一种基于权能标识的三方安全协议的设计和分析*)

黄建忠[1,2] 谢长生[1] 朱光喜[2] 罗东健[1]
(华中科技大学数据存储教育部重点实验室 武汉 430074)[1]
(华中科技大学电子与信息工程系 武汉 430074)[2]

摘 要 随着数据密集型应用的发展,网络存储的安全性成为研究热点。针对大规模存储系统的可扩展性和安全性的要求,本文提出了一种基于三方传输模式的存储安全系统框架,并设计了一种基于权能标识的三方安全协议,该协议的最大特点是传输安全性和访问控制机制二者独立。通过对三方安全协议的形式化分析和推导,从逻辑上验证请求中权能标识的完整性、协议传输过程的正确性,以及消息的真实性,从而确保了三方安全协议的可行性。

关键词 海量存储系统,权能标识,三方安全协议,形式化分析

Design and Analysis of a Capability-based Third-Party Security Protocol

HUANG Jian-Zhong[1,2] XIE Chang-Sheng[1] ZHU Guang-Xi[2] LUO Dong-Jian[1]
(Key Laboratory of Data Storage System, Huazhong University of Science and Technology, Wuhan 430074)[1]
(Dept. of Electronic & Information Engineering, Huazhong University of Science and Technology, Wuhan 430074)[2]

Abstract With the growth of data-intensive application, networked storage security becomes hotpot of research. Aiming at the scalability and security requirement of large-scale storage system, a storage security framework basing on third-party transferring mode is proposed, and a capability-based third-party security protocol separating transfer security from access control mechanism is designed. After the formal analysis, it is clear that the deducing results logically validate the integrity of requested capability, correctness of transfers process and authenticity of message, thus guaranteeing the feasibility of the third-party security protocol.

Keywords Massive storage system, Capability, Third-party security protocol, Formal analysis

1 引言

随着数字信息量的爆炸式增长,出于简化管理和经济省钱等因素的考虑,网络存储系统成为一个必然的趋势。网络存储系统的网络特性和数据共享特性很容易遭致外来的攻击,如果没有一个良好的安全防范措施,就很容易造成存储资源的非法泄漏。针对这种情况,近年来,网络存储安全问题逐步成为计算机存储领域的研究热点[1]。

目前,主要从4个方面对网络存储安全进行研究:1)增强文件服务器的安全,如使用 Kerberos[2] 认证机制来增强 AFS 服务器的安全性;2)客户端加密文件系统,例如在 CFS[3] 中,客户端文件系统对数据加密,并存放到存储设备上,从而保护数据;3)客户端直接存取磁盘的认证机制,例如 NASD[4] 利用设备本身的处理能力对请求进行验证,防止非授权用户访问特定的数据,达到限制访问的目的;4)高度可扩展的文件系统,例如 OceanStore[5] 将文件或目录的加密副本存放到多个存储设备上,避免恶意用户删除数据。

下一代互联网具有更大、更快、更安全的特点,对存储系统提出了相应的要求:更大存储规模、更大存储容量、更大存储带宽、更强存储安全性、更多的智能性[6]。这些要求是上述几种存储安全方案所无能为力的:采用 Kerberos 的 AFS 只能保证安全性中的机密性;CFS 是一个客户端的加密文件系统,主要用于个人计算机;NASD 在控制器上添加一定的处理能力,受控制器处理能力的限制,只能处理一些简单的、既定的安全任务;OceanStore 侧重于数据的可用性,对机密性、完整性考虑较少,而 SFS-RO 针对的是只读应用环境。

如前可知,这些系统的研究出发点一般是保证数据安全性的某一方面,如机密性、完整性,并且针对特定的应用环境。对于可扩展的大规模存储环境,不管从方案的框架,还是从方案的出发点看,都无法满足。针对这种情况,本文从系统级的角度进行考虑,给出了一种基于三方传输模式的存储安全系统框架,并结合该框架给出了一种基于权能标识的三方安全协议。

