基于 Spi 演算的 Kerberos 认证协议形式化研究*)

李国强　顾永跟　傅育熙
(上海交通大学计算机系　上海 200030)

摘　要　网络安全已成为世人关注的问题，安全协议的形式化验证显得越来越重要，基于 Spi 演算的验证是一种很好的模型检测方法。我们介绍了 Spi 演算并扩展了两个基本原语，描述和验证 Kerberos 协议的认证性，同时指出了该协议的不足之处，最后分析了基于 Spi 演算的形式化研究的今后发展方向。

关键词　Spi 演算，Kerberos 协议，测试等价

Formal Research of Kerberos Protocol Based on Spi Calculus

LI Guo-Qiang　GU Yong-Gen　FU Yu-Xi
(Computer Science Dept. Shanghai Jiaotong University, Shanghai 200030)

Abstract　With network security drawing the society's attention widely, the formalized verification of security protocols turn out to be more and more important. Verification based on Spi calculus is a commendable method of model checking. In this thesis, we introduce Spi calculus and expand its primitives, describing and verifying the authenticity of the Kerberos protocol and pointing out its demerits. Finally, we analyze and show the future developing trend of the formalized research work with the Spi calculus as its base.

Keywords　Spi calculus, Kerberos protocol, Testing equivalence

1　引言

网络数据安全主要有两个方面的内容：密码算法和安全协议。它们构成了网络安全体系的两个层次：密码算法为网络上传递的消息提供高强度的加密解密操作和其他辅助算法，而安全协议则在这些算法的基础上为各种网络安全性方面的需求提供实现方案。安全协议是基于密码算法的更高一层的算法，它为有安全需求的各方提供了一个步骤序列，以使它们能够安全地完成电子交易、身份验证或者零知识证明等过程。参与协议的各方被称为主体，协议运行过程中在网络上传输的数据称为消息。近年来，许多安全协议在提出之初被认为是足够安全的，然而却在很短的时间内被证明有漏洞。目前越来越多的安全协议不断涌现，关于协议安全性方面的讨论正在成为热点。现在国际上的研究集中在对安全协议的形式化验证方面。

π 演算是 Robin Milner 在 CCS[6]等并发计算模型的基础上提出的一种移动计算模型。它作为移动计算的基础，引入了通道的概念。通道具有确定的作用域，作用域以外的进程不能对该通道进行存取，这在一定程度上确保了通道通信的安全性。由于 π 演算在描述并发、同步、通信等方面有很大的优势，同时，π 演算提供了各种强弱不同的等价定义，因此，可以用来很准确地描述和验证协议的安全性。然而，π 演算并没有为数据加密和解密的描述提供相应的原语，因此，在用 π 演算来描述基于密码学的安全协议时，需要增加支持密码学的原语。Spi 演算[1]实现了这一原语。本文介绍了 Spi 演算，扩展了两个基本原语，并利用 Spi 演算来描述并验证安全认证协议中的 Kerberos 协议。

2　Kerberos 协议

Kerberos 协议[7,8]是为 TCP/IP 网络设计的第三方认证协议。它可以提供安全的网络认证，检查是否允许客户访问网络中的应用服务器，目前已经开发到了第 5 版。

Kerberos 协议很简明，客户从 Kerberos 服务器请求 TGS (Ticket Granting Service)票据，作为访问 TGS 服务器的凭证。Kerberos 服务器将 TGS 票据用 TGS 服务器的私有密钥加密后，发送给客户。客户将该票据提供给 TGS 服务器，然后从 TGS 服务器请求应用服务器票据。申请成功后，客户将持有的应用服务器票据发送给应用服务器，如果票据内的信息得到应用服务器确认，应用服务器便会让客户访问该服务。

为了便于描述 Kerberos 协议，本文使用了一些缩写，如下表所列：

缩写	代表内容
c	客户机名
tgs	TGS 服务器名
s	应用服务器名
a	客户的网络地址
v	票据的有效起止时间
t	时间标记
key	备用密码
K_x	x 的私有密钥
$K_{x,y}$	x 与 y 的会话密钥
$\{m\}_K$	以密钥 K 加密 m 后的密文
$T_{x,y}$	使用 y 的 x 的票据
$A_{x,y}$	从 x 到 y 的鉴别码

