ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.9, September 2006, pp.1933−1948
DOI: 10.1360/jos171933

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

自动信任协商及其发展趋势*

廖振松+, 金 海, 李赤松, 邹德清

(华中科技大学 计算机科学与技术学院 集群与网格计算实验室,湖北 武汉 430074)

Automated Trust Negotiation and Its Development Trend

LIAO Zhen-Song+, JIN Hai, LI Chi-Song, ZOU De-Qing

(Cluster and Grid Computing Laboratory, College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

+ Corresponding author: Phn: +86-27-62250217, Fax: +86-27-87557354, E-mail:zsliao@hust.edu.cn, http://www.hust.edu.cn

Liao ZS, Jin H, Li CS, Zou DQ. Automated trust negotiation and its development trend. *Journal of Software*, 2006,17(9):1933−1948. http://www.jos.org.cn/1000-9825/17/1933.htm

Abstract: Exchange of attribute credentials is an important means to establish mutual trust between strangers who wish to share resources or conduct business transactions. Automated trust negotiation (ATN) is an approach to regulate the exchange of sensitive credentials by using access control policies, so as to protect sensitive credentials, policies and private privacy, and to improve negotiation efficiency and successful negotiation establishment rate. A detailed review of the research on ATN is presented based on the survey and classification of the current negotiation techniques. According to the precise investigation on ATN, the pitfalls of ATN are pointed out and the reasonable rules that ATN should observe are put forward, meanwhile, the development trend for ATN is proposed.
Key words: ATN (automated trust negotiation); authentication; credential; authorization; access control policy

摘 要: 属性证书交换是一种在不同安全域里共享资源或进行商业事务交易的陌生人之间建立双方信任关系的重要手段.自动信任协商则通过使用访问控制策略提供了一种方法来规范敏感证书的交换,从而保护了用户的敏感证书信息、敏感访问控制策略与个人隐私,以及提高协商效率和协商成功率.对自动信任协商的研究动态进行了调研,对该领域的相关技术进行了归类及介绍.在认真分析现有技术的基础上,总结了当前自动信任协商的不足,并指出了自动信任协商继续发展应遵守的原则以及自动信任协商的未来发展趋势.
关键词: 自动信任协商;认证;证书;授权;访问控制策略
中图法分类号: TP301 文献标识码: A

自动信任协商(automated trust negotiation,简称 ATN)是指通过逐步向对方暴露数字证书,以在陌生者之间建立信任关系的一种访问控制方法[1−4].当访问者与资源/服务提供方不在同一个安全域时,常规的访问控制方法(如 MAC[5],RBAC[6]等)不能有效地对访问者的行为进行控制,自动信任协商则可以为合法用户访问资源提供安全保障,防止非法用户的非授权访问.

* Supported by the National Natural Science Foundation Important Special Project of China under Grant No.90412010 (国家自然科学基金重大项目); the National Natural Science Foundation of China under Grant No.60503040 (国家自然科学基金)
Received 2005-12-21; Accepted 2006-04-12

自动信任协商是在传统的信任管理基础上发展起来的,它与信任管理的主要区别在于是否事先知道对方的访问控制策略.传统的信任管理,双方在协商之前,访问者并不清楚对方的访问控制策略,用户通过提交访问请求来获得访问控制策略信息;而在自动信任协商里,访问控制策略是对外公布的,用户在访问前就可以事先准备好证书,协商过程一般不需要或仅需少量的人工参与.

与基于身份的访问控制系统相比,自动信任协商的优势体现在[7]:1) 陌生者之间的信任关系是通过参与者的属性信息交换进行确立,这是通过数字证书的暴露来实现的;2) 协商双方都可定义访问控制策略,以规范对方对其敏感资源的访问;3) 协商过程中,并不需要可信第三方(如 CA)的参与.

本文对自动信任协商领域的相关技术和项目进行了调研,分析了当前自动信任协商中所使用的主流技术以及自动信任协商的研究动态.在分析自动信任协商发展现状的基础上,总结了自动信任协商发展的不足以及自动信任协商继续发展所应遵循的原则,并指出了自动信任协商的未来发展趋势.

1 自动信任协商中的基本概念

自动信任协商提供了一种跨安全域访问资源的途径,许多专业术语与单域的安全概念类似.本节仅对自动信任协商中的一些基本概念进行介绍.

1. 数字证书

数字证书(digital credential)[8]是用来携带用户身份/属性等相关特征的数字化工具.由于证书代表着用户的身份,因此,证书必须具有可证实性和不可伪造性.按照在不同系统中的用途,证书可分为身份证书和属性证书.身份证书主要应用于对安全级别要求高的系统,如军事系统、邮件系统等,其主要代表为基于 PKI 身份认证系统的 X.509[8];属性证书则主要应用于方便管理和易于操作的系统,如图书资源共享管理系统、投票系统等,其典型代表为 SPKI/SDSI[9].

2. 认证

认证(authentication)[8,10]用来确认参与方身份的真实性,通过对用户身份进行一致性检查,防止冒名顶替现象的发生.在信任协商开始前,确定通信双方的身份是否合法,是系统安全得以维持的保障,也是检查用户授权、证书交换以及系统审计的前提.认证的方法主要是检查用户所提交的“用户名-密码”是否属实.对于安全级别高的系统,则还需从用户提交的证书中提取身份信息来验证用户身份.在特殊应用中,需要检查一些额外的信息,如与生物特征相关的指纹信息等.

3. 授权

授权(authorization)[11]是指分析用户提交的证书,根据证书上的属性值,为用户分配访问资源的权限.用户对资源具有什么样的操作权限,或者能够享受到什么样的服务,都体现在系统对用户的授权上.在基于身份认证的信任管理系统中,对用户的授权主要是激活用户对资源的相应控制操作.如在 MAC[5]系统中,用户身份直接对应着权限,系统对用户的授权则直接为其分配操作权限.在基于属性认证的信任管理系统中,对用户的授权则是激活用户在系统中的角色.如在 RBAC[6]系统中,系统对用户的授权,则表现在将其关联到一定的角色.

4. 委托

委托(delegation)[12−14]是一种重要的安全策略,主要思想是:系统中的主动实体将权限委托给其他主动实体,以便以前者的名义执行一些工作.在如下几种情况下会发生委托[14]:

(1) 角色备份:某人出差或度假,其负责的工作需要继续执行,这样就需要将其工作权力委托给其他人,使工作可以继续进行.

(2) 协同工作:在组织中和组织之间进行的工作需要相互协作.在这种情况下,需要赋予协作对方一定的访问权限以便进行信息共享.

(3) 权力下放:组织初始构建或重组织时,需要按照组织结构从高级到低级分配权限.

在委托中有一个隐含的原则,就是委托人要对被委托人的行为负责.这就意味着委托人需要对其委托的角色进行严格限制.所以,委托的限制是委托策略的重要组成部分.从情形(1)、情形(2)中可以看到,委托具有临时

性,即角色的委托只是在某个时间段内有效.在这个时间段内,使用角色的次数可能是有限的,并且可能仅在该时间段的一些周期性时间片内可用.在情形(2)的协作工作中,不一定需要委托角色的所有权限,即委托的部分性.而在所有这些情形中,都可能出现委托角色权限传播的问题.为便于管理,必须限制权限的传播.在各种情形中,委托角色都可能具有常规角色关联性.如在角色备份的情况下,委托角色只有在委托角色的用户没有激活角色时方可激活角色;而在协同工作的情况下,委托角色可能仅在特定用户激活特定角色时方可激活.

5. 访问控制策略

访问控制策略(access control policy)[2,15]是用来保护资源不被合法用户非授权访问,从而规范合法用户对资源的操作.访问控制策略决定了在自动信任协商中暴露哪些证书以及这些证书暴露的先后顺序.

根据描述的复杂程度,访问控制策略可分为简单策略(元策略)与复合策略.简单策略是组成复合策略的基本元素,它们的关系类似于元数据与数据的关系.一般地,系统中提供一些操作,如“AND/OR/MofN()”或“∧/∨/!”等,来实现复合策略的组成与分解.

6. 信任协商模型

信任协商模型(trust negotiation model)[1,2,16]是协商双方在建立信任关系中所采取的暴露证书和访问控制策略的方式.信任协商模型的选择,决定了协商双方将采用什么样的方式来释放证书和访问控制策略信息,对敏感信息、个人隐私保护具有极大的影响.

