典型安全网关的形式化设计与证明

王瑞云 赵国磊 常朝稳 王雪健
(中国人民解放军信息工程大学密码工程学院 郑州 450001)

摘 要 传统上依靠经验设计的安全网关侧重于功能实现,缺少严格的安全模型。对此,针对一种典型安全网关,首先根据其安全需求给出相应的安全策略,然后利用BLP模型对给出的安全策略进行形式化建模并对安全模型的内部一致性进行证明,最后对安全网关的功能规约和安全模型的一致性进行验证。为保证推理过程的正确性,使用定理证明器Isabelle/HOL对上述过程进行描述和推理,保证了安全网关顶层设计的安全性。研究结果为安全网关的形式化设计提供了一定的借鉴意义。

关键词 典型安全网关,形式化设计,BLP模型,功能规约,一致性验证,Isabelle/HOL
中图法分类号 TP309.1 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.09.028

Formal Design and Verification for Typical Security Gateway

WANG Rui-yun ZHAO Guo-lei CHANG Chao-wen WANG Xue-jian
(Department of Password Engineering, The PLA Information Engineering University, Zhengzhou 450001, China)

Abstract Security gateway which is designed in experience usually focuses on function implementation and is usually not designed according to security model. To solve this problem, we proposed a method of formally designing and verifying a typical security gateway. Firstly, we designed the typical security gateway's security policy according to its security requirements. Secondly, we formally modeled the security policy and verified the security model's internal consistency by means of BLP model. In the end, we verified the consistency between the security gateway's functional specifications and its security model. To make sure the reasoning procedure's correctness, we used the theorem prover Isabelle/HOL to formally describe the above work and help us deduce. Our work ensures the security of a typical security gateway in terms of its top-level design and plays certain referential significance on formal design of security gateway.

Keywords Typical security gateway, Formal design, BLP model, Functional specification, Consistency verification, Isabelle/HOL

1 引言

为安全连接内外网,通常需要在两个网络之间部署具有加密功能的安全网关,常见的有IPSEC VPN[1],SSL VPN[2]等设备,这些设备有效提高了网络的整体安全性。但是依靠传统经验开发的网关产品存在诸多问题,以很多安全网关采用的开源安全组件openssl为例,近几年出现的openssl心脏滴血漏洞[3]导致很多安全网关受到致命攻击,造成了大量信息泄露,一个主要原因是模块之间缺乏形式化技术保证的严格隔离性,一个模块的安全漏洞严重影响了其他模块的安全性。传统上依靠经验设计的安全网关缺乏严格的安全模型,而系统能否达到高安全级别,取决于对安全控制的设计和实施投入的精力,若对系统的安全需求不了解,即使运用最好的软件技术,投入最大的精力,也很难满足安全要求。安全模型的目的在于明确地表达这些需求,为设计开发安全系统提供坚实基础。Pierre Bieber等[4]描述了改进FOX安全网关过程中BLP模型[5]和B方法[6]等形式化技术的使用,但因其着眼于FOX安全网关ITSEC-E4级别[7]的评估,因此这些方法并不适用于安全网关的形式化设计和证明。目前,国内对安全网关的形式化设计与验证工作相对较少。因此,本文提出对一种典型安全网关的安全策略进行形式化建模,并对其功能规约与安全模型的一致性进行形式化验证的方法。

首先介绍一些重要的、有关形式化设计的基本概念和术语[8]。1)安全策略是针对系统面临的安全威胁所采取的应对办法,包括有关管理、保护和发布敏感信息的法律、规定和实施细则。安全策略由一整套严密的规则组成,这些确定授权访问的规则是决定访问控制的基础。2)安全模型是对安全策略所表达的安全需求的简单、抽象和无歧义的描述,它为安全策略和安全策略实现机制的关联提供了一种框架。虽然目前经典的信息安全模型有BLP模型[5]、Biba模型[9]、安全信息

到稿日期:2017-04-28 返修日期:2017-06-13 本文受面向用户的可信云计算环境安全研究(61572517)资助。
王瑞云(1992-),女,硕士生,主要研究方向为信息安全,E-mail:wry0068@126.com(通信作者);赵国磊(1979-),男,博士,讲师,主要研究方向为信息安全、形式化验证,E-mail:glz0371@163.com;常朝稳(1966-),男,博士,教授,博士生导师,CCF会员,主要研究方向为信息安全,E-mail:ccw@xdja.com;王雪健(1993-),男,硕士生,主要研究方向为信息安全、形式化验证,E-mail:1368154434@qq.com。

流模型[10]和无干扰安全模型[11]，但由于大多数工程项目人员对 BLP 模型更为熟悉，且 BLP 模型本身较直观且易于理解，因此本文利用 BLP 模型对典型安全网关的安全策略进行形式化建模。3)形式化规约的目标在于使用一种简洁、无二义且可被计算机处理的方式描述系统的功能行为。

