基于 SOAP 和 SRP-6 协议的认证与密钥交换方案

李 冬 郭荷清

(华南理工大学计算机科学与工程学院 广州 510640)

摘 要 SRP(Secure Remote Password)是一种基于密码的强认证协议。本文分析了 SRP 协议的密钥交换机制,提出了一种基于 SOPA 和 SRP-6 协议的密钥交换与认证方案 SRP-over-SOAP,并将该方案用于 Web Service,实现了服务器和客户机间的双向身份认证。据我们所知,该方案是第一个将 SOAP 用于 SRP-6 协议的方案。

关键词 SRP 协议,SOAP,认证,密钥交换

Authentication and Key Exchange Scheme Based on SRP-6 and SOAP

LI Dong GUO He-Qing

(Dept. of Computer Science and Engineering, South China University of Technology, Guangzhou 510640)

Abstract Secure Remote Password(SRP)protocol is a strong network authentication mechanism. In this paper, we analyze the SRP protocol and propose an authertixation and key exchange scheme which is based on SRP-6 and SOAP (SRP-over-SOAP). As far as we know, this is the first implementation of SRP-6 over SOAP protocol. We also point out how to use this scheme to Web service for secure key exchange and authentication.

Keywords SRP, SOAP, Authentication, Key exchange

1 引言

认证和密钥交换是网络安全的重要组成部分。通常的用户认证方案是通过比较〈用户名,密码〉对来实现的。由于需要在通信线路上传送密码以及用户可能选择一个简单的密码,这种方法很容易遭到窃听和字典攻击。

SRP(Secure Remote Password)是一种基于密码的强认证协议,SRP 的密钥交换机制使得认证双方既不需要以任何形式存储密码,也不需要在公众网上传输密码,并且允许用户选择一个简单的密码用于身份认证而不必担心密码遭受字典攻击。

本文分析了 SRP 协议的密钥交换机制,提出了一种基于 SOPA 和 SRP-6 协议的密钥交换与认证方案 SRP-over-SOAP,并将该方案用于 Web Service,实现了服务器和客户机间的双向身份认证。

2 SRP 认证及密钥交换机制

SRP 协议是斯坦福大学开发的口令认证和密钥交换体制,提供客户端和服务器间的强相互认证,能同时抵制来自网络的主动攻击和被动攻击。

SRP 协议只在服务器端存储密码的 Hash 值,不需要认证双方存储原始密码。即使攻击者获得了密码的 Hash 值,也需要通过耗时的字典攻击才能获得原始密码。SRP 协议认证过程不需要通过网络传输原始密码,使得攻击者无法通过通信线路窃听来获得原始密码。

在认证过程中,服务端和客户端交换各自的临时公钥,这些临时公钥在每一次认证会话过程中都不相同,避免了重放攻击。

SRP 协议认证过程描述如下:

1. 协议初始化过程:产生一个安全的大素数 n,所有的加、乘、幂运算都基于有限域 $GF(n)$,g 是 $GF(n)$的一个生成元;

2. 客户端注册过程:客户端自由选择口令 P 和盐值(salt)s,计算 $x=H(s,P)$,$v=g^x$,然后将用户名 U,校验符 v、盐值 s:〈U,v,s〉提交给服务器的校验符数据库;

3. 客户机向服务器发送用户名 U 作为认证请求;

4. 服务器从数据库中查找该用户的校验符 v 和盐值 s,发送 s 给客户机;

5. 客户机使用用户从终端输入的 P 和 s 计算出 x;

6. 客户机生成随机数 a,计算出临时公钥 $A=g^a$,发送给服务器;

7. 服务器生成随机数 b,计算出临时公钥 $B=v+g^b$,将 B 与另一随机参数 u 一起发送给客户机;

8. 客户机计算会话密钥 $S=(B-g^x)^{a+ux}$,服务器计算 $S=(Av^u)^b$。如果用户口令 P 没有错误,则双方计算出的 S 应该是相等的;

9. 客户机和服务器分别计算强会话密钥 $K=H(S)$;

10. 客户机计算 $M1=H(A,B,K)$,发送给服务器,以证明自己已经正确获得会话密钥 K,服务器用自己计算的 K 来验证 $M1$,以确认客户机身份;

11. 服务器计算 $M2=H(A,M1,K)$,将其发送给客户机,以证明自己也已经正确获得 K,客户机验证 $M2$,以确认服务器身份;