Winsborough 等人[2]提出了两种极端信任协商模型:热心模型(eager strategy)与吝啬模型(parsimonious strategy).热心模型采用的是 PUSH 方式,即在信任协商前,访问者一次性地将所拥有证书全部提交.其优点是减少证书交换次数,降低网络开销,提高协商效率.但由于用户将所有的证书(包括协商相关的与不相关的)都提交给了对方,这样无疑会导致无关证书的暴露.同时,不加条件地释放证书,会泄露一些个人隐私或商业信息;吝啬模型采用的是 PULL 方式,即资源方要求什么证书,访问者才提交相关的证书,直到建立起信任关系.吝啬模型可以克服热心模型中信息保护的不足,但信任关系的建立需要多次的证书交换,网络开销大,协商效率低.

2 策略语言

策略语言是描述和实现自动信任协商的重要工具.一方面,自动信任协商是在分布式环境下工作的,具有跨平台、操作系统异构等特点,许多常用语言并不满足这些需求;另一方面,自动信任协商对策略语言和系统都有相应的要求,并不是所有的策略语言都可用来为自动信任协商服务.

2.1 自动信任协商对策略语言的需求

自动信任协商对策略语言的需求[16−18]主要体现在:

(1) 定义良好的语义:一个定义良好的策略语言应该具有简单、紧凑和定义规范的语义,即使用该策略语言编写的策略,其含义与该语言的特殊应用无关.

(2) 单调性:自动信任协商对策略语言的单调性需求表现在:证书与策略的暴露,应对用户的授权产生影响;额外证书/策略的暴露,只能让用户获得额外资源/服务操作的权限.

(3) 证书结合:不同证书描述了特定主体不同的特征.自动信任协商要求策略语言具有很强的表达能力,能够使用“交”、“并”等操作将不同的证书结合起来,以满足需要提交多个证书的策略.

(4) 认证:信任协商的参与方均有多个证书,以便通过证书交换来建立信任关系.在系统运行过程中,证书提交者需证明其拥有与证书签名使用的公钥相对应的私钥信息,以确保证书的有效性.

(5) 属性值约束:通常,一个证书就是一个结构化的对象,它包含关于主体属性的信息,name-value 就是属性信息的典型代表.证书可关联到某种指定的证书类型,用来简化证书规范和管理.

(6) 内部证书约束:为了更好地评估远程参与方的属性,即使参与方使用了不同的密钥,策略也应该可以表达一些约束,用来比较属于同一主体的不同证书的值.

(7) 证书链:当某一证书中的主体是证书链中下一证书的发布者时,策略语言应提供足够的描述能力来表达和约束证书链.

(8) 传递闭包:在特定的环境中,信任关系具有传递性.这要求策略语言允许策略编写者来描述信任链中的数量和类型约束.

(9) 外部函数:在协商过程中,需要一个标准的函数库来规范对形如日期、时间、货币等的操作和比较.

(10) 本地证书变量:当处理资源的标准离线策略时,本地证书变量可使这些策略自动地与其证书关联起来,提高策略与证书的匹配效率.

(11) 检测提交者:策略编写者可以指定策略中哪些原子策略应该由访问者提交的哪些证书来满足.

(12) 敏感策略保护:敏感策略里可能包含一些个人隐私或商业机密.策略语言具有敏感信息保护机制,以避免/防止重要信息外泄.

(13) 具有互操作语言的统一形式和使用:这种需求强调了协商方法的应用能力,即在设计策略语言时,须充分考虑其是否可以在真实的环境中使用,以及是否可以集成到已有的上下文中.

目前,已有多种策略语言用于实现跨域的资源共享,如 PSPL[16],TPL[16,19],X-Sec[16,20,21],RT[22−24],KeyNote[16],Trust-X[17,25],TrustBuidler[17]等.这里仅对有影响的 RT 语言进行介绍,表 1 是这些语言对自动信任协商需求的满足情况的对比(表中的 Y 表示支持/满足,P 表示部分支持,N 表示不支持).

Table 1 Comparison of policy languages for ATN

表 1 策略语言对自动信任协商需求的满足情况对比

Requirements	Typical policy languages						
	PSPL	TPL	X-Sec	RT	KeyNote	Trust-X	TrustBuilder
Well-Defined semantics	Y	Y	Y	Y	Y	Y	Y
Monotonicity	Y	P	N	Y	Y	Y	Y
Credential combinations	Y	Y	Y	Y	Y	Y	Y
Authentication	Y	N	N	Y	P	N	N
Constraints on attribute values	Y	Y	Y	Y	N	Y	Y
Inter-Credential constraints	Y	Y	Y	Y	N	Y	Y
Credential chains	Y	Y	N	Y	Y	P	N
Transitive closure	P	Y	N	Y	P	Y	Y
External functions	Y	Y	Y	Y	N	Y	Y
Local credential variables	Y	N	N	Y	N	Y	Y
Who submits the credentials	N	N	N	P	N	Y	Y
Protecting sensitive policies	Y	P	Y	Y	N	Y	P
Unified formalism and use of interoprable languages	N	Y	Y	N	Y	Y	N

RT 是基于角色信任管理(role-based trust-management)的简写.RT 定义了两类实体:主体(principal)与角色(role).主体是指由个体、程序公钥等唯一证明的实体.在信任管理系统里,RT 语言[23,24]是一类语言的集合,包括:RT_0,RT_1,RT^T 与 RT^D.RT_0 是 RT 的基础,它主要用来定义角色.一个角色定义由两部分组成:开头(HEAD)与主体(BODY),用谓词连接起来.根据处理类型来划分,可分为如下 4 种:

类型 1: $A.r \leftarrow B$,定义主体 B 是角色 $A.r$ 的成员,即 $B \in members(A.r)$.

类型 2: $A.r \leftarrow B.r_1$,定义角色 $B.r_1$ 的成员是角色 $A.r$ 的成员,即 $members(B.r_1) \subseteq members(A.r)$.

类型 3: $A.r \leftarrow A.r_1.r_2$,定义角色 $A.r$ 包含所有的角色 $B.r_2$,即 $members(A.r_1.r_2)=\cup_{B \in memberS(A.r_1)} members(B.r_2) \subseteq members(A.r)$;其中 B 是角色 $A.r_1$ 的成员;$A.r_1.r_2$ 是一个链接角色(linked role).

类型 4: $A.r \leftarrow B_1.r_1 \cap \ldots \cap B_k.r_k$,定义角色 $A.r$ 的成员包含了角色集$\{B_i.r_i\}$交集的所有成员,即$(members(B_1.r_1) \cap \ldots \cap members(B_k.r_k)) \subseteq members(A.r)$.

此外,RT_0 还支持简单的委托,如 $A.r \Leftarrow B:C.r_2$,表示 A 将其对角色 r 的权限委托给 B,B 又将该权限委托给 $C.r_2$,即等价于 $A.r \leftarrow B.r \cap C.r_2$.$RT_1$ 在 RT_0 的基础上增加了携带参数的功能,即允许增加参数来约束角色.RT^T 语言通过提供两种角色操作(角色集成⊙与角色互斥⊕)支持职责分离(separation of duty,简称 SoD)[5,6].RT^D 语言主要用于处理角色激活的委托,以处理角色之间的关系.

2.2 自动信任协商对系统的需求

出于对多域身份联合互操作的需求,自动信任协商对系统的要求主要体现在信任协商系统必须提供相关协议,保证信任协商的顺利进行,这就要求协议必须支持[17]:

(1) 证书所有权:当接收到远程证书时,系统要求发送者提供与证书公钥相对应的私钥所有权证明.

(2) 证书有效性:在系统收到证书时,通过数字签名来检查证书内容的完整性以及证书是否过期与被吊销.

(3) 证书链发现:在协商过程中,有些证书并没有保存在本地,系统应该提供额外的机制和工具来实时发现与查找证书链.

(4) 隐私保护机制:保护资源与策略的暴露是保护用户隐私的前提,包括对敏感信息、敏感策略的保护.

(5) 支持多种协商策略:提供多种协商策略,用户有更大的空间来选择协商机制,也可以支持多种应用.

(6) 快速协商策略:系统在提供可供选择的协商机制的同时,应该根据不同的应用对协商机制的性能区别加以对待.这表现在:1) 系统应为广泛使用的资源(如 VISA 卡等)提供标准的非定制的策略;2) 协商中涉及到一些公共资源,相同序列的证书将会多次用到.此时,系统应提供有效的缓存机制,避免相同的协商多次进行;3) 为了加快协商,系统应接受一些预先计算的值或先前已经确认了的协商序列.