一般来说，形式化验证可分为两个基本阶段：形式化安全策略模型和形式化规约之间的一致性验证，以及形式化规约和系统具体代码实现之间的对应性验证[8]。本文研究工作基于第一个基本阶段，第 2 节给出安全网关的安全策略；第 3 节基于 BLP 模型的访问控制规则，在 Isabelle/HOL[12] 中给出安全模型；第 4 节给出安全网关功能规约与安全模型的一致性验证；第 5 节总结全文，并对进一步的研究方向进行展望。

2　安全网关的安全策略

安全网关是基于 TCP/IP 协议的网关[13-14]，用于连接外网和内网，保护内网数据的机密性和完整性。本文研究的安全网关采用嵌入式物理平台，其逻辑组成结构如图 1 所示。其中，安全芯片是组件的硬件平台；微型操作系统提供基本的任务调度与存储管理功能；配置模块负责整个组件的参数配置；过滤模块对通信模块产生的数据包依据配置规则(端口、IP 地址等)进行过滤，确保只有符合要求的通信数据包才能向下一个任务模块传递；数据加解密模块完成明密文之间的转换，同时还进行基于序列号的重放攻击检测、数据包完整性检测，以确保由外网进入的数据包中只有合法数据包能被正确解密并被转发到内网；内网发送出去的所有数据包需经过密码算法的正确处理，以确保信息的机密性。因此数据加解密模块的实质是过滤器。

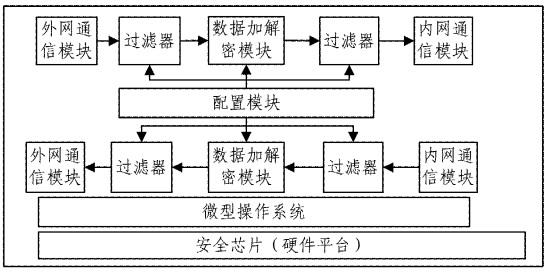

图 1　典型安全网关的逻辑组成结构

图 1 示出了信息由外网进入内网和由内网流向外网的流动情况。由外网进入的信息首先被外网通信模块根据 TCP/IP 协议进行处理，经处理后的信息流向过滤器和数据加解密模块进行过滤；最后，过滤得到的合法数据包被内网通信模块处理后进入内网。信息由内网发送到外网的流动情况与此类似。

为了确保安全网关对内外网交换的信息进行过滤，需要满足以下两个安全需求。

(1)信息隔离：在安全网关内部，来自内网的信息与来自外网的信息应相互隔离，即没有模块既可以处理来自外网的数据又可以处理来自内网的数据。

(2)信息过滤：由外网到内网或由内网到外网的信息必须经过安全网关内部过滤器的过滤。

为了满足上述安全需求，给出典型安全网关的安全策略，以由外网进入内网的信息流动方向为例进行说明。首先定义赋予主体或客体的安全级别以及安全级别间合法的信息流(本文的信息流是指两个安全级别间信息的流动)，如下所示。

high：具有高保密性的安全级别，来自内网的信息应具有级别 high。

low：具有低保密性的安全级别，来自外网的信息应具有级别 low。

middle：安全网关配置信息的安全级别。

为满足信息隔离的安全需求，安全级别 high 和 low 之间的信息流非法；流向自身级别的信息流合法；由 middle 流向 high 或 low 级别的信息流合法，但是，级别 high 或 low 流向 middle 的信息流非法，否则信息可以通过级别 middle 间接地由级别 high 流向级别 low。合法信息流如表 1 所列(顺序为第一列的各级别流向第一行的各级别；1 表示安全级别间的信息流合法，空表示非法)。

表 1　满足信息隔离的合法信息流

	high	low	middle
high	1		
low		1	
middle	1	1	1

根据信息过滤的安全需求，来自外网的信息被过滤后向内网传递，即级别 low 流向级别 high 之间的信息流合法，这与信息隔离的安全需求冲突。为了解决这个问题，本文重新定义安全级别，其由隔离级别(high，low，middle)和过滤级别(in，ok，out)两部分组成，例如：

〈low，in〉是来自外网且待过滤的信息的级别。

〈low，ok〉是来自外网且已过滤的信息的级别。

〈high，out〉是经过过滤且待转发至内网的信息的级别。

表 2 列出了所有合法信息流，其中〈low，ok〉到〈high，out〉是从〈low，?〉到〈high，?〉的唯一合法信息流(其中，?代表过滤级别)。

表 2　满足信息隔离的合法信息流

	〈low，in〉	〈low，ok〉	〈high，out〉	〈middle，?〉
〈low，in〉	1			
〈low，ok〉		1	1	
〈high，out〉			1	
〈middle，?〉	1	1	1	1