至此,客户机和服务器都完成了相互认证,并建立了会话密钥 K,用以保护后续会话的机密性。

SPR 协议认证过程如图 1 所示。

可以通过数学证明:当客户机提供正确的〈U,P〉时,客户机和服务器最后得到的会话密钥是相同的[1,2]。

1998 年提出的 SRP 协议被称为 SRP-3 协议。SRP-3 协议具有较强的安全特性,但也存在两个方面的问题[3]:

其一是“Two-for-One guessing”问题，由前述可知：临时公钥 $B=v+g^b$，而 $v=g^x$，由此可知 $B=g^x+g^b$，其中 $x=H(s,P)$，b 是一个随机数。攻击者可能通过猜测 x 或 b 来伪装成服务器，以达到欺骗客户机的目的。

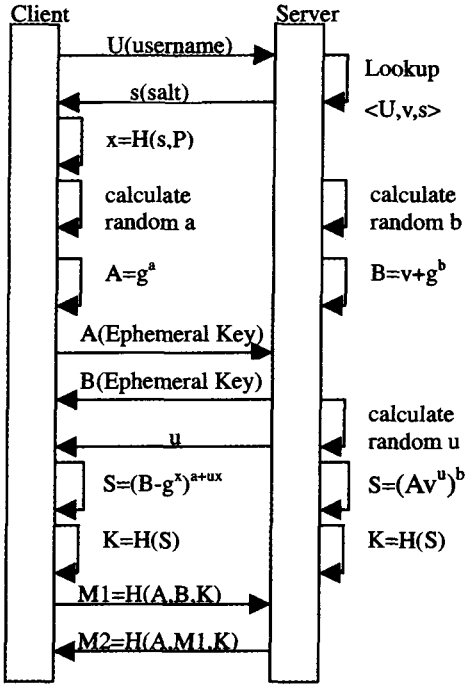

图 1　SRP 认证和密钥交换过程

其二是“Message Ordering”问题，SRP-3 协议的认证过程必须严格按照图 1 所示来进行，否则有可能遭到伪装攻击。考虑如下情况：在客户机发送 A 之前，服务器已经发送了随机数 u。攻击者事先窃取服务器校验数据库获得了〈U，v，s〉，然后截获 u，这样攻击者就可以发送 $A=g^a v^{-u}$ 代替 $A=g^a$ 来哄骗服务器，这种攻击方法可以使攻击者绕过执行耗时的字典攻击。

关于 SRP-3 协议的讨论在文[3]中有详细论述。斯坦福大学于 2002 年提出了 SRP-3 的改进协议 SRP-6，解决了上述两个问题。SRP-6 协议不需要在服务器和客户机之间发送随机数 u，而是由双方自行计算出 $u=H(A,B)$。SRP-6 协议流程如图 2 所示。

	Client		Server
1.	U	→	(Lookup<U,s,v>)
2.	x=H(s,P)	←	s
3.	$A=g^a$	→	
4.		←	$B=3v+g^b$
5.	u=H(A,B)		u=H(A,B)
6.	$S=(B-3g^x)^{a+ux}$		$S=(Av^u)^b$
7.	M1=H(A,B,S)	→	(verify M1)
8.	(verify M2)	←	M2=H(A,M1,S)
9.	K=H(S)		K=H(S)

图 2　SRP-6 协议流程

SRP-6 杜绝了攻击者伪装成合法的客户机或服务器的可能，并且允许开发者定义更灵活的客户机-服务器间的握手规则。

3　基于 SOAP 的 SRP-6 认证和密钥交换(SRP-over-SOAP)

根据前文对 SRP 协议的认证过程分析，我们设计了基于 SOAP 的 SRP-6 认证和密钥交换方案(SRP-over-SOAP)，SRP-over-SOAP 将 SOAP 消息进行扩展，通过在 SOAP 头中加入〈SRPAuth〉标记，实现了使用 SOAP 消息传递 SRP 认证信息的目的。

一个从客户端(Client)发出的请求建立认证过程的 SOAP 消息如下：

```
〈SOAP-ENV：Header〉
  〈SRPAuth：SRPAuthentication〉
    〈SRPAuthticationRequest〉
  〈client〉username〈/client〉
  〈/SRPAuthticationRequest〉
  〈/SRPAuth：SRPAuthentication〉
〈/SOAP-ENV：Header〉
〈SOAP-ENV：Body〉
  〈auth：getSalt〉
  〈user〉username〈/user〉
  〈/auth：getSalt〉
〈/SOAP-ENV：Body〉
```

