







基础的身份基私钥.需要指出的是,本文设计的通信协议基于双方处于同一个 PKG 管理的可信域.

标识管理部分主要包括两个步骤,是整个基于身份密码系统运行的基础,分别为 PKG 的初始化和私钥管理,如图 2 所示.本文以 BF-IBE 方案<sup>[10]</sup>为例,对密钥管理的流程进行介绍,该方案为目前被研究和应用最广泛的方案,本文所使用的密钥协商算法也以该方案为基础.

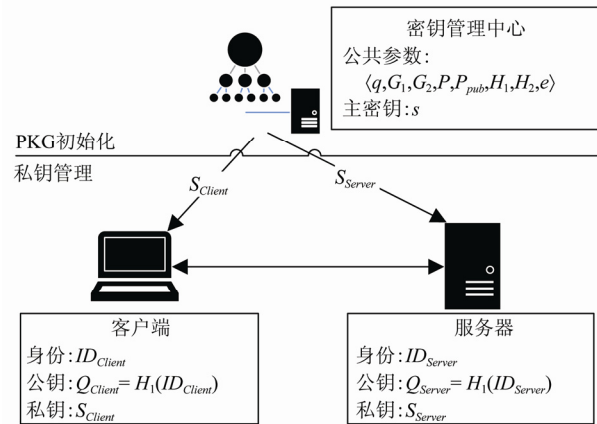


Fig.2 IBC-Based identity management

图 2 基于 IBC 的标识管理

**PKG 的初始化:**作为一个域的可信管理机构,PKG 在整个系统中最先启动和建立.首先选择某条特定的椭圆曲线,并由其上的点构成  $p$  ( $p$  为素数)阶加法循环群  $G_1$ , 其中生成元为  $P$ . 随机选择  $s \in \mathbb{Z}_p^*$ , 作为 PKG 的主私钥,计算  $P_{pub} = sP$ . 再根据群  $G_1$  选择双线性映射  $e$ , 使得  $e: G_1 \times G_1 \rightarrow G_2$ ,  $G_2$  为  $p$  阶乘法群. 最后选择相关哈希函数  $H_1: \{0,1\}^* \rightarrow G_1$ ,  $H_2: G_2 \times G_1 \times G_2 \rightarrow \{0,1\}^n$ , 其中,  $n$  为常量,表示密钥长度. 完成初始化后,在该域内广播自己的公共参数  $\langle p, G_1, G_2, e, P, P_{pub}, H_1, H_2 \rangle$ .

**私钥管理:**初始化步骤完成之后,PKG 为其域内所有生成私钥,并通过安全通道进行分发.另外,PKG 还需要负责私钥的撤销和更新.例如,管理域内的某个客户端 Client 请求加入系统,它首先向 PKG 证明自己的身份.这里的认证方式可以通过预留密钥,也可以基于传统的证书机制和 PKI 体系.虽然传统方法中存在计算和开销大的缺陷,但只需在加入系统时进行 1 次,对大的整体来说,复杂性可以忽略.

如果 Client 通过了身份认证,PKG 会对应其身份生成私钥.首先将其身份  $ID_{Client}$  (例如邮箱地址)映射到椭圆曲线上一个点  $Q_{Client} = H_1(ID_{Client})$ . 而后,将自己的主私钥与该点做点乘运算,生成 Client 的私钥,即  $S_{Client} = sQ_{Client}$ . 最后,PKG 将私钥  $S_{Client}$  通过一种安全的方式颁发给 Client 作为其私钥,供解密、签名、密钥协商等时候使用.

## 2.2 握手协议设计

Sakai 基于 BDH 问题提出了基于身份的零交互密钥协商算法(identity-based non-interactive key agreement, 简称 IBNIKA),Chen 等人<sup>[11]</sup>对其安全性使用挑战者模型和形式化方法进行了详细地证明.本文利用 IBNIKA 算法进行密钥协商,充分利用 IBNIKA 零交互的特性,设计简单、方便的握手过程,免去证书的交换和验证的过程,节省内存和带宽资源.

### 2.2.1 握手消息格式

握手过程主要完成安全参数的协商和安全会话的建立,本文新设计的握手过程中使用的消息有 ClientHello, HelloVerifyRequest, ServerHello, ChangeCipherSpec 和 Finished 这 5 种.

