低原先 DTLS 安全性的基础上,很大程度地缩短了连接建立时间.下一步工作将从物联网应用层协议入手,将本文设计的无证书方案与应用层协议集成起来并实现基于无证书 DTLS 协议的应用层协议的加密.

参考文献

References

[1] Grammatikis P R,Sarigiannidis P G,Moscholios I D.Securing the Internet of Things:challenges,threats and solutions[J].Internet of Things,2019,5:41-70

[2] Iglesias-Urkia M,Orive A,Urbieta A,et al.Analysis of CoAP implementations for industrial Internet of Things:a survey[J].Journal of Ambient Intelligence and Humanized Computing,2019,10(7):2505-2518

[3] Kim Y S,Kim K T,Lee B K.CoAP/6LoWPAN-based smart home network system using DTLS[J].The Journal of the Institute of Internet,Broadcasting and Communication,2018,18(6):53-61

[4] Park C S.Security architecture for secure multicast CoAP applications[J].IEEE Internet of Things Journal,2020,7(4):3441-3452

[5] 张振超,刘亚丽,殷新春,等.无证书签名方案的分析及改进[J].密码学报,2020,7(3):389-403
ZHANG Zhenchao,LIU Yali,YIN Xinchun,et al.Analysis and improvement of certificateless signature schemes[J].Journal of Cryptologic Research,2020,7(3):389-403

[6] Boneh D,Franklin M.Identity-based encryption from the Weil pairing[J].SIAM Journal on Computing,2003,32(3):586-615

[7] 李鹏坤,王小峰,苏金树,等.基于标识密码的数据报传输层安全协议[J].软件学报,2017,28(增刊 2):90-97
LI Pengkun,WANG Xiaofeng,SU Jinshu,et al.Datagram transport layer security protocol with identity-based cryptography[J].Journal of Software,2017,28(sup2):90-97

[8] Baek J,Safavi-Naini R,Susilo W.Certificateless public key encryption without pairing[C]//International Conference on Information Security,2005:134-148

[9] 姚永军.基于离散对数的无证书密码及其在 MANET 密钥管理中的应用[D].南京:南京理工大学,2014
YAO Yongjun.Discrete logarithm-based certificateless cryptography and its application to MANET key management[D].Nanjing:Nanjing University of Science and Technology,2014

[10] Hwang M S,Chang C C,Hwang K F.An ElGamal-like cryptosystem for enciphering large messages[J].IEEE Transactions on Knowledge and Data Engineering,2002,14(2):445-446

[11] Kumar S,Singh R K.Secure authentication approach using Diffie-Hellman key exchange algorithm for WSN[J].International Journal of Communication Networks and Distributed Systems,2016,17(2):189-201

[12] Wahyudi E,Efendi M M,Subli M,et al.Penerapan digital signature scheme dengan metode schnorr authentication[J].Explore,2020,10(1):23.DOI:10.35200/explore.v10i1.360

[13] Shamir A.Identity-based cryptosystems and signature schemes[C]//Workshop on the Theory and Application of Cryptographic Techniques,1984:47-53

[14] Kumar P M,Gandhi U D.Enhanced DTLS with CoAP-

based authentication scheme for the Internet of Things in healthcare application [J]. The Journal of Supercomputing,2020,76(6):3963-3983
[15] Granlund T. GNU MP 6.0 multiple precision arithmetic library[M].London:Samurai Media Limited,2015
[16] wolfSSL.wolfSSL embedded SSL/TLS library[EB/OL]. [2021-04-01]. https://www.wolfssl.com/products/wolfssl/
[17] Arpaia P, Bonavolontá F, Cioffi A. Problems of the advanced encryption standard in protecting Internet of Things sensor networks[J]. Measurement, 2020, 161: 107853.DOI:10.1016/j.measurement.2020.107853
[18] Jayaraghavendran K. TLS/DTLS PSK identity extension [S/OL]. [2021-04-01]. TLS Working Group, 2016. https://datatracker.ietf.org/doc/html/draft-jay-tls-psk-identity-extension-01.txt
[19] Gilmore J, Weiler S, Kivinen T. Using raw public keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)[R]. RFC Editor, 2014. DOI: 10.17487/rfc7250

Lightweight DTLS protocol design based on improved certificateless public key cryptography

XU Guodong[1] LIU Guangjie[1] QIAO Sen[1] LU Saijie[2] ZHAO Huawei[3]
1 School of Electronic & Information Engineering, Nanjing University of Information Science & Technology, Nanjing 210044
2 Nanjing Metro Construction Co., Ltd, Nanjing 210000
3 Beijing Urban Construction Design & Development Group Co., Limited, Beijing 100037

Abstract The rapid development of the Internet of Things further makes its data interaction vulnerable to various attacks. To ensure the security of data transmitted by UDP, the transport layer protocol of the Internet of Things, namely the DTLS (DatagramTLS) protocol, which supports the secure transmission of UDP datagrams, has been formed on the basis of the TLS protocol architecture. However, based on certificate public key cryptography, the existing DTLS protocol has disadvantages such as complex certificate management as well as high network communication overhead, thus cannot meet the secure communication requirements of resource-constrained networks such as the Internet of Things. Here, we propose an improved certificateless public key cryptographic scheme based on discrete logarithm, and design a lightweight DTLS protocol adaptable to resource-constrained networks, and then implement the protocol based on the embedded SSL library of wolfSSL. Finally, experiments are conducted to compare the DTLS protocol based on improved certificateless public key cryptography proposed in this article with the DTLS protocol based on traditional public key certificates and the DTLS protocol based on identity markers, and experimental results verify the superiority of the proposed protocol in terms of communication overhead and handshake delay.

Key words internet of things(IoT); discrete logarithm; certificateless; lightweight; DTLS protocols