可信计算环境下的 Canetti-Krawczyk 模型

李兴华[1], 马建峰[1,2], 马 卓[1]
(1. 西安电子科技大学计算机网络与信息安全教育部重点实验室, 陕西西安 710071;
2. 天津工业大学计算机技术与自动化学院, 天津 300160)

摘 要: 在可信环境下, 我们对密钥协商协议的形式化方法—Canetti-Krawczyk(CK)模型进行研究, 对该模型中定义的攻击者三种攻击能力重新进行分析. 发现在可信环境下, 如果用户的签名/验证公私钥对是由 TPM 生成的, 则 CK 模型中的攻击者只有一种攻击能力: 会话密钥查询(session-key query); 否则攻击者有两种攻击能力: 会话密钥查询和一种新的攻击能力—长期私钥攻陷攻击(long-term private key corruption). 另外, TPM 克服了 CK 模型中基于加密算法认证器的安全缺陷. 在此基础上, 我们提出了可信环境下的 CK 模型—CKTC. 之后, 通过一个使用 CKTC 模型进行密钥协商的例子可以看出该模型简化了可信环境下密钥协商协议的设计与分析. 另外, 通过分析我们发现: 为了提高密钥协商协议的安全性, 不同国家应该根据各自的需要在 TPM 内部增加对称加解密模块; 用户的签名/验证公私钥对也尽可能由 TPM 来生成.

关键词: 可信计算; Canetti-Krawczyk 模型
中图分类号: TP309.2 文献标识码: A 文章编号: 0372-2112 (2009) 01-0007-06

The Canetti-Krawczyk Model Under the Trusted Computation

LI Xing-hua[1], MA Jian-feng[1,2], MA Zhuo[1]
(1. *Key Laboratory of Computer Networks and Information Security, Ministry of Education, Xidian University, Shaanxi, Xi'an* 710071, *China*;
2. *College of Computer Technology and Automation, Tianjin Polytechnic University, Tianjin* 300160, *China*)

Abstract: Under the trusted environment, we rethink the three attack abilities defined in the Canetti-Krawczyk (CK) model which is a formal method for the design and analysis of key agreement protocols. We find that under the trusted environment if the signature/verification key pair is generated by TPM, the attacker in the CK model has only one attack ability: session-key query. Otherwise, he has two abilities: session-key query and one new attack ability: long-term private key corruption. In addition, TPM overcomes the weakness of the encryption algorithm based authenticator in the Canetti-Krawczyk model. Based on these, we proposed a new CK model under the trusted environment—CKTC. Thereafter, through an example of utilization of CKTC to design a key-agreement protocol, it can be seen that this formal model prediges ts the difficulty of the design and analysis a key-agreement protocol under trusted environment. In addition, we find that in order to enhance the security of a key agreement protocol, every country should add their own symmetric encryption modules in the TPM, and a user's signature/verification key pair should be generated by TPM.

Key words: trusted computation; Canetti-Krawczyk model

1 引言

随着计算机技术和网络的迅猛发展, 信息安全问题日趋复杂, 系统安全问题, 特别是计算机平台的开放框架所带来的威胁层出不穷. 传统信息安全系统是以防外为重点, 与目前信息安全主要威胁源自内部的实际状况不相符合. 另外, 从组成信息系统的服务器、网络、终端三个层面上来看, 现有的保护手段是逐层递减的. 人们往往把过多的注意力放在对服务器和网络设备的保护上, 而忽略了对终端的保护. 随着安全研究的不断深入, 人们认识到计算实体内部的攻击是一种重要的安全威胁, 因此越来越重视这些攻击所造成的危害. 为此, 研究人员提出了可信计算的概念[1~6]. 可信计算的本质主要是通过增强现有终端体系结构的安全性来保证整个系统的安全. 其主要思路是在各种终端(包含 PC, 手机以及其它移动智能终端等)硬件平台上引入可信架构, 通过其提供的安全特性来提高终端系统的安全性. 终端可信的核心是 TPM (Trusted Platform Module)芯片. TPM 通过存储、度量、报告等一系列手段来建立一个可信的计算环境, 以解决内部攻击的问题.

