第 35 卷 第 8 期 计 算 机 工 程 2009 年 4 月
Vol.35 No.8 Computer Engineering April 2009
·安全技术· 文章编号：1000—3428(2009)08—0182—04 文献标识码：A 中图分类号：TP393.08

基于改进型 OCSP 的交叉认证方案

张 茜[1,2]，朱艳琴[1,2]，罗喜召[1,2]
（1. 苏州大学计算机科学与技术学院，苏州 215006；
2. 江苏省计算机信息处理技术重点实验室，苏州 215006）

摘 要：针对在线证书状态协议（OCSP）存在的安全、证书信息源及响应器寻址等问题，提出一种改进型 OCSP 协议以及一个用于交叉认证系统的设计方案。该方案提高了响应器的性能，在检测证书状态的同时还可建立证书路径并验证其是否有效，避免了因信任域结构不同产生的构建证书路径难的问题。

关键词：公钥基础设施；在线证书状态协议；交叉认证

Cross-certification Scheme Based on Improved OCSP

ZHANG Qian[1,2], ZHU Yan-qin[1,2], LUO Xi-zhao[1,2]
(1. School of Computer Science and Technology, Soochow University, Suzhou 215006;
2. Jiangsu Provincial Key Laboratory of Computer Information Processing, Suzhou 215006)

【Abstract】Aiming at the problems in Online Certificate Status Protocol(OCSP) such as security, the information source of certificate and searching address of OCSP responder, this paper proposes an improved OCSP and a scheme based on the improved OCSP for the cross-certification system. The scheme improves the function of the responder, constructs and validates the certificate path when the status of the certificate is given. The scheme avoids the difficulty of constructing the certificate path due to the different architecture of each trust domain.

【Key words】 Public Key Infrastructure(PKI); Online Certificate Status Protocol(OCSP); cross-certification

1 概述

公钥基础设施(Public Key Infrastructure, PKI)是指利用公钥概念与技术来实施和提供安全服务的、普适性的安全基础设施[1]。认证中心(Certification Authority, CA)的主要功能是颁发和管理公钥证书。公钥证书是用户的身份与之所持有的公钥的结合，这种结合通常在证书的整个生命周期是有效的。但由于某些原因(证书用户身份的改变或私钥泄漏等)，证书在到期之前会被撤销，证书将不再有效。证书状态查询是 PKI 系统中的一个重要问题。证书状态查询的方法有很多种，目前广泛使用的证书状态查询机制主要分为 2 种：基于证书撤销列表(Certificate Revoke List, CRL)[1]的查询机制和基于在线证书状态协议(Online Certificate Status Protocol, OCSP)[2-3]的实时查询机制。

目前，CRL 存在的主要问题是[4]：(1)CRL 的规模性；(2)CRL 所含撤销信息的及时性。OCSP 协议是 PKIX 工作组在 RFC2560[2]中提出的协议，是检查数字证书在某一交易时间是否有效的标准，可实时在线地向用户提供证书状态，以及满足比 CRL 提供的证书撤销信息更及时的操作要求。因此，OCSP 可以作为周期性 CRL 的一种替代机制或补充机制。目前该协议只是简单地给出证书的撤销状态，并没有给出证书路径是否有效的信息。

本文提出一种 OCSP 协议在交叉认证系统中的应用机制，使 OCSP 响应器不仅可以检验待测证书的撤销状态，也可以建立并且检验证书路径是否有效。本文对该协议进行分析，在此基础上给出 OCSP 协议在交叉认证系统中的实现方案，并对该方案进行测试及分析。

2 OCSP 协议分析

2.1 协议概述

OCSP 协议作为 CRL 的补充，是一种用于 OCSP 请求者(客户端)和 OCSP 响应者(服务器)之间相对简单的请求/响应协议。OCSP 客户端发送证书状态查询给 OCSP 响应器，并且等待直到响应器返回其响应。

一个 OCSP 请求包含以下数据：协议版本，服务请求，目标证书标识和可选的扩展项。OCSP 协议服务器端的响应消息由响应状态域和响应字节域 2 个部分组成。当 OCSP 响应器返回出错信息时，不对该响应进行签名。出错信息包括以下类型：请求编码格式不正确，内部错误，稍候再试，请求需要签名，未授权。当 OCSP 响应器返回确定的回复时，该响应必须进行数字签名。

