# CPK真值逻辑

**南湘浩** **李益发**

鉴别逻辑有三种，一是基于模型的相信逻辑，二是基于行为的信任逻辑，三是新创建的基于证据的真值逻辑。

# 1 相信逻辑

### 1989年，M Burrows, M. Abadi and R. M. Needhanm提出了相信逻辑（belief logic）[1]。

### **相信逻辑特点**

在上面的推理证明中不难看出相信逻辑的特点：

(1)相信逻辑不能证明主体A的真实性，只能以假设主体A是可信的前提条件下才能推理证明客体的真实性。显然遵循“互相信任”的原则，直接违背网际空间安全“互相怀疑”的原则；

(2)相信逻辑把实体划分为主体和客体，按着给定的模型推理证明。但是“一切模型都是错误的，尽管有些是有用的”（有一数学家语）；

(3)现在PKI系统，由“B相信X”直接导出“B相信A”，如果这种导出成立，那么相信逻辑是在客体事件发生之后才能证明主体真实性，简称“事后证明”；

# 2 信任逻辑

2000年，由Trusted Computing Platform Alliance（TCPA，信托计算联盟）形成了基于行为的信任逻辑（trust logic）[2]，公布了Design Philosophies and Concepts Version 1.0。

信任计算平台TCP的信任逻辑有以下特点：

1. 实体分为主体和客体；

2. 主体是信托的TCP，对其他主体（软件）测量信任度，；

3. 测量与对比方式进行，得出“是”或“否”的结论；

4. 事件发生之后才能判别；

很显然，可信平台的技术路线是上世纪90年代形成的，不可能是满足网际空间安全要求的新技术路线，从理论和技术体系来说，它仍然贯穿了“基于信任”的信息安全原则，停留于信息安全年代水平。PKI的理论就是建立在信任纵向和横向扩展的理论上[3]。

**3 真值逻辑**

美国总统信息技术顾问委员会（PITAC）《网际安全－优先项目的危机》的报告准确地归结了安全原则的变化，指出：信息安全沿用的安全原则一直是“互相信任”，而网际安全的原则是“互相怀疑”(mutual suspicion)[4]。传统的“相信逻辑”、“理论可证安全”等都是在假设主体(subject)是可信的前提下，推理证明客体(object) 的真实性。这种证明方法，显然不满足新的安全原则。基于行为的信任逻辑则是根据行为跟踪的历史纪录来作出判断的，称“事后证明”，不能用于需要当场判别的场合。

于是便产生了创建新的证明逻辑的需求，即不依赖前提假设而能直接证明的“事前证明”或当场判别的证明方法，这就是基于证据鉴别的“真值逻辑”（truth logic）。真值逻辑只是用来判别真或假，结论是客观的，且没有二意性。信任不是交易的必备条件，如买卖双方互不相识，没有任何信任关系，交易照样能做成。因此，在真值逻辑中，不再使用“信任”，“可信”，“相信”等主观性概念。

真值逻辑是基于证据的证明逻辑，只能建立在基于标识的数字签名技术的基础之上。CPK刚好满足真值逻辑要求，且只有CPK才能实现，因此，将真值逻辑可称CPK鉴别逻辑。真值逻辑由实体（entity）鉴别逻辑和事件（event）鉴别逻辑构成。

### **3.1 实体鉴别**

在真值逻辑中，一个实体（entity）由实体标识（identity）和本体（body）构成。实体真实性证明是实体标识的真实性和实体本体的真实性的交

AUTH (entity) = AUTH (ID)∩AUTH (body)

**定义1：**实体标识真实性证明是实体标识对实体时间的签名，而实体标识被实体时间所验证。实体时间客观性和公认性，由此可证明本时刻的实体标识的真实性。由于标识鉴别是可以在本体出现之前独立进行，因此称“事前鉴别”。

设：实体Alice的公钥为*ALICE*，私钥为*alice*，那么

标识Alice真实性函数是标识（私钥）对时间的签名：

AUTH (Alice) = SIG*alice*(time) = sign1 = (*s*1,*c*1)

标识验证函数则是：

VER*ALICE*(time, *s*1) = *c*1’

如果*c*1 = *c*1’，验证方证明了这个时刻的实体标识Alice为真。

这是因为*ALICE* 是公钥，由标识 Alice产生，*alice*是私钥，满足 *ALICE*= (*alice*)*G*.

