自动信任协商中的攻击与防范

李 开 卢正鼎 李瑞轩 刘百灵
(华中科技大学计算机科学与技术学院 武汉 430074)

摘 要 自动信任协商主要解决跨安全域的信任建立问题,使陌生实体通过反复的、双向的访问控制策略和数字证书的相互披露而逐步建立信任关系。由于信任建立的方式独特和应用环境复杂,自动信任协商面临多方面的安全威胁,针对协商的攻击大多超出常规防范措施所保护的范围,因此有必要对自动信任协商中的攻击手段进行专门分析。按攻击特点对自动信任协商中存在的各种攻击方式进行分类,并介绍了相应的防御措施,总结了当前研究工作的不足,对未来的研究进行了展望。

关键词 自动信任协商,攻击,防范措施,敏感信息
中图法分类号 TP309 **文献标识码** A

Attacks and Defenses in Automated Trust Negotiation

LI Kai LU Zheng-ding LI Rui-xuan LIU Bai-ling
(College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China)

Abstract The purpose of Automated Trust Negotiation (ATN) is mainly to establish trust among different security domains. ATN is an approach to establish mutual trust between strangers wishing to share resources or conduct business by gradually requesting and disclosing access control policies and digital credentials. Special attacks can be initiated to ATN according to the characteristics of the way of trust establishment, which cannot be effectively tackled by the measures preventing normal network attacks. Therefore, it is essential to analyze all kinds of attacks existing in ATN. A comprehensive survey of research on attacks in ATN was presented based on the classification and introduction of different attacking manners and corresponding defenses, the shortcomings of the current related research were pointed out and the development trend was also discussed.

Keywords Automated trust negotiation, Attack, Defense, Sensitive information

自动信任协商(Automated Trust Negotiation, ATN)是继信任管理之后提出的一种跨安全域的信任建立方法,即通过陌生实体间逐步地请求、披露访问控制策略和数字属性证书而达成相互信任[1]。由于自动信任协商建立在开放的互联网中,其建立信任的方式独特,例如信任通过数字证书的逐步披露而建立、证书的分布式存储、允许授权、每个敏感的证书和访问控制策略都有相应的访问控制策略来保护等,攻击者会根据这些特点对自动信任协商有针对性地进行攻击[2,3]。很多情况下,传统的抗攻击方法无法抵御这些攻击,因此有必要对自动信任协商中存在的各种攻击进行专门分析。文献[4]仅简单地提到了DoS攻击、推理攻击以及硬件攻击,并未针对自动信任协商的具体背景对其攻击方式进行分析。目前还没有文献对自动信任协商中存在的各种攻击给出一个完整和深入的分析。本文对自动信任协商中存在的攻击手段进行分类,在分析这些攻击方式的同时介绍相应的防御措施,在此基础上指出现有研究工作的不足,对未来的工作进行了展望。为了叙述方便,本文将主动发起协商请求的一方称为客户端,而协商另一方称为服务器端。

1 拒绝服务攻击

拒绝服务攻击(Denial of Service, DoS)主要利用合理的服务请求来占用过多的服务资源,从而使合法用户无法得到相应的服务。在自动信任协商中,其攻击方式有以下几种:

1) 客户端和服务器端建立多个协商进程后,客户端就不再给予进一步的响应。这样使得服务器一直等待对方的回应,为该攻击者保留这些协商进程直到超时。这样会耗费大量的服务资源,从而没有足够的资源响应合法用户的协商请求,妨碍了与合法用户之间的信任建立。目前缓解这种攻击的主要措施包括:

(1)动态地调整超时和响应时间间隔[5]。某客户端一次超时现象还不能断定是拒绝服务攻击,也有可能是因为网络故障。但是连续地超时很可能就是服务器端遭到了拒绝服务

到稿日期:2009-09-18 返修日期:2009-12-11 本文受国家自然科学基金项目(60773191,70771043),国家高技术研究发展计划(863计划)项目(2007AA01Z403)资助。

李 开(1968—),男,讲师,主要研究方向为分布式系统安全,E-mail:kli@hust.edu.cn;卢正鼎(1944—),男,教授,博士生导师,主要研究方向为分布式计算、软件集成环境、信息安全等;李瑞轩(1974—),男,博士,副教授,主要研究方向为分布式计算、分布式系统安全等;刘百灵(1983—),女,博士生,主要研究方向为网络与信息安全。

攻击。用 P_{DoS} 表示遭受这种拒绝服务攻击的概率，其初始值为 0。在协商过程中根据某客户端出现超时的次数动态地增加 P_{DoS} 的值，例如超时一次、两次，P_{DoS} 的值分别为 0.1，0.2。如果 3 次超时，P_{DoS} 直接增加到 0.5；若 4 次超时，则很有可能受到拒绝服务攻击，将 P_{DoS} 设置为 1。服务器端的超时为 $ServerTimeout = DefaultServerTimeout * (1 - P_{DoS})$。对同一客户端的连续请求之间最小响应时间间隔限制为 $MinSpan = 1\ \text{sec} * P_{DoS}$。