当前,支持自动信任协商的系统很多,如 KeyNote[16],TrustBuilder[17],PolicyMaker[26],SD3[27],PRUNES[28],Trust-X[17,25]等.这里仅对应用广泛的 KeyNote 进行介绍,表 2 是这些语言在满足自动信任协商需求上的对比(表中的 Y 表示支持/满足,P 表示部分支持,N 表示不支持).

Table 2 Comparison of trust-management systems for ATN

表 2 信任管理系统对自动信任协商需求的满足情况对比

Requirements	Typical trust management systems					
	KeyNote	TrustBuilder	PolicyMaker	SD3	PRUNES	Trust-X
Credential ownership	N	N	P	P	Y	P
Credential validity	N	Y	Y	N	Y	Y
Credential chain discovery	N	N	N	Y	P	P
Privacy protection mechanism	Y	Y	N	N	N	Y
Support for alternative negotiaton strategies	N	Y	Y	P	N	Y
Fast negotiation strategies	N	N	N	N	N	Y

KeyNote 是信任管理系统中一个十分成功的系统.KeyNote 的基础设计与 PolicyMaker 类似,但 KeyNote 的简单性特征可更直接地支持类 PKI 的应用.KeyNote 与 PolicyMaker 的主要区别在于:1) KeyNote 谓词是在类 C 描述和规范描述的基础上用一种简单的标记编写的;KeyNote 为策略和证书提供了一种统一的语言;2) 在 KeyNote 里,证书统称为断言(assertion),其返回值全是布尔量,易于计算机处理;3) 断言语法是基于通用的 RFC-822 规范;4) 可信活动由简单的“属性-值”描述.

KeyNote 的简单性特征体现在:1) 重点突出——KeyNote 旨在使用应用具体的证书与策略,为用户提供一种通用的、独立于应用的机制;2) 易于描述的系统——整个 KeyNote 规范只有 15 页,而且包括演示实例;3) 宣言易于理解与创建;4) 易于实现——系统的代码量小,机器易读.

2.3 顺从检测器

顺从检测器(compliance checker)[16,26]是检测数字证书是否匹配访问控制策略的模块.系统根据顺从检测器的匹配返回值来建立信任级别以及确定信任协商是否成功.在顺从检测器对证书和策略进行匹配之前,顺从检测器须对证书与策略的语义进行检查,过滤掉无效的数字证书和有语义冲突的策略,从而提高协商效率.使用不同语言开发的自动信任协商系统,其顺从检测器的设计都有所差别,主要体现在匹配算法以及返回值上.按照操作请求模式的不同,顺从检测器可分为两类:传统顺从检测器与智能顺从检测器.

传统顺从检测器即指顺从检测器按照传统的信任管理环境模式进行工作.假定 Alice 是资源提供方,Bob 是欲访问资源的一般用户.在传统模式中,Alice 的协商管理器唤起顺从检测器,以决定是否允许 Bob 对敏感资源的访问.其中,Alice 的协商管理器向顺从检测器提供资源的本地访问控制策略、Bob 所释放的数字证书以及 Alice 可以公开的一些本地信息(如本地时间等).顺从检测器根据输入的信息进行比较,产生一系列的比较结果(布尔值 True 与 False),用以显示证书是否满足策略.

智能顺从检测器在传统的信任管理环境中并不是必需的,只是在策略暴露时才需要.假定 Alice 必须拒绝

Bob 对某项资源的访问,原因在于 Bob 没有提交充足的证书.智能顺从检测器并不是简单地拒绝 Bob 的资源访问请求,而是暴露相关的访问控制策略以引导协商成功.Bob 在接收到 Alice 返回的策略信息后,其协商管理器将唤起顺从检测器来检测是否 Bob 在本地已缓存了满足策略的证书,并获得满足策略所需求的最小证书集.这样,Bob 可以使用最少的证书来满足 Alice 的访问控制策略,从而避免了无关证书的暴露,提高了协商效率.

3 分布式证书链的发现

自动信任协商的过程,是通过不断交换证书进行认证的过程,当协商者拥有证书时,用户直接提交证书即可.而在分布式环境里,更多时候证书是非集中存储的,则信任协商过程中需要根据证书链进行证书查找与检索.下面通过一个实例来说明证书链发现的工作模式.

例 1:PUB 是一个出版社,可向其上级组织 EDU 的主要顾客提供优惠出版服务.EDU 认为 HUST 学校的学生是 EDU 的主要顾客.HUST 将发放学生证的权限委托给学位办公室(XWB),且 XWB 为学生 Tom 发放了学生证.我们用 RT_0 语言来描述这 4 个证书:

(1) *PUB.discount←EDU.mainCustomer*; (2) *EDU.mainCustomer←HUST.student*;

(3) *HUST.student←XWB.student*; (4) *XWB.student←Tom*.

逻辑上,证书(1)由 PUB 保存,证书(4)由 Tom 保存,而证书(2)、证书(3)则由非协商方保存.Tom 需要打印论文,为了享受优惠,他提交了证书(4).PUB 查看证书,发现 Tom 的证书是由 XWB 发布的,于是发送查询请求到 XWB 以核实证书(4)的有效性;XWB 又将该请求转发到 HUST 等.于是,证书通过不断地委托授权便构成了一个证书链.

Li 等人基于 RT_0 语言提出了一些意图明确的证书链发现算法.这些算法通过向协商方进行访问请求,查询与该请求相关的证书,不考虑可能存在的长证书链;同时,这些算法对分布存储的证书也是有效的[1,24].本节在 RT_0 的基础上,对证书的存储方式、证书链的发现等影响协商效率与成功率的问题进行探讨.

3.1 证书的分布式存储

分布式环境下证书的存储方式对证书链发现影响很大.关于证书的存储,要做到存储证书,而且保证对证书高效的访问.协商过程中,为达到认证的目的,需要查询证书链中的各种证书,对证书的有效性进行核实;而在查找证书时,需明确证书的存储位置,这样才能做到有的放矢,提高协商效率.

在证书中,每个角色 r 均具有两种存储类型:发布方存储(issuer-side)与接收方存储(subject-side),每种类型均有不同的可选值,表 3 对这些情况进行了汇总.由表 3 可知:角色存储的两种类型,分别有 3 种与两种可选状态,这样就有 6 种组合.不同的组合决定了该类角色存储类型是否良好,以及是否容易被发现与检索.表 4 是角色 r 在各种组合下的类型状态.由表 4 可知:角色的存储必须相互约束,有利于证书链的发现与证书的认证.

Table 3 Credential's storage in distributed environment

表 3 分布式环境下证书的存储

Role's storing type	Possible values	Illustration (Towards a credential $A.r \leftarrow B.r_1$)
Issuer-Side	Issuer-Traces-None	Issuer *A* does not store the credential
	Issuer-Traces-Def	Issuer *A* must store the credential, but does not ensure that negotiator can discover the members of *A.r*, but it requires *B* be issuer-side
	Issuer-Traces-All	Issuer *A* and subject *B* both store the credential, and subject *B* must have the value of issuer-traces-all
Subject-Side	Subject-Traces-None	Subject *B* does not store the credential
	Subject-Traces-All	Issuer *A* and subject *B* both store the credential

Table 4 Storage combination of role r

表 **4** 角色 r 存储方式的组合

Issuer-Side	Subject-Side	Types of role r
Issuer-Traces-All	Subject-Traces-All	Weakly well-typed
Issuer-Traces-All	Subject-Traces-None	Strongly well-typed
Issuer-Traces-Def	Subject-Traces-All	Strongly well-typed
Issuer-Traces-Def	Subject-Traces-None	Weakly well-typed
Issuer-Traces-None	Subject-Traces-All	Strongly well-typed
Issuer-Traces-None	Subject-Traces-None	Ill-Typed

单个角色具有不同的存储状态,角色表达式(角色的交、并以及角色连接等)则继承了这些存储状态,并根据单个角色的状态进行定义.一个具有良好类型(well-typed)的角色表达式满足:

(1) 实体 A 同时具有 issuer-traces-all 与 subject-traces-all 的存储形式;

(2) 角色 $A.r$ 具有与 r 相同的存储类型;

(3) 连接角色 $A.r_1.r_2$:当 r_1 与 r_2 同时 issuer-traces-all 时,$A.r_1.r_2$ 为 issuer-traces-all;当 r_1 与 r_2 同时 subject-traces-all 时,$A.r_1.r_2$ 为 subject-traces-all;当 r_1 为 issuer-traces-all 而 r_2 为良好类型,或者 r_1 为良好类型而 r_2 为 subject-traces-all 时,$A.r_1.r_2$ 为弱良好类型;其他情况时,$A.r_1.r_2$ 为问题类型(ill-typed);

(4) 角色表达式的组合 $f_1\cap\ldots\cap f_k$:当存在一个 f_i 为 issuer-traces-all,而其他均为良好类型时,$f_1\cap\ldots\cap f_k$ 为 issuer-traces-all;当存在一个 f_i 为 subject-traces-all,而其他均为良好类型时,$f_1\cap\ldots\cap f_k$ 为 subject-traces-all;当所有均为弱良好类型时,$f_1\cap\ldots\cap f_k$ 为弱良好类型;其他情况下为问题类型.