响应字节是由编码成 OCTET 字符串的响应内容与响应类型标识组成的。响应内容的语法由响应类型决定。对于基本的 OCSP 响应，对应的响应内容为基本 OCSP 响应的 DER 编码。基本 OCSP 响应由证书状态响应数据、签名算法标识、签名值和帮助请求者验证签名的证书组成，其中，签名值是对证书状态响应数据的数字签名。证书状态响应数据由版本号、响应器标识、响应产生时间、响应列表和可选的响应扩展项组成。响应列表中包含对每一张被请求证书的回复，在对每一张被请求证书的回复中包含有证书状态值：正常，撤

基金项目：国家自然科学基金资助项目(60673041)
作者简介：张 茜(1984－)，女，硕士研究生，主研方向：计算机网络，信息安全；朱艳琴，教授；罗喜召，讲师、在职博士研究生
收稿日期：2008-09-28 **E-mail：**zhangqian841103@126.com

销，未知。“正常”状态表示这张证书没有被撤销；“撤销”状态表示证书已被撤销；“未知”状态表示响应器不能判断请求的证书状态。

2.2 协议局限性

OCSP 协议作为周期性 CRL 的一种替代或补充机制，克服了 CRL 的规模性和其所含撤销信息的及时性 2 个缺陷，但仍有一定的局限性，目前 OCSP 协议主要存在以下 4 个方面的问题[4]：

(1)安全问题。OCSP 是一种典型的客户/服务器模式的协议，由于查询请求是随机的，因此当大量请求消息到达 OCSP 响应器时，对每个响应进行实时签名将明显影响 OCSP 响应器的性能，最终可能使 OCSP 响应器完全崩溃；同时响应器对每个确定状态的响应做实时签名，使 OCSP 系统容易遭受拒绝服务攻击[5]。

(2)证书信息源问题。目前 OCSP 协议并没有规定响应器用来检索证书状态的信息源，OCSP 响应器提供的信息的实时性取决于获取这些信息来源的延时，如果实现者没有合理选取获取证书状态信息源的方式，协议的优势就无法体现。

(3)响应器寻址问题。OCSP 在查询证书状态之前必须先确认证书的颁发机构，在多数情况下，一个 CA 域内的 OCSP 响应器仅具有本地 CA 的授权，如果待查询的证书由其他 CA 签发，它只能返回“未知”状态的响应，因此，给客户端的查询带来不便。

(4)响应器提供的服务问题。目前 OCSP 响应器只检测证书的撤销状态，没有给出证书路径是否有效的信息。

2.3 协议改进

为了解决 OCSP 协议存在的安全和证书信息源问题，本文首先对该协议进行改进，利用改进的协议实现的 OCSP 响应器以 CA 的证书目录库作为响应器的信息采集数据源，并依据该证书目录库采用预签名技术，将预产生的响应放入缓存，不但解决了其安全和证书信息源问题，而且能有效提高响应器的性能。

本文实现的 OCSP 响应器预产生一个短有效期的响应并对其进行数字签名，当响应的 nextUpdate 时间小于服务请求的时间，OCSP 响应器使用单向散列函数[6](One Way Hash Function, OWHF)来更新响应，而不用对响应进行重新签名。为了预先产生一个响应，响应器产生一个随机数 R_0，对 R_0 进行 d 次散列运算 $h^d(R_0)$，得到基本更新值(baseUpdate Value)R，其中 d 是最大更新时间段数(maximumUpdateIndex)，是响应器选定的参数，表示一个响应在生存期后可以继续被缓存的时间，该时间定义为 $d\times$(nextUpdate-thisUpdate)，然后将 R 和 d 分别作为 2 个扩展项包含在预先产生响应的 Single Extensions 中，响应器对该预产生的响应进行签名，将此基本 OCSP 响应(BasicOCSPResponse)和 R_0 存入缓存中。当用户在时刻 $t(t\in$[nextUpdate+(i-1) × (nextUpdate-this Update), nextUpdate+$i\times$(nextUpdate-thisUpdate)]) 发送一个证书状态查询请求时，若满足以下条件：(1)在缓存中有该证书基本 OCSP 响应(BasicOCSPResponse)；(2)查询请求发生在该响应的有效期外、$d\times$(nextUpdate-thisUpdate)范围以内；(3)证书状态没有发生改变，则响应器根据缓存中存储的该证书的 Basic OCSPResponse 和 R_0，计算出当前时间的更新值(current Update Value)R_i：$h^{d-i}(R_0)$，生成 A 类型的响应并将该响应发给用户。若不能同时满足上述 3 个条件，则响应器生成基本类型响应。