(2)设置协商数和协商时间上限[6]。限制服务器端最多能同时处理的协商数目。同时根据所要处理的访问控制策略和数字证书，估计协商持续时间的先验值，并设置为协商时间的上限。

2)客户端向服务器端披露大量的与协商不相干或无效的数字证书。当服务器端接收到对方的数字证书时，对其签名进行验证。一般签名都采用非对称密码算法，例如 RSA，DSA 等。对签名的验证过程相当于对这些密码算法进行解密，其计算量是相当大的。服务器如果在短时间内要验证大量数字证书，很有可能耗尽自身的计算资源。为了缓解这类进攻，可采取以下防治措施[5]：

(1)在协商过程中，当服务器端发现客户端披露了与协商不相关或无效的数字证书时，服务器端视为遭受拒绝服务攻击，立刻中断此次协商。

(2)服务器端若在协商中发现客户端披露的证书数量异常大，则向系统发出攻击警告，将此客户端直接阻止在防火墙外。

3)客户端向服务器端披露过于复杂的访问控制策略进行协商。目前缓解这类攻击的基本思想是让客户端也必须付出一定的代价，例如：

(1)客户端发送过于复杂的访问控制策略时，要向服务器端披露与目前协商相关的、合法的且先前没有披露过的数字证书[7]。

(2)服务器端执行此复杂策略之前，客户端需要先解答服务器端给出的问题。此方法迫使攻击者自己付出计算资源的代价，从而限制拒绝访问攻击的规模[8,9]。

4)证书链验证攻击[10]。客户为了耗尽服务器的 CPU 周期，发送很长的伪造证书链给服务器端验证。为确保其有效性，服务器端需要依次验证证书链中的每个证书，所需付出的代价是很大的。而证书链的长度从理论上来说可以无限长，可以很容易通过公共信息或网络监听等方式获得一些证书，攻击者还可以自己生成新的公/私钥对，创造新的证书。几个客户同时向服务器发送足够长的证书链，足以使服务器瘫痪。攻击者通常通过以下两种方式向服务器发起攻击：(a)向服务器端发送一条合法的证书链，但该证书链并不属于攻击者自己；(b)攻击者用某些不合法的证书将不同的证书链连接起来，发送给服务器端。例如，攻击者首先收集一条证书链 $v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_k$，再制造一条新的证书链 $v_k' \rightarrow v'_{k+1} \rightarrow \cdots \rightarrow v_n'$，然后伪造一证书 $v_{k-1} \rightarrow v_k'$，将这两条证书链连接起来。在这条证书链里，除了证书 $v_{k-1} \rightarrow v_k'$ 是伪造的外，其它都是合法的。攻击者利用这种方式可以轻易生成无限长的证书链。目前解决证书链验证攻击的有效措施主要有以下几种：

(1)服务器端等待以下条件满足后再对证书链进行验证：收到一条完整的证书链，这条证书链的起始端是信任根，末端是客户端，且满足服务器端的访问控制策略。这样可以有效抵御攻击方式(a)。

(2)设置证书链长度的上限值[10]。如果服务器端收到的证书链长度超过某个事先设定的阈值，则拒绝处理。由于服务器端很难知道所有合法证书链的长度，因此设定这个上限值需要在拒绝服务攻击的威胁和拒绝合法服务请求之间进行权衡。

(3)采用 fail-stop 模型[11]。服务器端在认证证书链时若发现无效证书，则立即终止验证操作并终止此次协商。

(4)证书链缓存[10]。借助信任树来描述证书链。例如一条证书链表示为 $v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_k$，其中结点表示实体，边“→”代表证书，即证书 $v_0 \rightarrow v_1$ 表示双亲结点 v_0 授权给孩子结点 v_1。其中该证书 c 在此证书链中的高度 $h(c)$ 定义为从 v_1 到离它最远的孩子结点 v_k 的距离，$S_i = \{c \in S \mid h(c) \geqslant i\}$。文献[10]证明了若服务器缓存 S_δ，那么无论攻击者选用什么攻击策略，服务器在验证任何证书链时所需要验证的证书数都不超过 $\delta+1$，其中 δ 为一正整数。也就是说，如果服务器能缓存在证书链中的高度大于、等于 δ 的所有证书，那么服务器在验证任何证书链时所需要验证的证书数都不超过 $\delta+1$。同时，为了节省存储空间，服务器端缓存合法 S_δ 的哈希值，即 hash(S_δ)。当服务器端验证证书链时，首先计算证书哈希值。若该哈希值已在缓存中，则无需验证；若不在缓存中，则进行验证；若证书合法，则将其哈希值缓存。当缓存器满了，就替换 $h(c)$ 最小的证书。所以为了替换方便，服务器对每个证书缓存其二元组(hash(c)，$h(c)$)。