为了实现信息过滤的目标，应禁止级别〈low，in〉到级别〈low，ok〉的信息流动，否则，待过滤的信息会流向已过滤的信息。对此，为过滤器引入过滤级别 f_{fi} 和 f_{tf}。

〈low，f_{fi}〉是实现过滤功能的主体的级别，该级别不赋予其他主体或客体。

〈low，f_{tf}〉是待过滤器过滤的信息的级别。

显然，由级别〈low，f_{tf}〉到级别〈low，f_{fi}〉的信息流为合法信息流。记安全网关的第一个过滤器为 $f1$，第二个过滤器为 $f2$，为满足信息过滤的安全需求，由级别〈low，in〉流向级别〈low，$f1_{tf}$〉的信息流合法，由级别〈low，$f1_{tf}$〉流向〈low，$f1_{fi}$〉的信息流合法。如果 $f2$ 是最后一个过滤器，那么，由级别〈low，$f2_{fi}$〉流向级别〈low，ok〉的信息流合法。表 3 列出满足信息隔离和信息过滤的合法信息流。

表 3 满足信息隔离和信息过滤的合法信息流

	⟨low,in⟩	⟨low,$f1_{tf}$⟩	⟨low,$f1_{fi}$⟩	⟨low,$f2_{tf}$⟩	⟨low,$f2_{fi}$⟩	⟨low,ok⟩	⟨high,out⟩	⟨middle,?⟩
⟨low,in⟩	1	1						
⟨low,$f1_{tf}$⟩		1	1					
⟨low,$f1_{fi}$⟩			1	1				
⟨low,$f2_{tf}$⟩				1	1			
⟨low,$f2_{fi}$⟩					1	1		
⟨low,ok⟩						1	1	
⟨middle,?⟩	1	1	1	1	1	1	1	1

基于上述安全级别和合法信息流的定义，给出安全网关满足信息隔离和信息过滤两个安全需求的安全策略：将安全网关的模块看作主体，将模块完成特定功能涉及到的数据流看作客体，并赋予主、客体不同的安全级别，如果主体能读访问客体，当且仅当客体的安全级别流向主体的安全级别合法；如果主体能写访问客体，当且仅当主体的安全级别流向客体的安全级别合法。第 3 节将利用 BLP 模型，在定理证明辅助工具 Isabelle/HOL 中给出表示安全策略的安全模型。

3 安全模型

本节利用 BLP 模型的访问控制规则对第 2 节给出的安全网关的安全策略进行形式化建模，并对形式化安全模型的内部一致性进行证明。为保证推理的正确性，对形式化模型的描述和内部一致性验证的工作均在高阶逻辑辅助证明系统 Isabelle/HOL 中进行。

3.1 Isabelle/HOL 符号系统[12]

Isabelle 是一个实现逻辑形式化的通用系统，Isabelle/HOL 是 Isabelle 应用于 HOL(Higher Order Logic)的规格化系统，采用函数式编程(functional programming)的方式提供交互式的验证环境。HOL 是一种类型逻辑，类型变量(type variable)可以采用$'a$，$'b$ 等表示；类型的项(term)$x::'a$ 表示变量 x 是类型$'a$ 的项。一般的 HOL 数据类型可用 datatype 定义，如定义隔离级别类型为 datatype LEVELi=low|high|middle，表示隔离级别由 3 种级别 low，high 和 middle 定义。定义一个类型的别名可用 type_synonym 命令，如 type_synonym LEVEL=" LEVELi * LEVELf"表示 LEVEL 是类型 LEVELi * LEVELf 的别名。构造非递归定义可用 definition 命令，如定义合法信息流为 definition flow::"(LEVEL * LEVEL) set" where "flow=={K. fst K=snd K}"，其中形式"{s. Ps}"定义了一个集合，其所有的元素满足谓词公式 P。声明常元而不予定义可用 consts 命令，如定义活跃客体的集合为 consts obj_active::"OBJECT set"。声明新的类型而不加以定义可用 typedecl 命令，如声明客体类型为 typedecl OBJECT。表示属性可利用命令 axiomatization 将属性声明为一个公理，如声明活跃客体集合与已删除客体集合不相交 axiomatization where obj_not_interaction [simp]:"obj_active∩obj_zombie ={} "。

3.2 BLP 模型[5]

Bell 和 LaPadula 于 1973 年提出并于 1976 年整合完善的 BLP 模型是最早的也是目前最流行的保密性访问控制模型。它从保密性策略的角度出发，结合了强制访问控制(MAC)和自主访问控制(DAC)，通过禁止上读下写操作有效地防止了主体读取安全级别比它更高的客体。