2.4 改进型协议的消息格式

改进型 OCSP 响应包括 2 种类型的 OCSP 响应：基本类型 OCSP 回复和 A 类型 OCSP 回复。当响应类型是基本类型 OCSP 回复时，响应内容是基本 OCSP 回复的 DER 编码。当响应类型是 A 类型 OCSP 回复时，响应内容是 A 类型 OCSP 回复(TypeAOCSPResponse)的 DER 编码。baseUpdateValue 和 maximumUpdateIndex 作为扩展项包含在 singleExtensions 中。响应类型分别定义如下：

id-pkix-ocsp-basic OBJECT IDENTIFIER::={id-pkix-ocsp 1}

id-pkix-ocsp-type-a OBJECT IDENTIFIER::={id-pkix-ocsp 9}

baseUpdateValue 和 maximumUpdateIndex 的扩展标识符分别定义如下：

id-pkix-ocsp-base-update-value OBJECT IDENTIFIER::={id-pkix-ocsp 8}

id-pkix-ocsp-maximum-update-index OBJECT IDENTIFIER::={id-pkix-ocsp 11}

以下是本方案所使用的 baseUpdateValue 和 maximum UpdateIndex 扩展项的 extnValue 的格式，定义如下：

baseUpdateValue::=OCTET STRING

maximumUpdateIndex::=INTEGER

A 类型回复是由基本类型响应和当前更新值构成，定义如下：

```
TypeAOCSPResponse::=SEQUENCE{
basicResponse          BasicOCSPResponse,
currentUpdateValue     OCTET STRING }
```

3 实现原理

3.1 交叉认证

交叉认证[7]是一种把以前无关的 CA 连接在一起的有用机制，从而使它们各自主体群之间的安全通信成为可能。交叉认证的实际构成除了最后交叉认证的主体和颁发者都是 CA 外，与其他认证相同。交叉认证可以是单向的，也可以是双向的，即 CA1 可以交叉认证(即签署了身份和公钥)CA2，而 CA2 没有交叉认证 CA1。这种单向交叉认证产生单个交叉证书，而 CA 层次结构是一个典型的实际运用。CA1 和 CA2 也可以互相交叉认证，这种相互交叉认证能力产生 2 个不同的交叉证书，并且更常见，例如，使安全通信成为可能的公司之间和它们的雇员之间。

针对响应器寻址和提供的服务问题，对于查询由其他 CA 颁发的证书，本文实现的 OCSP 响应器通过返回不同的证书状态达到不同信任域之间的证书状态查询，不仅能检测出证书的撤销状态，而且还提供证书路径是否有效的信息。

3.2 实现原理

由于每个信任域的结构不同，因此构建一条跨越不同信任域的信任路径将非常复杂。文献[8]提出了不同信任域间证书状态验证机制，实现了不同信任域间路径的构建及验证，从而不仅能检测证书的撤销状态，而且还给出证书路径是否有效的信息。根据文献[8]的思想，本文基于改进型 OCSP 协议所设计的响应器返回的证书状态有 3 种：“正常，用户可以信任该目标证书”，“撤销”，“未知”。

以图 1 为例，单项箭头为证书签发，双向箭头为双向交叉认证，左边的 PKI 域为层次结构的信任模型，右边的 PKI 域为网状信任模型。2 个信任域通过双向交叉认证连接在一起。每个 CA 都有一个被该 CA 授权的响应器，因此，图 1 的交叉认证系统转变为图 2 的 OCSP 系统结构图。若 User1 收到 User2 的一个数字签名的电子邮件时，User1 必须验证