3.2 证书链的发现

3.2.1 集中式证书链发现

在证书集中存储的假设下,进行证书链发现主要解决如下 3 类查询问题[24]:

(1) 任意给定一个角色表达式 e 和一个实体 D,检查 $D\in expr[S_C](e)$是否成立;

(2) 任意给定一个角色表达式 e,确定其成员集合 $expr[S_C](e)$;

(3) 任意给定一个实体 D,明确所有包含实体 D 的角色.

其中:$[S_C](e)$表示角色表达式 e 所包含的实体(S_C 是将角色映射到实体的函数);$expr[S_C](e)$是指角色表达式 e 所包含的各类成员的集合.

处理集中式证书链发现的算法很多,如文献[9]使用 SPKI/SDSI 提出自底向上搜索方法;文献[24]提出的后向查询算法、前向查询算法以及双向查找算法[24]等.这里,仅对具有代表意义的后向查询算法进行介绍.

后向查询算法通过构建一个证据图(图中的节点代表实体、角色或角色表达式,边代表证书),维持着一个等待处理的节点队列.初始化时,每个队列仅包含一个节点,通过对所有节点逐个进行处理,直到队列里的节点为空.对于每一个节点 e,算法均存储着与 e 相关的实体集.同时,e 也存储着需要通知的客体,这些客体包括节点 e 可直接到达的实体或角色,以及用来处理连接角色和交叉角色的对象等.算法处理节点的方法为:

(1) 角色节点 $A.r$:查找所有定义了 $A.r$ 的证书,对证书 $A.r\leftarrow e$,算法为 e 创建一个节点,并增加边 $A.r\leftarrow e$;

(2) 实体节点 B:算法通知节点将 B 作为它的一个对象节点;

(3) 连接节点 $A.r_1.r_2$:算法为 $A.r_1$ 创建一个节点,同时创建连接监视器并将其加入到 $A.r_1$ 的监视对象中.通过观察 $A.r_1$ 是否接收到新的对象 $B.r_2$,来决定是否创建节点 $B.r_2$ 以及增加边 $A.r_1.r_2\leftarrow B.r_2$;

(4) 交叉节点 $e=f_1\cap\ldots\cap f_k$:算法为 e 创建一个交叉监视器,为每个 f_i 创建一个节点,共 k 个节点,并将监视器加为每个 f_i 的对象监视器.监视器统计实体 D 被加入的次数,当次数达到 k 时,则增加边 $e\leftarrow D$.

3.2.2 分布式证书链发现

分布式证书链发现算法的前提是:存在某种机制能够代替实体 A 连接到拥有与 A 相关证书的服务器主机上.涉及到 A 的证书是指:证书由 A 发布,或使用了实体 A、角色 $A.r$ 等.在分布式证书链发现算法中,有两个主类:ProofGraph 与 ProofNode,以及 3 个辅助类:BlinkingMonitor,BintersectionMonitor 与 FlinkingMonitor.

ProofGraph 具有 4 类临时变量:

(1) nodes:维护图中所有的节点.可使用哈希图来实现角色表达式到节点的映射;

(2) edges:维护图中所有的边.支持定时存储检查与新边的增加;同时也能检索所有边的状态,包括边离开、节点加入产生新边等;

(3) B-proc-queue:后向处理队列.该状态表示节点等待后向处理;

(4) F-proc-queue:前向处理队列.该状态表示节点等待前向处理.

ProofNode 具有 6 类临时变量:

(1) B-proc-state:后向处理状态.该状态有 3 种可选值:未处理、等待处理与已处理.创建节点时,状态为未处理;当节点进入 B-proc-queue 队列时,状态为等待处理;当节点从图上撤出或处理时,状态为已处理;

(2) F-proc-state:前向处理状态.与 B-proc-state 状态类似,也有 3 种可选的值,与 B-proc-state 定义类似;

(3) B-solutions:后向查找实体的集合;

(4) F-solutions:前向查找实体的集合;

(5) B-sol-monitors:后向查询对象集合,包括后向连接监视器、后向交叉监视器等;

(6) F-sol-monitors:前向查询对象集合,与 B-sol-monitors 定义类似.

算法在具体的实现中,通过将节点与边赋予不同的状态,以达到对所有的实体与证书进行处理的目的,可充分解决证书链发现的 3 类问题(见第 3.2.1 节),从而达到证书链发现与检索的目的.

4 自动信任协商信息的泄露/攻击保护及敏感信息保护

在确立陌生人之间的信任关系期间,敏感证书、访问控制策略的保护对自动信任协商提出了重要的挑战.本节对自动信任协商系统中用来防止信息泄露及敏感信息保护的典型技术进行汇总,这些技术围绕着相同的主题,从不同的侧面进行展开,在防止信息泄露和免受攻击的同时,也起到了保护敏感信息的功效.

4.1 隐藏证书

隐藏证书的概念,最初由 Holt 等人提出[29];随后,Bradshaw 等人提出了使用隐藏证书来隐藏复杂策略的解决方案,并分析该方案的协议性能等相关问题[30];Frikken 等人提出了使用隐藏证书隐藏访问控制策略[31].隐藏证书基于椭圆曲线加密的原理(即大素数相乘容易因式分解困难),具有很好的安全保密性与数据完整性.将隐藏证书引入到自动信任协商系统,则可充分保护证书不受攻击、协商不被破坏,以及防止敏感信息的泄露.

在使用隐藏证书时,需要根据不同的应用来定义系统参数.一般地,系统参数可定义为

$$params=\langle q,G_1,G_2,\hat{e},n,P,H_1,H_2,\varepsilon,D\rangle.$$

其中:q 是一个大素数;G_1 与 G_2 为两组群,其生成元为 q;$\hat{e}$ 为线性映射,满足:

1) 线性:$\hat{e}:G_1\times G_1\to G_2$;

2) 非退化性:$\forall P\in G_1,\hat{e}(P,P)\in G_2$;

3) 可计算性:存在着有效的算法能够快速计算 $\hat{e}(P,Q)$,且满足公式 $\hat{e}(aP,bQ)=\hat{e}(P,Q)^{ab}$;$n$ 指明了消息(M)和密文(C)空间大小;P 是 G_1 中的随机元素 $\forall P\in G_1$,用来产生公钥与私钥;H_1,H_2 为两个哈希函数,满足 $H_1:\{0,1\}^n\to G_1^*$,$H_2:G_2\to\{0,1\}^n$;ε,D 为对称的加密/解密函数.

一个隐藏证书系统包括 4 个函数:两个安装函数(创建与发布)和一对加/解密函数.

(1) 证书创建函数 $CA\_Create()$:创建证书机构或发布者;

(2) 证书发布函数 $CA\_Issue(nym,attribute)$:创建证书,证书的主体为 nym,其具有属性 $attribute$;

(3) 加密函数 $CT=HC_E(R,nym,Policy)$:对资源 R 进行加密,R 由策略 $Policy$ 进行保护,R 的接收者为 nym,CT 为加密后的密文;

(4) 解密函数 $R=HC_D(CT,\{Cred_1,\ldots,Cred_n\})$:对密文 CT 进行解密,当且仅当证书 $\{Cred_1,\ldots,Cred_n\}$ 包含有 nym 的证书且满足策略 $Policy$ 时,才可恢复资源 R.