服务器端(Server)接收到认证请求后，根据客户机提供的用户名〈user〉 username〈/user〉查询自己的 SrpDb，找到属于该用户名的三元组：用户名 U，校验符 v、盐值 s，并将 s 作为响应消息的参数发回给客户端：

```
〈SOAP-ENV：Header〉
  〈SRPAuth：SRPAuthentication〉
    〈SRPAuthticationResponse〉
      〈server〉servername〈/server〉
    〈/SRPAuthticationResponse〉
  〈/SRPAuth：SRPAuthentication〉
〈/SOAP-ENV：Header〉
〈SOAP-ENV：Body〉
 〈auth：getSaltResponse〉
   〈salt〉user's salt〈/salt〉
 〈/auth：getSaltResponse〉
〈/SOAP-ENV：Body〉
```

客户端接收到 salt 后计算出临时密钥 A，并发送给服务器端，并请求服务器的临时密钥 B：

```
〈SOAP-ENV：Body〉
  〈auth：setPublicKeyA〉
    〈A〉ephemeral A〈/A〉
  〈/auth：setPublicKeyA〉
  〈auth：getPublicKeyB〉
    〈server〉servername〈/server〉
  〈/auth：getPublicKeyB〉
〈/SOAP-ENV：Body〉
```

服务器根据客户机的请求作出发送临时密钥 B 的响应：

```
〈SOAP-ENV：Body〉
 〈auth：getPublicKeyBResponse〉
   〈B〉 ephemeral B〈/B〉
 〈/auth：getPublicKeyBResponse〉
〈/SOAP-ENV：Body〉
```

至此，服务器和客户机完成了临时公钥的交换，双方各自计算出 u、$M1$、$M2$ 的值，并交换 $M1$ 和 $M2$ 各自进行匹配，交换 $M1$、$M2$ 时仍然是使用 set 方法和 get 方法进行。通信时使用的 SOAP 消息格式与前文相似，在此不再赘述。

若任何一方的匹配结果不成功，则发出〈SOAP：Fault〉消息，提示认证过程出错，密钥交换失败。如果匹配成功，则各自计算出会话密钥 K，用于后继的通信。

SRP-over-SOAP 将 SOAP 用于 SRP 认证与密钥交换过程。由于 SOAP 协议已经是 WebService 的事实上的基本通信协议，因此 SRP-over-SOAP 使得在 Web Service 中使用 SRP 机制进行强身份认证和安全密钥交换成为可能。据我们所知，SRP-over-SOAP 第一次实现了基于 SOAP 消息的 SRP-6 认证和安全密钥交换过程。

4 基于 SRP-6 的认证服务器系统 SrpAuth

具有 SRP 认证机制的 Web Service 系统应该解决两个问题，其一是扩展 SRP 协议使之适合应用于 Web Service 中，本文已经通过 SRP-over-SOAP 解决了此问题；其二是该认证服务必须部署简便和使用简单；本文在 J2EE 平台上实现了一个使用 SRP-6 认证机制的 Web Service 身份认证和密钥交换系统 SrpAuth。实际使用时，可将该系统置于普通 Web Service 系统之上，实现服务提供者和服务请求者之间的强身份认证。SrpAuth 结构如图 3 所示。

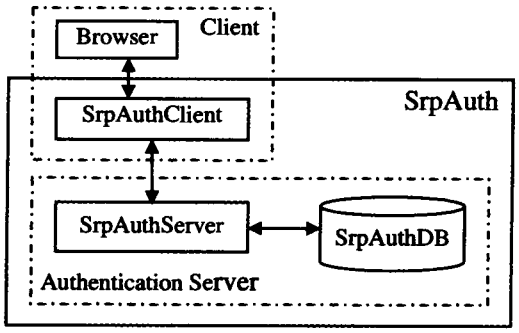

图 3 SrpAuth 系统结构

当客户端通过浏览器访问服务时，首先执行 SrpAuthClient 程序，SrpAuthClient 按照 SRP-6 协议的握手程序与服务器端的 SrpAuthServer 通信，实现身份认证和建立安全会话密钥。SrpAuthDB 存储了三元组 $\langle U,v,s\rangle$，当 SrpAuthServer 接收到 SrpAuthClient 发来的用户名时，在 SrpAuthDB 中查询响应的 $\langle U,v,s\rangle$，以进行后续通信。SrpAuth 各组件的工作流程如图 4 所示。

