面向Web服务的SAML路径验证协议及其性能分析

张 斌 王 曦

(解放军信息工程大学电子技术学院 郑州 450004)

摘 要 基于PKI的签名机制在保护SAML断言传递时存在增加SOAP消息长度、显著降低Web服务响应速度的问题。为此,提出了基于身份聚合签名的SAML路径验证(IBASPV)协议,该协议通过缩短SOAP消息中签名值和验证公钥的长度来减少SOAP消息的传输时间,提高服务响应速度。采用SVO逻辑形式化证明了IBASPV协议具有断言完整性、源不可伪造性、传递路径不可篡改和抗重放攻击的安全特性。然后,采用安全模块Rampart测试分析了密码运算时间和数据传输时间随网络数据传输速率的变化趋势,比较了基于IBASPV协议与基于PKI签名的服务认证调用协议的性能。

关键词 Web服务认证,SAML,基于身份的聚合签名,SVO逻辑,性能分析

中图法分类号 TP393.08 **文献标识码** A

SAML Path Verification Protocol for Web Service and its Performance Analysis

ZHANG Bin WANG Xi

(Institute of Electronic Technology, PLA Information Engineering University, Zhengzhou 450004, China)

Abstract To resolve the problem of increasing the length of SOAP message and reducing the respond speed of Web service seriously during the PKI-based signature to protect the SAML assertion, the paper proposed the SAML path verification protocol based on identity aggregate signature(IBASPV), which improveds the respond speed of Web service by shortening the length of the signature and the public key to reduce the transport time of SOAP message. By using SVO logic, we proved that the IBASPV protocol can ensure the integrity of the SAML assertion and source unforgeability, and can protect the transmission path which can not be tampered with, and can prevent anti-replay attacks. By measuring the cryptographical calculation time and transport time based on Rampart module, we analyzed the trend that these factors which cause response time increase with the network data transport speed. Finally, we compared the performance of Web service based IBASPV protocol with that based on PKI signature.

Keywords Web service authentication, SAML, Identity-based aggregate signature, SVO logic, Performance analysis

1 引言

Web服务调用认证是防止攻击者通过服务接口非法使用资源的必要安全措施。服务提供者SP(Service Provi-der)通过验证携带在SOAP请求消息头部的SAML[1]断言实现对服务调用者的身份认证。身份提供者IDP(Identity Provider)为通过认证的用户生成SAML断言,并随SOAP请求消息传递给SP。在Web服务认证调用时,需要保证SAML断言传递过程中的完整性、源不可伪造性以及传递路径不被篡改。

目前WS-Security[2]标准中采用基于PKI的签名机制保证SAML断言安全传递。该方案需要传递路径上的每个SP对SAML断言依次签名并传递签名值和验证签名所需的公钥证书,其不足是:随着断言传递路径增加,需要传递的签名值和公钥证书增多,增加了SOAP请求消息的长度,从而导致数据传输时间的增加,降低了Web服务响应速度。

针对以上问题,本文将基于身份聚合签名(Identity-Based Aggregate Signature, IBAS)[3]算法应用在SAML断言传递过程的安全性保护中,提出了基于身份聚合签名的SAML路径验证(IBAS-based SAML Path Verification, IBASPV)协议。IBASPV协议利用IBAS算法将多个不同的签名值压缩成一个聚合签名的特点以及身份就是公钥的特性,通过缩短签名值和验证公钥的长度,减少签名后SAML断言的增长幅度,来减少携带断言的SOAP请求消息长度,达到减少数据传输时间、提高Web服务响应速度的目的。

2 基于身份聚合签名的SAML路径验证协议

2.1 IBASPV协议

本节基于文献[3]提出的IBAS算法设计IBASPV协议。IBAS算法基于椭圆曲线上双线性对设计,由参数生成、私钥生成、签名、签名聚合以及签名验证子算法组成。协议的基本思想是:以Web服务的地址作为公钥,由SP为其所提供的

到稿日期:2012-05-29 返修日期:2012-08-27 本文受国家973重点基础研究发展计划(2011CB311801)资助。

张 斌(1969—),男,博士,教授,主要研究方向为网络信息安全,E-mail:zhangyym@xinhuanet.com;王 曦(1983—),男,硕士生,主要研究方向为网络认证技术。