*)国家杰出青年科学基金资助项目(编号：60225012)。**李国强**　在读工学硕士，主要研究方向：进程演算；**顾永跟**　在读工学博士，主要研究方向：进程演算、网络安全；**傅育熙**　教授，主要研究方向：理论计算机科学、并行理论、类型理论。

Kerberos 协议的过程如下：

(1)客户到 Kerberos 服务器：客户向 Kerberos 服务器发送请求消息，该消息包括客户名及 TGS 服务器名：(c,tgs)。

(2)Kerberos 服务器到客户：Kerberos 服务器如果鉴别客户合法，便产生一个在客户和 TGS 服务器之间使用的会话密钥，并且用客户的私有密钥将之加密；同时产生 TGS 票据，用来向 TGS 证实客户的身份，并用 TGS 的私有密钥对其加密。TGS 票据的格式如下：$T_{c,tgs}=(c, a,v,K_{tgs})$。Kerberos 服务器将这两种加密的消息发送给客户：$(\{K_{c,tgs}\}K_c, \{T_{c,tgs}\}_{K_{tgs}})$。

(3)客户到 TGS 服务器：客户用自己的私有密钥解开传来的第一个加密消息，得到 Kerberos 服务器传来的它与 TGS 服务器的会话密钥；接着产生鉴别码，并用会话密钥加密。产生的鉴别码格式如下：$A_{c,tgs}=\{c,t,key\}$。同时将 Kerberos 传来的第二种加密消息一起传给 TGS 服务器，请求获取应用服务器票据。(这一项工作一般由程序自动完成，对于客户来说是透明的)，它发送给 TGS 服务器的消息如下：$(\{A_{c,tgs}\}_{k_{c,tgs}}, \{T_{c,tgs}\}_{K_{tgs}})$。

(4)TGS 服务器到客户：TGS 服务器接收到请求后，用自己的私有密钥解密 TGS 票据，然后再用票据中的会话密钥解密鉴别码。最后，TGS 比较鉴别码中的信息与 TGS 票据中的信息，如果两者之间的客户名、客户地址吻合，并且鉴别码中的时间标记在票据有效起止时间范围内，便允许处理该请求。它产生应用服务器票据，并用服务器的私有密钥加密，返回给客户。应用服务器票据的格式如下：$T_{c,s}=(c,a,v,K_{c,s})$。同时，TGS 还为客户和应用服务器产生一个新的会话密钥，此密钥由客户和 TGS 共享的会话密钥加密。然后将应用服务器名、加密后的会话密钥和应用服务器票据一起返回给用户：$(s, \{K_{c,s}\}_{K_{c,tgs}}, \{T_{c,s}\}_{K_s})$。

(5)客户到应用服务器：客户再次产生一个鉴别码，鉴别码由客户名、客户网络地址和时间标记组成，格式如下：$A_{c,s}=\{c,a,t\}$。用客户和应用服务器会话密钥对其加密后，连同应用服务器票据一起发给应用服务器：$(\{A_{c,s}\}_{K_{c,s}}, \{T_{c,s}\}_{K_s})$。应用服务器进行相关确认，通过客户认证后，就可以在应用服务器票据所规定的有效起止时间内用它与客户的会话密钥加密消息，进行会话了。

3 Spi 演算

Spi 演算[1]的项(term)定义如下：

$L,M,N::= n|x|(M,N)|0|suc(M)|\{M\}_k$

在 π 演算中，名是唯一的项。在 Spi 演算中增加了对数字的结构、密文项 $\{M\}_K$ 和对 (M,N) 的描述，这些结构并不能增加 π 演算的描述能力，引入它们的目的仅仅在于简化安全协议的描述。

进程(process)定义如下：

$P,Q,R::=\overline{M}\langle N\rangle.P|M(x).P|P|Q|(vn)P|!P|[M \text{ is } N]P|0|$

let $(x,y)=M$ in P|case M of $0:P\ suc(x):Q$|case L of $\{x\}_K$ in P

在 $(vn)P$ 中，P 中的名 n 称为是受限的(bounded)；在 $M(x).P$ 中，P 中的变量 x 是受限的；在 case M of $0:P\ suc(x):Q$ 中，变量 x 在第二个分支 Q 中是受限的。如果项中的名 n 不是受限的，则称 n 是自由的(free)。记进程 P 中所有的自由名的集合为 $f_n(P)$，所有自由变量的集合为 $f_v(P)$。如果一个进程中 $f_v(P)=\phi$，则称该进程为闭进程。