2 基于三方传输模式的存储安全系统

2.1 三方传输模式

我们将配备专门控制服务器的通信模式称为三方传输模式,该控制服务器称为元数据服务器(Metadata Server, MDS),三方指代客户端、MDS、存储端。三方传输模式的基本思想是在不同通道上传输元数据和数据信息,并由 MDS 来管理元数据。元数据是关于数据的数据,这里特指数据的属性信息和相关的控制信息。属性信息包括如大小、文件组、拥有者、修改者、创建时间等信息,控制信息包括访问控制列表、读/写操作请求等信息。

*)本文得到国家"973"重大基础研究项目(2004CB318203)和国家自然科学基金(60603074)资助。**黄建忠** 博士,主要研究方向为网络存储安全和对象存储安全;**谢长生** 教授,博导,主要研究方向为网络存储系统、采用新原理的超高密度、超高速存储技术从事;**朱光喜** 教授,博导,主要研究方向为图形处理、多媒体通信和第四代移动通信等。

三方传输的工作流程是：客户端向 MDS 发出元数据请求，MDS 返回元数据，客户端利用该元数据直接与存储端交互，存储端将请求结果直接返回给客户端，即数据通道绕过 MDS，如图 1 所示。

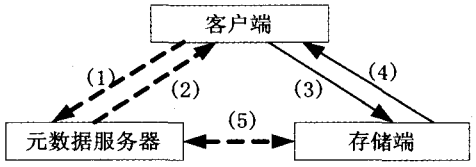

图 1　三方传输模式示意图

图 1 中(1)、(2)指代并发的元数据访问；(3)、(4)为并发的数据访问；(5)指代元数据的管理，如查询、创建元数据。

在 MDS 控制和协调下，三方传输模式具有下列优点：1)对存储系统的所有元数据进行全局的管理，保证了系统的整体功能和性能；2)元数据和数据信息相分离，数据的传输无需经过服务器，使得数据 I/O 路径缩短，提高了数据传输率；3)元数据和数据信息在不同通道上异步传输，避免重传机制消耗网络带宽，保证了网络传输带宽。

2.2　权能标识机制

Van Horn 于 1966 年提出标识(Capability)的概念，其基本思想是：应用程序若要访问一个对象，它必须拥有某一特定的标记(Token)，该标记能指定一个对象以及该对象上的一组操作[7]。本文借鉴了 Capability 的概念，并针对三方传输模式进行定义，记为权能标识。这里将权能标识视为自描述证书，同时具备属性信息和行为信息，对存储空间的某一区域赋予特定类型和特定权限的访问，其结构如图 3 所示。图中安全方法指所采用的安全策略，如加密、*MAC* 等。

R	W	A	T	C	D	I
读取	写入	增加	截断	创建	删除	信息

图 2　访问权限的结构示意图

ID	AL	AR	SM	TS	Res
标志符	地址列表	访问权限	安全方法	时间戳	保留项

图 3　权能标识的结构示意图

图 4 所示为采用 *MAC* 和加密机制的权能标识 $C_{Crypt+MAC}$，图中 *Ks* 指权能标识的私钥，MDS 在生成权能标识时同时产生私钥 *Ks*。*Ks* 具有唯一性；*Ke* 和 *Km* 分别称为加密密钥和认证密钥，用于加密和消息认证码计算，并在存储端和 MDS 之间交换、共享；$E_{Ke}(Ks)$ 表示用 *Ke* 对 *Ks* 进行加密；MAC_{Km} 指使用 *Km* 对权标识进行 *MAC* 计算。

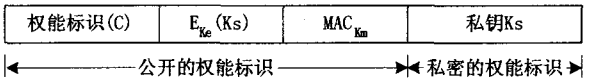

图 4　含认证码和加密的权能标识

权能标识的完整性和真实性由 MAC_{Km} 保证，其 *MAC* 值由认证密钥 *Km* 计算得出。当存储端收到用户请求和加密后的权能标识时，存储端先使用 *Km* 验证权能标识的真实性，再用 *Ke* 解密出私钥 *Ks*，利用 *Ks* 进一步验证通道是否安全可靠。