2 敏感信息的推理攻击

自动信任协商中存在 5 类敏感信息的非授权推理[12]。

1)向前肯定推理：假设攻击者 A 知道在协商者 S 与 C 之间存在 $A.t \leftarrow B.r$，若 A 知道 S 具有属性 $B.r$，则 A 可推理出 S 也满足 $A.t$ 的需求。

2)向前否定推理：假设攻击者 A 知道在协商者 S 与 C 之间存在 $A.t \leftarrow B.r$，若 A 知道 S 不具有属性 $B.r$，则 A 可推理出 S 也不满足 $A.t$ 的需求。

3)向后肯定推理：假设攻击者 A 知道在协商者 S 与 C 之间存在 $A.t \leftarrow B.r$，若 A 知道 S 满足 $A.t$ 的需求，则 A 可推测出 S 具有 $B.r$ 的需求。

4)向后否定推理：假设攻击者 A 知道在协商者 S 与 C 之间存在 $A.t \leftarrow B.r$，若 A 知道 S 不满足 $A.t$ 的需求，则 A 可推测 S 也不具有 $B.r$ 的需求。

5)概率推理：基于社会知识或者专业知识，在看起来没有联系的信息之间找出逻辑上的联系，从而推理出敏感信息。例如，用户拥有某个城市图书馆的图书证(library_card)，那么攻击者能够以较大的概率推理该用户也是该城市居民(citizenship)。

对于前 4 种推理攻击，目前提出的防范措施主要有：

(1)属性确认策略(Attribute Acknowledgment Policy，简称 Ack Policy)[12,13]。其基本思想是：对于给定的敏感属性 a，任意主体无论是否拥有 a，都披露相同的 $Ack[a]$ 策略。协商者在没有满足 $Ack[a]$ 策略之前，无法通过对方的行为来推理其否具有属性 a，从而阻止敏感信息的非授权推理。

(2)策略数据库(policy database)[14]。它允许用户从策

略数据库中随机地选取某敏感属性的访问控制策略，使得被选择的访问控制策略和拥有的属性之间没有必然的联系，从而攻击者无法从用户响应中推理出任何有意义的信息。其形式化定义为 random:$a->f(P)$，其中 a 为某属性，$f(P)$为策略集函数。random:$a->f(P)$ 理解为将属性 a 随机地映射到策略集函数 $f(P)$。

(3)文献[15]在属性确认策略的基础上，通过对制定相关属性 Ack 策略之间的强度关系加以制约，使得 $Ack[B.r]\supseteq Ack[A.t]$，即 $Ack[A.t]$被满足，$Ack[B.r]$不一定被满足；$Ack[A.t]$不被满足，$Ack[B.r]$肯定不被满足，这样使得前 4 类信息推理不成立。

类似地，方法(3)也适用于防范概率推理攻击。例如 5)中的例子，在制定 *Ack* 策略时，使得 $Ack[library\_card]\supseteq Ack[citizenship]$。

3 属性盗窃攻击

所谓属性盗窃攻击是指攻击者从其他协商者那里窃取证书，并使用这些证书与对方建立信任。目前防止这种攻击的方法有[6]：

1)检查证书拥有者。例如利用证书链追溯到证书拥有者的信息。

2)使用短期证书。即使证书被盗，由于证书的有效期很短，攻击者在该证书过期之前也不可能用于过多的协商。

3)请求来自不同证书颁发者颁发的证书。虽然某用户的单个证书可能会被盗，但是该用户所有的证书都被盗的可能性是比较小的。所以攻击者要能满足披露几个不同证书颁发者颁发的证书，他必须破解几对公/私钥对来伪造证书，这对于攻击者来说难度是很大的。

4 证书集合攻击

证书集合(credential pooling)攻击是指多个攻击者将各自的证书收集起来，联合获得某项服务，而他们中的任何一个都无法单独得到此服务。例如某出租车公司提供一项优惠服务，要求客户是芝加哥大学的本科生且拥有芝加哥地区颁发的驾照。Alice 拥有芝加哥地区颁发的驾照，但不是芝加哥大学的本科生；Carol 是芝加哥大学本科生，但由于违规驾驶，驾照证被吊销了。Alice 和 Carol 各自都不能获得此项服务，但是他们若将各自的证书联合起来并设法向服务提供者证明它们属于同一个实体拥有者，就可以获得此服务了。匿名证书系统尤其容易遭受这种攻击。目前防御这种攻击的主要思想是将证书拥有者的信息嵌入证书，使得证书拥有者不愿或无法和别人合用自己的证书，例如：

1)将证书拥有者的主要特征嵌入证书，例如指纹等。这样，当 Alice 披露多个证书的时候，服务提供者就通过校验嵌入这些证书的主要特征是否相同，来探测证书集合攻击[16]。