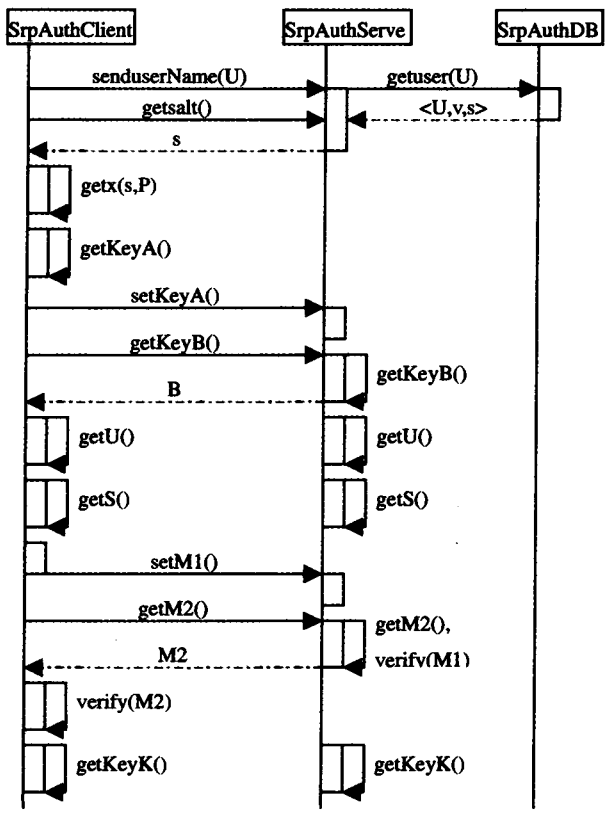

图 4 SrpAuth 认证和密钥交换过程

5 SrpAuth 与 Web Service 的集成方案

实际应用中，可将 SrpAuth 置于 Web Service 系统之上，当客户请求服务时，必须先通过 SrpAuth 的身份认证，根据身份认证结果，由 SAML 做出身份断言，XACML 根据断言作出该客户的访问决策。图 5 展示了将 SrpAuth 用于 Web Service 的情形。

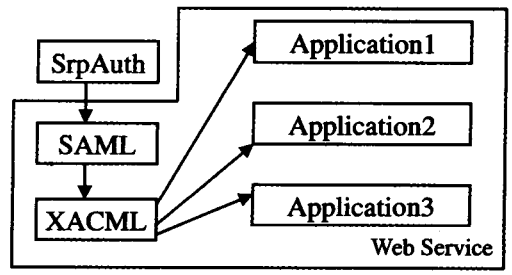

图 5 SrpAuth 和 Web 服务的集成

当客户身份认证成功后，由 SrpAuthServer 向 SAML 发送一个标识客户身份的 SOAP 消息，该 SOAP 消息的内容包含了客户身份及认证结果，SAML 根据该消息做出身份断言。下面是 SrpAuthServer 发往 SAML 的 SOAP 消息：

```
〈SOAP-ENV：Header〉
  〈SRPAuth：SRPAuthentication〉
   〈SRPAuthtication〉
    〈Authserver〉
     Servername
    〈/Authserver〉
   〈/SRPAuthtication〉
  〈/SRPAuth：SRPAuthentication〉
〈/SOAP-ENV：Header〉
〈SOAP-ENV：Body〉
 〈auth：AuthResult〉
  〈user〉user's Name〈/user〉
  〈result〉success〈/result〉
  〈sessionKey〉
  sessionKey's value
  〈/sessionKey〉
 〈/auth：AuthResult〉
〈/SOAP-ENV：Body〉
```

该 SOAP 消息包含了进行身份认证的服务器名、被认证者信息、认证结果及最后达成的会话密钥。比如上例中的〈result〉success〈/result〉则表示认证结果是成功的。

结论与展望 用 SOAP 消息承载 SRP 认证信息，实现了 SRP 协议与 Web Service 的无缝集成，同时也便于在 Web Service 上部署 SRP。

本文提出了一种实现 SOAP 承载 SRP 的方法 SRP-over-SOAP，该方法采用 SRP-6 协议完成客户机与服务器之间的握手。据我们所知，这是 SRP-6 协议第一次在 SOAP 中实现。

SrpAuth 作为一个认证服务器，为 SRP 应用于 Web Service 提供了现实的可能性。SrpAuth 提供了客户端接口和服务器接口，使得系统设计者能够方便地在 Web Service 上部署 SRP。

本文下一步的研究工作将围绕 SRP-over-SAML 展开，即考虑如何采用 SRP 机制认证 SAML 断言的可信性。

由于采用了复杂的 SRP 协议进行身份认证，SRP-over-SOAP 的效率相比传统的身份认证来说，差距较大。如何提高 SRP-over-SOAP 的效率也将是下一步研究的重点。

Schnorr 签名方案的一种攻击*)

刘景美　王新梅
(西安电子科技大学综合业务网国家重点实验室　西安 710071)