收稿日期: 2007-11-14; 修回日期: 2008-09-04
基金项目: 国家 863 高技术研究发展计划(No. 2007AA01Z429, 2007AA01Z405, 2007AA01Z472); 国家自然科学基金重点项目(No. 60633020), 国家自然基金(No. 60702059, 60573036); 华为公司科技基金(No. YJCB2008053MT); 天津科技攻关计划(No. 06YFGZGX17500)

在密钥协商协议的研究中,人们同样是更加关注网络上的攻击,也发现了许多攻击手段,如:未知密钥共享[7],交错攻击(interleave attack)[8],DoS 攻击[9,10],重放攻击等[8]. Delov-Yao 模型[11]中给出的攻击者的攻击能力全部是网络上的攻击. 对于这些攻击人们提出了不少解决方法. 但对于计算实体内部的攻击,并没有提出有效的方法. 而可信计算正是为了解决这个问题,它能够为协议提供一个比较安全的运行环境. 因此如何利用可信计算所提供的安全环境来简化密钥协商协议的设计,使协议变得更加实用就变得非常有意义了.

为了使问题更具有普遍性,我们在可信计算环境下重新研究密钥协商协议的一个形式化模型—Canetti-Krawczyk(CK)模型[12],该模型是目前非常流行的密钥协商协议设计与分析的形式化方法. 它将攻击者对计算实体内部的攻击能力分为三类. 而可信计算的发展为密钥协商提供了一个相对安全的运行环境,使得某些针对计算实体内部的攻击是几乎不可能成功的. 因此有必要将该理论模型同工业界的可信计算相结合,给出一个更加实用的模型,来简化密钥协商协议的设计及分析,使协议在现实中更具有实用性和安全性. 具体的方式是:保持 CK 模型中的安全定义(也即:密钥协商协议设计的安全目标)不变,在可信环境下重新考虑 CK 模型中的三种攻击,分析哪些攻击是有可能发生的,哪些攻击是不可能的. 对于在可信环境下不可能发生的攻击,在协议设计时就不用考虑,这样就降低了协议设计的复杂性.

2 背景知识

在本章中我们对相关的背景知识进行介绍,主要包括可信计算、TPM,以及 CK 模型.

2.1 可信计算及 TPM

1999 年 10 月,由 Intel、Compaq、HP、IBM、Microsoft 发起了一个"可信计算平台联盟"(TCPA:Trusted Computing Platform Alliance). 截止至 2002 年 7 月,已经有 180 多家硬件及软件制造商加入 TCPA. 该组织致力于促成新一代具有安全、信任能力的硬件运算平台.

可信计算的目的主要是增强现有计算终端体系结构的安全性,它通过在计算实体内部建立一个基于可信平台模块 TPM 的信任传递模式,来解决内部安全威胁. 使得终端具有对恶意代码,如病毒、木马、蠕虫的免疫能力. 保证程序代码没有受到完整性的破坏. 另外,可信计算还提供了防止进程之间内存直接访问的能力.

可信计算的主要功能是由可信平台模块 TPM 完成的.TPM 起着核心控制的作用,它主要提供两个功能[13]:(1) 安全存储;(2) 平台完整性测量、存储和报告. TPM 中包括的部件如图 1 所示.

TPM 中与密码运算相关的操作有 RSA 引擎、密钥生成器、SHA-1 引擎和随机数生成器等. RSA 引擎提供对内对外的数字签名功能和传输数据的加密解密功能. 密钥生成器负责生成对称密码的密钥和非对称密码运算的密钥对.

可信软件栈 TSS 是可信平台模块的支撑软件,它在 TPM 之上又扩充了 TPM 的功能. 由 TCG 设备驱动库 TDDL、TCG 核心服务 TCS 和 TCG 服务提供者 TSP 三层组成. 它主要提供了 TPM 管理、上下文管理、密钥管理、数据加解密、数据 HASH、策略和属性管理等功能. 根据 TCG 给出的架构,TSS 与 TPM 之间进行交互的时候,凡影响到安全、隐私和泄漏平台秘密的命令必须被授权才能使用;需要保护的消息在 TPM 和 TSS 之间要进行加密传输[14].