2)将证书拥有者的隐私信息(如信用卡密码)嵌入证书私钥[17]。如果 Carol 为实现证书集合攻击将自己的证书私钥告诉了 Alice，就相当于将自己的信用卡密码告诉 Alice，这迫使 Carol 不愿公开其私钥，也就无法进行证书集合攻击。

5 敏感信息收集攻击

自动信任协商中，攻击者与对方协商的目的并不是为了建立信任，而是以获取收集对方的敏感信息为主要目的。攻击者引诱对方披露敏感信息的方式及其对应的抵御方式分析如下：

1)网络钓鱼攻击。攻击者利用欺骗性的电子邮件和伪造的 Web 站点来进行网络诈骗活动，受骗者往往会泄露自己的私人资料。协商者可通过访问控制监控器来鉴别攻击者。当监控器发现攻击者的证书不合理时，便向用户发出警告[18]。

2)need-to-know 攻击[19]。攻击者隐藏访问控制策略的某些部分，从而限制对方只有两种选择：要么披露攻击者所想要的证书，要么终止协商。用户为了能够获得自己需要的服务，只有选择披露攻击者要求的证书来推动协商的进行，因此攻击者就可以得到与协商不相关但自己感兴趣的信息。假设客户 Bob 想获取攻击者 Alice 的资源 R，Alice 通过重写访问控制策略为 $a'=a\vee\phi_1\vee v$ 进行攻击，其中 a 为保护资源 R 的访问控制策略，ϕ_1 的定义来自攻击策略 ϕ_i，ϕ_i 定义如下：

$$\phi_i=(P_i\wedge b_i)\vee\phi_{i+1},i<n$$

$$\phi_n=(P_n\wedge b_n)$$

$$P_i=a,1\leqslant i\leqslant n$$

v 定义为：

$$v=\bigvee_{i=1}^{m}(Q_0\wedge Q_i)$$

$$Q_0=a$$

$$Q_i=b_i,1\leqslant i\leqslant m$$

式中，Q_i 是 Alice 用来防止 Bob 在协商过程中有不止一个证书披露的选择，$b_1,\cdots,b_n$ 为 Bob 可能拥有的且 Alice 感兴趣的证书，$b_{n+1},\cdots,b_m$ 是 Bob 可能拥有的但 Alice 并不感兴趣的证书。每一个访问控制策略都由策略 τ_i 保护，τ_i 的内容为 true，但 τ_i 又由策略 false 来保护。

攻击过程如下：Bob 向 Alice 请求 R，Alice 披露策略 $a'=a\vee\phi_1\vee v$ 和 $\phi_1=(P_1\wedge b_1)\vee\phi_2$。若 Bob 请求 Alice 披露 P_1 或 ϕ_2，虽然它们的内容都可以披露，但该事实又被策略 false 保护，所以 Bob 只有披露证书 b_1；若 Bob 没有 b_1，则 Alice 向 Bob 披露 ϕ_2。以此类推，直到 Alice 获得了 Bob 拥有的并且她所感兴趣的所有信息后，才披露资源 R 的真正访问控制策略 a。目前有些用于防止 need-to-know 攻击的方法不一定能完全阻止它，但可以起到一定的缓解作用，或作为抵御方法的一部分。

(1)允许协商者请求访问控制策略的定义。这种方法虽然不能完全防止该攻击，但可以作为抵御措施的一部分。

(2)在协商过程中，当访问控制策略定义被请求或策略名被披露时，协商者必须立刻披露相应策略的定义。但是这样容易遭受拒绝服务攻击，抵抗效率很低。

(3)引入零知识证明。Bob 可以要求 Alice 用零知识证明 Bob 满足了所请求策略的披露要求时，Alice 就会披露该策略。这样可以防止 Alice 使用策略 τ_i，但是增加了计算负担。

(4)审计代理的介入。审计代理检查资源拥有者的访问控制策略是否存在 need-to-know 攻击。若不存在，则签发其一个证书。用户在与陌生服务器协商前，可以先查找自己所信任的审计代理签发给该服务器某资源的证书。但审计代理的介入存在隐私权问题。

(5)限制策略定义的层数不超过两层。即服务器端可以定义访问控制策略来保护自己的资源(第一层)，并定义访问控制策略来保护这些控制资源被访问披露的策略(第二层)，

除此之外，再不能定义更多层的策略。该方法对于抵制这种攻击是很有效的。

(6)规定策略本身的定义要比保护该策略的策略定义更严格。即若用户能够满足某访问控制策略，那么他一定也能够满足保护该策略的策略。这样可以防止攻击者使用 false 策略来保护 r_i，但以牺牲策略定义的自治性为代价。

(7)用户为特定资源的请求所披露的证书划定范围。例如 Bob 可以预先定义请求驾驶执照更新服务时只披露自己目前的驾驶执照，其它信息一概不披露。但是这种方法不现实，因为用户不可能对所有的资源请求规定证书披露的范围，这既耗时又费神，而且需要对这些范围做实时修改。