Web服务向PKG(Private Key Generator)申请对应的私钥。IDP为通过认证的用户生成SAML断言并初始签名,经过签名的SAML断言和签名验证公钥随SOAP请求消息传送给SP,SP验证断言的签名。如果本SP提供的服务调用了另一个SP提供的服务,则本SP使用该服务对应的私钥对断言进行签名,生成新的聚合签名值,将本跳公钥加入验证公钥列表并和新的聚合签名值向下一跳传播,协议流程如图1所示。

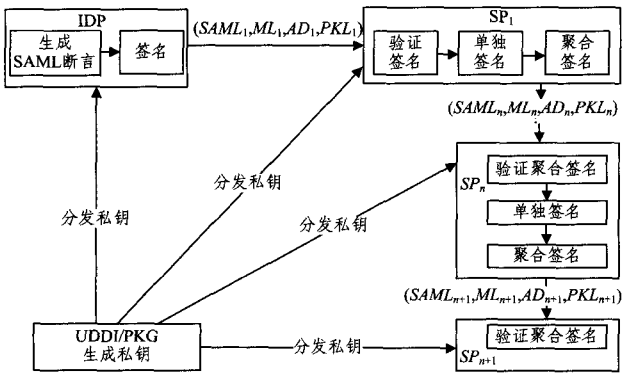

图1 IBASPV协议流程

2.1.1 协议初始化

基于身份的密码系统中,通信实体的身份作为公钥,私钥由通信双方都信任的第三方——私钥生成器PKG生成。本文将PKG与服务注册和发布中心UDDI相结合,系统参数的生成如下:选定阶为素数q的加法循环群G_1、乘法循环群G_2,双线性映射$e:G_1\times G_1\to G_2$;选择生成元$G\in G_1$。定义Hash函数:$H_1,H_2:\{0,1\}^*\to G_1$;$H_3:\{0,1\}^*\to Z/qZ$。PKG随机选取$s\in Z/qZ$,计算$Q=sG$。系统的公开参数是$(G_1,G_2,e,G,Q,H_1,H_2,H_3)$,$s$是系统主密钥。

2.1.2 私钥生成

将UDDI发布的Web服务地址URI_{WS}作为公钥(IDP使用URL_{IDP}作为公钥),由UDDI/PKG使用IBAS私钥生成算法生成对应的私钥,并通过安全的信道向IDP和SP传递私钥。由SP管理其所提供Web服务的私钥,私钥生成如下:给定Web服务地址为URI_{WS},PKG计算私钥为$sP_{i,j}$,$i\in\{0,1\}$,$j\in\{0,1\}$,其中,$P_{i,j}=H_1(URI_{WS},j)$。

2.1.3 协议步骤

(1)初始断言生成

IDP生成初始断言报文:$(SAML_1,ML_1,AD_1,PKL_1)$,其中:

断言$SAML_1=(m,idp,N_c,ts,PL_1)$,$m$是用户的身份和属性,$idp$是生成断言的身份提供者地址,$N_c$是不重复使用的随机字符串,$ts$是断言生成时间,$PL_1=(URI_{WS_1})$是路径列表。

$ML_1=(MAC_1)$是摘要列表,$MAC_1=hash(SAML_1)$。

$PKL_1=(URL_{IDP})$是验证签名的公钥列表。

AD_1是初始断言签名值,采用IBAS签名算法计算签名值的过程如下:

$AD_1=Sig(MAC_1)=(w,S_1,T_1)$,$w$是所有签名者共享的未曾使用过的随机字符串。IDP随机选取$r_1\in Z/qZ$,计算:

$P_w=H_2(w)$,$c_1=H_3(MAC_1,URL_{IDP},w)$

$S_1=r_1P_w+sP_{1,0}+c_1sP_{1,1}$,$T_1=r_1P$

最后得到断言的初始签名(w,S_1,T_1)。

(2)断言转发过程

设:

$SAML_n=(m,idp,N_c,ts,PL_n)$

$PL_n=(URI_{WS_1},URI_{WS_2},\cdots,URI_{WS_n})$

$ML_n=(MAC_1,MAC_2,\cdots,MAC_n)$

$PKL_n=(URL_{IDP},URI_{WS_1},URI_{WS_2},\cdots,URI_{WS_{n-1}})$

其中,断言传播的跳数$n\geqslant 1$。当SP_n收到报文$(SAML_n,ML_n,AD_n,PKL_n)$时,执行以下验证操作:

①判断$MAC_n=hash(SAML_n)$是否成立。

②若成立,根据ML_n,PKL_n使用IBAS签名验证算法验证聚合签名AD_n的正确性。聚合签名验证过程如下:

已知验证公钥集合$(URL_{IDP},URI_{WS_1},URI_{WS_2},\cdots,URI_{WS_{n-1}})$,摘要集合$(MAC_1,MAC_2,\cdots,MAC_n)$对应的聚合签名值$AD_n=(w,S_n,T_n)$,验证下式是否成立:

$$e(S_n,P)=e(T_n,P_w)e(Q,\sum_{i=1}^{n}P_{i,0}+\sum_{i=1}^{n}c_iP_{i,1})$$

其中,$P_{i,j}=H_1(URI_{WS_i},j)$;

$P_w=H_2(w)$;

$c_i=H_3(MAC_i,URI_{WS_i},w)$。

若验证通过,则聚合签名AD_n正确。

③SP_n要继续向外传递断言,需要完成以下操作:

1)将下一个需要调用的服务地址加入路径列表PL_n,生成$PL_{n+1}=(URI_{WS_1},\cdots,URI_{WS_n},URI_{WS_{n+1}})$,得到新的$SAML_{n+1}$。

2)计算摘要$MAC_{n+1}=hash(SAML_{n+1})$,生成$ML_{n+1}$。

3)生成断言$SAML_{n+1}$的聚合签名值AD_{n+1}。

首先,使用IBAS签名算法计算$SAML_{n+1}$的单独签名:

$D_{n+1}=Sig(MAC_{n+1})=(w,S_{n+1},T_{n+1})$

然后,使用IBAS签名聚合算法将$AD_n=(w,\sum_{i=1}^{n}S_i,\sum_{i=1}^{n}T_i)$和$D_{n+1}=(w,S_{n+1},T_{n+1})$聚合成新的聚合签名值$AD_{n+1}=(w,\sum_{i=1}^{n}S_i+S_{n+1},\sum_{i=1}^{n}T_i+T_{n+1})$。

4)将当前调用服务的地址加入公钥列表:

$PKL_{n+1}=(URL_{IDP},URI_{WS_1},\cdots,URI_{WS_{n-1}},URI_{WS_n})$

5)生成和发出新的断言验证报文:

$(SAML_{n+1},ML_{n+1},AD_{n+1},PKL_{n+1})$

2.2 IBASPV协议安全性证明

SVO逻辑[4]是目前较完善的BAN类逻辑语义分析系统,是密码协议安全性证明的主要方法。文献[3]证明了IBAS算法的安全性,本节在IBAS算法安全的基础上使用SVO逻辑证明IBASPV协议的安全性。证明所需的推理公理如下:

MP规则:由φ和$\varphi\supset\psi$可以推导出ψ

Nec规则:由$\vdash\varphi$可以推导出$\vdash P\ believes\ \varphi$

接收公理

A_1:$P\ received\ (X_1,\cdots,X_n)\supset P\ received\ X_i$

临时值验证公理

A_2:$(fresh(X)\wedge P\ said\ X)\supset P\ says\ X$

原始SVO逻辑中没有针对聚合签名定义推理公理,聚合签名算法通过验证聚合签名值可以一次验证多个签名者的有

效性，通过扩展 SVO 逻辑中的消息来源公理，定义推理公理：

A_3^*：$PK_\sigma(\{P_i\}_{i=1}^n,\{K_i\}_{i=1}^n)\wedge R\ received\ AD\wedge SV(AD,\{K_i\}_{i=1}^n,\{Y_i\}_{i=1}^n)\supset\{P_i\}_{i=1}^n\ said\ \{Y_i\}_{i=1}^n$