2.2 CK 模型

CK 模型是目前一种非常流行的用来分析密钥协商协议的形式化方法[12]. 它有三个重要的组成部分:非认证链路攻击模型(UM),认证链路模型(AM)和认证器(authenticator). 下面分别对它们进行介绍.

(1) 非认证链路攻击模型(UM)

UM 也即真实的网络模型,该模型中的攻击者不但具有被动的攻击能力,而且具有主动的攻击能力,他能够篡改,添加消息等.

(2) 认证链路模型(AM)和认证器(authenticator)

AM 的定义方式和 UM 的完全一样,但一个根本不同之处是:AM 中的攻击者只能够传递由参与者产生的真实消息,而不能够改变或增添消息的内容. 也即 AM 中的攻击者只具有被动的攻击能力,对于网络上传输的消息只能够窃听,而不能够修改.

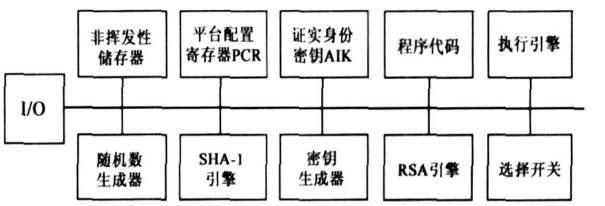

图1 TPM内部结构

认证器是一种特殊的算法,其作用就是一个自动的编译器,它能够将 AM 中的协议转化为 UM 中等价(安全性相同)的协议. 目前提出的认证器有:基于签名算法和消息认证码的认证器,以及基于加密算法的认证器[15]. 但基于加密算法的认证器被指出存在安全缺陷[16]. 由于认证器是在计算实体内部执行的算法,所以其安全性是建立在计算实体内部安全性的基础上的.

用 CK 模型来指导协议设计的时候,可以首先在 AM 中来设计和分析一个协议,然后利用一个认证器将该协议转化为现实 UM 中具有相同安全属性的协议.

(3) 攻击者的能力

CK 模型对攻击者的攻击能力进行了形式化的描述. 该模型中定义的攻击者反映了开放网络中攻击者真实的攻击能力. 它除了能够控制通信链路和控制协议事件的调度外, 还能够通过明确的攻击手段来得到协议参与者存储器中的秘密信息. 为了区分各种攻击和确保信息在被暴露情况下尽可能多的安全性, CK 模型根据攻击者能够得到的信息将攻击分为三类:

攻陷参与者(party corruption): 攻击者能够随时决定攻陷一个实体, 在这种情况下, 攻击者能够得到该实体内部存储的所有信息, 这些信息包括长期的秘密以及和会话具体相关的信息.

会话密钥查询(session key query): 攻击者通过该查询能够得到一个已经完成会话的会话密钥.

会话状态暴露(session state reveal): 攻击者通过此攻击能够得到一个还未完成会话的内部状态.

如果一个会话或者其匹配会话遭受到以上的攻击, 我们称该会话被"暴露"了.

(4) 会话密钥安全的定义

定义 1　会话密钥安全(SK-security): 如果对于任何密钥协商协议的攻击者 $\mathcal{A}$, 协议能够满足以下两条性质的话, 我们称该协议是会话密钥安全的.

1. 如果两个未被攻陷的参与者完成了匹配的会话, 它们将输出相同的会话密钥;

2. $\mathcal{A}$ 区分协议产生的会话密钥和一个随机数概率不超过 $0.5+\varepsilon$, 其中 ε 为一个在安全参数下可忽略的概率. (其中 ε 称之为"优势".)

3　可信环境下的 CK 模型

CK 模型通过模块化的协议设计方法(首先设计 AM 下安全的协议, 然后利用认证器将其转化为 UM 下安全的协议), 大大简化了协议的设计. AM 中的攻击者对于网络上的消息只具有被动攻击能力, 即: 它不能够发起网络上的攻击. 但它具有攻陷参与者、会话密钥查询及会话状态暴露能力. 而这三种攻击都是针对计算实体内部的攻击, 不涉及对网络上传输消息的攻击. 也即: AM 保证了攻击者只具有内部攻击能力下协议的安全性, 而认证器则保证了协议消息在网络上传输时不会遭受攻击, 从而保证了 AM 中协议转化为现实环境(UM)中协议的安全性. 所以从本质上来说, CK 模型将攻击分为两大类, 针对计算实体内部的攻击以及网络上的攻击.