隐藏证书具有证书不可区分性(indistinguishability),表现为:若元策略 P 与 P' 满足 $P\neq P'$,CT 与 CT' 为其对应的密文.对任何不拥有满足策略 P 与 P' 对应证书的用户,CT 与 CT' 必须是不可区分的,即若消息的接收者不拥有

解密消息的证书,则该接收者不可获得任何有关证书的信息.从攻击-保护角度来看,证书的不可区分性体现在:有且仅有一个接收者可以决定使用哪些证书来解密消息,这些证书满足保护消息的策略.这意味着,对于任何攻击者 A,要在非多项式时间内获得敏感信息的可能性很小.攻击者 A 向 CA 发送 t 个请求,获取随机产生的 CA 公钥或私钥,A 选择一个接收者 nym 以及两个元策略 P_0 和 P_1(A 并未收到相关的证书).A 向挑战者 C 发送策略 P_0 和 P_1,nym 以及选取的消息 M; C 随机选取字符 $b\in\{0,1\}$,并使用策略 P_b 和 nym 来解密消息 M,将解密的明文结果返回给 A;A 重复刚才的工作,继续请求 CA 公钥或私钥,向其他挑战者发送消息,使用新的 $b'\in\{0,1\}$ 接收新的解密明文.若 $b'=b$,则 A 攻击成功.在隐藏证书系统中,A 成功的概率满足 $|\Pr[b'=b]-1/2|<1/f(t)$,其中,函数 $f(t)$ 为任意 t 多项式.

隐藏证书对自动信任协商的贡献在于:

(1) 可解决基于 PKI 认证系统中“谁先”的问题:在信任协商系统中,协商双方都不愿意先暴露自己的策略,这样就妨碍了用户对资源/服务的正常访问;

(2) 访问控制策略不会让非授权的接收者所了解:策略、资源信息都是密文,窃听者无法解开,或者说解密代价太大,获得信息后已失去意义;

(3) 隐藏证书没有固定的格式,可随意创建和使用:没有分布式证书链发现的网络开销,也没有保存大量证书的存储代价等;

(4) 良好的协商协议保证了协商效率:在协商过程中,除一次身份交换外,隐藏证书只需一个回合的消息交换,极大地降低了证书交换的网络开销;

(5) 较高的安全级别保护信息不被泄露和攻击.

4.2 无记忆属性证书

无记忆属性证书(oblivious attribute certificates,简称 OACerts)[32]是通过允许证书拥有者选择使用其中的哪些属性以及如何使用这些属性,以达到防止信息外泄和免受攻击的目的.OACerts 使用零知识协议,协商双方通过输入不同的属性值计算授权函数,以决定能获得什么样的信息.

OACerts 的模型可描述为(假定 Alice 为资源提供方,Bob 为资源访问者)

$$F[commit,pred]_{\text{Alice}}(Params,M,a,r)=0 \tag{1}$$

$$F[commit,pred]_{\text{Bob}}(Params,c,M,a,r)=\begin{cases}M, & \text{if } c=commit_{Params}(a,r)\wedge pred(a)=\text{True}\\ 0, & \text{otherwise}\end{cases} \tag{2}$$

其中:$commit$ 为提供零知识协议的算法;$Params$ 为供 $commit$ 使用的公共安全参数;$pred$ 是公共谓词,$pred\in\{\geq,\leq,\neq,\in,\notin,>,<,=\}$;$a$ 是一个私有数字,代表 Bob 的属性值;r 为随机数字;M 是 Alice 欲传输给 Bob 的消息,即需要保护的资源;$c=commit_{Params}(a,r)$ 理解为对属性值的数字签名;F 是由 $commit$ 与 $Pred$ 参数化的函数,这里,$F[commit,pred]_{\text{Alice}}$,$F[commit,pred]_{\text{Bob}}$ 分别表示 Alice 与 Bob 的输出.整个模型中,$commit$,$pred$,$Params$ 以及 c 是公共参数;Alice 输入消息 M;Bob 输入 a 与 r.式(1)表示:资源提供方 Alice 不能从协商过程中获得任何关于 Bob 的信息;式(2)则说明:只有当 Bob 提交了正确的属性值以及用来签名的随机数时,他才能获得资源信息;否则,Bob 无法获取其他更多的信息.

OACerts 可以避免信息泄露并保护信息免受攻击,OACerts 的优势在于:

(1) 使用谓词,对属性值的范围进行检查,避免了用户属性值的直接泄露.在信任协商过程中,为了达到某种信任关系,用户将要暴露一些证书.而若策略设计不完善或有缺陷,则很有可能泄露一些信息.如对于策略“*CGCL* 实验室的高性能机器允许实验室的老师、博士以及项目领导者直接使用”,如果策略描述为复合策略 $P=P_1\vee P_2\vee P_3$(其中:$P_1=\{role=Teacher\}$,$P_2=\{role=PhD\}$,$P_3=\{role=Teamleader\}$),则可通过 $P_i(Cred)=\text{True}(i=1,2,3)$ 是否成立来确定用户的角色;而如果把策略描述为简单策略 $P=\{Teacher,PhD,Teamleader\}$,使用谓词 $\in$ 来匹配,若 $\in(user.role)=\text{True}$,则允许访问.使用谓词来判断属性值,只能确定一个范围,不能确定一个具体的属性值;

(2) 使用零知识协议,避免了非授权猜测.OACerts 在 Pedersen Commitment 模式[32]的基础上,提出了 OCBE(oblivious commitment based envelope)协议来提供零知识协议.同时,OACerts 针对每一个谓词提出一种协

议,如使用 EQ-OCBE,GE-OCBE,LE-OCBE 来保证谓词“=,≥,≤”的工作.

4.3 OSBE协议

OSBE(oblivious signature-based envelope,无记忆基于签名信封)[33]是通过检查用户提交的数字签名是否为指定证书的签名,来决定用户是否有权获取信息.OSBE 从数据完整性出发,使用数字签名的唯一性来保证非合法无法获取传输的信息.

OSBE 的基本思想可描述为

$$F[Verify,M,PK]_{\text{Alice}}(P,\sigma)=\perp \tag{3}$$

$$F[Verify,M,PK]_{Bob}(P,\sigma)=\begin{cases}P, & \text{if } Verify_{PK}(M,\sigma)=\text{True}\\ \perp, & \text{otherwise}\end{cases} \tag{4}$$

在该描述中,PK 是指 CA 的公钥;M 与 P 是两条消息,均是 Bob 证书的内容,但 M 没有数字签名,而 P 是具有数字签名的完整消息;σ是使用 PK 对 M 的数字签名;$Verify$ 是检验签名是否属实的函数,$Verify(M,\sigma)$=True 表示σ是对 M 的数字签名,其公钥为 PK; F 是以 $Verify$,M 和 PK 为参数的函数;⊥表示空的意思,即没有输出;$F[Verify,M,PK]_{\text{Alice}}$,$F[Verify,M,PK]_{\text{Bob}}$ 代表 Alice 与 Bob 的输出.在整个协商过程中,Alice 输入其证书 P,Bob 输入签名σ.式(3)表示:资源提供方 Alice 不能从协商过程获得任何信息;式(4)表示只有当 Bob 提交了正确的数字签名,他才可以获得证书信息;否则将一无所获.

OSBE 协议可以由 3 个函数来实现:安装(setup)、交互(interaction)以及开启(open).假设 S 为消息发送者,R_1 与 R_2 为两个不同的消息接收者(R_1 具有数字签名,而 R_2 不具有),Sig 为签名函数,则 OSBE 协议可描述为:

(1) 安装函数:安装算法随机选取安全参数 t,并创建系统参数:签名公钥 PK. M 与 P 是欲传输的消息,M 是没有数字签名的消息,P 是完整的消息.PK 与 M 将发送给 3 个参与方:S,R_1 与 R_2.同时,S 获得消息 P, R_1 获得签名$\sigma=Sig_{PK}(M)$.

(2) 交互函数:系统从 R_1 与 R_2 中随机选择一个作为消息接受者 R,R 并不为 S 所知.S 与 R 同时运行一个 OSBE 的交互协议.

(3) 开启函数:在运行交互协议后,若 $R=R_1$,则 R 获得消息 P;否则,R 无输出结果.

OSBE 协议提供了一种使用签名防止信息泄露的框架,可使用多种签名算法来实现.综合起来,OSBE 协议的主要贡献在于:

(1) 单轮回的消息交换协议,降低消息受威胁或攻击的几率,减少了网络传输开销,提高了协商效率;

(2) 提供了一种使用签名防止信息泄露的框架,有利于根据不同的应用场景,设计不同的签名协议;

(3) OSBE 协议与签名算法具有相同的安全级别和极好的消息保密性.

4.4 ACK策略

ACK 策略(acknowledgement policy,确认策略)[1,3]主要用来避免协商者对另一方的非授权猜测,以达到保护证书里属性目的.一般地,当访问者拥有满足访问控制策略的证书时,其行为和表现与没有证书时有很大的差别.从用户对访问控制策略的反应里,协商方可推测该用户是否具有某些信息,以及哪些信息是敏感的.ACK 策略的目标在于:协商者在没有满足 ACK 策略之前,是不清楚对方是否具有某些属性的.