其中，符号$\{P_i\}_{i=1}^n$表示签名者序列$\{P_1,\cdots,P_n\}$，$\{K_i\}_{i=1}^n$表示签名密钥序列$\{K_1,\cdots,K_n\}$，$\{Y_i\}_{i=1}^n$表示被签名的报文序列$\{Y_1,\cdots,Y_n\}$，AD表示聚合签名值。

A. 协议的安全性目标

IBASPV 协议的安全目标是保证 SAML 断言从域 D_n 传递到域 D_{n+1} 的过程中源不可伪造性、传递路径不可伪造性以及抗重放攻击。基于 SVO 逻辑，将安全目标分别定义如下：

G_1：$D_{n+1}\ believes\ D_1\ says\ IDP$

G_2：$D_{n+1}\ believes\ \{D_i\}_{i=1}^n\ says\ \{PL_i\}_{i=1}^n$

G_3：$D_{n+1}\ believes\ D_n\ says\ SAML_n$

B. 协议的初始假设

P_1：$D_{n+1}\ believes\ fresh(N_C)$

P_2：$D_{n+1}\ believes\ fresh(T)$

P_3：$PK_\sigma(D_n,PKL_n)$

P_4：$D_{n+1}\ received\ (SAML_n,ML_n,AD_n,PKL_n)$

C. 安全性证明

下面基于 SVO 规则、公理以及协议初始假设，证明协议满足定义的安全目标：

C_1. 由 P_4，A_1，MP，Nec 得出：

$D_{n+1}\ believes\ D_{n+1}\ received\ AD_n$

C_2. 由 C_1，P_3，A_3^*，MP 得出：

$D_{n+1}\ believes\ D_i\ said\ SAML_i, i\in[1,n]$

C_3. 由 C_2，P_1，P_2，A_2，MP 得出：

$D_{n+1}\ believes\ D_i\ says\ SAML_i, i\in[1,n]$

C_4. 由 C_3 和聚合签名的定义得出：

$D_{n+1}\ believes\ D_1\ says\ SAML_1$

C_5. 由 C_4，A_1，MP 得出：

$D_{n+1}\ believes\ D_1\ says\ IDP$（安全目标 G_1）

C_6. 由 C_2 和聚合签名的定义得出：

$D_{n+1}\ believes\ \{D_i\}_{i=1}^n\ says\ \{SAML_i\}_{i=1}^n$，即

$D_{n+1}\ believes\ \{D_i\}_{i=1}^n\ said\ \{(ID_C,IDP,N_c,T,PL_i)\}_{i=1}^n$

C_7. 由 C_6，A_1，MP 得出：

$D_{n+1}\ believes\ \{D_i\}_{i=1}^n\ says\ \{PL_i\}_{i=1}^n$（安全目标 G_2）

C_8. 由 C2 和聚合签名的定义得出：

$D_{n+1}\ believes\ D_n\ says\ SAML_n$（安全目标 G_3）

2.3 IBASPV 协议性能分析

IBASPV 协议结合了身份密码体制中公钥长度短和聚合签名可以压缩多个签名的特点，缩短了 SAML 断言中签名元素的长度。下面对采用 PKI 机制和 IBASPV 机制生成的签名断言结构和长度进行比较。假设 Web 服务调用中断言传播路径为 IDP→SP_1→⋯→SP_n。

基于 PKI 的签名先对签名内容用 SHA 函数生成长度为 20 Byte 的摘要 MAC_i；对摘要用 RSA 签名算法生成长度为 128 Byte 的签名值 D_i，其长度为 L_{D_i}。X.509 证书 PK_i 包含用户标识、公钥和证书签名，假定用户标识为 50Byte，公钥为 128Byte，证书签名 128Byte，则证书长度 $L_{X.509}$ 为 306Byte。在上述场景下，基于 PKI 机制签名的断言传递的签名值和公钥长度为：

