第34卷 第23期 计 算 机 工 程 2008年12月
Vol.34 No.23 Computer Engineering December 2008
·安全技术· 文章编号：1000—3428(2008)23—0167—03 文献标识码：A 中图分类号：TP393.08

OCSP协议的改进和实现

张 茜，朱艳琴，罗喜召
(苏州大学计算机科学与技术学院，苏州 215006)

摘 要：对标准在线证书状态协议(OCSP)进行分析，发现该标准协议存在一定的局限性。在此基础上对其进行改进，改进型OCSP响应包括基本类型OCSP回复和A类型OCSP回复。改进型OCSP响应器采用预签名技术，能提高性能且有效抵御重传攻击。对该响应器进行效率和安全性分析。实验结果表明，改进后的响应器的平均响应时间减少27%，提高了响应器的响应速度。

关键词：公钥基础设施；在线证书状态协议；证书状态；预签名

Improvement and Implementation of OCSP

ZHANG Qian, ZHU Yan-qin, LUO Xi-zhao
(School of Computer Science and Technology, Soochow University, Suzhou 215006)

【Abstract】 This paper analyzes the Online Certificate Status Protocol(OCSP), and some limitations of the protocol are found. It improves the normal protocol: the improved OCSP response includes basic OCSP response and type-A OCSP response. The improved responder adopts signature in advance technology to improve efficiently the functionality based on the improved protocol. The responder resists the replay attack efficiently. It analyzes the efficiency and security of the improved OCSP responder. Experimental result shows that the average response time of the improved responder is reduced by 27%, and the response speed is improved.

【Key words】 Public Key Infrastructure(PKI); Online Certificate Status Protocol(OCSP); status of certificate; signature in advance

1 概述

随着电子商务的迅速发展，公钥基础设施(Public Key Infrastructure, PKI)[1]技术的应用越来越广泛。PKI是指利用公钥的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。在PKI框架中，认证中心(Certification Authority, CA)[1]的主要功能是颁发和管理证书。PKI应用系统在使用数字证书之前，必须先验证数字证书的有效性，包括：证书是否过期，证书是否被撤销，证书中的CA签名是否正确等。因此，证书状态查询是PKI系统的关键问题之一。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)[2-7]的实时查询机制。

CRL是由颁发证书的CA定期签发的一个签名的数据结构，内含被该CA撤销的证书列表。目前CRL存在的主要问题[3]如下：

(1)CRL的规模性。随着该CA的终端实体数目、证书的生命期以及证书撤销频率的增加，CRL的大小也随之增加。这可能导致在该CA域内CRL的规模变得非常庞大。

(2)CRL所含撤销信息的及时性。由于CRL定期发布，因此不能保证证书撤销信息的实时性和准确性。

OCSP协议是PKIX工作组在RFC2560[2]中提出的协议，它可以满足比CRL提供的证书撤销信息更及时的操作要求。因此，OCSP可以作为周期性的CRL的一种代替机制或补充机制。基于安全性需求，OCSP协议要求响应器对响应消息进行实时签名。但签名运算所消耗的资源较多、时间较长，因此，对响应消息进行实时签名将影响响应器的性能，使其成为OCSP服务器实现中的性能瓶颈。采用预签名技术[3]，即预先签名技术，虽能提高响应器的性能，但易引起重传攻击[3]，因此，本文提出一种改进型的OCSP协议，使响应器采用预签名技术提高性能的同时，又能有效地抵御重传攻击。

2 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。

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。响应消息的格式如下(ASN.1语法表示)：

```
OCSPResponse::=SEQUENCE{
responseStatus OCSPResponseStatus,
responseBytes [0]EXPLICIT ResponseBytes OPTIONAL}
```

响应字节由编码成OCTET字符串的响应内容与响应类型标识组成。响应内容的语法由响应类型决定。对于基本的OCSP响应，响应类型id-pkix-ocsp-basic定义如下：

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

对应的response域的值为基本OCSP响应(BasicOCSPResponse)的DER编码。ResponseBytes的格式如下：

基金项目：江苏省自然科学基金资助项目(BK2004039)
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、博士研究生
收稿日期：2008-04-06 **E-mail：**zhangqian841103@126.com

```
ResponseBytes::=SEQUENCE{
responseType OBJECT IDENTIFIER,
response OCTET STRING}
```

BasicOCSPResponse 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。BasicOCSPResponse 的格式如下：

```
BasicOCSPResponse::=SEQUENCE{
tbsResponseData      ResponseData,
signatureAlgorithm   AlgorithmIdentifier,
signature            BIT STRING,
certs      [0]EXPLICIT SEQUENCE OF Certificate OPTIONAL}
```