在信任协商过程中存在着 4 类非授权推测[3]:

(1) 向前肯定推测:假设攻击者 O 知道在协商者 M 与 P 之间存在 $A.t\leftarrow B.r.O$ 向 M 询问是否满足 $B.r$,若 M 具有属性 $B.r$,则 O 可推测 M 满足 $A.t$ 的需求;

(2) 向前否定推测:假设攻击者 O 知道在协商者 M 与 P 之间存在 $A.t\leftarrow B.r.O$ 向 M 询问是否满足 $B.r$,若 M 不具有属性 $B.r$,则 O 可推测 M 亦不满足 $A.t$ 的需求;

(3) 向后肯定推测:假设攻击者 O 知道在协商者 M 与 P 之间存在 $A.t\leftarrow B.r.O$ 向 M 询问是否满足 $A.t$ 的需求,若 M 满足 $A.t$,则 O 可推测 M 具有 $B.r$ 的需求;

(4) 向后否定推测:假设攻击者 O 知道在协商者 M 与 P 之间存在 $A.t\leftarrow B.r.O$ 向 M 询问是否满足 $A.t$ 的需

求,若 M 不满足 $A.t$,则 O 可推测 M 亦不具有 $B.r$ 的需求.

ACK 策略的原理是:对于需要保护的证书,使用 ACK 函数进行标记,表示不暴露标记的证书或证书中的信息,达到不泄露信息的目的.

4.5 TTG协议

TTG(trust target graph,信任目标图)协议[1,3]是将协商过程构建成一棵查询树.TTG 协议支持 ACK 策略,是保护敏感信息和访问控制策略的重要方法.TTG 是一个直连图,节点表示协商者的查询,每一个信任目标都与唯一的节点关联;边则具有不同的类型,代表着不同的依赖关系.

在 TTG 中,节点的类型包括:

(1) 标准目标节点(standard target):该类节点的形式为$\langle V:\xleftarrow{?} S\rangle$.其中:$V$ 是协商者;f 为普通角色或链接角色;S 通常是与 V 进行通信的实体,也可以是其他实体.该类节点的含义为:V 检测 S 是否具有角色 f.

(2) 交叉目标节点(intersection target):该类节点的形式为$\langle V:A_1.r_1\cap\ldots\cap A_k.r_k\xleftarrow{?} S\rangle$.其含义为:$V$ 检测 S 是否满足 $A_1.r_1\cap\ldots\cap A_k.r_k$ 的要求.

(3) 细节目标节点(trivial target):该类节点的形式为$\langle V:S\xleftarrow{?} S\rangle$,用来检测实体 S 是否是另一实体的成员.细节目标为 TTG 中的边提供了占位符.

(4) 链接目的节点(linking goal):该类节点的形式为$\langle V:?X.r_2\xleftarrow{?} S\rangle$,用来检测实体 S 是否是具有 $B.r_2$ 的角色,其中,B 为任意拥有 r_2 角色的实体.

在上述 4 类节点中,V 为节点的检验者,S 是节点的实体.目标节点均有一个满足状态,用来表示该节点是否被满足,有 3 种可选值:satisfied,failed 与 unknown;链接目的节点则有一个完成状态,用来表示查询的路径是否完成,有两种可选值:complete 与 incomplete.不论是哪一类节点,都有一个处理状态,用来标识该节点是否处理过,有 3 种可选值:verifier-processed,opponent-processed 与 fully-processed.

在 TTG 中,边具有如下类型:

(1) 标准隐含边(standard implication):形式为$\langle V:f\xleftarrow{?} S\rangle\leftarrow\prec\langle V:e\xleftarrow{?} S\rangle$,其中:$e$ 为任意的角色表达式;f 是普通角色或链接角色.标准隐含边终止于标准目标接点,对起点不作限制.从边的结构来看,如果 $f\leftarrow e$ 成立,则边是有效的.

(2) 链接监视边(linking-monitor):形式为$\langle V:A.r_1.r_2\xleftarrow{?} S\rangle\leftarrow\prec\langle V:?X.r_2\xleftarrow{?} S\rangle$,表示为便于 V 检测 $A.r_1.r_2\leftarrow S$ 是否成立,V 需检查 B 是否满足 $B.r_2\leftarrow S$.链接监视边总是证明合理的.

(3) 链接解答边(linking-solution):形式为$\langle V:?X.r_2\xleftarrow{?} S\rangle\leftarrow\prec\langle V:B.r_2\xleftarrow{?} S\rangle$.从边的结构来看,这是唯一进入链接目的节点的边,该边总是证明合理的.

(4) 链接隐含边(linking implication):形式为$\langle V:A.r_1.r_2\xleftarrow{?} S\rangle\leftarrow\prec\langle V:A.r_1\xleftarrow{?} B\rangle$.当节点$\langle V:A.r_1.r_2\xleftarrow{?} S\rangle$是链接监视边的终点,且节点$\langle V:A.r_1\xleftarrow{?} B\rangle$是链接解答边的起点时,则链接隐含边是证明合理的.

(5) 交叉边(intersection):形式为$\langle V:A_1.r_1\cap\ldots\cap A_k.r_k\xleftarrow{?} S\rangle\leftarrow\prec\langle V:A_i.r_i\xleftarrow{?} S\rangle$.从结构上可以看出,交集边总是有效的.

(6) 控制边(control):形式为$\langle V:f\xleftarrow{?} S\rangle\leftarrow\prec\langle opp(V):f`\xleftarrow{?} V\rangle$,其中,$opp(V)$表示与 V 进行通信的另一协商者.控制边用来处理 ACK 策略以及 AC 策略,总是证明合理的.

当用户访问资源时,TTG 协议根据访问控制策略与证书的暴露情况构成一个策略图,通过检查证书对各节点的匹配情况,为协商生成最短路径.只有在证书满足访问控制策略时才可访问资源,从而达到了保护敏感信息的目的.

4.6 UniPro模式

UniPro(unified scheme for resource protection)模式是 Yu.等人提出的一种在信任协商过程中保护资源(包括敏感证书与访问控制策略)统一模式[7].UniPro 在早先自动信任协商工作的基础上将策略当作最优资源来看待,采用保护资源的方式来保护访问控制策略;同时为策略的暴露提供了细粒度的控制,对策略暴露与策略满足进

行了明确的区分,这为用户描述授权请求提供了极大的灵活性.

UniPro 对策略的形式、内容描述等进行了重新定义,使用谓词逻辑语言来实现,如使用谓词 Credential()来指定对证书的约束;使用谓词 Cert_authority()来指定发布证书的机构等.同时,UniPro 协议支持 3 种类型的暴露:资源、策略 ID、策略与证书之间的关系.对资源的暴露,分 3 种情况讨论:(1) 若该资源是一项服务,则另一参与方可直接调用该服务;(2) 若该资源为证书,则证书的内容将发送给另一参与方;(3) 若该资源为一项策略,则策略的定义(即策略的内容)将发送给另一参与方.对于策略 ID 的暴露,则直接将(*R*:*P*)发送给另一参与方(其中:*R* 代表资源 ID;*P* 代表为 *R* 服务的策略 ID).值得一提的是,策略 ID 的暴露仅释放策略的 ID 而已,并不泄露策略的内容.策略证书之间的关系所释放的形式为 *P.x=C*.其中,*P* 为策略 ID;*x* 为策略 *P* 内容中的变量;*C* 为与策略潜在相关的证书 ID.

5 自动信任协商未来发展趋势

在了解自动信任协商的各项研究领域以及相关技术的同时,必须充分意识到自动信任协商存在的不足;另外也应看到,自动信任协商的继续发展是有一定规律可循的.

5.1 自动信任协商当前的不足

自动信任协商在为用户跨域资源共享和网上商业交易提供安全保障的同时,也暴露出一些不足.这既有技术上的因素,也有设计的原因,还有政策的影响.概括起来,主要有以下几个方面:

(1) 自动信任协商模型的安全等级受到限制

这是由自动信任协商自身工作模型所决定的.安全模型一般分为访问控制模型与信息流模型.访问控制模型的出发点是控制主体与客体的合法访问来达到安全要求;而信息流模型则是通过分析信息在实体之间的流动来控制安全.自动信任协商是在单一安全域工作模式的基础上进行发展的,属于访问控制模型,这决定了自动信任协商是一种抽象的工作模型,很难将系统进行形式化描述,其安全级别受到了限制.