2.3　基于三方传输模式的存储安全系统

根据三方传输模式下元数据和数据信息在不同通道上传输的思想，并在权能标识的基础上，本文提出了一种基于权能标识的三方存储安全系统框架，如图 5 所示。

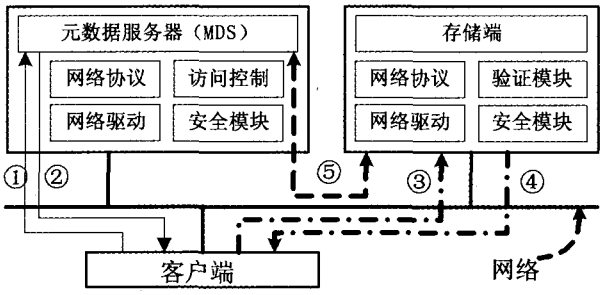

图 5　基于三方传输模式的存储安全系统框架

三方传输的存储安全系统框架的工作流程如下：1)客户端向 MDS 发出请求，以获取权能标识，如图中①；2)MDS 根据客户端的请求和信任程度，给每个请求发出权能标识及其密文哈希值，如图中②；3)得到权能标识的客户端向存储端发出数据请求，如图中③；4)存储端解密请求中的权能标识，并根据权能标识发回响应，如图中④，对于读数据请求，返回请求的数据以及相应的新权能标识；5)MDS 和存储端周期性地进行元数据更新，如元数据的创建、删除、修改，如图中⑤。

采用权能标识的三方传输存储框架除了具备三方传输框架的可扩展性外，还具备如下的安全特性：1)元数据通道和数据通道是两条不同的传输通道，符合“区别对待不同重要性的数据”这一安全原则；2)存储端通过权能标识检查请求的访问权限，防止客户端访问未授权数据，同时对权能标识进行加密和消息认证码操作，可保证访问数据的机密性和完整性。

如图 3 所示，上文定义的权能标识是一个结构灵活、功能完善的数据结构：在安全方法项中指定安全策略，如使用加密算法来保证机密性，使用哈希操作来确保完整性；使用扩展项来防范不同的攻击，如采用时间戳来避免重放攻击。

3　基于权能标识的三方安全协议

根据上节的三方存储安全框架的工作流程，将传输过程划分成 3 个独立的流程(即安全协议)：1)客户端-MDS 协议，客户端需要新的权能标识时，由客户端发起；2)MDS-存储端协议，用于更新共享密钥 *Ke* 和 *Km*；3)客户端-存储端协议，存储端用来验证来自客户端的请求是否经过 MDS 的授权。

3.1　客户端-MDS 协议

客户端为某操作向 MDS 请求一个权能标识，MDS 根据预定的安全策略判定是否同意该请求操作。若允许的话，MDS 生成私钥 *Ks*，并将执行加密操作和 *MAC* 操作的权能标识 $C_{Crypt+MAC}$ 发送给客户。另外，私钥 *Ks* 是未加密的(如图 4 所示)，所以必须在安全通道上传送。客户端-MDS 的交互过程如图 6 所示。