$$L_{SAML_n}=n(L_{D_i}+L_{X.509})=434n\ \text{Byte}$$

基于 IBASPV 机制的断言签名若采用 ECC[5] 实现，聚合签名 AD_n 为三元组(w,S_n,T_n)，其中，w 为随机字符串取 16Byte；S_n、T_n 是 ECC 上的点，分别取 128Byte，则聚合签名的长度 L_{AD_n} 为 272Byte。假设用户标识为 50Byte，则用户公钥长度 L_{PK} 为 50Byte。在上述场景下，基于 IBASPV 签名的断言的签名值和公钥长度为：

$$L'_{SAML_n}=L_{AD_n}+nL_{PK}=272+50n\ \text{Byte}$$

假设，RSA 使用 128Byte 私钥签名。在 IBASPV 生成私钥时选取 Hash 函数为 SHA 函数，则用于 ECC 签名的私钥的长度为 20Byte。一般认为，使用 128Byte 密钥的 RSA 签名和使用 20Byte 密钥的 ECC 签名具有同等的安全强度，下面在保证同等安全强度的情况下比较以上断言长度的增加值，则 IBASPV 方案比 PKI 方案的报文长度的减少幅度 λ 为：

$$\lambda=\frac{L_{SAML_n}-L'_{SAML_n}}{L_{SAML_n}}=0.88-\frac{0.63}{n}$$

3 基于 IBASPV 协议的 Web 服务认证调用性能分析

文献[6]基于 Web 服务安全模块 Rampart 测量并分析了 Web 服务安全调用响应时间，本节改进文献[6]的实验方案对基于 IBASPV 协议的服务认证调用性能进行测试分析。测试分析的总体思路是：首先通过测量基于 PKI 签名的服务调用响应时间，分析响应时间增加的原因，并结合 2.3 节中分析得到的 IBASPV 方案比 PKI 方案的报文长度的减少幅度，对比分析基于 IBASPV 协议的 Web 服务调用性能。

3.1 测试环境

Web 服务认证调用测试环境如图 2 所示，测试用基准服务发布在服务端应用服务器 Tomcat 的 Web 服务引擎 Axis2 容器中，其功能是服务端将客户端发送的字符串返回给客户端。通过配置安全模块 Rampart[7] 实现基于 PKI 签名的 Web 服务调用认证。将服务端和客户端部署在同一台计算机上，配置 Web 服务使用的端口为 8080，使用 TCP/IP Monitor 对服务调用过程中传递的 SOAP 消息进行监测，设置监测端口为 8081。本机的 CPU 为双核 1.73GHz，内存为 2GB。

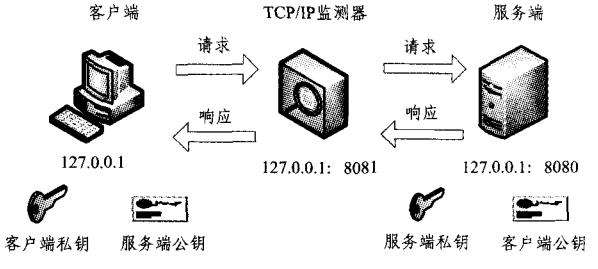

图 2 Web 服务认证调用测试环境

3.2 测试步骤

为了比较不同报文长度下服务调用的响应时间，设置以下两个测试场景：

测试场景 1 客户端向服务端发送长度为 1000 字节的字符串。

测试场景 2 客户端向服务端发送长度为 10000 字节的字符串。

对以下 Web 服务调用类型进行测试：

调用类型 1 没有认证的 Web 服务调用，对 SOAP 消息

不进行签名操作。

调用类型 2　基于 PKI 签名的服务认证调用。首先对 SOAP 消息使用哈希函数 SHA-1 生成 160 比特的报文摘要，随后对摘要使用 RSA 算法进行签名，签名私钥长度为 1024 比特。

使用 TCP/IP Monitor 测试服务调用的响应时间为 T'_{rsp}，请求报文长度为 L_{rqt}，响应报文长度为 L_{rsp}，为了消除误差，每个服务调用测试 30 次，T'_{rsp} 取平均值，分别记录报文的长度。测试场景 1 的结果如表 1 所列，测试场景 2 的结果如表 2 所列。

表 1　服务调用测试场景 1 结果

服务调用类型	响应时间 T'_{rsp}(ms)	请求报文长度 L_{rqt} (byte)	响应报文长度 L_{rsp} (byte)
1	5	1706	1604
2	42	4585	5211