3.2.1 BLP 模型的定义

$S=\{s_1,s_2,\cdots,s_m\}$ 是主体的集合，主体是用户、进程等，$O=\{o_1,o_2,\cdots,o_n\}$ 是客体的集合，客体是数据、文件、程序、设备等。BLP 模型是一种状态机模型，状态 $v\in V$ 由一个有序四元组(b,M,f,H)表示，其中：

1)$b\in(S\times O\times A)$表示在某个特定状态下，哪些主体以何种访问属性访问哪些客体，其中 S 是主体的集合，O 是客体的集合，$A=\{r,w,a,e\}$是访问属性集；

2)M 为访问控制矩阵，表示系统中所有主体对所有客体所拥有的访问权限，其中元素 M_{ij} 是主体 S_i 对客体 O_j 的访问权限；

3)$f\in F$ 表示访问类函数，记作 $f=\{f_s,f_o,f_c\}$，其中，f_s 表示主体的安全级函数，f_c 表示主体当前安全级函数，f_o 表示客体的安全级函数；

4)H 表示当前的层次结构，即当前客体的树型结构，$O_j\in H(O)$表示在此树型结构中 O_j 为子节点，O 为父节点。

3.2.2 BLP 模型的公理

每一个主体都有一个安全许可，每一个客体都有一个安全密级。设$L=(l_1,l_2,\cdots,l_q)$是主体或客体的密级(密级从高到低有绝密、机密、秘密、公开等)。设 $L(S)$是主体 S 的安全许可，而 $L(O)$是客体 O 的敏感等级，BLP 模型满足：

1)简单安全属性(ss-property)。S 可以读 O，当且仅当 $L(S)\leqslant L(O)$，且 S 对 O 具有自主型读权限，即一个主体只能读不高于自身安全级别的客体。

2)*-属性(star-property)。S 可以写 O，当且仅当 $L(O)\leqslant L(S)$，且 S 对 O 具有主动型写权限，即一个主体只能写不低于自身安全级别的客体。

3)基本安全定理。设系统 Σ 的某一个初始状态为 σ_0，T 是状态转换的集合，那么 T 的每个元素都遵循简单安全属性和*-属性，因此对于每个 $j\geqslant 0$，状态 $\sigma_j(j=0,1,2,\cdots)$都是安全的。