ResponseData 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，其格式如下：

```
ResponseData::=SEQUENCE{
version          [0]EXPLICIT Version DEFAULT v1,
responderID      ResponderID,
productAt        GeneralizedTime,
responses        SEQUENCE OF SingleResponse,
responseExtensions   [1]EXPLICIT Extensions OPTIONAL}
```

SingleResponse 域指明了单个证书的状态，即“正常”、“已撤销”、“未知”3 种状态，如果是“已撤销状态”，则在撤消消息中指明证书撤消的时间，且有可能指明证书撤消的原因。SingleResponse 由证书标识、证书状态、本次更新时间、下次更新时间和扩展项构成。SingleResponse 格式如下：

```
SingleResponse::=SEQUENCE{
certID               CerID,
certStatus           CertStatus,
thisUpdate           GeneralizedTime,
nextUpdate           [0]EXPLICIT Generalized Time OPTIONAL,
singleExtensions     [1]EXPLICIT Extensions OPTIONAL}
```

2.2 协议局限性

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，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[4]。

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 改进型 OCSP 协议的设计

为进一步提高OCSP响应器的性能，减少响应的生成时间，本系统实现的OCSP响应器的信息采集数据源为CA的证书目录库，依据该证书目录库采用预签名技术。但预签名技术容易引起重传攻击，即攻击者利用旧的响应消息对请求者提供虚假服务。若预产生一个短有效期的响应，虽然避免了重传攻击，但响应器需要很多的处理资源来更新这些短有效期的响应，这样就会使响应器容易遭受拒绝服务攻击。因此，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链OWHF[5](One Way Hash Function)来更新响应，而不用对响应进行重新签名，OWFH 计算速度比签名响应至少快 10 000 倍，从而避免重新签名响应所引起的处理资源的大量耗费。

3.1 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 时回复(BasicOCSP Response)的是 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}

本方案所使用的 baseUpdateValue 和 maximumUpdate Index 扩展项的 extnValue 的格式定义如下：

```
baseUpdateValue::=OCTET STRING
maximumUpdateIndex::=INTEGER
```