表 2　服务调用测试场景 2 的结果

服务调用类型	响应时间 T'_{rsp}(ms)	请求报文长度 L_{rqt} (byte)	响应报文长度 L_{rsp} (byte)
1	6	10720	10612
2	44	13599	14219

3.3　结果分析

3.3.1　Web 服务响应时间

响应时间 T_{rsp} 是指在 Web 服务调用过程中，从发出 SOAP 请求消息到收到 SOAP 响应消息的时间。响应时间包括 SOAP 消息处理时间 T_p 和 SOAP 消息在网络中的传输时间 T_t，计算公式如下：

$$T_{rsp}=T_p+T_t$$

本文是在同一台计算机的不同端口之间进行的测试，数据网络传输时间 T_t 可以忽略。但在实际网络环境下，T_t 不能够忽略。假设服务调用发生在位于网络环境的客户端 A 和服务端 B 之间，A 与 B 之间的通信链路长度为 L_{AB}(单位：km)，网络中数据传输速率为 v(单位：Mbps)，电磁波传输速率 $c=3\times10^5$ km/s，则 T_t 的计算公式如下：

$$T_t=\frac{L_{rqt}}{v}+\frac{L_{rsp}}{v}+2\frac{L_{AB}}{c}$$

基于网络环境下 A 与 B 之间 Web 服务调用的响应时间 T^{AB}_{rsp} 为：

$$T^{AB}_{rsp}=T'_{rsp}+T_t$$

对表 1 和表 2 中的测量时间进行修正，得到测试场景 1 和测试场景 2 在网络环境下的 Web 服务调用响应时间 T^{AB}_{rsp}，如表 3 所列。

表 3　网络环境下的 Web 服务调用响应时间

服务调用类型	T^{AB}_{rsp}(ms)（测试场景 1）	T^{AB}_{rsp}(ms)（测试场景 2）
1	$5+\frac{26.4}{v}+6.7L_{AB}\times10^{-3}$	$6+\frac{170.7}{v}+6.7L_{AB}\times10^{-3}$
2	$42+\frac{78.4}{v}+6.7L_{AB}\times10^{-3}$	$44+\frac{222.5}{v}+6.7L_{AB}\times10^{-3}$

3.3.2　响应时间增加的因素

对 SOAP 消息添加签名运算的 Web 服务调用响应时间增加的因素有以下两个：

a)签名运算本身增加的时间开销；

b)由于签名运算导致的 SOAP 消息长度增加造成的传输时间增加，主要是在 SOAP 消息头部增加了签名值以及签名验证公钥等安全参数，导致报文长度增加。

为了分析以上因素对响应时间增加的影响程度，分别定义因素 a 和因素 b 对响应时间增加的贡献率为 η_a 和 η_b：

$$\eta_a=\frac{T_a}{\Delta T_{rsp}},\eta_b=\frac{T_b}{\Delta T_{rsp}}$$

式中，T_a 为签名运算的时间开销，T_b 是由签名运算造成的报文长度增加而导致的传输时间的增加值，ΔT_{rsp} 是添加了签名运算的服务比普通服务增加的响应时间。

η_a 计算过程如下：

首先计算 T_a，取表 1 和表 2 中添加安全操作的服务响应时间测量值 T'_{SS} 减去普通服务响应时间测量值 T'_{NSS}，即：

$$T_a=T'_{SS}-T'_{NSS}$$

计算 ΔT_{rsp}，取表 3 中添加安全操作的服务响应时间 T_{SS} 减去普通服务响应时间 T_{NSS}，即

$$\Delta T_{rsp}=T_{SS}-T_{NSS}$$

则得到 η_a：

$$\eta_a=\frac{T'_{SS}-T'_{NSS}}{T_{SS}-T_{NSS}}$$

由因素 a 和因素 b 的互补关系，得到：

$$\eta_b=1-\eta_a$$

针对测试场景 1 和 2，经过计算得到因素 a 和因素 b 分别对响应时间增加的贡献率，如表 4 所列。

表 4　因素 a 和 b 对响应时间增加的贡献率

序号	η_a	η_b
测试场景 1	$\frac{v}{v+1.4}$	$\frac{1.4}{v+1.4}$
测试场景 2	$\frac{v}{v+1.4}$	$\frac{1.4}{v+1.4}$