对于 Spi 演算的语义，定义了闭进程下的三种关系：

闭进程上的归约关系(reduction relation)是由如下规则定义的关系：

$$!P > P|!P$$
$$[M \text{ is } M]P > P$$
$$\text{let}(x,y)=(M,N) \text{ in } P > P[M/x][N/y]$$
$$\text{case } 0 \text{ of } 0:P\ suc(x):Q > P$$
$$\text{case } suc(M) \text{ of } 0:P\ suc(x):Q > Q[M/x]$$
$$\text{case } \{M\}_K \text{ of } \{x\}_K \text{ in } P > P[M/x]$$

结构等价(structural equivalence)是满足下列等式和规则的闭进程上的最小关系：

$$P|0 \equiv P$$
$$P|Q \equiv Q|P$$
$$P|(Q|R) \equiv (P|Q)|R$$
$$(vm)(vn)P \equiv (vn)(vm)P$$
$$(vn)0 \equiv 0$$
$$(vn)(P|Q) \equiv P|(vn)Q \text{ if } n\notin f_n(p)$$

$$\frac{P>Q}{P\equiv Q} \quad \frac{}{P\equiv P} \quad \frac{P\equiv Q}{Q\equiv P}$$

$$\frac{P\equiv Q\ Q\equiv R}{P\equiv R} \quad \frac{P\equiv Q}{P|R\equiv Q|R} \quad \frac{P\equiv Q}{(vn)P\equiv(vn)Q}$$

交互关系(reaction relation)是 Milner 在 π 演算[5]中引入的一种简单的关系，交互关系由组合在一起的两个进程 $\overline{M}\langle N\rangle.P$ 和 $M(x).Q$ 发生的一次交互引起，它的推导和规则如下：

$$\overline{M}\langle N\rangle.P|M(x).Q\rightarrow P|Q[N/x]$$

$$\frac{P\equiv P' \quad P'\rightarrow Q' \quad Q'\equiv Q}{P\rightarrow Q}$$

$$\frac{P\rightarrow Q}{P|R\rightarrow Q|R} \quad \frac{P\rightarrow Q}{(vn)P\rightarrow(vn)Q}$$

仅仅有如上的定义，还不足以清楚地表达 Kerberos 协议。于是我们对上述部分进程语义进行了扩展，定义了多元组和时间戳的概念。

在 Spi 演算的项中，只定义了对(Pair)，无法表述 Kerberos 协议中的票据和鉴别码。因此我们使用多元组(tuple)：$P=(x_1,x_2,x_3,\cdots,x_n)$，多元组可以由多个二元组嵌套而成：$P=(\cdots((x_1,x_2),x_3),\cdots,x_n)$，因此并没有改变 Spi 算的表达能力，只是为了表达方便。

同时，我们也扩展了拆对进程(pair splitting)的语义：$let(x_1,x_2,\cdots,x_n)=M$ in P，含义为如果 M 是一个多元组 $(N_1,N_2,\cdots,N_n)$，则整个进程的结果是 $P[N_1/x_1][N_2/x_2]\cdots[N_n/x_n]$，否则进程停止，等价于空进程(nil)。

为了对多元组的表述更为简洁，我们使用了下面几个简写来描述拆对进程的输入和解密：

$c(x_1,x_2,\cdots,x_n).P \triangleq c(y).let(x_1,x_2,\cdots,x_n)=y$ in P

case L of $\{x_1,x_2,\cdots,x_n\}_K$ in $P \triangleq$ case L of $\{y\}_k$ in $let(x_1,x_2,\cdots,x_n)=y$ in P

Kerberos 协议利用时间戳来保证协议的安全性。在本协议中，我们需要描述时刻(t)是否在某个有效起止时间(v)中。在文[2]中，作者提出了定义新的原语来描述时间戳，但没有定义这一原语。本文通过扩展匹配进程(match)的语义来完成时间戳的验证：$[t \text{ is } v].P$ 表示：如果 t 在 v 内，则可以进行 P 进程，否则进程阻塞。