(8)协商过程中，用户不披露任何信息，使得服务器端披露尽可能多的访问控制策略，直到最后披露用户所请求资源的真正策略。但是用户的这种行为很容易被服务器端识破，那么服务器端也会拒绝披露任何信息。

(9)用户可以要求利用一些自动信任协商技术来避免直接披露证书。例如 hidden credentials[20,21]，资源拥有者将资源访问控制策略加密发送给用户。只有用户满足其访问控制策略时才能解密出资源，不管该加密信息是否能解开，用户都不需要再给服务器端以任何回复；OACerts[22]，实现证书和属性披露的分离，即协商过程中，即使用户披露了证书，服务器也无法得到该证书里的任何属性信息；CIPPE[23] 既不需要服务器端披露资源的访问控制策略，也不需要客户端披露自己的证书，双方就能够知道是否能够建立信任，这样可以防止攻击者使用 need-to-know 访问控制策略。

6 证书互斥攻击

某些证书之间存在着互斥关系，激活一个新的角色需要撤销先前激活的某个角色。由于自动信任协商是通过信息的逐步披露建立信任，攻击者可以在同一个协商但不同的时间使用互斥的证书来获得资源，这就是证书互斥攻击。例如 Bob 在经济公司 A 工作，但同时也是经济公司 B 的兼职顾问，这两个角色是互斥的。Bob 想访问由 C 公司提供的一个在线数据库，该数据库只允许 B 公司有权进行 10 万元以上交易的职员访问。Bob 是 B 公司的职员，但只是兼职顾问，他没有权力进行 10 万元以上的交易，所以他无权访问该数据库，但是他在 A 公司有权进行 10 万元以上交易。所以 Bob 先向 C 提交了自己在公司 B 的工作证，C 验证其证书有效之后，Bob 为能够在 A 公司申请证明其有权进行 10 万元以上交易的证书，将 B 公司的工作证注销，申请到 A 公司的证书并发送给 C 后，Bob 就能够访问该数据库。因此在自动信任协商中，证书的有效性不仅仅局限于语法上的有效，即证书格式正确、证书签名有效以及在有效期内，而且还有语义上的有效性，即当某时刻 t 得知证书 c 在将来的时间 t' 内都不会无效，则称 c 在时间 t 语义有效[7]。如果忽略了语义有效性(见上例)，攻击者会利用此漏洞进行证书互斥攻击，以获得非授权访问。

为了防止该攻击，提出证书内部状态一致的概念[7]。首先证书状态定义为 $s=\langle c, r, syn, sem_v, sem_i\rangle \in C\times T\times B\times (T\cup\{\text{NULL}\})\times(T\cup\{\text{NULL}\})$，其中 T 表示时间戳的集合，r 表示证书 c 被接收的当地时间。当证书 c 语法上有效时，布尔值 syn 为 true，sem_v 表示 c 被验证在语法上有效的最近时间，sem_i 表示 c 第一次被验证在语法上无效的时间，sem_v 和 sem_i 的初始值都为 NULL。如果 $\phi_{int}(V)$ 为 true，则称证书内部状态一致。$\phi_{int}(V)$ 定义如下：

$$\phi_{int}(V)\equiv(\forall s\in V: checked(s))\wedge(\max(\{\alpha(s)\mid s\in V\})<\min(\{s.sem_i\mid s\in V\}))\wedge(\max(\{\alpha(s)\mid s\in V\})<end(V))\wedge(\min(\{\omega(s)\mid s\in V\})<start(V))$$

其中，

$$start(V)=\min(\{s.r\mid s\in V\})$$

$$end(V)=\max(\{s.r\mid s\in V\})$$

V 表示证书状态的集合，$\alpha(s)$ 为证书 $s.c$ 开始有效的时间，$\omega(s)$ 为证书 $s.c$ 有效期满的时间。

证书内部状态一致保证了自动信任协商中所有相关证书在某个时候同时都是合法的，因此防止相互排斥的角色被交替激活、撤销，以获得某个资源，即有效抵御了证书互斥攻击。

7 传输信道攻击

自动信任协商依靠双方属性信息的逐步披露从而在陌生者之间建立信任。披露的信息在通信信道的传输过程中有可能被非授权用户窃听，从而导致重放攻击或中间人攻击。阻止这类攻击的措施有：

1)重放攻击。自动信任协商中，Alice 和 Bob 通过交换属性信息建立了信任。但在信任建立过程中，Bob 发送给 Alice 的信息被攻击者 Marry 得到。Marry 将 Alice 之前发送给 Bob 的信息再发送一遍给 Bob，因此 Marry 便和 Bob 建立信任并获得本不该获得的资源。防止重放攻击，目前采用的主要思想是将随机数或时间戳嵌入交换的信息内，使得交换过的信息就算被攻击者截获，也不能够重新使用[6,18]。此外，也可使用短期证书，即证书的有效期很短，但这种方法耗费比较大，因为每次使用都要重新发放证书。