3.3 安全模型在 Isabelle/HOL 中的描述

为了形式化描述安全网关的安全策略，创建理论 Formal Model，其公理部分对应 BLP 模型的安全属性(security property)，操作部分对应 BLP 模型中主客体创建删除、当前级别改变、权限赋予等规则。下面以赋权和创建客体两个操作为例进行说明，具体验证过程以代码注释形式(* *)给出。

```
theory Formal Model
imports Main
begin
typedecl OBJECT (*定义客体*)
typedecl SUBJECT (*定义主体*)
  datatype LEVELi = low|high|param (*定义隔离级别*)
  datatype LEVELf=out|inn|ok|f1fi|f1tf|f2fi|f2tf (*定义过滤级别*)
  type_synonym LEVEL=" LEVELi * LEVELf"
  datatype RIGHT = obs|alt (*定义权限=读|写*)
  definition flow::"(LEVEL * LEVEL) set" where"flow == {K. fst K = snd K}∪{K. fst(fst K) = param} ∪ {((low,inn),(low,f1tf)), ((low,f1tf),(low,f1fi)),((low,f1fi),(low,f2tf)),((low,
```

```
f2tf),(low, f2fi)),((low, f2fi),(low, ok)),((low, ok),(high, out))}"(*定义合法信息流*)
  consts obj_active::"OBJECT set"(*定义活跃客体的集合*)
  consts obj_zombie::"OBJECT set"(*定义已删除客体的集合*)
  consts subj_active::"SUBJECT set"(*定义活跃主体的集合*)
  consts subj_zombie::"SUBJECT set"(*定义已删除主体的集合*)
  consts origin::"(SUBJECT * LEVEL)set"(*赋予主体固有级别*)
  consts c_level::"(SUBJECT * LEVEL)set"(*赋予活跃主体当前级别*)
  consts clasf::"(OBJECT * LEVEL)set"(*赋予活跃客体范畴级别*)
  consts c_access::"((SUBJECT * OBJECT) * RIGHT)set"(*主体对客体拥有读|写权限*)
(*************公理部分*************)
  axiomatization where obj_not_interaction[simp]:"obj_active ∩ obj_zombie={}"(*活跃客体集合与已删除客体集合交集为空*)
  axiomatization where subj_not_interaction[simp]:"subj_active ∩ subj_zombie={}"
  definition subj_obj1::"(SUBJECT * OBJECT)set" where "subj_obj1 == {K. ∃m. m∈c_access∧fst m = K ∧snd m = obs }"(*主体对客体有读权限的(主体×客体)集合*)
  definition subj_obj0::"(SUBJECT * OBJECT)set" where "subj_obj0 == {K. ∃m n. m∈origin∧n∈clasf∧(fst m ,fst n) = K ∧(snd n ,snd m) ∈ flow}"(*客体范畴级别流向主体固有级别合法的(主体×客体)集合*)
  axiomatization where obs1[simp]:"subj_obj1⊆subj_obj0"(*如果主体对客体有读权限,那么由客体的范畴级别到主体的固有级别的信息流应是合法信息流,对应 BLP 模型中的简单安全属性*)
  definition subj_obj00::"(SUBJECT * OBJECT)set" where "subj_obj00 == {K. ∃m n. m∈c_level∧ n∈clasf∧ (fst m ,fst n)=K∧(snd n ,snd m) ∈ flow }"(*客体的范畴级别流向主体的当前级别合法的(主体×客体)集合*)
  axiomatization where obs2[simp]:"subj_obj1⊆subj_obj00"(*如果主体对客体有读权限,那么由客体的范畴级别到主体的当前级别的信息流应是合法信息流,对应 BLP 模型中的*-属性*)
  definition subj_obj2::"(SUBJECT * OBJECT)set" where "subj_obj2 == {K. ∃m. m∈c_access∧fst m=K∧snd m = alt}"(*主体对客体有写权限的(主体×客体)集合*)
  definition subj_obj000::"(SUBJECT * OBJECT)set" where "subj_obj000== {K. ∃m n. m∈c_level∧n∈clasf∧(fst m ,fst n)=K∧(snd m ,snd n) ∈ flow}"(*主体的当前级别流向客体的范畴级别合法的(主体×客体)集合*)
  axiomatization where alt[simp]:"subj_obj2⊆subj_obj000"(*如果主体对客体有写权限,那么由主体的当前级别到客体的范畴级别的信息流应是合法信息流,对应 BLP 模型中的*-属性*)
(*************操作部分*************)
  definition get_access::"SUBJECT⇒OBJECT set⇒OBJECT set⇒((SUBJECT * OBJECT) * RIGHT)set" where "get_access ss oo_obs oo_alt =
  (let subj_obj3={ss}×oo_obs;
    subj_obj4={ss}×oo_alt
  in (
    if (ss∈subj_active) ∧ (oo_obs⊆obj_active)∧ (oo_alt⊆obj_active)∧(subj_obj3⊆subj_obj0)∧(subj_obj3⊆subj_obj00)∧(subj_obj4⊆subj_obj000)
    then
        c_access∪ subj_obj4×{alt}∪subj_obj3×{obs}
    else
        c_access
  ))"(*如果满足条件(ss∈subj_active)∧(oo_obs⊆obj_active)∧(o o_alt⊆obj_active )∧(subj_obj3⊆subj_obj0)∧(subj_obj3⊆subj_obj00)∧(subj_obj4⊆subj_obj000),那么赋权成功,否则赋权失败*)
  definition creat_object::"SUBJECT⇒OBJECT set⇒LEVEL⇒OBJECT set" where "creat_object_ok ss oo nn =
    ( if (ss∈subj_active)∧(oo ∩ obj_active={})∧(o o∩obj_zombie = {})∧(∃K. K∈c_level∧ (fst K = ss)∧{snd K} ×{nn}⊆flow)
    then
        obj_active∪oo
    else
        obj_active
)"(*如果满足条件(ss∈subj_active)∧(oo ∩ obj_active={})∧(oo∩obj_zombie={})∧(∃K. K∈c_level∧ (fst K = ss) ∧ {snd K}×{nn}⊆flow),那么创建客体 oo 成功,否则失败*)
```

3.4　安全模型内部一致性的证明

内部一致性证明即证明每一个操作保持简单安全属性和*-属性，为此需证明每一个可达状态都满足 3.3 节的 5 个公理，以操作 get_access 为例，在 Isabelle/HOL 中自动证明如下。

```
lemma "({K. ∃m. m∈get_access ss oo_obs oo_alt∧fst m=K ∧snd m=obs}⊆subj_obj0)∧({K. ∃m. m∈get_access ss oo_obs oo_alt∧fst m=K ∧snd m = obs }⊆subj_obj00)∧({K. ∃m. m∈get_access ss oo_obs oo_alt∧fst m=K ∧snd m = alt }⊆subj_obj000) "
apply (simp add:get_access_def)
apply (simp add:Let_def)
apply (rule conjI)
apply (insert obs1 obs2 alt)
apply (simp_all add:subj_obj1_def subj_obj2_def)
apply auto
done
```

其中，apply 表示使用规则或者已证明的引理和定理进行验证；使用 simp 方法对子目标根据有关定义进行展开并化简；根据 rule 证明方法对子目标使用合取引入规则(conjI)；使用 insert 方法在所有子目标中插入定理 obs1，obs2 和 alt 作为新的假设；auto 方法试图简化所有子目标，并根据已有结论和定义进行搜索验证。