4 Kerberos 协议的形式化

定义 Kerberos 服务器接受客户请求的通道为 c_{ker}，服务器由此通道接收客户 c 一个消息，如果消息是二元组 $(x_c; x_{tgs})$，那么通过客户的 c_{x_c} 通道送出由两上密文组成的二元组 $(\{K_{x_c,x_{tgs}}\}_{K_{x_c}}, \{T_{x_c,x_{tgs}}\}_{K_{x_{tgs}}})$。Kerberos 服务器的形式化描述如下：

$$KERBEROS \triangleq c_{ker}(x_c, x_{tgs}).\ \overline{cx_c}(((\{K_{x_c,x_{tgs}}\}K_{x_c}, \{T_{x_c,x_{tgs},K_{xtgs}}\})))$$

其中 $T_{x_c,x_{tgs}}$ 是一个四元组，由客户名称(由通道 c_{ker} 传来的变量)、客户地址、票据有效起止时间和会话密钥组成：$(x_c, a_{x_c}, v_{x_c}, K_{x_c,x_{tgs}})$。

TGS 服务器接收到二元组 (x_{cipher}, y_{cipher})，它首先用自己的私有密钥解开第二个密文，从中取得会话密钥；再解开第一个密文，如果两个密文中的几个信息吻合(使用 match 原语)，则由 c_{x_c} 通道送出一个三元组：应用服务器名、加密的会话密钥和应用服务器票据：$(s, \{K_{x_c,s}\}_{K_{x_c}}, \{T_{x_c,s}\}_{K_s})$。TGS 服务器的形式化描述如下：

$$TGS \triangleq c_{tgs}(x_{cipher}, y_{cipher}).\ case\ y_{cipher}\ of\ \{x_c, x_a, x_v, x_k\}_{K_{tgs}}\ in$$
$$case\ x_{cipher}\ of\ \{y_c, y_t, y_{key}\}_{x_k}\ in\ [x_c\ is\ y_c].[y_t\ is\ x_v].$$
$$\overline{c_{x_c}}((s, \{K_{x_c}, s\}_{x_k}, \{T_{xc,s}\}_{K_s}))$$

与 Kerberos 服务器一样，$T_{x_c,s}$ 也是一个四元组，由客户名称(由密文中得知)、客户地址、票据有效起止时间和会话密钥组成：$(x_c, a_{x_c}, v_{x_c}, K_{x_c,s})$。

应用服务器接收到二元组 (x_{cipher}, y_{cipher})，它首先用自己的私有密钥解开第二个密文，从中取得会话密钥；再解开第一个密文，如果两个密文中的几个信息都吻合则可以验证通过，让客户来使用自己的服务，应用服务器的表式化描述如下：

$$S \triangleq C_s(x_{cipher}, y_{cipher}).\ case\ y_{cipher}\ of\ \{x_c, x_a, x_v, x_k\}_{K_s}\ in$$
$$case\ x_{cipher}\ of\ \{y_c, y_a, y_t\}_{x_k}\ in\ [x_c\ is\ y_c].[x_a\ is\ y_a].[y_t\ is\ x_v].$$
$$c_s(req_{cipher}).case\ req_{cipher}\ of\ \{x_{req}\}_{x_k}\ in\ F(x_{req})$$

如果客户通过了验证，应用服务器就可以按客户的需求处理事务，这里用 $F(x)$ 来表示处理事务。

客户首先向 Kerberos 服务器发出请求，得到二元组后，用自己的私有密钥解开第一个密文，内容是与 TGS 服务器的会话密钥；然后使用这个密钥加密鉴别码，将鉴别码和得到的第二个密文发往 TGS 服务器，如果成功，得到另一个二元组；它使用与 TGS 服务器的会话密钥解开第一个密文，内容是与应用服务器的会话密钥；接着使用这个密钥加密新产生的鉴别码，将此鉴别码和第二次得到的第二个密文发往应用服务器，如果成功，则可以访问服务器。

$$C(REQ) \triangleq \overline{c_{ker}}((c, tgs)).\ c_c(x_{cipher}, y_{cipher}).\ case\ x_{cipher}\ of$$
$$\{k_{c,tgs}\}_{K_c}\ in$$
$$\overline{c_{tgs}}((\{A_{c,tgs}\}_{k_{c,tgs}}, y_{cipher})).\ c_c(x_s, z_{cipher}, w_{cipher}).$$
$$case\ z_{cipher}\ of\ \{k_{c,x_s}\}_{kc,tgs}\ in\ \overline{c_s}((\{A_c, x_s\}_{k_{c,x_s}}, w_{cipher})).\overline{c_{x_s}}$$
$$(REQ)$$