2)中间人攻击。中间人攻击的主要方式是对信息进行篡改或窃取。对这类攻击，主要思想是将传输的信息加密。例如在自动信任协商中，开发或集成一些具有保密功能的通信协议，比如 SSL/TSL，SSH，SOAP，IPSec 等[24-26]。为防止披露的访问控制策略被篡改，可以将其进行数字签名[24]。

8 硬件攻击

对自动信任协商的硬件攻击主要源于对终端设备的使用。硬件攻击主要分为两类，即主动攻击和被动攻击。主动攻击表现在攻击者篡改或操纵智能卡，而被动攻击是攻击者通过分析智能卡的功率消耗、错误信息等来获取密钥。目前可通过降低智能卡的功率消耗，使软件程序的执行方式不可预测以及给与安全操作系统支持，来防止硬件攻击[27]。

结束语 目前对自动信任协商中的攻击和防范的研究还有诸多不足，有待进一步深入和拓展。具体分析如下：

1)尚没有全面检测自动信任协商中所有攻击的系统

由于目前缺乏对自动信任协商中存在的攻击进行系统的研究，因此还没有一种系统能够检测所有自动信任协商攻击。入侵检测系统[28]虽然可以检测某些常规的网络攻击，但对于大部分自动信任协商的攻击却束手无策。

2)信任协商系统防范攻击能力薄弱

TrustBuilder，Trust-X 等协商系统的抗攻击能力都很差。例如 TrustBuilder[29] 防范拒绝服务攻击的措施仅简单地设置

每轮接收证书数的上限值，对于对方通过披露无关证书、无效证书等方式发起的拒绝服务攻击都没有很好的抵御措施。而Trust-X[30]只考虑到防范中间人攻击，所以在自动信任协商系统的设计中充分考虑针对自动信任协商攻击的防御是至关重要的。

3)自动信任协商中的攻击防范措施有待进一步完善

某些攻击的抵御方法，虽然在一定程度上缓解了攻击造成的危害，但系统仍不可避免遭受了一定程度的攻击。例如证书链验证攻击，虽然通过证书链缓存将需要验证的证书数控制在$\delta+1$范围内，但被攻击者仍浪费了一定的计算资源。还存在一些针对自动信任协商的攻击，但对这些攻击目前没有很有效的防范措施。例如敏感信息的推理攻击，Ack策略也只能预防向后否定和向前肯定推理，对于另外两种推理却不能完全避免；对于need-to-know攻击的检测和防范也都有待进一步的开发。

自动信任协商中攻击的防范技术在未来的研究中还有很多工作需要完善。首先，有待发掘一体化的抵抗自动信任协商攻击的系统，该系统能够检测到自动信任协商中的各类攻击并自动采取最优措施。而"一体化"是指能够将防范各种自动信任协商攻击的最有效的方法融合到一个系统中，并能灵活地作为自动信任协商系统的一个模块，同时它的使用对现有的协议产生尽量小的影响。其次，在研究策略语言、协商策略以及协商协议时，将自动信任协商攻击的抵御也考虑在内。第三，受限环境下抵抗攻击的研究也是必要的。目前随着无线网络、普适计算的发展，信任在其中的分量越来越重，这些环境计算能力比较有限，所以效率是它们最关注的问题。因此需要专门针对这些受限环境中信任协商攻击的防范，研究出适用于它们的高效抗攻击方法。目前自动信任协商中可能还存在某些攻击未被发掘出来，因此还需做进一步的测试和探究。

自动信任协商解决跨安全域的信任建立问题，确保协商自身的安全可靠尤其重要。本文全面分析了自动信任协商中存在的各种攻击方式以及相应的防范措施，并对现有工作的不足进行了总结，对其研究前景进行了展望，希望能够对自动信任协商抗攻击的研究有所帮助，使自动信任协商技术的实用价值得到更好的体现。

参 考 文 献

[1] Winsborough W H, Seamons K E, Jones V E. Automated trust negotiation[C]// Proc. of DARPA Information Survivability Conference and Exposition. Piscataway, New Jersey: IEEE Press, 2000: 88-102

[2] Lee A J, Winslett M, Basney J, et al. The Traust Authorization Service [J]. ACM Transactions on Information and System Security, 2008, 11(2): 1-33

[3] Skogsrud H, Motahari N H R, Benatallah B, et al. Modeling trust negotiation for web services [J]. IEEE Computer, 2009, 42 (2): 54-61

[4] 廖振松，金海，邹德清，等. 自动信任协商抗攻击能力分析[J]. 计算机研究与发展，2006，43(增刊)：13-17

[5] Ryutov T, Zhou L, Neuman C, et al. Adaptive Trust Negotiation and Access Control[C]// Proc. of the Tenth ACM Symposium on Access Control Models and Technologies. Stockholm, Sweden, 2005