User2 的证书状态。于是 User1 向本地响应器 R_2 发出状态查询请求，R_2 收到查询请求后，提取出待查询目标证书颁发者的名称散列(issuerNameHash)和密钥散列(issuerkeyHash)，看是否能识别，R_2 不能识别此颁发机构，将请求转发给它所信任的响应器 R_0，并设置等待时间。R_0 不能识别待查询证书的颁发者，将请求转发给它信任的响应器 R_1, R_3 和 R_4。R_1 和 R_3 不能识别待查询证书的颁发者，因此，R_1 和 R_3 不能处理该请求，也没有其他信任的响应器传递，于是将该请求挂起。R_4 收到该请求后，将该请求转发给它所信任的响应器 R_5 和 R_6。R_5 不能识别待查询证书的颁发者，将请求转发给它所信任的响应器 R_6。R_6 收到 2 个同样的请求，于是将 R_5 传来的请求挂起。R_6 可以识别待查询证书的颁发机构，于是处理该请求，将生成的响应传递给 R_2。R_2 收到响应，将其发送给客户端 User1，若超过了等待时间，则发回“未知”响应给客户端。由于响应器只将请求传递给它所信任的响应器，因此若接收客户端请求的初始响应器收到其他响应器发来的响应，则表明一条信任路径已经成功建立。

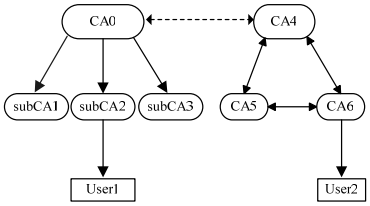

图 1　交叉认证系统结构

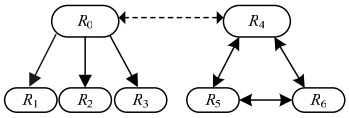

图 2　交叉认证系统的 OCSP 结构

4　方案实施

4.1　信任响应器地址的获取

在响应器端保存 CA 证书和一张响应器节点列表，表项是颁发机构标识符和得到该颁发机构授权的 OCSP 响应器的 IP 地址。CA 证书具有一些扩展项，可以用来提供 LDAP 目录的 URI[9]信息。

主体信息访问扩展(Subject Info Access, SIA)可获得由该证书所颁发的下级证书。当该证书为 CA 证书时，可以用的 accessMethod 为 id-ad-caRepository。通过该扩展项可获得一个或多个被签发的证书。其定义如下:

SubjectInfoAccessSyntax ::= SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription ::=SEQUENCE {

accessMethod OBJECT IDENTIFIER,

accessLocation GeneralName }

权威信息访问扩展(Authority Info Access, AIA)，可用来获得上级签发者证书信息或 OCSP 服务器的 URI 信息，其定义类似于 SIA。当 accessMethod 为 id-ad-caIssuers 时，表示可以获得签发者信息；当 accessMethod 为 id-ad-ocsp 时，表示为 OCSP 服务器信息。通过该扩展项，可以获得一个或多个签发者的证书，该证书可为 CA 证书也可为交叉证书。其定义如下:

AuthorityInfoAccessSyntax ::= SEQUENCE SIZE (1..MAX) OF AccessDescription

AccessDescription ::= SEQUENCE {

accessMethod OBJECT IDENTIFIER,

accessLocation GeneralName }

若响应器存储的 CA 证书是自签发证书，获取其 SIA 扩展项，通过 SIA 扩展项获得该 CA 所签发的证书，从而获得该证书的主体名，通过该主体名查询存储的响应器节点列表，获得该响应器信任的响应器的 IP 地址；若响应器存储的 CA 证书不是自签发证书，获取其 SIA 和 AIA 扩展项，通过 SIA 扩展项获得该 CA 所签发的下级证书，通过 AIA 扩展项获得上级签发者证书，从而获得这些证书的主体名，通过主体名查询存储的响应器节点列表，以获得该响应器信任的响应器的 IP 地址。