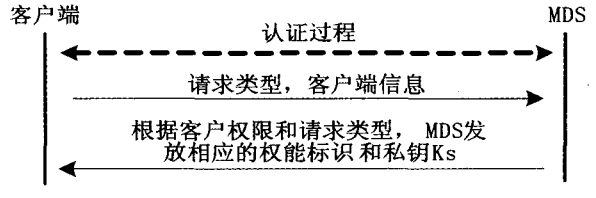

图 6　客户端-MDS 的安全协议

3.2　MDS-存储端协议

该协议实际上是一个密钥交换协议，用于更新存储端和

MDS 共享的两种密钥：加密密钥 *Ke* 和 *MAC* 密钥 *Km*，如图 7 所示。

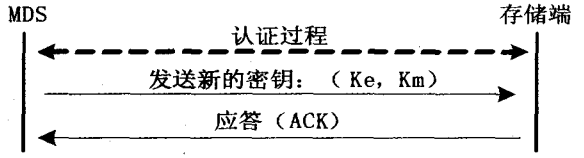

图 7　MDS-存储端的安全协议

3.3　客户端-存储端协议

客户端-存储端协议是三方安全协议的核心部分，它由 3 个阶段组成：握手、会话、注销，其示意图如图 8 所示。

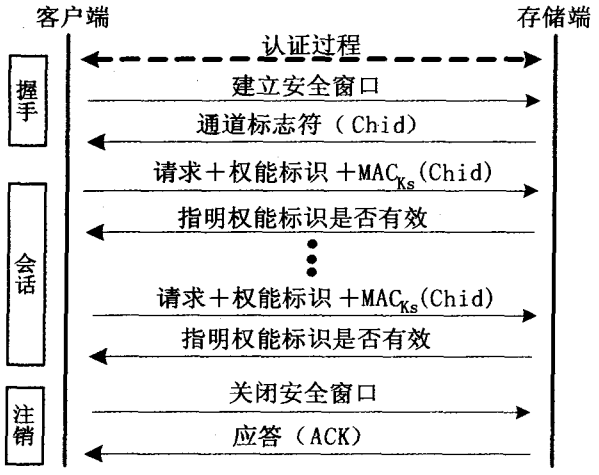

图 8　客户端-存储端的安全协议

(1)握手阶段：客户端请求建立安全窗口，存储端以一组随机选择的通道标志符 *Chid* 作为回应，并将 *Chid* 和会话二者关联起来。

(2)会话阶段：客户端将一个附有权能标识 $C_{Crypt+MAC}$ 和加密的通道标志符 $MAC_{Ks}(Chid)$ 的请求发送给存储端；存储端先用 *Km* 验证 $C_{Crypt+MAC}$ 的完整性，再用 *Ke* 从 $C_{Crypt+MAC}$ 解密出私钥 *Ks*，利用 *Ks* 来验证 *Chid* 的真实性。只要任一验证过程失败，请求将被拒绝；对于多个连续请求，只要安全窗口没有关闭，请求过程就不断重复。

(3)注销阶段：客户端请求关闭安全窗口，存储端关闭安全窗口，并发送应答。

会话期间，客户端可能发送多个请求给存储端，将握手开销分摊到多个请求上；三方安全协议采用对称密钥，以获取快捷的验证过程和较低的安全开销。

4　三方安全协议的形式化分析

4.1　形式化分析方法

形式化方法是指用数据方法描述和推理基于计算机的系统，并用于理解密文协议的强度和局限性[8]。过去几年，大量的分析技术被开发出来，如 GNY 逻辑[9]、串空间[10]等。其中 GNY 逻辑能处理非加密消息，并支持密钥哈希函数，因此本文采用 GNY 逻辑对三方安全协议进行分析。

GNY 分析产生一系列逻辑步骤，允许主体(Principal)在协议的结论阶段理性地持有一组起始信念(Belief)，使得分析过程可以进行。在本文中，主体分别是存储端、客户端和 MDS。首先是将协议理想化成形式化的描述；然后当一个主体收到一条消息，该主体根据逻辑假设可推导出另一套信念。在协议完成时，该主体应该能达到一个事先预期的目标信念。无法达到意味着协议存在错误或者还需要额外的假设。

GNY 逻辑包括 6 条接收法则(T1～T6)、8 条拥有法则(P1～P8)、11 条新鲜度法则(F1～F11)、6 条识别法则(R1～R6)、7 条消息解释法则(I1～I7)、3 条管辖法则(J1～J3)和 3 条“不在这里产生”法则。GNY 通常将使用私钥 *S* 的哈希函数表示为 $H(\langle S\rangle, X)$。为了与三方安全协议表达一致，本文用 $MAC_S(X)$ 来替代该哈希函数。

4.2　符号定义

协议的分析不涉及协议中加密算法、哈希算法的机密性，主要是通过形式化分析来证明协议传输过程的正确性，以及消息的真实性。

GNY 只处理已命名主体。从存储端的角度看，三方传输模式中的客户端是匿名的，存储端只知道请求来自拥有特定权能标识的某个用户，并不知道其真正身份。本文将持有特定权能标识的客户端表示为 $Clt_C(Ri)$，通过该符号，存储端能推断出某请求来自某特定权能标识的客户端。表 1 所示为下文分析所使用的符号，以及符号的含义。