(1) ClientHello 消息中包括版本号、客户端生成随机数、会话标识、cookie、所支持密码套件、所支持压缩方式和扩展部分,其中扩展部分包含客户端的身份信息.其结构定义如下.

```

struct {
    ProtocolVersion client_version;
    Random random;
    SessionID session_id;
    opaque cookie<0..32>;
    CipherSuite cipher_suites<2..216-1>;
    CompressionMethod compression_methods<1..28-1>;
    Extension extensions<0..216-1>;
} ClientHello;

```

(2) HelloVerifyRequest 消息中包括版本号和服务端产生的 cookie,结构定义如下.

```

struct {
    ProtocolVersion server_version;
    opaque cookie<0..32>;
} HelloVerifyRequest;

```

(3) ServerHello 消息中包括版本号、服务端生成随机数、会话标识、密码套件、压缩算法和扩展部分,其中扩展部分包含服务端的身份信息.其结构定义如下.

```

struct {
    ProtocolVersion server_version;
    Random random;
    SessionID session_id;
    CipherSuite cipher_suite;
    CompressionMethod compression_method;
    Extension extensions<0..216-1>;
} ServerHello;

```

(4) ChangeCipherSpec 为密码变更协议,长度为一个字节,其值为 1,用于通知密码规格的改变,其结构如下.

```

struct {
    enum { change_cipher_spec(1), (255) } type;
} ChangeCipherSpec;

```

(5) Finished 消息在各自 ChangeCipherSpec 之后发送,用于验证密钥交换过程是否成功,并校验握手过程的完整性,其结构如下.

```

struct {
    Opaque verify_data[verify_data_length];
} Finished;

```

其中,verify\_data 为校验数据,该数据产生方法如下:

$$\text{PRF}(\text{master\_secret}, \text{finished\_label}, \text{Hash}(\text{handshake\_messages})).$$

表达式中,

① finished\_label

对于由客户端发送的结束消息,该标签是字符串“client finished”.对于服务端,该标签是字符串“server finished”.

② handshake\_messages

指 ClientHello 消息开始直到本消息为止(不包括本消息、密码规格变更消息和 hello 请求消息)的所有与握手有关的消息,包括握手消息的类型和长度域.

### 2.2.2 握手流程

本文设计的新的握手流程如图 3 所示.

客户端首先发送未添加 cookie 的 ClientHello 消息给服务器,当服务器收到后,发送 HelloVerifyRequest 消息给客户端,并在其中放置新生成的 cookie.客户端从收到的 HelloVerifyRequest 消息中提取出 cookie,将它添加到新的 ClientHello 消息中,然后将信息 ClientHello 消息发送给服务端.这个 cookie 交换过程的目的是为了防止 DOS 攻击.由于本文使用的 IBNIKA 密钥协商算法计算简单,不需要密钥材料的交换,发送完 ServerHello 消息,即可进行共享密钥的计算.倘若服务器对每个 ClientHello 消息都进行密钥计算,自身会因为计算量过大而宕机,更不能抵御 DoS 攻击.所以需要通过对 cookie 的发送和验证来防止这种情况.服务端将收到的 cookie 与自己保存的 cookie 进行对比,如果验证的 cookie 的结果是有效的,则继续握手过程.如果恶意用户一直发送未添加 cookie 或伪造 cookie 的 ClientHello 消息,服务器的握手过程则最多只停留在发送 HelloVerifyRequest 阶段,而不需要过多的计算量.

本文采用基于 BDH 问题提出的 IBNIKA 算法为通信双方建立共享密钥.客户端和服务端首先通过 ClientHello 消息和 ServerHello 消息交换各自的身份  $ID_{Client}$  和  $ID_{Server}$ .客户端通过服务器的身份计算其公钥  $Q_{Server} = H_1(ID_{Server})$ ,并用自身私钥  $S_{Client}$  与其进行双线性映射得到共享密钥  $e(S_{Client}, Q_{Server})$ .同样地,服务器将自身私钥  $S_{Server}$  与通过客户端的身份计算出的公钥  $Q_{Client}$  进行双线性映射得到共享密钥  $e(Q_{Client}, S_{Server})$ .根据双线性映射性质可证明其正确性:

$$e(S_{Client}, Q_{Server}) = e(sQ_{Client}, Q_{Server}) = e(Q_{Client}, Q_{Server})^s = e(Q_{Client}, sQ_{Server}) = e(Q_{Client}, S_{Server}).$$

IBNIKA 算法包含通信双方的身份信息,因此具有认证性,攻击者冒充任意一方都不能获得正确的共享密钥.

重新设计的握手协议的具体流程如下.