A 类型回复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 }
```

3.2 协议的设计原理

改进型OCSP响应器设计原理为：预先产生 1 个响应，响应器产生 1 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R，其中，d是最大更新时间段数，是响应器选定的参数，表示 1 个响应在生存期后可以继续被缓存的时间，该时间定义为$d\times(nextUpdate\text{-}thisUpdate)$，将$R$作为 1 个扩展项包含在预先产生响应的SingleExtensions中，响应器对该预产生的响应进行签名，将此基本类型的响应存入缓存中。当用户在时刻$t(t\in[nextUpdate+(i\text{-}1)\times(nextUpdate\text{-}\ thisUpdate),\ nextUpdate+\ i\times(nextUpdate\ \text{-}thisUpdate)])$发送一个证书状态查询请求，若满足以下条件：

(1)在缓存中有该证书基本类型的响应；

(2) 查询请求发生在该响应的有效期外，$d\times(nextUpdate\text{-}thisUpdate)$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基本类型响应计算出当前时间的更新值$R_i{:}h^{d-i}(R_0)$，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。图 1 为改进型 OCSP 响应器端流程。

改进型 OCSP 客户端设计原理为：

(1)若用户收到基本类型响应，则检查当前时间是否在响应有效期内；

(2)若用户收到A类型响应，则从该响应中获得R_i值，计算出$h^i(R_i)$，并把它和该响应中SingleExtensions包含的扩展项的值R进行比较，若匹配，则表明该响应是由响应器产生的；若不匹配，则表明该响应为无效响应。

图 2 为改进型 OCSP 客户端流程。

本文提出的更新响应方法仅适用于有效期过时，但响应中的证书状态没有改变的响应。若预产生的响应中的证书状

态发生改变，则须产生一个新的响应并对其进行签名。因为大部分响应是由于其有效期过时而须更新，所以本文提出的改进协议具有一定的实用价值。

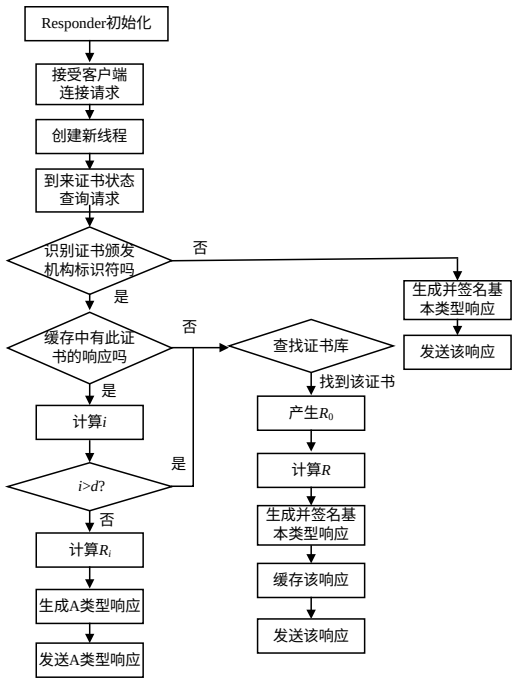

图 1　改进型 OCSP 响应器端流程

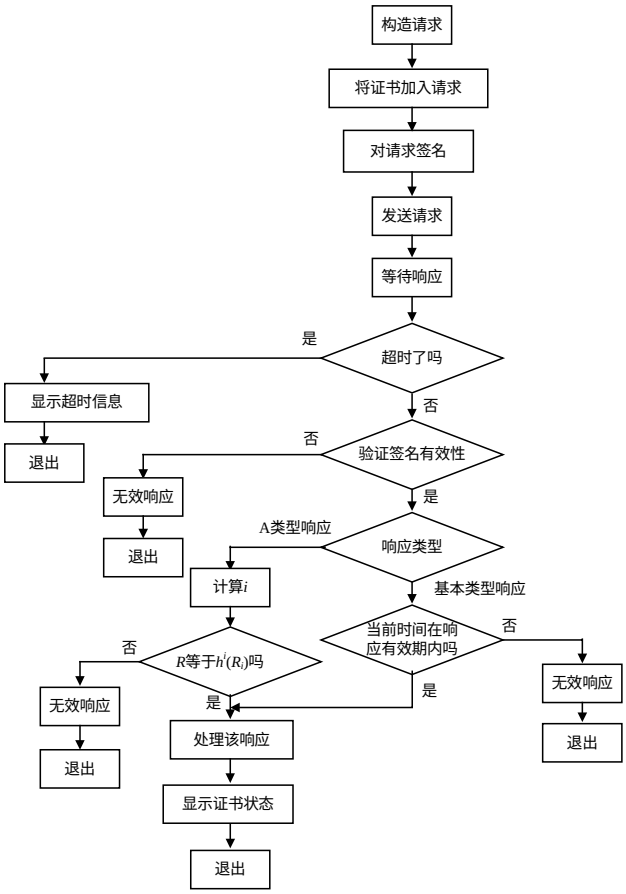

图 2　改进型 OCSP 客户端流程

4　实验结果

在本文的实验中，客户端分别向改进型响应器和标准响应器反复发出证书状态查询请求，记录每次查询花费的响应时间，得出响应器预签名和实时签名分别所需要的最小响应时间、最大响应时间和平均响应时间。采用 Java 语言编程，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 机上进行实验。

实验结果为：响应器实时签名所需的响应时间最小值为 1 087 ms，最大值为 1 250 ms，平均响应时间为 1 153.5 ms，采用预签名技术的改进型响应器所需的响应时间最小值为 812 ms，最大值为 969 ms，平均相应时间为 841.166 7 ms。改进后的响应器的平均响应时间降低 27%。可见，按本方案实现的改进型 OCSP 响应器可提高响应速度，降低响应时间。

5　效率和安全性分析

本方案实现的 OCSP 响应器采用预签名技术减少了响应器的计算负载，提高了 OCSP 响应器的响应速度，使 OCSP 响应器具有较高的实时响应能力，进一步提高了响应器的性能。传统采用预签名技术的 OCSP 响应器在提高响应器性能的同时是以牺牲一定的安全级别为代价的，容易引起重传攻击。而本文中实现的 OCSP 响应器预产生一个短有效期的响应并对其签名，同时在响应的最大保护期内($d \times (nextUpdate\text{-}thisUpdate)$)利用单向散列链机制以很少的处理资源来更新这些响应。攻击者如果想要利用以前旧的响应消息进行重传攻击，就必须用旧的响应消息中的当前更新值来计算此刻的更新值，而这在计算上是不可能的，因为 OWHF 是单向不可逆的。因此，本文设计的响应器在使用预签名技术的同时有效避免了重传攻击，同时，响应器用很少的处理资源来更新预先生成的响应，减少了遭受拒绝服务攻击的风险。

6　结束语

OCSP 是一种用于查询证书当前状态的在线实时协议。本文对标准的 OCSP 协议进行改进，实现改进型 OCSP 客户端和改进型 OCSP 响应器，给出改进型 OCSP 协议的消息格式，并对该改进型 OCSP 响应器进行效率和安全性分析。证书状态查询是 PKI 应用系统中一个关键问题，可分为离线和在线 2 种，由于在线验证比离线验证更符合商业模型的基本需要，因此在线验证是未来 PKI 证书验证的重要发展方向。

参考文献

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

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

[3] 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.

[4] 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.

[5] 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.

[6] 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 Computer Society, 2002.

[7] Munoz-Tapia J L, Forne-Munoz J. CPC-OCSP: An adaptation of OCSP for M-Commerce[Z]. (2002-12-01). http://isg.upc.es/cervantes/papers/data/upgrade02.pdf.