**定义**2：本体真实性证明是实体标识对实体本体的签名，而本体被标识所验证。由于本体鉴别总是在本体事件发生之后进行，因此称“事后鉴别”。：

AUTH (body)=SIG*alice*(CHR) = sign2 = (*s*2,*c*2) 或

AUTH (body)=SIG*alice*(MAC) = sign2 = (*s*2,*c*2)

其中，本体特征(CHR)是代表本体的特征码，如果本体是物理的，则将数据化的物理特征(指纹、照片)作为本体特征；如果本体是逻辑的，则将数据的完整性码或抽样码作为本体特征。完整性码是Hash(data) = MAC，抽样码是SAMP(data)=sam。本体鉴别在事件发生之后才能进行，因此称“事后鉴别”

本体验证函数是：

VER*ALICE*(MAC, *s*2) = *c*2’

如果*c*2 = *c*2’，则证明了本体的真实性，也证明了标识和本体的一体性。

将实体分解为标识和本体证明的方法是理论意义上的完整的证明方法，但在实际应用中，可以省去标识鉴别，只用本体鉴别代表实体鉴别。特别是在身份鉴别、防伪、票据鉴别中，标识和本体是天然绑定在一起的，总是同时发生。标识鉴别只有在基于标识的公钥体制下可省略。

### **3.2 事件鉴别**

实体的活动形成事件，事件以过程的形式而存在，而过程均纷纷为受理进程(accessing process)和采信进程(adopting process)，因此，事件真实性是受理进程真实性和采信进程真实性的交

AUTH (Procedure) = AUTH(access) ∩ AUTH(adopt)

**定义3：**受理进程的真实性函数是事件标识（identity)对事件的时间(time)的签名，受理（接入）鉴别是可以在采纳（接收）进程之前进行，因此称“事前鉴别”：

AUTH *(*accept) = SIG*identity* (time) = sign1 = (*s*1,*c*1)

对事件标识的验证函数则是：

VER*IDENTITY*(time, *s*1) = *c*1’

如果*c*1 = *c*1’，验证方证实标识是真实的，可以受理。

**定义4：**采信进程真实性函数是事件标识对数据(data)的签名，采纳鉴别是在事件发生之后才能进行，因此称“事后鉴别“：

AITH(adopt) = SIG*identity* (data)=sign2= (*s*2,*c*2)

对数据的验证函数是：

VER*IDENTITY*(data, *s*2) = *c*2’

如果*c*2= *c*2’，验证方证实数据是真实的，可以采纳。

实际上，实体鉴别和事件鉴别，其工作原理相同，但是事件的受理（接入）采纳（接受）过程的发生存在时间上的差异，不能同时发生。在需要“事前鉴别”的场合，如通信鉴别、软件鉴别等，不能省掉受理鉴别，如果没有事前鉴别的需要，则受理鉴别可以省掉。

### **4 鉴别协议**

### **4.1 通信鉴别协议**

通信鉴别由接入鉴别和采纳鉴别构成:

AUTH (comm) = AUTH (access) ∩ AUTH (adopt)

与交易鉴别相比较，接入鉴别相当于受理鉴别, 接收鉴别相当于采信鉴别。根据通信体制的不同, 接入鉴别分为离线鉴别和在线鉴别，其中离散鉴别又分为单向鉴别和双向鉴别。

1）离线鉴别

单向鉴别：用于非交互式交换中，如eMail通信等，假设Alice给Bob发送数据那么Alice要证明发送者（Alice）、接受者（Bob）真实性证明。

SIG*alice* (time) = sign1 = (*s*1,*c*1)

SIG*alice* (Bob) = sign2= (*s*2,*c*2)

Bob 检查签名：

VER*ALICE* (time,*s*1) = *c*1’

VER*ALICE* (Bob,*s*2) = *c*2’

如果*c*1 = *c*1’, *c*2 = *c*2’, 则证明了Alice,Bob, data都是真是的。

双向鉴别：假设发方电话号码为Alfa, 收方号码为Beta，那么双向鉴别是：

发方发送：

SIG*alfa* (Alfa) = sign1 = (*s*1,*c*1)

SIG*alfa* (Beta) = sign2= (*s*2,*c*2)

收方验证：

VER*ALFA*(*s*1,Alfa) = *c*1’

VER*ALFA*(*s*2,Beta) = *c*2’

收方再发送：

SIG*beta* (Beta) = sign3 = (*s*3,*c*3)