4.2　实现方案

响应器与客户端之间以及响应器之间均通过 SSL 握手协议建立安全通信通道，完成对数据的加密保护。

响应器的流程如图 3 所示。

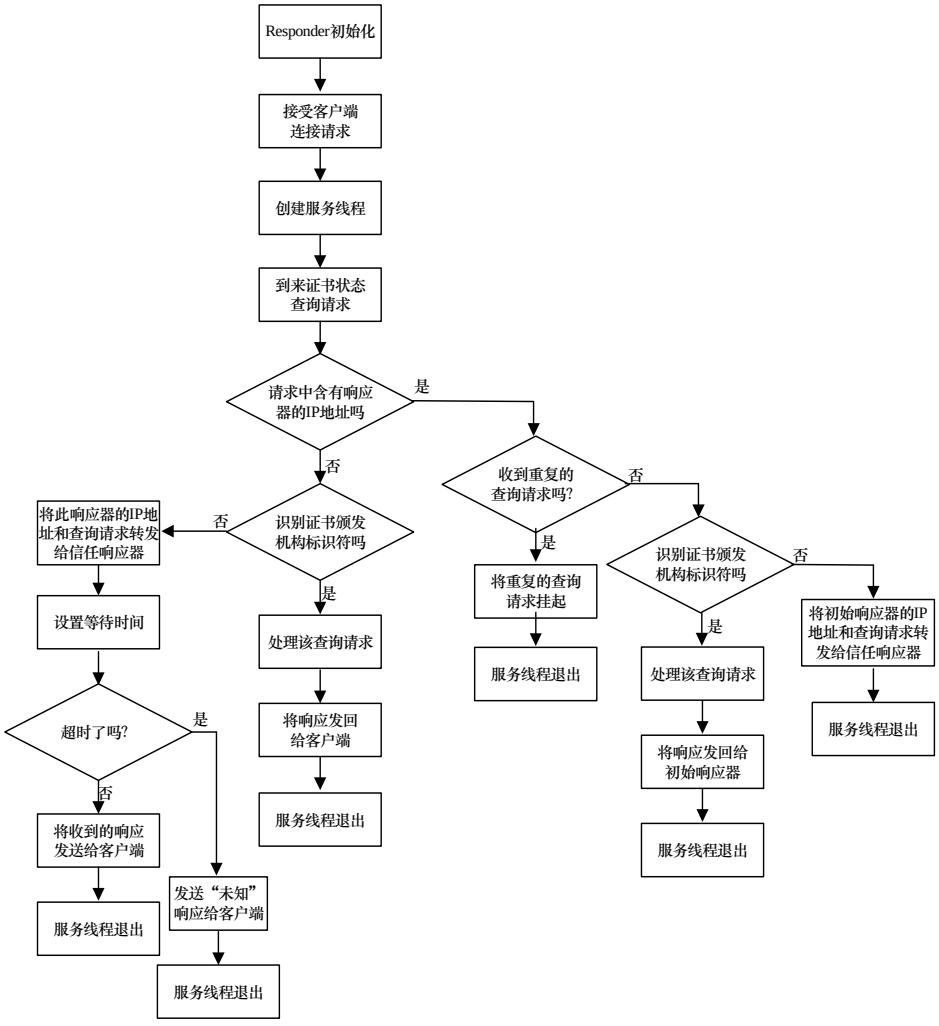

图 3　响应器端流程

响应器的具体工作过程如下:

(1)响应器初始化，获取其信任响应器的地址，创建侦听客户端连接端口。

(2)若有客户端连接请求，则创建服务线程来处理客户端的证书状态查询请求。若请求中含有响应器的 IP 地址，则表明接收请求的响应器为转发响应器；若请求中不含有响应器的 IP 地址，则表明此响应器为初始响应器。

(3)若是转发响应器，判断收到的查询请求是否为重复的查询请求，若是，则将收到的重复查询请求挂起，服务线程退出；否则，从查询请求中提取待查询证书的发布者散列和密钥散列。若此响应器识别此颁发机构，则表明该响应器不仅为转发响应器同时也是最终响应器，处理该查询请求，执行(5)；否则，将初始响应器的 IP 地址和查询请求转发给除传入响应器的 IP 地址以外的其他信任响应器，服务线程退出。

(4)若是初始响应器，则从客户请求中提取待查询证书的发布者散列和密钥散列。若此响应器识别此颁发机构，表明此响应器不仅是初始响应器，也是最终响应器，处理该查询请求，直接将响应发回给客户端，服务线程退出；否则，此响应器创建响应等待端口，设置等待时间，将此响应器的 IP 地址和到来的查询请求转发给它信任的响应器，执行(6)。