(2) 缺乏对相关概念统一的安全定义

在自动信任协商中,没有对证书、访问控制策略、协商者等给出安全定义,即没有回答“什么样的证书才算是安全的”、“什么样的访问控制策略才可提供安全保障”等问题.一般地,为了提高证书的安全级别,科研者增加了签名算法的研究力度;为了保证访问控制策略的安全,则规范访问控制策略的暴露,对敏感信息的释放进行严格限制等.但高安全级别的证书增加了加密/解密的开销,谨慎的访问控制策略暴露,降低了协商效率.

(3) 研究面广,但有些方面的研究还欠缺一定的深度.

当前在自动信任协商方面的研究,涉及到策略语言、协商机制、证书链查找、隐私保护等.然而,在某些方面还缺乏深层次的研究,如当前的研究工作缺乏有效的基础信任模型,没有有效的方法从协议角度研究协商策略的安全性,缺乏适用的自动信任协商系统中间件的设计技术等.

(4) 自动信任协商的研究大多停留于理论,缺乏大众化的系统

目前,科研者对自动信任协商的研究如火如荼,各种理论与技术相继提出,开发的项目与系统也不断增加.然而,许多研究(如策略语言、协商机制等)仅停留于理论,且自动信任协商系统一般都是重量级的,大多数系统中都使用其自行开发的策略语言,各机构公布的系统因缺乏相应的工作平台或移植困难而无法运转,系统的安装与部署费力且应用范围小.这极大地限制了自动信任协商系统的普及.

(5) 协商效率与协商成功率相互制约

协商效率与系统操作的灵活性相关,而协商成功率受系统的安全级别的影响.一方面,自动信任协商系统为保证用户访问资源的安全性,势必使用许多手段来保护协商的各个环节,包括提高证书签名算法的安全度、增强访问控制策略暴露的条件等,这制约了系统操作的灵活性;另一方面,降低信息保护标准,当系统不支持敏感信息保护时,系统协商的成功率很高,但这不能有效地保护协商双方的个人隐私以及商业交易的机密信息,系统的安全级别将受到挑战.

5.2 自动信任协商发展应遵守的原则

在自动信任协商继续发展的过程中,应本着“有所为,有所不为”的方针,避免重复开发和不必要的投入,以及减少人力、物力的浪费等.总的来说,自动信任协商的发展应遵循以下原则:

(1) 自动信任协商的过程应逐渐实现智能化

当前的自动信任协商,大多数都需要人工参与,远没有做到协商自动化与智能化.尽管人工操作可以避免程序的机械化,降低了一些恶意攻击的机会,能有效地防止“拒绝服务”等类似的网络攻击;但从自动信任协商的长远发展来看,协商智能化是一种趋势,必然会逐渐取代人工参与.且人工参与效率低下,出错的主观因素更不可控制.

(2) 自动信任协商应避免协商过程复杂化

自动信任协商通过协商双方不断的证书交换来建立信任关系:先暴露一些不携带敏感信息的证书,当达到一定的安全级别时,再释放敏感信息.在协商前,用户清楚资源方的访问控制策略,但资源方并不清楚用户释放证书的策略以及根据用户的策略可能会制定哪些新的访问策略等.因此,协商过程具有许多未知的因素,使得协商过程复杂化.当前,有效避免协商复杂化的方法主要有:1) 使用高效且保证安全级别的协商协议,减少协商轮回,降低因信息交互而带来的网络开销;2) 开发新的算法,或对现有算法进行改进,提高证书-策略匹配效率.

(3) 自动信任协商应做到安全级别高,系统操作灵活

一方面,安全级别高可确保协商过程不受攻击,保证用户隐私和商业机密不被泄露,这势必会给用户操作带来一些不便,从而降低协商效率,影响协商成功率;另一方面,用户使用方便,协商过程简单,随意暴露证书、访问控制策略的信息,这确实提高了协商效率与成功率,但却不能保证用户隐私和重要信息的安全,用户访问资源将受到威胁.因此,在设计和开发进程中,必须根据具体的应用,在系统的安全需求与操作需求之间进行折衷,在提高协商效率的同时,兼顾协商成功率.

5.3 自动信任协商的发展趋势

自动信任协商的目的是解决用户跨域资源访问安全问题,从自动信任协商过去的研究历程与当前的研究状况来看,自动信任协商的未来发展趋势将体现在:

首先,自动信任协商将继续设计与开发一系列安全、高效的协商协议.

自动信任协商通过协商双方不断暴露证书与访问控制策略来达到建立信任关系的目的,这样,安全、高效的协商协议必不可缺少.尽管当前已有许多安全协议,但这些协议仅从某个方面解决了一定的问题,如隐藏证书系统的协议具有较高安全级别与较少的证书交换轮回,但证书与策略匹配效率不佳,并且加密/解密开销较大,这使得整个协商效果并不理想.因此,自动信任协商在未来的发展中,将继续致力于安全、高效的协商协议的设计与开发.同时,自动信任协商的发展,将更多地借鉴密码系统或混沌加密技术,通过引进高安全级别的算法等办法来提高协议的安全性.亦可使用可信芯片,以有效防止外界对协商的破坏与攻击.

其次,自动信任协商中将增加对访问控制策略与授权的一致性分析与检测.

在实际应用中,应避免访问控制策略与授权的不一致性.造成策略与授权相互冲突的原因有多方面,主要表现在:1) 不加限制的委托:在使用 SPKI/SDSI 证书的系统中,用户可将权限授予他人,当多人向同一用户进行授权时,很容易造成用户对同一实体拥有不同的权限;2) 策略描述模糊:为了达到保护一些重要资源的目的,许多资源方将其访问控制策略设置得很复杂,有时甚至是相互冲突的,类似于“没有明确规定允许访问就是被拒绝”与“没有明确规定禁止访问就是被接受”之间的矛盾.因此,自动信任协商在未来的发展中,策略与授权的一致性分析将不可忽视.

第三,自动信任协商中将加入良好的反馈机制,以提高协商成功率.

在协商过程中,当用户提交的证书不足以让资源方授权访问资源时,系统应给予一定的反馈,以引导协商继续进行.关于反馈信息,一般存在着两种极端:

(1) 系统只是简单地给出“访问拒绝”,而没有任何其他信息.这种方法极好地保护了系统策略信息的安

全,因为没有必要给非授权用户更多的信息.然而,该方法缺乏灵活性,对合法用户并不友好,用户并不清楚拒绝访问的原因,如到底是系统程序出错,还是访问权限不够等.

(2) 系统给出所有策略信息.这种方法具有极大的灵活性,提高了协商的成功率,同时也增加了用户点击率,扩大了系统的影响力.同样,不加区分地暴露策略,势必会危及系统的安全性,如攻击者可能根据这些策略信息对系统进行入侵,策略中的信息也会暴露一些合法用户的隐私等.

当前,对自动信任协商的反馈机制研究得并不多,许多系统都没有完整的反馈机制,或者采用第 1 种极端方法,这样远不能满足用户对自动信任协商的需求.因此,自动信任协商在未来的发展中,引入良好的反馈机制是很必要的.在实际的系统中,可引入成熟智能 Agent 技术来实现.

第四,自动信任协商将向无线与普适计算方面发展.

当前对自动信任协商的研究,主要是针对有线的、对等的计算机系统而言,而在无线与普适计算方面的研究尚未拉开序幕.随着科技的高速发展,高性能芯片与集成电路不断涌现,以及移动电话、手提电脑等无线设备的普及,人们对自动信任协商在无线与普适计算方面的需求越来越大,这将迫使自动信任协商向无线与普适计算方面发展.如使用可信计算芯片来实现自动信任协商的智能化,通过使用智能卡来实现与无线设备的互连互通等.

最后,自动信任协商将更注重应用与实践.

一项技术或者理论如何发展或成熟,如果不应用于实际,不能为人们的工作和生活带来方便,它将不会为人们所接受.因此,自动信任协商的高度发展(包括理论研究与系统开发),最终都应体现在实际的应用与系统中.当前,自动信任协商相关的项目与系统比较多,但尚未有大众化的系统.自动信任协商在未来的发展中,应更多地与一些商业机构进行联合开发,设计一些具有通用接口的安全中间件,研究开发一些商业化的产品,这样更有利于自动信任协商的进一步发展与成熟.

6 结束语

自动信任协商代表着一种新型的安全技术,为实现跨域资源共享与互访提供安全保障.目前,自动信任协商已引起了科学界的足够重视,研究者针对该领域中的问题,从不同的角度进行研究,相关的理论、系统与项目也不断面世.