可信计算及 TPM 就是来加强计算实体内部的安全性, 为协议的执行提供了一个相对安全的运行环境, 从而防止内部攻击. 而认证器的安全也是建立在计算实体内部安全性的基础上的. 所以我们要重新考虑在可信计算环境下攻击者的攻击能力及认证器的安全性.

3.1　可信环境下 CK 模型中攻击者的攻击能力

基于计算实体内部所有秘密信息都有可能被攻陷或泄漏这样的考虑, CK 模型给出针对计算实体内部的以上三种攻击方式. 这在理论上的考虑是有一定意义的, 但随着可信计算的发展, 计算实体内部的安全性得以增强, 许多针对实体内部的攻击就不可能发生. 在可信环境下用 CK 模型进行协议设计的时候, 如果还要将这些不可能发生的攻击考虑进来的话, 就会导致设计出来的密钥协商协议不够简练; 协议的分析也比较复杂(用 CK 模型对协议进行形式化分析的过程本身就比较复杂), 并有可能导致错误的产生. 因此对于可信环境能够防止的攻击在 CK 模型中就不用考虑了, 这样就简化了协议的设计与分析. 下面, 我们分别对 CK 模型给出的三类攻击进行分析.

session state reveal 是用来暴露会话状态信息的, 该攻击能力在非可信环境下是可能发生的, 如: 通过木马、后门程序等恶意软件或直接的内存访问来获得会话的状态信息. 但在可信环境下, 密钥协商协议程序本身是通过完整性检测的, 是没有被修改, 没有病毒、木马、蠕虫及恶意软件. 也即协议程序没有遭到完整性破坏, 因此它本身不会向外泄漏自己的状态信息. 另外, 可信计算也确保了进程之间不能够直接访问对方的内存. 这样别的实体也不能够得到协议进程的状态信息. 综上所述, session state reveal 在可信环境下是无效的.

对于 session key query 来说, 在协议双方完成消息的交互后, 各自计算出会话密钥, 这时会话密钥作为一个会话状态存在. 由上面分析可知, 该密钥是不可能被泄漏的. 在协议结束后, 协议执行双方如果都将得到的会话密钥存放在 TPM 中, 攻击者不可能攻破 TPM 来获得会话密钥. 但为了不受出口的限制, 在可信计算组织 TCG 给出的 TPM 结构中没有对称加解密模块, 所以在密钥协商之后的保密通信中, 会话密钥必须从 TPM 模块中取出, 在使用过程中该会话密钥就有可能被暴露. 所以在可信环境下面 session key query 攻击还是存在的. 为了防止该攻击, 建议不同国家根据自己的需要, 在 TPM 中增加对称加解密模块. 据我们所知, 中国在自己的 TPM 的设计中, 就增添了该模块.

而对于 party corruption, 由上面分析可知, 凡是和会话状态相关的信息攻击者是得不到的. 对于用户长期的秘密, 如: 签名私钥, 它有可能是由用户自己或别的实体(如: 证书权威 CA 或密钥管理中心 KMC)生成. 对于第一种情况, 用户可以利用 TPM 中的密钥生成器来产生. 在这种情况下, 签名运算都可以在 TPM 内部进行, 用户的长期私钥不会脱离 TPM, 这样用户的私钥就不会丢失. 而对于第二种情况, 这和没有引入 TPM 的情

形是一样的,所以存在 party corruption 的攻击.但在这种情况下,该攻击不能够获得实体内部的会话状态信息,而只能够得到用户的长期私钥,这样 party corruption 蜕化为一种新的攻击方式—长期私钥暴露 long-term private key corruption.下面是该能力的一个定义.

定义 2 长期私钥暴露(Long-term private key corruption):攻击者可以在会话开始前或者完成后攻陷一个实体,得到该实体的长期私钥.