可以看出：1)在测试 1 和测试 2 中，当 SOAP 消息传输的应用数据长度变化时，因素 a(或因素 b)对响应时间增加的贡献率 η_a(或 η_b)保持基本一致的变化规律；2)因素 a 和因素 b 对响应时间增加的贡献率随着网络数据传输速率 v 而变化，当 v 较小时因素 b 起主要作用，当 v 较大时因素 a 起主要作用，变化规律如图 3 所示，可以得出，当 $v=1.4$Mbps 时，$\eta_a=\eta_b=50\%$；当 $v<1.4$Mbps 时，$\eta_a<\eta_b$；当 $v>1.4$Mbps 时，$\eta_a>\eta_b$。

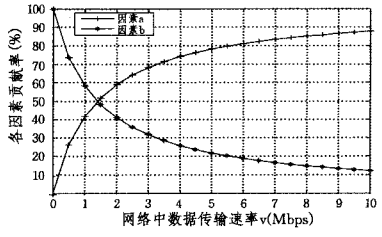

图 3　因素 a 和 b 对响应时间增加的贡献率

3.3.3　基于 IBASPV 协议的 Web 服务认证调用性能

从 Web 服务响应时间的角度，分析基于 IBASPV 协议的服务调用的性能。以下从因素 a 和 b 分析当使用基于身份聚合签名(IBAS)算法的 IBASPV 方案替换基于 RSA 算法的 PKI 多次签名方案时，服务响应时间的变化。

首先，分析因素 a 对服务响应时间带来的影响。通过表 5 可知，软件实现的基于配对运算的 IBAS 算法和基于 RSA 算法的运算时间基本一致。因此，当使用 IBASPV 方案替换 PKI 方案时，可以认为因素 a 对服务响应时间造成的变化不

大。

表 5　软件实现的 IBAS 算法和 RSA 算法运行时间

算法	运行时间(ms)
IBAS 签名[8]	43
RSA 签名[9]	50

分析因素 b 对服务响应时间带来的影响。由 2.3 节分析可知：随着 SAML 断言传递路径长度 n 的增加，IBASPV 方案比 PKI 方案的报文传送效率提高：

$$\lambda=0.88-\frac{0.63}{n}$$

与 PKI 方案相比，IBAS 方案减少了密码运算引起的报文长度增加，从而减少了报文传输时间，进而缩短了服务响应时间，则由因素 b 带来的响应速度提升率为 λ。

综合因素 a 和 b 对响应时间的影响，可知因素 b 对基于 IBASPV 方案的服务响应速度提升率 μ 起主要作用，μ 的计算公式为：将 η_b 作为影响因子与由因素 b 带来的响应速度提升率 λ 相乘，得到：

$$\mu=\eta_b\cdot\lambda=\left(\frac{1.4}{v+1.4}\right)\left(0.88-\frac{0.63}{n}\right)$$

服务响应速度提升率 μ 与网络数据传输速率 v 和服务调用链长度 n 的关系如图 4 所示：1)当网络数据传输速率 v 固定时，随着服务调用链长度 n 的增加，IBASPV 方案比 PKI 方案有明显的性能提升；2)网络数据传输速率 v 越小，IBASPV 方案的性能提升幅度越大，随着网络传输速率 v 的增加，IBASPV 方案性能提升的幅度逐渐减小。