[6] Squicciarini A, BertinolE, Ferrari E, et al. PP-Trust-X: A System for Privacy Preserving Trust Negotiations[J]. ACM Transactions on Information and System Security (TISSEC). New York: ACM press, 2007, 10: 1-48

[7] Lee A J, Winslett M. Enforcing Safety and Consistency Constraints in Policy-bBased Authorization Systems [J]. ACM Transactions on Information and System Security, New York: ACM press, 2007, 9: 1-30

[8] Dean D, Stubblefield A. Using Client Puzzles to Protect Tls[C]// Annual USENIX Security Symposium. Washington, DC, 2001

[9] Wang X, Reiter M. Defending Against Denial-of-Service Attacks with Puzzle Auctions[C]// IEEE Symposium on Security and Privacy. Berkeley, CA, 2003

[10] Li J, Li N, Wang X, et al. Denial of Service Attacks and Decentralized Trust Management[C]// Securecomm and Workshops. 2006: 1-12

[11] Gong L, Syverson P. Fail-stop protocols: An approach to designing secure protocols[C]// Proc. of the 5th International Working Conference on Dependable Computing for Critical Applications. 1995

[12] Winsborough W H, Li N. Protecting sensitive attributes in automated trust negotiation[C]// Proc. of the ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2002: 41-51

[13] Winsborough W H, Li N. Towards Practical Automated Trust Negotiation[C]// 3rd International Workshop on Policies for Distributed Systems and Networks. Monterey, California, 2002

[14] Irwin K, Yu T. Preventing Attribute Information Leakage in Automated Trust Negotiation[C]// Proc. of the 12th ACM Conference on Computer and Communications Security (CCS-12). New York: ACM Press, 2005: 41-51

[15] 杨秋伟，洪帆，郑明辉，等. 自动信任协商中的推理攻击分析[J]. 计算机科学，2007，34(7)：76-79

[16] Camenisch J, Lysyanskaya A. Efficient Non-transferable Anonymous Multi-show Credential System with Optional Anoymity Revocation[C]// EUROCRYPT 2001. volume 2045 of Lecture Notes in Computer Science. Springer, 2001

[17] Goldreich O, Pfitzman B, Rivest R. Self-delegation with Controlled Propagation or What If You Lose Your Laptop[C]// CRYPTO'98. Volume 1642 of Lecture Notes in Computer Science. Springer, 1998

[18] Hess A, Holt J, Jacobson J, et al. Content-triggered Trust Negotiation [J]. ACM Transactions on Information and System Security, 2004, 7 (3): 428-456

[19] Olson L, Rosulek M, Winslett M. Harvesting Credentials in Trust Negotiation as an Honest-But-Curious Adversary[C]// Workshop on Privacy in the Electronic Society (WPES 2007). 2007

[20] Bradshaw R, Holt J, Seamons K. Concealing complex policies with hidden credentials[C]// Proc. of 11th ACM Conference on Computer and Communications Security. New York: ACM Press, 2004: 146-157

[21] Holt J E, Bradshaw R W, Seamons K E, et al. Hidden credentials [C]// Proc. of the 2nd ACM Workshop on Privacy in the Electronic Society. New York: ACM Press, 2003: 1-8

(下转第 98 页)

到步骤1)；

3)计算 $k^{-1} \bmod n$；

4)计算 $e=MD5(m)$；

5)计算 $s=k^{-1}(e+dAr) \bmod n$，且 $s\neq 0$，否则重新选择 k 进行计算；

6)输出消息 m 的签名(r,s)。

为了验证上述消息 m 的签名(r,s)，验证者必须预先获得一份签名者公布的参数，一旦得到了这些参数，对签名的验证过程如下：

1)确认 r,s 是区间$[1,n-1]$中的整数；

2)计算 $e=MD5(m)$；

3)计算 $w=s^{-1} \bmod n$；

4)计算 $u_1=ew \bmod n$ 和 $u_2=rw \bmod n$；

5)计算 $X=u_1G+u_2QA$，如果 X 为零点则拒绝签名，否则计算 $v=x_1 \bmod n$，其中 $X=(x_1,y_1)$；

6)如果 $v=r$ 则验证成功。

以上的数字签名算法只是举了一个例子，读者还可以根据情况来改变算法，使其更安全。

结束语 本文基于对椭圆曲线理论方法的研究，提出了一种新的椭圆双曲线密码的加/解密方法，该方法的一个重要优点在于它在计算的过程中不增加计算的难度便可提高密码的安全性。我们可以从坐标图上看出椭圆双曲线密码比原椭圆曲线密码多了一条曲线，点的选择范围要大很多，而且在代数表达式中可以看到椭圆双曲线密码的随机性要大得多等等，这些都是密码体制的良好特性，从而为信息增加了更大的安全系数。另外，椭圆双曲线计算出的数据比原椭圆曲线计算出的数据多，这样就给我们提供了灵活的操作方法，从而使攻击方法找不到追踪的线索。鉴于椭圆曲线密码对信息安全的重要作用，本文所提出的椭圆双曲线密码方法对该领域的研究具有推动作用。