该攻击和会话状态无关,发生在会话开始之前或完成之后.它和 party corruption 的区别是:在 party corruption 攻击中,攻击者可以随时对某一个实体发起该攻击,可能在协议执行过程中进行,所以有可能获得一个会话的状态信息.而长期私钥暴露则只在会话开始前或完成后由攻击者对某个实体发起该攻击,它不能够得到和会话状态相关的信息,而只能够得到用户的长期私钥.与该攻击相关的是密钥协商协议的前向保密性.

综上所述,如果用户的签名/验证公私钥对是由 TPM 生成的,则在可信环境下攻击者只有一种攻击能力:会话密钥查询(session key query).否则,在可信环境下攻击者有两种攻击能力:会话密钥查询和长期私钥暴露(long-term private key corruption).所以我们建议用户的签名/验证公私钥对应该由 TPM 来生成.

3.2 可信计算克服了 CK 模型中基于加密算法认证器的安全缺陷

Canetti 等人在 1998 年提出了一个基于加密的认证器[15],如图 2 所示.其中 sid 为会话标识符,$\{v_A\}_{K_B}$ 是公钥加密,K_B 为 B 的公钥.$MAC_{v_A}(m,A)$ 为消息认证码.

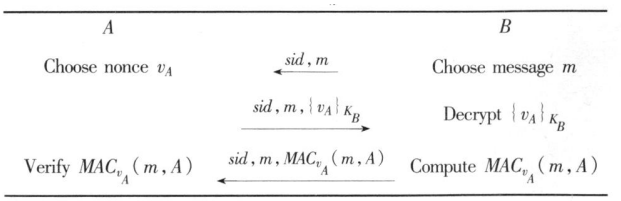

图 2 基于加密算法的认证器

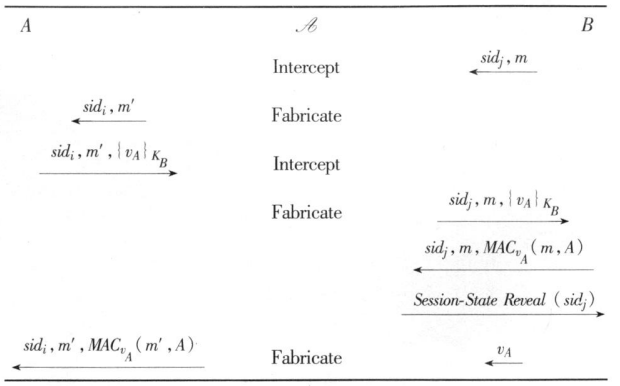

图 3 针对基于加密算法的认证器的攻击

但该认证器被指出存在安全缺陷[16].其攻击过程如图 3 所示.在 B 发起同 A 密钥协商会话(其会话标识符为 sid_j)的过程中,攻击者 $\mathscr{A}$ 同 A 进行另外一个会话 sid_i 的密钥协商,而同 B 进行 sid_j 的协商.由于 A 和 B 所完成的不是匹配的会话,即 $sid_i \neq sid_j$.所以攻击者可以将 B 上的会话 sid_j 发起 session state reveal 攻击,得到 v_A,这样攻击者就能够伪造消息 sid_i, m', $MAC_{v_A}(m', A)$.因此该认证器是不安全的.

为了避免该认证器的安全缺陷,文献[16]对其进行了改进,规定在 B 在发送消息$\{sid_j, m, MAC_{v_A}(m,A)\}$前将 v_A 从内部状态中删除.

从上面的分析可以看出,导致该安全缺陷的根本原因是在非可信环境下 session state reveal 攻击是可以实现的,而在可信计算环境下,该攻击是不存在的.所以基于加密算法认证器的安全缺陷在可信环境下自然就不存在了.