摘　要　Schnorr 签名算法计算量少，速度快，在灵巧卡中具有大量的应用，因此必须具有足够的安全性。本文给出了选择消息下对 Schnorr 签名方案的一种攻击方法，攻击者可以假冒签名者进行签名；另外给出了一种攻击签名者私钥的选择消息攻击方法，其攻击性不依赖于离散对数的求解问题。

关键词　Schnorr，签名，安全性分析

Cryptanalysis of Schnorr Signature Scheme

LIU Jing-Mei　WANG Xin-Mei
(National Key Lab. of Integrated Service Networks, Xidian Univ., Xi'an 710071)

Abstract　Schnorr signature is widely used in smart card with little computation and high rate, so it should be enough secure. In this paper a method is presented to forge the signature of Schnorr signature scheme, and we also present a key-recovery attack against the Schnorr signature algorithm under the chosen messages. All the attack actions do not depend on the computation of discrete logarithm.

Keywords　Schnorr, Signature, Cryptanalysis

1　引言

数字签字在信息安全，包括身份认证、数据完整性、不可否认性以及匿名性等方面都有重要应用，特别是在大型网络安全通信中的密钥分配、认证以及电子商务系统中具有重要作用，它是实现电子贸易、电子支票、电子货币、电子购物、电子出版及知识产权保护等系统安全的重要保证。目前大多数的数字签字都是基于大整数分解或离散对数计算的困难性等数学难题，如 1978 年出现的基于大整数分解困难性的 RSA[1] 签字体制和 1985 年出现的基于离散对数计算困难性的 ElGamal[2~7] 签字体制和 ElGamal 签字体制变形后的 Schnorr[8] 签字体制。已被定为美国数字签字标准 DSS 的签字算法 DSA 就是在 ElGamal 和 Schnorr 签字体制的基础上设计的。Schnorr 签名算法计算量少，速度快，在灵巧卡中具有大量的应用，因此必须具有足够的安全性。本文继续对 ElGamal 签字体制的变形 Schnorr 签字体制进行了分析，发现 Schnorr 签字体制在选择消息的情况下是不安全的，不仅可以假冒签名者进行签名，而且可以求出签名者的私钥，攻击性不依赖于离散对数的求解问题。

2　Schnorr 签名方案

1989 年 C. Schnorr 提出的基于离散对数的数字签名方案是一种随机化的签名机制，其签名方案是：

1. 体制参数

p，q：大素数，q|p−1，q 是大于等于 160bit 的整数，p 是大于 512bit 的整数，保证 Z_p 中求解离散对数为困难问题；

g：Z_p^* 中元素，且 $g^q \equiv 1 \bmod P$，也就是说 g 是 Z_p^* 中子群 Z_p^* 的一个生成元；

μ：消息空间，为 Z_p^*；

ζ：签字空间，为 $Z_p^* \times Z_p$；

x：用户秘密钥 $x \in Z_q^*$；

$y \equiv g^x \bmod p$

密钥：$\kappa=(p,g,a,y)$，其中 p,g,y 为公钥，x 为秘密钥。

2. 签字过程

1)选择秘密随机数 $\kappa \in Z_q$；

2)计算 $r=g^k \bmod p$

$s=k+xH(r \| M) \bmod (p-1)=k+xe \bmod (p-1)$，式中 $e=H(r \| M)$；

3)将 $Sig_k(M,k)=S=(r,s)$ 作为签字，将 $M,(r \| s)$ 送给对方，其中 $r \| s$ 表示 r 与 s 的数字串的级连。

3. 验证过程

收信人收到消息 M 及签字 $(r \| s)$ 后，(1)计算 $e=H(r \| M)$；(2)计算 $r'=g^s y^{-e} \bmod p$；(3)验证 $Ver_k(M,r,s) \Leftrightarrow H(r' \| M) \equiv e \bmod p$，若等式成立，则 (r,s) 是对消息 M 的签字，理

*)基金项目：国家 973 项目(G1999035804)；“十五”国家密码发展基金。刘景美　博士研究生，研究方向：网络安全和密码分析学。

参 考 文 献

1　Wu T. The secure remote password protocol. In: Proceedings of the Internet Society Network and Distributed System Security Symposium, 1998. 97～111

2　Wu T. The SRP Authentication and Key Exchange System. Stanford University, September 2000. Request For Comments (RFC)2945

3　Wu T. SRP-6: Improvements and Refinements to the Secure Remote Password Protocol. 2002

4　OASIS. Web Service Security: SOAP Message Security 1. 0. OASIS Standard 200401, 2004