表 1　三方安全协议中的符号及其含义

符号	符号的含义
Ri	请求标志符(Request Identifier)
Ts	时间戳(Timestamp)，用于检查请求的新鲜度，免遭重放和延迟攻击
Chid	通道标识符，用于保证通道的真实性
Mds	元数据服务器(Metadata Server)
Stg	存储设备端(Storage)
Clt	客户端(Client)
$Clt_C(Ri)$	拥有由 *Clt* 所描述权限的客户端
Ke	*Mds* 和存储端共享的加密密钥
Km	*Mds* 和存储端共享的 *MAC* 密钥
Ks	由 *Mds* 发送给客户端的私钥，客户端通过加密的方式发送到存储端
C(*Ri*)	请求标识符函数，该函数能根据权能标识来识别用户
$C_{Km}(Ri)$	指代对 *C*(*Ri*)用 *Km* 进行消息认证码计算，并包含 *C*(*Ri*)

4.3　逻辑 GNY 假设

分析前，做如下设定。

假设 1：$Clt \ni Ks \quad Clt \models Clt \overset{Ks}{\leftrightarrow} MDs$

假设 2：$Stg \ni Ke \quad Stg \models Stg \overset{Ke}{\leftrightarrow} Mds$

假设 3：$Stg \ni Km \quad Stg \models Stg \overset{Km}{\leftrightarrow} Mds$

假设 1 表示客户端和 MDS 共同拥有私钥 *Ks*；假设 2、3 具有相应的含义。

假设 4：$Clt \models \#(Ts) \quad Stg \models \#(Ts)$

假设 4 表示时间戳 *Ts* 是新鲜的，并用于防范消息请求的重放攻击。

假设 5：$Clt \models \phi(C(Ri)) \quad Stg \models \phi(C(Ri))$

假设 5 表示客户端和存储端能够识别出有效权能标识，这是因为客户端专门请求权能标识。存储端需要根据权能标识执行操作，而权能标识有特定的结构。

假设 6：$Clt \models Mds \Rightarrow Clt_{C(Ri)} \overset{Ks}{\leftrightarrow} Stg$

假设 6 表示客户端相信 MDS 能管辖私钥 *Ks*。另外，私钥 *Ks* 是以加密包 $\{Ks\}_{Ke}$ 的方式从客户端传到存储端。从这一角度看，私钥 *Ks* 是共享的。

假设 7：$Clt \models Mds \Rightarrow Mds \models *$

$Stg \models Mds \Rightarrow Mds \models *$

假设 7 表示客户端和存储端相信 MDS 是诚实而且是有能力的。

4.4 三方安全协议的理想化

根据图 6 可以将安全协议描述成：

步骤 1：

$Clt \rightarrow Mds : Clt, Ri$

步骤 2：

$Mds \rightarrow Clt : C(Ri), MAC_{Km}(C(Ri)), \{Ks\}_{Ke}, Ts$

根据图 8，可将协议描述为：

步骤 3：

$Stg \rightarrow Clt : Chid$

步骤 4：

$Clt \rightarrow Stg : C_{Km}(Ri), \{Ks\}_{Ke}, Request, Ts, MAC_{Ks}(Request, Ts, Chid)$

步骤 5：

$Stg \rightarrow Clt : Reply, P(Ts), MAC_{Ks}(Reply, P(Ts))$

将上述协议分别抽象化成 *GNY* 逻辑，各逻辑描述如下：

逻辑 1：$Mds \triangleleft * Clt, * Ri$

逻辑 2：

$Clt \triangleleft * \{C(Ri) \square\!\!> Mds \models Clt_{C(Ri)} \overset{Ks}{\leftrightarrow} Stg, MAC_{Km}(C(Ri)), \{Ks\}_{ke}, Ts\}_{Ks}$

逻辑 3：$Clt \triangleleft * Chid$

逻辑 4：

$Stg \triangleleft * C_{Km}(Ri), * \{Ks\}_{Ke}, * Request, * Ts, * MAC_{Ks}(Request, Ts, Chid)$