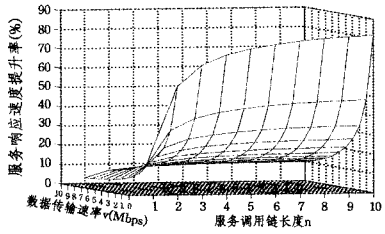

图 4　IBASPV 方案比 PKI 方案服务响应速度提升率

以下基于不同网络传输速度的 Web 服务调用环境，举例说明基于 IBASPV 协议的服务调用响应速度比基于 PKI 多次签名方案的服务调用响应速度的提升率。根据实践经验，SAML 断言传递路径长度多为 2 到 7 之间。权威机构 Akamai 在 2011 年第三季度公布的中国互联网用户终端平均网速约为 1.4Mbps，全球互联网用户终端平均网速约为 2.7Mbps[10]。据此假设，$2\leqslant n\leqslant 7$；当 $v=1.4$Mbps 时，响应速度提升约为 28%～40%；当 $v=2.7$Mbps 时，响应速度提升约为 19%～27%。若校园网环境中终端网速 $v=10$Mbps，则响应速度提升约为 7%～9%。

结束语　为了解决基于 PKI 签名机制的服务认证调用显著降低响应速度的问题，提出了基于身份聚合签名的 SAML 路径验证协议，并采用 SVO 逻辑证明了协议的安全性。该协议在保证断言传递过程中完整性、源不可伪造性和传递路径不可伪造的同时，通过减少签名后断言的增长幅度，达到提高服务响应速度的目的。通过实验分析得到了引起响应时间增加的主要因素随网络数据传输速率的变化趋势，然后对比分析了基于本文协议与基于 PKI 多次签名协议的 Web 服务认证调用性能，结果表明本文提出的 IBASPV 协议能够有效提高服务响应速度。

参 考 文 献

[1] OASIS. Assertions and Protocols for the OASIS Security Assertion Markup Language(SAML) V2.0[S]. 2005

[2] OASIS identifier: wss-v1.1-spec-os-SOAPMessage Security, Web Services Security: SOAP Message Security 1.1[S]. 2006

[3] Gentry C, Ramzan Z. Identity Based Aggregate Signatures[C]// Proc. of Public Key Cryptography. Springer, 2006

[4] Syverson P, Van Oorschot P. On Unifying Some Cryptographic Protocol Logics[C]// IEEE Computer Society Symposium on Principles of Distributed Computing. ACM Press, 1994, 5: 14-28

[5] Michael C. Pairing Calculation on Supersingular Genus 2 Curves [C]// Proc. of SAC'06. 2006

[6] Rodrigues D, Estrella J C. Analysis of security and performance aspects in service-oriented architectures [J]. International Journal of Security and its Applications, 2011, 5(1): 13-30

[7] The Apache Rampart Project[Z]. http://axis.apache.org/axis2/java/rampart, 2011

[8] Berreto. A Note on Efficient compution of cube roots in charateristic3[EB/OL]. http://eprint.jacr.org/2004/305, 2004

[9] Zhao M, Smith S, Nicol D. Aggregated Path Authentication for Efficient BGP Security[C]// ACM Conference on Computer and Communication Security. Alexandria, USA, 2005: 128-138

[10] Akamai. State of Internet report[R]. http://www.akamai.com/stateofinternetreport/, 2011

(上接第 169 页)

[5] Sider A, Couturier R. Fast load balancing with the most to least loaded policy in dynamic networks[J]. Supercomput, 2009, 49: 291-317

[6] Bahi J M, Couturier R, Sider A. Design and analysis of the M2LL policy distributed algorithm for load balancing in dynamic, networks. 2006[C]// Heidelberg, Springer, Proc of the 2006 int symp on parallel and distributed processing and applications (ISPA'06). LNCS, vol 4331, 2006: 195-204

[7] 杨夏妮，覃海生. 基于 Petri 网的动态负载平衡双层调度模型研究[J]. 广西科学院学报，2008，24(2)：296-299

[8] 王少峰，周忠，吴威. 一种面向分布式虚拟环境的分层迭代负载平衡算法[J]. 软件学报，2008，19(9)：2471-2482

[9] 王宏宇，何利娟，杜晓丽. 基于计算场的网格动态负载平衡算法[J]. 河北大学学报：自然科学版，2011，31(2)：208-213

[10] Aakanksha, Bedi P. Load balancing on dynamic network using mobile process groups[C]// 15th International Conference on Advanced Computing and Communications. 2007

[11] Xu C Z, Lau F C M. Optimal parameters for load balancing with the diffusion method in mesh networks[J]. ParallelProcess, 1994, 4(2): 139-147