(5)从到来的查询请求中获取初始响应器的 IP 地址，将生成的响应传给初始响应器，服务线程退出。

(6)若未超出等待时间，则将收到的响应转发给客户端；否则，发送“未知”响应给客户端，服务线程退出。

5 测试与分析

本文采用 Java 语言编程，操作系统为 Windows XP，使用 jdk1.5.0_03，证书库使用 openLDAP 目录服务器，在配置为 P4 3.0 GHz 处理器、512 MB 内存的 PC 上对该方案进行实施。本文按照图 1 所示的交叉认证系统搭建的测试环境，subCA2 的用户 Daniel 验证 CA6 的用户 Amy 的证书 Amycert.der，OCSP 服务器地址为 192.168.150.105，OCSP 服务器端口号为 5432，查询的目标证书为 Amycert.der，CA 证书为 CA6cert.der，用户证书为 Danielcert.der，用户私钥为 Danielkey.pem，客户端密钥库为 Danielstore，客户端密钥库密码为 123456，客户端信任密钥库为 Danieltruststore，客户端信任密钥库密码为 123456。查询结果如图 4 所示。

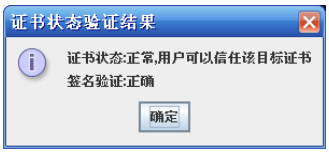

图 4　证书状态查询结果

上述结果表明待测证书没被撤销，并且该证书的路径是有效的，从而既能检测证书的撤销状态，同时也提供证书路径处理服务。由于响应器只将请求传递给它所信任的响应器，因此转发请求的过程就是建立信任路径的过程，若接收客户端请求的初始响应器收到其他响应器发来的响应，表明一条信任路径已经成功建立，从而避免了因信任域结构不同产生的构建证书路径的复杂性。

6 结束语

证书状态查询是 PKI 应用系统中的一个关键问题，作为一种在线证书状态查询方式，OCSP 协议存在一些局限性，针对这些局限性，本文提出了一种改进型 OCSP 协议，利用改进型 OCSP 协议设计的响应器极大地提高了性能。本文设计的改进型 OCSP 协议的应用方案使改进后的响应器不仅能检验待测证书的撤销状态，同时还可建立并且检验证书路径是否有效。当交叉认证系统中的 CA 较少时，响应器数目相应较少，响应速度较快，但当系统中的 CA 数目较多时，响应速度较慢。在系统 CA 数目较多时，如何提高响应器的响应速度将是下一阶段的研究重点。

参考文献

[1] Housley R, Polk W, Ford W, et al. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List(CRL) Profile[S]. RFC 3280, 2002-04.

[2] IETF. X.509 Internet Public Key Infrastructure Online Certificate Status Protocol OCSP[S]. RFC 2560, 1999-06.

[3] Berbecaru D, Lioy A, Marian M. Security Aspects in Standard Certificate Revocation Mechanisms: A Case Study for OCSP[C]// Proceedings of the 7th International Symposium on Computers and Communications. [S. l.]: IEEE Press, 2002: 484-489.

[4] 张　岩, 曹秀英. 一种改进型 OCSP 系统的设计与实现[J]. 信息安全与通信保密, 2005, (7): 277-281.

[5] 林璟锵, 余　婧, 曹　政, 等. 高性能 OCSP 服务器的实现[J]. 计算机工程, 2005, 31(4): 74-76.

[6] 李景峰, 潘　恒, 祝跃飞. 基于单向散列链的公钥证书撤销机制[J]. 小型微型计算机系统, 2006, 27(4): 642-645.

[7] 谢冬青, 冷　健. PKI 原理与技术[M]. 北京: 清华大学出版社, 2003.

[8] Zhang Shaomin, Gong Huitao, Wang Baoyi. An Extended OCSP Protocol for Grid CA Cross-certification[C]//Proceedings of the 2nd International Conference on Semantics, Knowledge and Grid. [S. l.]: IEEE Press, 2006.

[9] Howes T, Smith M. The LDAP URL Format[S]. RFC 2255, 1997-12.

编辑　张正兴

(上接第 181 页)

参考文献

[1] Weiser M, Gold R, Brown J S. The Origins of Ubiquitous Computing Research at PARC in the Late 1980s[J]. IBM Systems Journal, 1999, 38(4): 693-696.

[2] Perrig A, Stankovic J, Wagner D. Security in Wireless Sensor Networks[J]. Communications of the ACM, 2004, 47(6): 53-57.

[3] Blundo C, Santis D, Herzberg A. Perfectly-secure Key Distribution for Dynamic Conferences[C]//Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology. California, USA: [s. n.], 1993: 471-486.

[4] Perrig A, Szewczyk R, Tygar J D, et al. SPINS: Security Protocols for Sensor Networks[J]. Wireless Networks, 2002, 8(5): 521-534.

编辑　索书志