参 考 文 献

[1] Cohen H, Frey G. Handbook of Elliptic and Hyperelliptic Curve Cryptography[Z]. Discrete Mathematics and its application. Chapman & Hall/CRC, 2006

[2] Sliverman J H. The Arithmetic of Elliptic Curves[M]. Springer Verlag, 1996

[3] Miller V. Uses of elliptic curve in cryptography[C]//CRYPTO'85, Lecture Notes in Computer Science, LNCS218. Springer-Verlag, 1986: 417-426

[4] Birch B J, Kuyk W. Modular Functions of One Variable IV[C]// Lecture Notes in Mathematics 476. New York-Berlin-Heidelberg: Springer-Verlag, 1975

[5] Ross R. K_2 of elliptic curves with sufficient torsion over Q[J]. Comp Math, 1992, 81: 211-221

[6] Koblitz N, Menezes, Vanstone S. The state of elliptic curve cryptography[J]. Designs, Codes and Cryptography, 2000, 19: 173-193

[7] IEEE P1363a. included ECDSA, ECNR, ECNR2, ECPV, ECDH and ECMQV. Draft Version D9[Z]. 2001

[8] Silverman J H. Advanced Topics in the Arithmetic of Elliptic Curves[M]. New York-Berlin-Heidelberg-Tokyo: Springer-Verlag, 1994

[9] Cremona J. Algorithms for Modular Elliptic Curves[M]. Cambridge: Cambridge University Press, 1997

[10] Knapp A W. Elliptic Curves[M]. Princeton: Princeton University Press, 1992

[11] Lang S. Elliptic Functions, GTM(112)[M]. New York-Berlin-Heidelberg: Springer-Verlag, 1987

[12] Ramakrishnan D. Regulators, Algebraic Cycles, and Values of L-functions[M]. Contemporary Mathematics, 83, Providence, RI: Amer Math Soc, 1989: 183-310

[13] Silverman J H. The arithmetic of elliptic curves[M]. New York-Berlin-Heidelberg-Tokyo: Springer-Verlag, 1986

[14] Silverman J H. Computing heights on elliptic curves [J]. Math Comp, 1988, 51: 339-358

[15] Akishita T, Takagi T. Zero-Value Register Attack on Elliptic Curve Cryptosystem[J]. IEICE, 2005, E88-A(1)

[16] Eicher J, Opoku Y. Using the Quantum Computer to Break Elliptic Curve Cryptosystems[Z]. CiteSeer, 1997

[17] 于飞. 对于有限域上椭圆曲线的一些算术问题的研究[D]. 合肥：中国科学技术大学, 2008

[18] 王海艳. 最优扩域上的椭圆曲线加密系统研究[D]. 太原：太原理工大学, 2007

[19] Kapoor V, Kapoor V. Ramesh Singh, Elliptic Curve Cryptography[J]. ACM Ubiquity, 2008, 20(9): 20-26

[20] Uhsadel L, Poschmann A, Paar C. An Efficient General Purpose Elliptic Curve Cryptography Module for Ubiquitous Sensor Networks[J]. Software Performance Enhancement for Encryption and Decryption(SPEED 2007), 2007

[21] Barbosa M, Moss A, Page D. Compiler Assisted Elliptic Curve Cryptography[C]//OTM Conferences, 2007(2): 1785-1802

(上接第71页)

[22] Li J, Li N. OACerts: Oblivious attribute certificates[C]// Proc. of the 3rd Conference on Applied Cryptography and Network Security (ACNS). Lecture Notes in Computer Science, volume 3531. Springer, 2005: 301-317

[23] Li J, Li N. Policy-hiding access control in open environment[C]// Proc. of the 24nd ACM Symposium on Principles of Distributed Computing (PODC). New York: ACM Press, 2005: 29-38

[24] Stallings W. Cryptography and Network Security: Principles and Practice(second ed)[M]. Prentice Hall, 1999

[25] Yu T. Automated trust establishment in open systems [D]. Illinois: University of Illinois, 2003

[26] Rescorla E. SSL and TLS: Designing and Building Secure Systems[M]. Addison-Wesley, 2001

[27] Witteman M. Advances in Smartcard Security [J]. Information Security, Bullentin, 2002

[28] Ryutov T, Neuman C, Kim D, et al. Integrated Access Control and Intrusion Detection for Web Servers [J]. IEEE Transactions on Parallel and Distributed Systems, 2003, 14(9): 841-850

[29] Winslett M, Yu T, Seamons K E, et al. Negotiating trust on the web [J]. IEEE Internet Computing, 2002, 6(6): 30-37

[30] Bertino E, Ferrari E, Squicciarini A C. Trust-X: A peer to peer framework for trust negotiations[C]// Proc. of the IEEE Trans. on Knowledge and Data Engineering. Washington: IEEE Computer Society Press, 2004: 132-138