Isabelle 的验证结果如图 2 所示。"No subgoals"说明 Isabelle验证逻辑完整，不存在任何未证明的子目标。

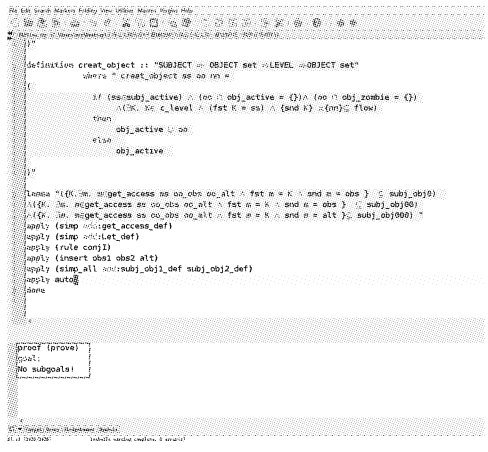

图 2　Isabelle 验证结果(一)

4 功能规约与安全模型的一致性验证

由图 1 可知,安全网关的逻辑框架由多个模块组成,不同模块完成不同功能。本节内容即是对这些模块的功能规约与安全模型的一致性进行形式化验证。限于篇幅,本文仅以过滤模块为例阐述其功能规约与安全模型的一致性验证方法。

首先给出功能规约的形式化描述,得到形式化规约。对于某一功能,确定完成该功能的模块,将其作为主体(记为主体 module_name),相关数据流作为客体(记为客体 data_flow_name_1,data_flow_name_2,…,data_flow_name_n),判断主体对客体的权限(记为 dd,dd 取值为读取、修改、创建或删除),此时可以得到功能规约在安全模型中的描述“主体 module_name 对客体 data_flow_name_1,data_flow_name_2,…,data_flow_name_n 有 dd 权限”,然后利用 Isabelle/HOL 对其进行形式化表示,即得到形式化规约。

如图 3 所示,对于功能规约“过滤进入的消息”,实现该功能的模块是过滤器,相关信息流是待过滤的信息 d_tf、已过滤的信息 d_ok 和配置信息 management。因此在安全模型中描述该功能规约为“主体 f 对客体 d_tf 有读权限、对客体 d_ok 有修改权限、对客体 management 有读权限”,利用 Isabelle/HOL 对其进行形式化表示,得到功能规约“过滤进入的消息”对应的形式化规约为 get_access_ok f {d_ok,management} {d_tf}。

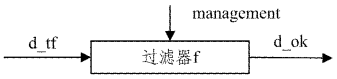

图 3 过滤进入的消息

但根据安全模型的规则,操作 get_access f {d_ok,management} {d_tf}是否成功尚不可知,为解决这个问题,应对形式化规约与安全模型的一致性进行验证,即验证形式化规约对应操作的前提条件是否满足,若满足,则一致性得到验证;若不满足,则说明功能规约违反安全策略,应禁止该功能。Isabelle/HOL 将待证明的前提条件表述为引理(lemma),举例说明如下。

要验证操作“get_accessf {d_tf,management} {d_ok}”是否成功,应验证其前提条件(f∈subj_active)∧({d_tf,management}⊆obj_active)∧({d_ok}⊆obj_active)∧({f}×{d_tf,management}⊆subj_obj0)∧({f}×{d_tf,management}⊆subj_obj00)∧({f}×{d_ok }⊆subj_obj000)是否满足。若证得满足,则一致性得到验证,说明功能规约“过滤进入的消息”合理;若不满足,则不具有一致性,说明功能规约“过滤进入的消息”违背安全网关的安全策略,应禁止。