- (1) 客户端发送 cookie 为空的 ClientHello 消息,询问服务端是否在线.
- (2) 服务器收到一个没有 cookie 的消息时,通过 HelloVerifyRequest 返回一个 cookie 值.
- (3) 客户端提取 cookie 并放在新的 ClientHello 消息中发送,还包括客户端所支持的密码套件和压缩算法,以及产生的随机数,扩展中附带自己的身份信息“client@company.com”.
- (4) 服务器验证 cookie,选择合适的密码套件(本文中选择的密码套件为 TLS\_IBC\_WITH\_AES\_128\_CCM\_8),选择合适的压缩算法,创建一个会话 ID,并产生一个随机数,通过 ServerHello 消息发送给客户端,扩展中附带自己的身份信息“server@company.com”.
- (5) 服务端发送 ChangeCipherSpec 消息通知密码规格的改变,接下来的数据将使用新协商的安全参数来保护.服务端通过对方的身份信息和 PKG 公布的公共参数产生其公钥,并通过 IBNIKA 算法产生预主密钥,预主密钥通过与随机数进行伪随机运算产生主密钥和会话密钥.
- (6) 服务端使用新协商的算法和密钥,加密并发送 Finished 消息,用于验证密钥交换过程是否成功,并校验握手过程的完整性.
- (7) 客户端 ServerHello 消息并且收到 ChangeCipherSpec 消息后,计算会话密钥;收到 Finished 消息后,对其进行解密并验证校验数据,验证通过后发送 ChangeCipherSpec 消息通知客户端接下来的数据将使用新协商的安全参数来保护.
- (8) 客户端使用新协商的算法和密钥,加密并发送 Finished 消息.服务端收到 Finished 消息后,对其进行解密

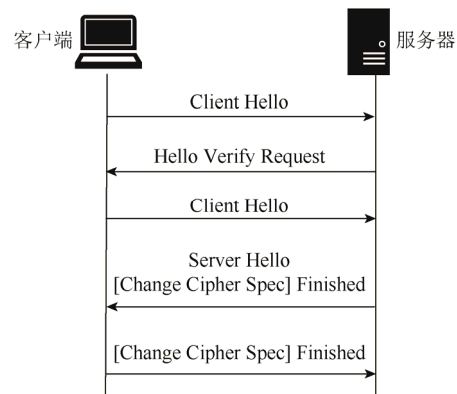


Fig.3 IBC-Based DTLS handshake

图 3 基于 IBC 的 DTLS 握手流程

并验证校验数据,验证通过后,二者正式建立连接.

### 3 实验分析

本文在 Linux 内核的协议栈中实现 DTLS,并将本文提出的基于 IBC 的握手协议以可选密码套件的形式提供给用户.客户端发送带有新密码套件的 ClientHello 消息,当服务端选择此套件,则进行本文设计的握手方案;当服务端未选择此套件,依旧使用自带的 PSK 和 ECDHE 方案.这样使得本文设计的握手协议可以与 DTLS 完全兼容.本文提供的密码套件为 DTLS\_IBC\_WITH\_AES\_128\_CBC\_SHA256,表示密钥交换和认证算法为 IBC,对称加密算法为 128 位 AES,加密模式为 CBC,完整性校验算法为 256 位 SHA.

本文实验环境为个人台式机,处理器为 Core i5-4570,内存为 4GB,虚拟机软件为 VMware Workstation Pro,虚拟内存为 1GB,实现系统为 Linux Ubuntu 3.13.0.

DTLS 配备两个密码套件:PSK 方法和 ECDHE 方法.我们将本文提出的基于标识密码的传输层安全协议与 DTLS 自带的两种方案进行对比,通过比较它们的通信开销和连接延迟来测试本文所提方案的性能.

PSK 方案使用的是预先配置的静态密钥,虽然在通信过程中可以认为对方能解密本方发出的消息,即为预期的通信对象,但是存在着密钥刚开始建立时就存在欺骗的情况,因此不满足认证性.PSK 方案不交换密钥协商材料,ClientKeyExchange 消息中只发送与密钥对应的标识码,因此没有进行密钥协商的过程.ECDHE 方案中,双方先交换证书验证身份,后交换密钥协商材料,因此完成了双方的认证和密钥协商过程.本文设计的基于 IBC 的安全传输协议,在密钥协商的同时,完成了通信双方的认证.其安全性等同于 ECDHE,强于 PSK.

#### 3.1 通信开销

通过抓取 3 种方案握手过程中传输的报文,我们对比了 3 种握手过程中产生的通信流量,结果见表 1,其中 ECDHE 中使用的证书大小为 89 字节.在本文设计的基于 IBC 的握手过程中,双方交互次数共 5 次,相对于 PSK 和 ECDHE 的方案,交互次数节省 1 次;握手消息数量共 8 条,比 PSK 方案消息数量节省 2 条,比 ECDHE 方案消息数量节省 7 条.由于在基于 IBC 的握手过程中不再需要处理和传输通信双方的证书,因此相对于 ECDHE 方案,其传输开销缩减了 63%,充分减少了握手通信流量.