SIG*beta*(Alfa)=sign4=(*s*4,*c*4)

发送方再验证，至此双方进行了互相证明。

2）在线鉴别

在线鉴别在客户段和门户之间发生, 门户包括网站、中站、服务器等，这是DOS攻击最容易发生的地方。在线鉴别协议由客户申请，门户问答，客户相应构成。

客户申请：客户对自方user和对方portal签名

SIG*user* (user) = sign1 = (*s*1,*c*1)

SIG*user* (portal) = sign2 = (*s*2,*c*2)

门户提问：Portal在检查发方和收方真实性证明以后，对portal签名回应，同时发随机数*r*, 并等待回应

SIG*portal* (portal) = sign3 = (*s*3,*c*3)

门户发送：｛*r*. (*s*3,*c*3)}

客户相应：对*r*签名发送：

SIG*user* (*r*) = (*s*4,*c*4)

### **4.2 软件鉴别协议**

软件鉴别有两个不同情形：第一种情形是软件的发行；第二种情形是软件的调用。软件的发行包括软件的下载和安装；软件的调用包括软件的加载和执行。这两种情形的鉴别过程完全相同。软件发行真实性用于软件下载和安装控制，软件调用真实性用于软件加载和执行控制。

AUTH (issue) = AUTH (download) ∩ AUTH (install)

AUTH(invoke)=AUTH (upload) ∩ AUTH (execute)

软件下载(加载)鉴别：包括软件发行者（issuer）的真实性和软件名(name)的真实性。其中，发行者真实性证明是发行者（issuer）对时间（time）的签名与验证，即标识签名：

SIG*issuer*(time) = (*s*1,*c*1)

VER*ISSUER*(*s*1,time)=*c*1’

如果*c*1=*c*1’，那么软件发行者是真实性。

软件名的真实性是发行者对软件名的签名，即数据签名。

SIG*issuer*(name)=(*s*2,*c*2)

VER*ISSUER*(*s*1,name)=*c*2’

如果*c*2=*c*2’，那么软件名是真实的。满足以上两个条件，软件是可以下载（加载）的。

软件的安装(执行)鉴别：只检查软件体(code)真实性就可以了。软件体的真实性是软件发行者（issuer）的对软件体的签名与验证。软件体一般用压缩码mac或抽样码sam取代。

SIG*issuer*(mac)=(*s*3,c3,)

VER*ISSUER*(*s*3,mac)=*c*3’

如果*c*3=*c*3’，那么软件就允许安装（执行）。

发行者可以分为三种：一是厂家，二是集团客户，三是个人。

### **5 小结**

以上证明中可看出，真值逻辑有一下特点：

(1) 真值逻辑把实体划分为标识和本体，分别证明其真实性；把交易划分为受理进程和采纳进程，分别证明其真实性；把通信划分为接入和接收进程没分别证明其真实性，把软件划分为加载和执行，分别证明其真实性，将所有证明均划分为事前证明和事后证明。

(2) 标识鉴别可以在本体事件发生之前进行，简称“事前鉴别”；而事前鉴别是实现“自主可控”的基础理论。

(3) 真值逻辑没有前提假设（互相怀疑原则），采用示证与验证对应的证明方式；

(4) 标识鉴别可以实现“我方识别”，能够实施“非我即敌”的策略，在交易中可防止非法交易，在通信中可防止非法接入，在计算机操作中可以分离各种恶意软件，等。

(5) 标识鉴别必须由基于标识的公钥体制的数字签名实现。

(6) 真实性证明是证明方提供证据，验证方验证证据的方式进行的。

**参考文献**

[1] M Burrows, M. Abadi and R. M. Needhanm, A Logic of Authentication, Proc. R. Soc. Lond. Dec. 1989 vol. 426 no. 1871 233-271

[2] Trusted Computing Platform Alliance (TCPA TCPA Design Philosophies and Concept Version 1.0

Copyright © 2000 Compaq Computer Corporation, Hewlett-Packard Company, IBM Corporation, Intel Corporation, Microsoft Corporation

[3] Information Assurance Technical Framework, issued by National Security Agency, Information Assurance Solutions Technical Directors, Release 3.0 Sup. 2000.

[4] President’s Information Technology Advisory Committee, Cyber Security: A Crisis of Prioritization, A Report to president, Feb. 2005

[5] Xiang-hao Nan, CPK Crypto-system and Identity Authentication, Publishing House of Electronics Industry, 2012