下面在 Isabelle/HOL 中给出功能规约与安全模型的一致性验证过程。

```
theory Consistency Verification
imports Main
begin
  datatype LEVELi = low|high|param
  datatype LEVELf = out|inn|ok|f1fi|f1tf |f2fi|f2tf
  datatype RIGHT = obs|alt
  type_synonym LEVEL =" LEVELi * LEVELf"
  definition flow::"(LEVEL * LEVEL) set" where "flow == {K. fst K = snd K} ⋃{K. fst(fst K) = param} ⋃{((low,inn),(low,f1tf)),((low,f1tf),(low,f1fi)),((low,f1fi),(low,f2tf)),((low,f2tf),(low,f2fi)), ((low,f2fi),(low,ok)),((low,ok),(high,out))}"
  datatype OBJECT = d_tf|d_ok |management(*定义客体集合*)
  datatype SUBJECT= f (*定义主体集合*)
  definition obj_active::"OBJECT set" where "obj_active == {d_tf,d_ok,management}" (*定义活跃客体集合*)
  consts obj_zombie::"OBJECT set"
  definition subj_active::"SUBJECT set" where "subj_active == {f}" (*定义活跃客体集合*)
  consts subj_zombie::"SUBJECT set"
  definition origin::"(SUBJECT * LEVEL)set" where"origin == {f}×{low} ×{f1fi}" (*赋予主体 f 固有级别(low,f1fi)*)
definition c_level::"(SUBJECT * LEVEL)set" where "c_level == {f}×{low} ×{f1fi} " (*赋予主体 f 当前级别(low,f1fi)*)
  definition clasf::"(OBJECT * LEVEL)set" where "clasf == {d_tf}×{low}×{f1tf}⋃{d_ok}×{low}×{f2tf}⋃{management}×{param}×{ok} " (*赋予客体范畴级别*)
  consts c_access::" ((SUBJECT * OBJECT) * RIGHT)set"
  definition subj_obj1::"(SUBJECT * OBJECT)set" where "subj_obj1 == {K. ∃m. m∈c_access∧fst m = K ∧snd m = obs }"(*主体对客体有读权限的(主体×客体)集合*)
  definition subj_obj0::"(SUBJECT * OBJECT)set" where "subj_obj0 == {K. ∃m n. m∈origin∧ n∈clasf∧ (fst m ,fst n) = K ∧(snd n ,snd m) ∈ flow}" (*客体的范畴级别流向主体的固有级别合法的(主体×客体)集合*)
  definition subj_obj00::"(SUBJECT * OBJECT)set" where "subj_obj00 == {K. ∃m n. m∈c_level∧n∈clasf∧(fst m ,fst n) = K ∧(snd n ,snd m) ∈flow }" (*客体的范畴级别流向主体的当前级别合法的(主体×客体)集合*)
  definition subj_obj2::"(SUBJECT * OBJECT)set" where "subj_obj2 == {K. ∃m. m∈c_access∧fst m = K ∧snd m = alt}"(*主体对客体有写权限的(主体×客体)集合*)
  definition subj_obj000::"(SUBJECT * OBJECT)set" where "subj_obj000== {K. ∃m n. m∈c_level∧n∈clasf∧(fst m ,fst n) = K ∧(snd m ,snd n) ∈ flow}"(*主体的当前级别流向客体的范畴级别合法的(主体×客体)集合*)
(******功能规约与安全需求的一致性验证*******)
    lemma"let
    ss=f;
    oo_obs={d_tf,management};
    oo_alt={d_ok}
        in(ss∈subj_active) ∧(oo_obs⊆obj_active)∧(oo_alt⊆obj_active)∧({ss} ×oo_obs⊆subj_obj0∧({ss}×oo_obs⊆subj_obj00)∧({ss} ×oo_alt⊆subj_obj000))"
    apply (simp add:Let_def)
    apply (simp add:subj_active_def obj_active_def)
    apply (simp only:subj_obj0_def subj_obj00_def subj_obj000_def)
    apply (simp only:habilt_def clasf_def flow_def c_level_def)
    apply auto
    done
```

Isabelle 的验证结果如图 4 所示。“No subgoals”说明Isa-

belle验证逻辑完整，不存在任何未证明的子目标。

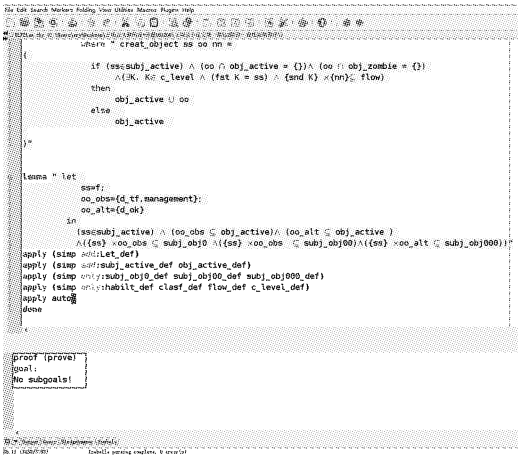

图 4　Isabelle 验证结果(二)

结束语　针对传统上依靠经验设计的安全网关缺乏严格安全模型的不足，本文提出了一种典型安全网关的形式化设计和证明方法。首先利用 BLP 模型对典型安全网关的安全策略进行形式化建模，然后使用定理证明器 Isabelle/HOL 对安全网关的功能规约和安全模型的一致性进行验证，保证了安全网关顶层设计的安全性。

本文仅给出了典型安全网关简化的逻辑结构，在项目的整个设计过程中，涉及到的模块有 20 多个，需要形式化证明的定理有 80 多条，通过采用形式化方法对其进行分析验证，避免了复杂情况下传统设计过程中人为错误的引入，增强了安全网关的可靠性。本文采用的 BLP 模型可以实现对安全网关及其功能模块的描述。接下来的工作将继续利用这种形式化设计和证明的方法对各个功能模块进行形式化设计和验证研究，最终得到采用形式化方法开发的典型安全网关。

参 考 文 献

[1] VIAPLANA A R. IpsecVPN[EB/OL]. [2017-4-10]. http://upcommons. upc. edu/handle/2099. 2/2117.