另外,即使在非可信环境下,我们也可以通过 TPM 来杜绝上述缺陷.由于 TPM 提供了公钥解密功能,可以将$\{v_A\}_{K_B}$交给 TPM 进行解密,TPM 在对它进行解密后,直接在其内部计算 $MAC_{v_A}(m,A)$(利用 SHA-1 引擎),然后将该值返回给协议会话,而不将 v_A 返回给会话(该值对协议会话没用).因此 v_A 不作为协议会话的一个状态存在.也就是说:TPM 作为一个独立的安全解密模块,计算实体内的公钥解密操作都是由 TPM 来执行,而不是由协议的实例来处理,且 TPM 只将最终值(如:$MAC_{v_A}(m,A)$)返回给会话,而不将中间值返回给会话.这样攻击者只有攻破 TPM 才能够得到解密的值.

在 CK 模型[12]第五章中基于公钥加密的协议的安全性就是基于计算实体内部存在一个独立的安全模块这一假设基础上的,该假设在 TPM 出现之前难以实现.而 TPM 的出现正好提供了这样的一个模块.这也充分说明了模型的理论研究同业界的安全考虑是完全一致的:理论研究有这样的一个安全需求,而业界的工业产品的出现也正好满足了理论上的要求.

3.3 可信环境下新的 CK 模型—CKTC

基于 3.1 和 3.2 节的分析,我们对可信环境下 CK 模型中的认证链路模型 AM、非认证链路模型 UM 及认证器 Authenticator 进行重新定义.

可信环境下的认证链路模型—AMTC(Authentication Link Module under Trusted Computation):在该模型下,如果用户公私钥对是由 TPM 本身生成,攻击者只有 session key query 能力;如果用户的公私钥对是由证书权威 CA 或密钥管理中心来产生的话则具有会话密钥查询(session key query)和长期私钥攻陷(long-term private key corruption)能力.攻击者对于网络上传输的消息只有被动攻击能力.

可信环境下的非认证链路模型—UMTC(Unauthenticated Link Module under Trusted Computation): 在该模型下,攻击者对计算实体内部的攻击能力同 AMTC 下一样,具有会话密钥查询这一种攻击能力或长期私钥暴露和会话密钥查询两种攻击能力. 但它在网络上具有主动的攻击能力,可以决定何时发送什么样的消息,可以修改消息或者任意插入自己产生的消息.

这样一来,在可信环境下设计协议时,最多只需要考虑长期私钥暴露(long-term private key corruption)和会话密钥查询(session-key query)这两种攻击能力,而不需要考虑原来 CK 模型下的那三种攻击了,从而简化了协议设计及分析的难度.

可信环境下的认证器—Authenticator under Trusted Computation: 可信环境下,TPM 克服了原 CK 模型中基于加密算法认证器的安全缺陷. 再加上 CK 模型中本来就有的基于签名的认证器、基于消息认证码的认证器,因此在可信环境中就存在这三类认证器.

另外,CKTC 模型下的安全定义同定义 1 完全一致,其安全目标同 CK 模型中的目标是一样的.

在可信计算环境下,密钥协商协议的设计仍然采用模块化的协议设计方法,首先在 AMTC 下设计一个安全的协议,然后通过可信环境下的认证器将其转化了 UMTC 下安全的协议.

4　CKTC 模型中的协议设计分析方法及其优势

在第 3 章中,我们给出了 CKTC 模型的定义,但如何利用 CKTC 来进行密钥协商协议的设计以及该形式化方法同传统的 CK 模型相比有何优势哪? 我们通过一个例子来进行说明.

首先我们给出 AM 环境下的一个密钥协商协议,证明该协议在非可信环境下是不安全的,但在 AMTC 下是安全的.

(1) AM 下的 DH 密钥协商协议[17]

协议的双方为 A 和 B,在协议执行前,双方首先协商好参数 g,以及两个大素数 p, q. 作为协议的发起方,A 首先选择随机数 x,计算 $g^x \bmod (p \times q)$,然后将 $\{A, S, g^x\}$ 发送给 B,其中 S 为会话标示符. A 将 x 作为一个会话状态保存在起来. B 在收到 A 发送过来的消息后选择随机数 y,计算 $g^y \bmod (p \times q)$ 和 $K = g^{xy} \bmod (p \times q)$,其中 K 是会话密钥. 之后 B 将 $\{B, S, g^y\}$ 发送给 A. A 在收到 B 发送过来的消息后将 x 提取出来,计算 $K = g^{xy} \bmod (p \times q)$. 其协议交互过程如图 4 所示.