Table 1 Transmission overhead of three schemes

表 1 3 种方案传输开销对比

方案	交互次数	消息数量	字节数
PSK	6	10	458
ECDHE	6	15	1 070
IBC	5	8	397

#### 3.2 连接延迟

我们测量了 3 种通信方案的连接延迟,测量标准为从发起连接的第 1 个报文开始到连接建立的时间.通过对 3 种方案通信连接时间进行多次测量,得到它们的平均连接建立时间,结果如图 4 所示.由于 PSK 方案预先设置好共享密钥,因此在握手过程中节省了计算密钥的时间开销.另外,PSK 方案也不需要传递和验证证书,因此安全性较低的 PSK 方案远远优于其他方案,仅需 2.18ms 即可完成连接建立;ECDHE 方案需要验证证书和交换密钥材料,建立连接时间变长,需要 199.78ms;与 ECDHE 方案安全性相当的 IBC 方案因为优化通信量,同时省去了验证证书的时间,仅需要 101.04ms 即可建立通信,比 ECDHE 方案节省了将近一半的时间.

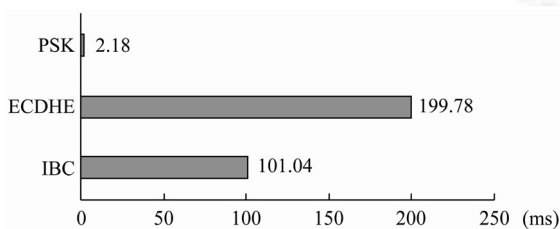


Fig.4 Connection setup time of three schemes

图 4 3 种方案连接建立时间对比

## 4 结束语

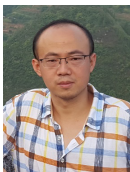
本文将标识密码应用于 DTLS,重新设计了 DTLS 的握手协议.本文采用 IBNIKA 进行密钥协商,用户的身份即为公钥,并参与会话密钥的计算,使得我们能够在计算会话密钥的同时验证对方身份的合法性,从而减少握手协议中通信双方的交互次数,简化了握手过程.实验结果表明,本文提出的基于 IBC 的 DTLS 在不降低原来 DTLS 安全性的同时,缩短了近 50%的通信连接时间.

### References:

- [1] Rosenberg J, Schulzrinne H, Camarillo G, Johnston A, Peterson J, Sparks R, Handley M, Schooler E. SIP: Session initiation protocol. RFC 3261, 2002.
- [2] Schulzrinne H, Casner S, Frederick R, Jacobson V. RTP: A transport protocol for real-time applications. RFC 3550, 2003.
- [3] Andreasen F and Foster B. Media gateway control protocol (MGCP). RFC 3435, 2003.
- [4] Cheng Y, Chu J, Radhakrishnan S, Jain A. TCP fast open. RFC 7413, 2014.
- [5] Bittau A, Hamburg M, Handley M, Mazières D, Boneh D. The case for ubiquitous transport-level encryption. In: Goldberg I, ed. Proc. of the 19th USENIX Conf. on Security (USENIX Security 2010). Washington: USENIX Association Berkeley, 2010. 403–418.
- [6] Petullo WM, Zhang X, Solworth JA, Bernstein DJ, Lange T. MinimalLT: Minimal-Latency networking through better security. In: Virgil Gligor, ed. Proc. of the 2013 ACM SIGSAC Conf. on Computer & Communications Security (CCS 2013). Berlin: ACM New York, 2013. 425–438.
- [7] Langley A, Riddoch A, Wilk A, Vicente A, Krasic C, Zhang D, Yang F, Kouranov F, Swett I, Iyengar J, Bailey J, Dorfman J, Roskind J, Kulik J, Westin P, Tenneti R, Shade R, Hamilton R, Vasiliev V, Chang W, Shi Z. The QUIC transport protocol: Design and Internet-scale deployment. In: Snoeren A. C, ed. Proc. of the Conf. of the ACM Special Interest Group on Data Communication (SIGCOMM 2017). New York: ACM, 2017. 183–196.
- [8] Rescorla E, Modadugu N. Datagram transport layer security version 1.2. RFC 6347, 2012.
- [9] Shamir A. Identity-Based cryptosystems and signature schemes. In: Blakley GR, ed. Proc. of the CRYPTO'84 on Advances in Cryptology. New York: Springer-Verlag, 1984. 47–53.
- [10] Boneh D, Franklin M. Identity-Based encryption from the Weil pairing. SIAM Journal on Computing, 2003,32(3):586–615.
- [11] Chen Y, Huang Q, Zhang Z. Sakai-Ohgishi-Kasahara identity-based non-interactive key exchange revisited and more. Int'l Journal of Information Security, 2016,15(1):15–33.



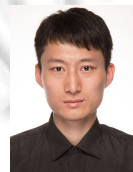
李鹏坤(1993—),男,河南周口人,博士生,主要研究领域为网络安全.



王小峰(1982—),男,博士,助理研究员,CCF 专业会员,主要研究领域为可信网络及系统,网络安全,分布智能数据处理.



苏金树(1962—),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机网络与通信,信息安全.



薛天(1991—),男,硕士,主要研究领域为计算机网络与通信,信息安全.