[2] PHIFER L. Tunnel Vision: Choosing a VPN-SSL VPN vs IPSec VPN[EB/OL]. http:/search security. techtarget. com/feature/Tunel-vision-Choosing-a-VPN-SLL-VPN-vs-IPSec-VPN.

[3] WU T, KUANG X H, FU Z L. Analysis and Suggestion of the Heartbleed [J]. National Defense Science & Technology, 2014, 35(5): 27-30. (in Chinese)
吴彤，匡兴华，傅中立. “心脏出血”漏洞的特点分析和对策建议[J]. 国防科技，2014，35(5)：27-30.

[4] BIEBER P. Formal techniques for an ITSEC-E4 secure gateway [C]// Computer Security Applications Conference, 1996. IEEE, 1996: 236-245.

[5] BELL D E, LAPADULA L J. Secure computer system: mathematical foundation: MTR2527 [R]. Belford: Mitrecorp, 1973.

[6] ABRIAL J R. Modeling in Event-B: System and Software Engineering[M]. Cambridge university Press, 2010.

[7] COMMUNITY E E. Information Technology Security Evaluation Criteria (ITSEC)[R]. Commission, 1990.

[8] 沈晴霓. 操作系统安全设计[M]. 北京：机械工业出版社，2013.

[9] BIBA K J. Integrity Considerations for Secure Computing Systems: Mitre Report MTR 3153[R]. Belford: Mitrecorp, 1975.

[10] DENNING D E. A lattice model of secure information flow[J]. Communications of the ACM, 1976, 16(5): 236-243.

[11] GOGUEN J A, MESEGUER J. Unwinding and Inference Control [C] // 1984 IEEE Symposium on Security and Privacy. IEEE, 1984: 75.

[12] NIPKOW T, PAULSON L C, WENZEL M. Isabelle/HOL: A proof assistant for higher-order logic[M]. Springer, 2002.

[13] CHAPMAN D B, ZWICKY E D. Building Internet firewalls [M]// O'Reilly & Associates, Inc., 1995.

[14] COMER D E. Internetworking with TCP/IP[M]. Beijing: Posts & Telecom Press, 2002.

(上接第 119 页)

[10] ZHANG W Z, LIU J, ZHANG L, et al. Non-cluster Based Topology Control Method in Wireless Sensor Networks[J]. Comptuer Science, 2010, 37(2): 44-47. (in Chinese)
张文铸，刘佳，张林，等. 无线传感器网络的非分簇拓扑控制方法研究[J]. 计算机科学，2010，37(2)：44-47.

[11] FAN Z P, XIE D Q, JIN Z Z. Energy-efficient and load-balancing multipath routing scheme for wireless sensor networks[J]. Journal of Chinese Computer Systems, 2013, 34(2): 253-257. (in Chinese)
樊志平，谢冬青，金政哲. 无线传感网络能量有效负载均衡的多路径路由策略[J]. 小型微型计算机系统，2013，34(2)：253-257.

[12] SUN Y Q, PENG J, LIU T, et al. Uneven clustering routing protocol based on dynamic partition for wireless sensor network [J]. Journal on Communications, 2014, 35(1): 199-206. (in Chinese)
孙彦清，彭舰，刘唐，等. 基于动态分区的无线传感器网络非均匀成簇路由协议[J]. 通信学报，2014，35(1)：199-206.

[13] FENG C X, LIU Z, LUO Y S. Optimal Cluster Numbers in Clustered Wireless Sensor Networks[J]. Journal of Huazhong University of Science and Technology, 2013, 41(10): 49-53. (in Chinese)
冯成旭，刘忠，罗亚松. 分簇传感器网络中最佳簇数的研究[J]. 华中科技大学学报，2013，41(10)：49-53.

[14] REN X L, WANG C. Load-balancing Routing Protocol Based on Dividing Node Neighboring Space in Wireless Sensor Networks [J]. Journal of Chinese Computer Systems, 2016, 6(6): 1222-1227. (in Chinese)
任秀丽，王冲. 基于节点邻域空间划分的无线传感网负载均衡路由协议[J]. 小型微型计算机系统，2016，6(6)：1222-1227.

[15] TIAN X Z, XIAO Y. Algorithm of Opportunistic Routing Based on Energy Harvesting Wireless Sensor Networks[J]. Comptuer Science, 2016, 43(6A): 288-290. (in Chinese)
田贤忠，肖赟. 一种能量捕获无线传感网络机会路由算法[J]. 计算机科学，2016，43(6A)：288-290.