在考虑到现实环境中的攻击时,该协议在 AM 下是不安全的,它不能够抵抗 session-state reveal. 随机数 x 作为 A 的一个会话状态,它有可能通过病毒、木马、蠕虫以及直接内存访问而被泄露. 一旦攻击者通过这些方式获取了 x,它就可以得到 A 和 B 协商出来的会话密钥 K 了. 这样的话该协议在 AM 下是不安全的.

(2) AMTC 下的 DH 密钥协商协议

在 AMTC 模型下,图 4 所示的协议是会话密钥安全的. 因为 AMTC 杜绝了 session-state reveal 攻击,所以攻击者就不可能够通过此攻击来得到 x 了. 这样在可信环境保障下,在 AM 模型下不安全的协议在 AMTC 模型下就变得安全了. 其具体的安全性证明可参考文献[12].

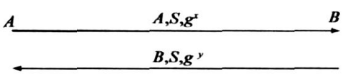

图4　AM下的DH密钥协商协议

(3) UMTC 下的 DH 密钥协商协议

利用 CK 模型中基于签名的认证器,我们将 AMTC 下的 DH 密钥协商协议转化为 UMTC 模型的下的协议,也即可信环境下现实中的协议. 其方法同 CK 模型中利用基于签名的认证器将 AM 下的协议转化为 UM 下协议的方法是一样的.

最终可信环境下的认证协议如图 5 所示,其中 $Sig_B\{B, S, g^x, g^y, A\}$ 为 B 利用自己的私钥对消息 $\{B, S, g^x, g^y, A\}$ 做的签名.

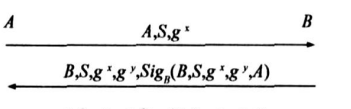

图5　UMTC下的DH密钥协协商协议

由于原协议在 AMTC 下是安全的,而且使用的是经过证明安全的认证器,根据 CKTC 可知,该协议在可信环境下面会话密钥安全的. 且 A 或 B 的长期私钥泄露对已经协商出来的会话密钥并不会造成影响. 因此在可信环境下,该协议是会话密钥安全的,且具有前向保密性 PFS.

从上面过程可以看出,由于 CKTC 模型里不存在 session-state reveal 攻击,而且 party corruption 可能蜕化为长期私钥暴露(long-term private key corruption),所以它简化了协议的设计及分析. 另外,可信计算还可通过其本身提供的增强的敏感数据保护功能来提高密钥协商协议执行时的安全性.

5　结论与下一步的工作

通过分析可信环境下 CK 模型中攻击者的三种攻击能力,我们发现在该环境下会话状态暴露(session-state reveal)攻击是不可能发生的. 但存在会话密钥查询攻击(session-key query). 另外如果用户的签名/验证公私钥对是由 TPM 生成的,则不存在实体攻陷攻击(party corruption)攻击; 否则该攻击退化为一种新的攻击方式—长期私钥攻陷攻击(long-term private key corruption). 所以可信计算简化了密钥协商协议的设计与分析. 另外,可信计算克服了原 CK 模型中基于加密算法认证器的

安全缺陷. 在此基础上, 我们提出了可信环境下的 CK 模型—CKTC. 通过一个例子, 我们给出了利用 CKTC 来进行可信环境下密钥协商协议的设计的方法. 该例子也证明了 CKTC 模型相对于 CK 模型简化了可信环境下密钥协商协议的设计与分析. 另外, 通过分析我们发现, 为了加强可信环境下密钥协商协议的安全性, 不同国家应该根据需要在 TPM 内部增加对称加解密算法; 而且用户的签名/验证公私钥对尽可能由 TPM 来生成.

我们下一步的工作打算是: 提出可信密钥协商协议的定义, 该定义从两个方面来考虑. 第一, 从协议的角度来保证其达到可证明的安全性; 第二, 从软件的角度来保证其可靠性, 而且该可靠性包括静态的可靠和动态的可靠. 并且对密钥协商协议的可信性进行分级, 按照不同场景对可信的要求来设计不同的密钥协商协议.