逻辑 5：

$Ctl \triangleleft * Reply, * F(Ts), * MAC_{Ks}(Reply, F(Ts))$

4.5 逻辑的形式化推导

4.5.1 逻辑 1 的推导与分析

根据接收法则 T1 和拥有法则 P1，从逻辑 1 可以推导出：

$$Mds \ni (Clt, Ri) \tag{1}$$

式(1)表示 MDS 在返回的应答中包含正确的请求标志符 *Ri*，从而能够生成客户端可识别的权能标识。

4.5.2 逻辑 2 的推导与分析

结合假设 1，根据接收法则 T1、T2、T3，拥有法则 P1，可从逻辑 2 推导出：

$$Clt \ni (C_{Km}(Ri), \{Ks\}_{Ke}) \tag{2}$$

式(2)表示客户端能拥有消息的内容，包括权能标识 $C(Ri)$。

结合假设 1、假设 4，应用拥有法则 P2 和新鲜度法则 F1，可从式(2)推导出：

$$Clt \models \#(C_{Km}(Ri), \{Ks\}_{Ke}, Ts) \tag{3}$$

式(3)表示客户端的消息是新鲜的，即客户端相信消息不是一个重放消息。

结合假设 1、4 和式(2)，应用法则 F1、I1、I7，可从逻辑 2 推导出：

$$Clt \models Mds \mid\sim C_{Km}(Ri), \{Ks\}_{Ke} \tag{4}$$

式(4)表示 MDS 曾经发送过权能标识 $C_{Km}(Ri)$ 和加密后的私钥 $\{Ks\}_{Ke}$。

结合假设 6、7 和式(3)，应用管辖法则 J1、J2，可从逻辑 2 推导出：

$$Clt \models Clt_{C(Ri)} \overset{Ks}{\leftrightarrow} Stg \tag{5}$$

综合式(3)、(5)可知，客户端认为元数据服务器会相信 *Ks* 是一个有效的私钥。客户端拥有私钥 *Ks* 后，将充当 $Clt_{C(Ri)}$ 的角色。

4.5.3 逻辑 3 的推导与分析

根据接收法则 T1 和拥有法则 P1，可从逻辑 3 推导出：

$$Clt \ni Chid \tag{6}$$

式(6)表示存储端在返回的应答中包含正确的通道标志符 Chid，从而保证客户端能在正确的通道上传输访问请求。

4.5.4 逻辑 4 的推导与分析

根据接收法则 T1、拥有法则 P1，可从逻辑 4 得到：

$$Stg \ni C_{Km}(Ri), \{Ks\}_{Ke}, Request, Ts, MAC_{Ks}(Request, Ts, Chid) \tag{7}$$

根据接收法则 T1、T3，结合假设 2，可从逻辑 4 推导出：

$$Stg \ni Ks \tag{8}$$

结合式(7)、(8)，并根据拥有法则 P5，可推导出：

$$Stg \ni Chid \tag{9}$$

从而保证请求消息在 *Chid* 通道上传输，防止请求遭非法截获。

根据式(8)和 $C_{Km}(Ri)$ 的定义，结合假设 3、假设 5，应用识别法则 R5，可推导出：

$$Stg \models \phi(MAC_{Km}(C(Ri))) \tag{10}$$

根据式(10)和 $Stg \ni MAC_{Km}(C(Ri))$ 便可判断权能标识 $C(Ri)$ 的完整性，防止遭到篡改。

结合假设 4 和式(7)，应有消息解释 I3、I7，可推导出：

$$Stg \models Clt_{C(Ri)} \mid\sim Request \tag{11}$$

根据假设 4，应用新鲜度法则 F1，可推导出：

$$Stg \models \#(Request, Ts) \tag{12}$$

式(11)、(12)可知客户端发出了请求 *Request*，并且 *Request* 不是重放的请求。

因此，存储端可以得出如下结论：请求的传输通道为先前随机挑选的通道(通道标志符为 *Chid*)，确保网络传输的安全性；请求中的权能标识 $C(Ri)$ 通过消息认证码 $MAC_{Km}(C(Ri))$ 保证了其完整性；请求 *Request* 来自客户端 $Clt_{C(Ri)}$，而且不是重放的请求，因此存储端应该执行该请求。