这里，$A_{c,tgs}$ 是一个三元组(c,t,key)，包括客户名(c)，当前时刻(t)和一个备用密码 key，在我们描述的系统中，备用密码并没有得到使用。$A_{c,s}$ 也是一个三元组(c,a,t)，包括客户名(c)，客户地址(a)和当前时刻(t)。REQ 代表客户对应用服务器的申请，通过应用服务器的通道发给应用服务器，以求得到应用服务器的响应。

除了各个密钥是保密以外，所有的通道都是公开的。因此，整个具有 Kerberos 协议的系统可以有如下定义。在这里，我们已经简化多个 TGS 服务器和应用服务器为一个(这一简化并不影响后面的验证)。假设这个系统需要处理 n 个客户请求 $REQ_1, REQ_2, \cdots, REQ_n$，则系统表示如下：

$$Sys(REQ_1, REQ_2, \cdots, REQ_n) \triangleq (\nu K_c)(\nu K_s)(\nu K_{tgs})$$
$$(\nu K_{c,tgs})(\nu K_{c,s})$$
$$(!KERBEROS\,|\,!TGS\,|\,!S\,|\,\Pi_{i\in 1,\cdots,n}C(REQ_i))$$

5 Kerberos 协议认证性(authenticity)的验证

我们对协议的认证性有如下定义：只有合法的客户申请服务，服务器才能提供；任何一个攻击者不能使服务器提供未经合法申请的服务。用 Spi 来验证一个协议的认证性，我们需要首先定义该协议规范(specification)，然后证明对于任意的客户请求，形式化描述的协议和协议规范满足测试等价(testing equivalence)关系：

对于任意的请求 $REQ_1, REQ_2, \cdots, REQ_n$，都有 $Sys(REQ_1, REQ_2, \cdots, REQ_n) \simeq Sys_{spec}(REQ_1, REQ_2, \cdots, REQ_n)$。

5.1 Kerberos 协议的规范

对于 Kerberos 服务器和 TGS 服务器来说，它们只负责认证的作用，并没有信息的传递。因此规范与形式化描述相同：

$$KERBEROS_{spec} \triangleq KERBEROS$$
$$TGS_{spec} \triangleq TGS$$

对于应用服务器和客户双方有信息传递，因此我们可以这样定义两者的规范，客户并没有发送 REQ 到服务器，它只是发送了一个私有通道 p；服务器收到信息以后，从该通道发送一个信号 TRI，客户接收此信号后就作 $F(REQ)$，因此，服务器和客户的规范如下：

$$S_{spec} \triangleq c_s(x_{cipher}, y_{cipher}) \cdot case\ y_{cipher}\ of\ \{x_c, x_a, x_t, x_k\}_{K_s}\ in$$
$$case\ x_{cipher}\ of\ \{y_c, y_a, y_t\}_{x_k}\ in\ [x_c\ is\ y_c] \cdot [x_a\ is\ y_a] \cdot [x_t\ is\ y_t].$$
$$c_s(ch_{cipher}).\ case\ ch_{cipher}\ of\ \{p\}_{x_k}\ in\ \overline{p}\langle TRI\rangle$$
$$C_{spec}(REQ) \triangleq (\nu p)(C(p)\,|\,p(x).F(REQ))$$
$$sys_{spec}(REQ_1, REQ_2, \cdots, REQ_n) \triangleq (\nu p)(\nu K_c)(\nu K_s)$$
$$(\nu K_{tgs})(\nu K_{c,tgs})(\nu K_{c,s})$$
$$(!\,KERBEROS_{spec}\,|!\,TGS_{spec}\,|!\,S_{spec}\,|\,\Pi_{i\in 1,\cdots,n}\,C_{spec}(REQ_i))$$