参考文献:

[1] 林闯, 彭雪海. 可信网络研究[J]. 计算机学报, 2005, 28(5): 751- 758.
Lin Chuang, Peng Xuehai. Research on trustworthy networks [J]. Chinese Journal of Computers, 2005, 28(5): 751- 758. (in Chinese)

[2] 郑宇, 何大可, 何明星. 基于可信计算的移动终端用户认证方案[J]. 计算机学报. 2006, 29(8): 1255- 1264.
Zheng Yu, He Dake, He Mingxing. Trusted computing based user authentication for mobile equipment[J]. Chinese Journal of Computers, 2006, 29(8): 1255- 1264. (in Chinese)

[3] S Pearson. Trusted computing: strengths, weaknesses and further opportunities for enhancing privacy[A]. In Proc. iTrust' 05 [C]. Rocquencourt: Springer Press, 2005. 305- 320.

[4] S Hilley. Trusted computing—path to security or road to servitude? [J]. Infosecurity Today, 2004. 1(4): 18- 21.

[5] D Safford, Mimi Zohar. Trusted computing and open source [J]. Information Security Technical Report 2005, 10(2): 74- 82.

[6] B Berger. Trusted computing group history[J]. Information Security Technical Report. 2005, 10(2): 59- 62.

[7] L law, A Menezes, M Qu, et. al. An efficient protocol for authenticated key agreement[J]. Designs, Codes and Cryptography. 2003, 28(2): 119- 134.

[8] A Menezes, P vanOorschot, S Vanstone. Handbook of Applied Cryptography[M]. Boca Raton, USA: CRC Press, 1996. 489- 541.

[9] E Bresson, O Chevassut, D Pointcheval. New security results on encrypted key exchange[A]. In Proc. PKC' 2004[C]. Singapore: Springer Press, 2004. 145- 158.

[10] W Aiello, S M Bellovin, M Blaze. Efficient, DoS resistant, secure key exchange for internet protocols[A]. In Proc. CCS' 2002[C]. Washington, DC, USA: ACM Press, 2002. 45- 58.

[11] D Dolev, A C YAO. On the security of public key protocols [J]. IEEE Transaction on Information Theory. 1983, 29(2): 198- 208.

[12] R Canetti, H Krawczyk. Analysis of key-exchange protocols and their use for building secure channels[A]. In Proc Eurocrypt' 2001[C]. Innsbruck (Tyrol), Austria: Springer Press, 2001. 453- 474.

[13] W Mao, H Jin, A Martin. Trusted Computing: Cryptographic Protocols Running With Inr Platform Trusted Third Party[M]. Boston, USA: Addison Wesley Press, 2008. 659- 700.

[14] TCG Specification Architecture Overview Specification, v1. 4 [OL]. https://www. Trustedcomputinggroup. org/groups/TCG_1_4_Architecture_Overview. pdf

[15] M Bellare, R Canetti, H Krawczyk. A modular approach to the design and analysis of authentication and key exchange protocols[A]. In Proc. STOC' 1998[C]. Dallas, Texas, USA: ACM Press, 1998. 419- 428.

[16] K Kwang, R Choo, C Boyd, Y Hitchcock. Errors in computational complexity proofs for protocols[A]. In Proc. Asiacrypt' 2005[C]. Chennai, India: Springer Press, 2005. 624- 643.

[17] W Diffie, M Hellman. New directions in cryptography[J]. IEEE Transaction on Information Theory. 1976, 22(6): 644- 654.

作者简介:

李兴华 男, 1978 年生于河南南阳, 博士, 西安电子科技大学副教授. 主要研究方向为无线网络安全、可信计算与协议形式化模型.
Email: lixingh@gmail.com

马建峰 男, 1963 年生于陕西西安, 西安电子科技大学计算机学院院长, 教授, 博导. 主要研究领域为计算机安全、密码学、移动与无线网络安全.
Email: jfma@mail.xidian.edu.cn

马 卓 男, 1980 年生于陕西延安, 西安电子科技大学博士生. 主要研究领域为网络与信息安全、可信计算.