本文回顾了自动信任协商领域的研究内容,对相关技术进行了总结,包括自动信任协商对系统与策略语言的需求、证书链的发现与检索、防止信息泄露与被攻击,以及对敏感信息与访问控制策略的支持等.同时,在回顾目前自动信任协商发展的基础上,对自动信任协商中存在的不足进行了总结,并指出了自动信任协商在发展中应遵守的原则,以及自动信任协商的未来发展趋势.在接下来的工作中,立足于自动信任协商的研究现状,从提高协商效率与协商成功率出发,对已有的协议进行改进或开发新的协议,研究具有高安全标准、低系统开销的自动信任协商机制与系统,为跨域资源共享与互访提供更可靠的安全保障.

References:

[1] Winsborough WH, Li NH. Towards practical automated trust negotiation. In: Michael JB, ed. Proc. of the 3rd Int'l Workshop on Policies for Distributed Systems and Networks. Washington: IEEE Computer Society Press, 2002. 92−103.

[2] Winsborough WH, Seamons KE, Jones VE. Automated trust negotiation. In: Hilton SC, ed. DARPA Information Survivability Conf. and Exposition. New York: IEEE Press, 2000. 88−102.

[3] Winsborough WH, Li NH. Protecting sensitive attributes in automated trust negotiation. In: Sushil J, Pierangela S, eds. Proc. of the ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2002. 41−51.

[4] Li JX, Huai JP, Li XX. Research on automated trust negotiation. Journal of Software, 2006,17(1):124−133 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/17/124.htm

[5] Li LX, Chen WM, Huang SL. Realizing mandatory access control in role-based security system. Journal of Software, 2000,11(10): 1320−1325 (in Chinese with English abstract).

[6] Saunders G, Hitchens M, Varadharajan V. Role-Based access control and the access control matrix. In: Hitchens M, ed. Proc. of the ACM SINGOPS Operating Systems Review. New York: ACM Press, 2001. 6−20.

[7] Yu T, Winslett M. A unified scheme for resource protection in automated trust negotiation. In: Chris B, ed. Proc. of the 2003 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2003. 245−257.

[8] Bosworth KP, Tedeschi N. Public key infrastructures—The next generation. Journal of BT Technology, 2001,19(3):44−59.

[9] Li NH. Local names in SPKI/SDSI. In: Lee ES, ed. Proc. of the 13th IEEE Computer Security Foundations Workshop. Washington: IEEE Computer Society Press, 2000. 2−15.

[10] Harbitter A, Menasce DA. A methodology for analyzing the performance of authentication protocols. In: Wallach DS, ed. Proc. of the ACM Trans. on Information and System Security. New York: ACM Press, 2002. 458−491.

[11] Thompson MR, Essiari A, Mudumbai S. Certificate-Based authorization policy in a PKI environment. In: Thompson MR, ed. Proc. of the ACM Trans. on Information and System Security. New York: ACM Press, 2003. 566−588.

[12] Zhang LH, Ahn GJ, Chu BT. A rule-based framework for role-based delegation. In: Sandhu RS, Jaeger T, eds. Proc. of the 6th ACM Symp. on Access Control Models and Technologies. New York: ACM Press, 2001. 153−162.

[13] Xu Z, Li L, Feng DG. A constrained role-based delegation model. Journal of Software, 2005,16(5):970−978 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/16/970.htm

[14] Zhang XZ, Oh S, Sandhu R. PBDM: A flexible delegation model in RBAC. In: Ferrari E, Ferraiolo D, eds. Proc. of the 8th ACM Symp. on Access Control Models and Technologies. New York: ACM Press, 2003. 149−157.

[15] Seamons KE, Winslett M, Yu T. Limiting the disclosure of access control policies during automated trust negotiation. In: Network and Distributed System Security Symp (NDSS 2001). Internet Society Press, 2001. http://isrl.cs.byu.edu/pubs/ndss2001.pdf

[16] Seamons KE, Winslett M, Yu T, Smith B, Child E, Jacobson J, Mills H, Yu L. Requirements for policy languages for trust negotiation. In: Michael JB, ed. Proc. of the 3rd IEEE Int'l Workshop on Policies for Distributed Systems and Networks. Washington: IEEE Computer Society Press, 2002. 68−79.

[17] Bertino E, Ferrari E, Squicciarini AC. Trust Negotiations: Concepts, Systems and Languages. Washington: IEEE Computer Society Press, 2004. 27−34.

[18] Bonatti P, Samarati P. Regulating service access and information release on the Web. In: Bagchi A, ed. Proc. of the 7th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2000. 134−143.

[19] Herzberg A, Mass Y, Mihaeli J, Naor D, Ravid Y. Access control meets public key infrastructure, or: Assigning roles to strangers. IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2000. 2−14.

[20] Bertino E, Castano S, Ferrari E. On specifying security policies for web documents with an XML-based language. In: Ferrari E, Ahn GJ, eds. Proc. of the 6th ACM SACMAT. New York: ACM Press, 2001. 57−65.

[21] Bertino E, Castano S, Ferrari E. Securing XML documents: The author-x poject demonstration. In: Keogh H, ed. Proc. of the ACM SIGMOD Conf. New York: ACM Press, 2001. 605−615.

[22] Li NH, Mitchell JC, Winsborough WH. Design of a role-based trust management framework. In: Heather H, ed. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2002. 114−130.

[23] Li NH, Mitchell JC. Datalog with constraints: A foundation for trust management languages. In: Verónica D, Philip W, eds. Proc. of the 5th Int'l Symp. on Practical Aspects of Declarative Languages. LNCS 2562, Berlin, Heidelberg: Springer-Verlag, 2003. 58−73.

[24] Li NH, Winsborough WH, Mitchell JC. Distributed credential chain discovery in trust management. In: Herbert AS, ed. Proc. of the IEEE Symp. on Computing and Communications Security. New York: ACM Press, 2001. 156−165.

[25] Bertino E, Ferrari E, Squicciarini AC. Trust-X: A peer to peer framework for trust negotiations. In: Agrawal R, ed. Proc. of the IEEE Trans. on Knowledge and Data Engineering. Washington: IEEE Computer Society Press, 2004. 132−138.

[26] Blaze M, Feigenbaum J, Strauss M. Compliance checking in the policymaker trust management system. In: Franklin M, ed. Proc. of the 2nd Financial Crypto Conf. New York: IEEE Press, 1998. 205−216.

[27] Jim T. SD3: A trust management system with certificate evaluation. In: Roger N, Martin A. eds. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society Press, 2001. 106−115.

[28] Yu T, Ma X, Winslett M. PRUNES: An efficient and complete strategy for automated trust negotiation over the Internet. In: Bagchi A, ed. Proc. of the 7th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2000. 210−219.

[29] Holt JE, Bradshaw RW, Seamons KE, Orman H. Hidden credentials. In: Jajodia S, Samarati P, Syverson PF, eds. Proc. of the ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2003. 1−8.

[30] Bradshaw RW, Holt JE, Seamons KE. Concealing complex policies with hidden credentials. In: Carl AB, ed. Proc. of the 11th ACM Conf. on Computer and Communications Security. New York: ACM Press, 2004. 146−157.

[31] Frikken K, Atallah M, Li JT. Hidden access control policies with hidden credentials. In: Vijay A, ed. Proc. of the 3rd ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2004. 27−28.

[32] Li JT, Li NH. OACerts: Oblivious attribute certificates. In: Han YF, ed. Proc. of the 3rd Conf. on Applied Cryptography and Network Security. New York: ACM Press, 2003. 108−121.

[33] Li NH, Du WL, Boneh D. Oblivious signature-based envelope. In: Elizabeth B, ed. Proc. of the 22nd ACM Symp. on Principles of Distributed Computing (PODC 2003). New York: ACM Press, 2003. 182−189.

附中文参考文献:

[4] 李建欣,怀进鹏,李先贤.自动信任协商研究.软件学报,2006,17(1):124−133. http://www.jos.org.cn/1000-9825/17/124.htm

[5] 李立新,陈伟民,黄尚廉.强制访问控制在基于角色的安全系统中的实现.软件学报,2000,11(10):1320−1325.

[13] 徐震,李斓,冯登国.基于角色的受限委托模型.软件学报,2005,16(5):970−978. http://www.jos.org.cn/1000-9825/16/970.htm

廖振松(1979－),男,湖北仙桃人,博士生,主要研究领域为网格计算,网格安全.

李赤松(1976－),女,博士生,讲师,主要研究领域为网格计算,网格安全.

金海(1966－)男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机系统结构,集群计算,网格计算,并行与分布式计算,对等计算,普适计算,语义网,存储与网格安全.

邹德清(1975－),男,博士,副教授,主要研究领域为系统安全,集群与网格计算.