5.2 测试等价

为了定义测试等价，首先需要定义一个操作子，描述进程可以与外界发生交互的能力。我们定义闭进程 $P \downarrow \beta$，如果 m 是自由名，并且 m 是 P 可以与外界交互的通道，也即：

$$m(x).Q \downarrow m \quad \overline{m}\langle M\rangle.Q \downarrow \overline{m}$$

如果 P 作了若干的动作以后变成 P'，并且 $P' \uparrow \beta$，则有 $P \Downarrow \beta$。因此，显然有：

$$\frac{P \downarrow \beta}{P \Downarrow \beta} \qquad \frac{P \to Q \quad Q \Downarrow \beta}{P \Downarrow \beta}$$

我们将二元组(R,β)定义为一个测试，这个二元组是由一个闭进程 R 和一个通道 β 组成，则有：

测试等价 $P \simeq Q$ 可定义为对于任意的测试(R,β)，$(P|R) \Downarrow \beta$ 当且仅当 $(Q|R) \Downarrow \beta$。

5.3 Kerberos 协议的认证性及缺陷

在这里，我们仅对 Kerberos 协议的认证性的证明作一简

要的说明。

对于任意的测试(R,β)，如果有$(Sys(REQ_1,REQ_2,\cdots,REQ_n)|R)\Downarrow\beta$，则$\beta$通道必定属于$Sys(REQ_1,REQ_2,\cdots,REQ_n)$或$R$，而不是两者作内部动作以后出现的新通道。这是因为有时间戳的验证$[y_i\ is\ x_v]$，任何非合法客户利用公有通道发出的消息都不能通过时间戳的检验，而使得进程阻塞。由于$Sys_{spec}(REQ_1,REQ_2,\cdots,REQ_n)$可以和$Sys(REQ_1,REQ_2,\cdots,REQ_n)$有相同的外部通道，则$(Sys_{spec}(REQ_1,REQ_2,\cdots,REQ_n)|R)\Downarrow\beta$。反之亦然，因此有$sys(REQ_1,REQ_2,\cdots,REQ_n)\simeq Sys_{spec}(REQ_1,REQ_2,\cdots,REQ_n)$。

但是，上面的测试等价是建立在时间戳的确起作用的基础上，如果R可以传递一个消息满足时间戳验证，则Kerberos协议是不安全的，它无法阻止重放(replay attack)。对于形式化而言，如果我们去掉$[x_i\ \text{is}\ y_i]$匹配项，则两者不是测试等价的，下面定义的一个测试可以说明这个问题：

我们首先假设$F(x)\triangleq\overline{c_t}\langle x\rangle$，其中$c_t$是一个新通道。令

$$R\triangleq c_c(u).\overline{c_c}\langle u\rangle.\overline{c_c}\langle u\rangle.c_t(x).c_t(y)[y\ \text{is}\ x].\overline{d}\langle *\rangle$$

则有$(Sys(REQ_1,REQ_2,\cdots,REQ_n)|R)\Downarrow d$，但没有$(sys_{spec}(REQ_1,REQ_2,\cdots,REQ_n)|R)\Downarrow d$，所以

$$Sys(REQ_1,REQ_2,\cdots,REQ_n)\not\simeq Sys_{spec}(REQ_1,REQ_2,\cdots,REQ_n)$$

结束语 Spi演算为认证协议的描述和论证提供了很好的支持，但这一领域还需要作更深入的研究和应用。

第一，在理论上，我们可以看到Spi的语义还不足以描述复杂的认证协议，如时间戳的描述。虽然在本文中使用了匹配(match)来模拟时间戳，但无法做到对其精确的描述，尚存在一些语义上的缺陷。为此，我们应该提出高阶(high order)的Spi演算来解决这个问题。

第二，为了把Spi演算应用于实际，我们需要结合Model checking的思想，来研究制作自动验证工具，利用这个工具来验证现存的和即将制订的协议的安全性，才是本研究所要达到的目的。

参考文献

1 Abadi M, Gordon A D. A calculus for cryptographic protocols: The spi calculus. In: the Proc. of the Fourth ACM Conf. on Computer and Communications Security, 1997

2 Abadi M, Gordon A D. A calculus for cryptographic protocols: The spi calculus: [Technical Report 414]. University of Cambridge Computer Laboratory, 1997