4.5.5 逻辑 5 的推导与分析

根据新鲜度法则 F1，假设 4 可表达成：

$$Clt \models \#(F(Ts)) \tag{13}$$

根据接收法则 T1 和拥有法则 P1，从逻辑 5 推导出：

$$Clt \ni (Reply, F(Ts), MAC_{Ks}(Reply, F(Ts))) \tag{14}$$

根据式(13)，应用新鲜度法则 F1，可推导出：

$$Clt \models \#(Reply, F(Ts)) \tag{15}$$

式(14)、(15)说明客户端收到了消息 *Reply*，并且 *Reply* 不是重放的消息。

结合式(14)、假设 1，应用拥有法则 P2，从逻辑 5 推导出：

$$Clt \ni (Reply, Ks) \tag{16}$$

根据式(5)、(16)，并应用解释法则 I3、I7，可推导出：

$$Clt \models Stg \mid\sim Reply \tag{17}$$

从式(15)、(17)可知，消息 *Reply* 来自存储端，并且不是重放的消息。

结论与展望 已有的存储安全方案受技术和出发点的限制，无法同时保证大规模存储系统的可扩展性和安全性要求。本文在分析三方传输模式的基础上，将三方传输模式引入大规模存储系统，提出了基于权能标识的三方存储安全系统框

(下转第 68 页)

率均有 20%～30%的提高。

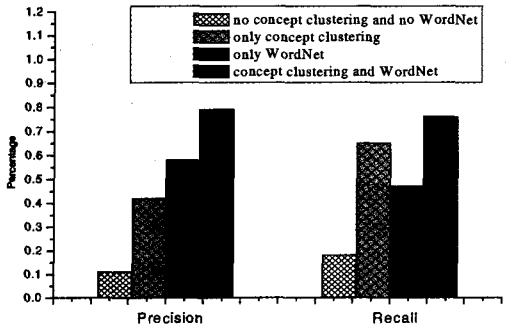

图 6　概念簇和 WordNet 对接口匹配性能的影响

结论　通过把服务抽象成为具有输入和输出接口的实体，本文提出了基于接口匹配的 Web 服务组合方法。它能够实现在现有服务描述标准基础上自动、动态发现潜在的 Web 服务交互。WSDL 是 Web 服务接口描述的标准语言，得到工业界的广泛支持。因此，我们利用 WSDL 文档已经提供的信息，进行服务接口匹配。接口匹配算法基于消息参数聚类形成的概念簇和现有的 WordNet 语义词典。实验结果证明二者的结合互为补充，从而使得匹配算法具有更优的查准率和查全率。在此基础上提出了 Web 服务组合的算法。

WSDL 对接口的描述只限于语法层次，不能表达语义信息。本文的接口匹配算法借助于参数聚类和 WordNet 语义词典，在一定程度上克服了这一缺陷。如果各个服务的描述都能够遵循统一的语义约定，在某个应用环境中共同遵守的语义词汇集合范围内描述 Web 服务，服务之间将能相互识别，实现自动的组合和协作。因此，在今后的工作中，我们将进一步研究基于语义的 Web 服务组合。

参 考 文 献

1　Benatallah B, Dumas M, Fauvet M, et al. Towards Patterns of Web Services Composition [M]. In: Patterns and Skeletons for Parallel and Distributed Computing, Springer Verlag, UK, 2003. 265～296

2　Rao J H, Su X M. A Survey of Automated Web Service Composition Methods [A]. In: Proceedings of the First International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004), San Diego, USA, July 2004. 43～54

3　Srivastava B, Koehler J. Web Service Composition - Current Solutions and Open Problems [A]. In: ICAPS 2003 Workshop on Planning for Web Services, Trento, Italy, June 2003. 51～59

4　Milanovic N, Malek M. Current Solutions for Web Service Composition [J]. IEEE Internet Computing, 2004, 8(6): 51～59

5　Casati F, Shan M C. Definition, Execution, Analysis, and Optimization of Composite E-Services [J]. IEEE Data Engineering Bulletin, 2001, 24(1): 29～34

6　Sheng Q Z, Benatallah B, Dumas M, et al. SELF-SERV: A Platform for Rapid Composition of Web Services in a Peer-to-Peer Environment[A]. In: Proceedings of 28th Very Large Data Bases, Hong Kong, China, August 2002. 1051～1054

7　The OWL-S Service Coallition. OWL-S: Semantic Markup for Web Services, version 0. 1. http://www. daml. org/services/owl-s/1. 0/owl-s. pdf

8　McIlraith S, Son T, Zeng H. Semantic Web Services [J]. IEEE Intelligent Systems, 2001, 16(2): 46～53

9　Christensen E, Curbera F, Meredith G, et al. Web Services Description Language (WSDL) 1. 1. http://www. w3. org/TR/wsdl, March 2001

10　Casati F, Shan M C. Models and Languages for Describing and Discovering E-Services (Tutorial) [A]. In: Proceedings of the International ACM SIGMOD Conference on Management of Data, Santa Barbara, California, USA, May 2001

11　Hand D, Mannila H, Smyth P. Principles of Data Mining[M]. Cambridge, MA, USA: The MIT Press, 2001

12　Kaufman L, Rousseeuw P J. Finding Groups in Data: An Introduction to Cluster Analysis [M]. New York: John Wiley & Sons, 1990

13　Zaremski M, Wing J M. Signature Matching: a Tool for Using Software Libraries[J]. ACM Transactions on Software Engineering and Methodology, 1995, 4(2): 146～170

14　Zaremski M, Wing J M. Specification Matching of Software Components[J]. ACM Transactions on Software Engineering and Methodology, 1997, 6(4): 333～369

15　Luckham D C, Vera J, Meldal S. Three Concepts of System Architecture[R]: [Technical Report. CSL-TR-95-674]. Stanford University, 1995

16　Miller G. WordNet: An On-line Lexical Database[J]. International Journal of Lexicography, 1990, 3(4): 235～312

17　Voorhees E M. Using WordNet for Text Retrieval. In: Fellbaum C, ed. WordNet: An Electronic and Lexical Database[M]. Cambridge, MA, USA: The MIT Press, 1998. 285～303

18　Yu S J, Le J J. Study and Development of Textile Enterprise Management System Based on Web. In: Proceeding of 8th Joint International Computer Conference (JICC), Ningbo, China, November 2002. 121～125

(上接第 53 页)

架和相应的三方安全协议，除了满足大规模存储系统的可扩展性要求，还保证该存储系统的系统级安全性。

借助形式化分析方法，对三方安全协议进行了逻辑推导，从推导结果可知：在一定的假设下，三方安全协议能保证请求中的权能标识的完整性、协议传输过程的正确性，以及消息的真实性，从逻辑上验证了三方安全协议的正确性和可行性。

网络技术和存储技术不断发展，网络存储面临新的安全挑战，比如新的攻击方式。为此，我们将在进一步工作中修改权能标识结构，并用形式化分析方法指导三方安全协议的详细设计。

参 考 文 献

1　Singh A, Voruganti K, Gopisetty S, et al. Security vs Performance: Tradeoffs Using a Trust Framework. In: Proceedings of the 22nd IEEE/13th NASA Goddard Conference on MSST, 2005

2　Neumann C, Ts'o T. Kerberos: An Authentication Service for Computer Networks. IEEE Communications Magazine, 1994, 32 (9): 33～38

3　Blaze M. A Cryptographic File System for UNIX. In: Proc. of 1st ACM Conference on Communications and Computing Security. USA: ACM Press, 1993. 9～16

4　Gibson G A, Nagle D F, Amiri K, et al. File Server Scaling with Network Attached Secure Disks. In: Proc. of the ACM ICMMCS. USA: ACM Press, 1997. 272～284

5　Kubiatowicz J, Bindel D, Chen Y, et al. Oceanstore: an Architecture for Global-Scale Persistent Storage. In: ASPLOS-1x. USA: ACM Press, 1999. 190～201

6　Butler M L. 1 Petabyte Production Storage Environments and File Systems. In: 2004 International Conference on Supercomputing. USA: ACM Press, 2004

7　Dennis J B, Van Horn E C. Programming Semantics for Multiprogrammed Computations. Communications of the ACM, Feb. 1966

8　卿斯汉. 安全协议 20 年研究进展. 软件学报, 2003, 14(10): 1740～1752

9　Gong L, Needham R, Yahalom R. Reasoning about belief in cryptographic protocols. In: Proc. of the 1990 IEEE Computer Society Symposium on Research in Security and Privacy. USA: IEEE Press, 1990. 234～248

10　Fabrega F J T, Hertzog J, Guttman J. Strand spaces: Proving security protocols correct. Journal of Computer Security, 1999, 7 (2-3): 191～230