3 Abadi M, Gordon A D. Reasoning about cryptographic protocols in the spi calculus. In: CONCUR'97: Concurrency theory, volume 1243 of Lecture Notes in Computer Science, 1997. 59～73

4 Abadi M, Gordon A D. A Bisimulation Method for Cryptographic Pro-tocols. Nordic Journal of Computing, 1998, 5(4): 267～303

5 Milner R, Parrow J, Walker D. A calculus of mobile processes, Parts I and Ⅱ. Information and computation, 1992

6 Milner R. Communication and Concurrency. Prentice-Hall International, 1999

7 Neuman B C, Ts'o T. Kerberos: An Authentication service for com-puter network. IEEE Communications Magazine, 1994

8 Schneier B. Applied Cryptography Second Edition: Protocols, algorithms and source code in c. Wiley, 1996

(上接第3页)

需要说明的是，这里的"逻辑值"与数学中的"概率值"是两个不同的概念，前者是基于泛逻辑原理而新提出的概念，主要用于复杂系统中对包含不确定性因素的参量进行柔性推理与精确控制；后者则是熟知的概率论中的概念；尽管二者有一定的联系，都从概率密度与分布函数进行求解，但二者的求解思想与计算方法是截然不同的。那么，已经有了概率论，为什么还要研究其泛逻辑的推理呢？范例的计算表明：$P(X<3.12)+P(X\geqslant3.12)=1$，其物理意义是：随机变量$X$落在区间$(-\infty,3.12)$的概率是0.5239，则落在区间$[3.12,\infty]$的概率必然是1－0.5239，其本质仍然是经典逻辑的推理，即变量值如不属于集合$(-\infty,3.12)$，则必然属于集合$[3.12,\infty]$。但是，由$Q(3.12)+\sim Q(3.12)\neq1$可知，基于泛逻辑的考虑了广义自相关性系数$k$的这种推理模式，能更好地描述变量$X$的不确定性，当且仅当$k=0.5$时存在$Q(x)+\sim Q(x)=1$。

小结 正分布是随机系统中许多参数所服从的一种分布规律，泛逻辑学为描述不确定性问题提供了新的思路和方法。文章在泛逻辑N范数和广义自相关性概念的基础上，主要针对正态分布随机变量的泛非运算性质进行了研究，其工作内容与意义在于：

(1)给出了正态分布参量的N范数、N性生成元、广义自相关性系数；建立了分布函数$F(x)$与广义自相关性系数k之间的重要关系式，这有助于加速泛逻辑在不确定性系统推理中的应用。

(2)拓展了泛逻辑中k值的用途，它不仅可以用于泛逻辑中各连接词的运算；也可以作为一个评价指标，从逻辑推理的角度，对一个数学模型的误差进行评估。计算表明：标准正态分布的数学模型也不是完美的，其广义自相关性系数$k<0.5$，在此模型中，所描述随机参量的逻辑值呈偏小估计。

参考文献

1 朱燕堂，赵选民，徐伟. 应用概率统计方法[M]. 西北工业大学出版社，2000

2 盛骤，谢式千，潘承毅. 概率论与数理统计[M]. 高等教育出版社，2001

3 何华灿，王华，刘永怀，等. 泛逻辑学原理[M]. 北京科学出版社，2001

4 王拥军. 需求工程中的不确定性研究[D]：[西北工业大学博士学位论文]. 2001

5 何华灿，刘永怀，何大庆. 经验性思维中的泛逻辑[J]. 中国科学(E辑)，1996，126(1)：72～78

6 He Huacan, Ai Lirong, Wang Hua. Uncertainties and the Flexible Logics[C]. In: IEEE Proc. of 2003 Intl. Conf. on Machine Learning and Cybernetics, vol4/5, Xi'an, 2003, 11: 2573～2578

7 Chen Zhicheng, He Huacan, Mao Mingyi. Correlation Reasoning of Complex System Based on Universal Logic[C]. In: IEEE Proceedings of 2003 International Conference on Machine Learning and Cybernetics, vol3/5, Xi'an, 2003, 11: 1831～1835

8 刘永怀. 基于广义范数的不确定性推理理论研究[D]：[西北工业大学博士学位论文]. 1996