基于有色 Petri 网的 DNP3-SA 协议形式化建模与分析

汪润[1]，赵磊§[1]，熊琦[2]
（1．武汉大学计算机学院，武汉 430072，中国；2．中国信息安全测评中心，北京 100085，中国）

摘要：监控与数据采集（SCADA）系统是一种主要的工业控制系统类型。DNP3 是标准的 SCADA 协议，主要用于主控站和节点之间通信，DNP3-SA 是 DNP3 协议中一种用于保证端到端通信安全的安全认证机制。本文利用有色 Petri 网对 DNP3-SA 协议进行形式化建模，文中提出的 DNP3-SA 有色 Petri 网模型可以测试和验证篡改等攻击场景。利用文中有色 Petri 网模型分析发现了 DNP3-SA 协议中的一个未知安全隐患，攻击者在没有共享密钥的条件下通过使用任意的变量可以重放认证命令，能够通过网络访问 DNP3 设备。
关键词：有色 Petri 网；监控与数据采集系统；安全分布式网络协议
中图分类号：TP309.2　　文献标志码：A

Formal Modeling and Analysis of DNP3-SA Protocol Based on CPN

WANG Run[1], ZHAO Lei§[1], XIONG Qi[1]
(1. School of Computer, Wuhan University, Wuhan 430072, China;
2. China Information Technology Security Evaluation Center, Beijing 10085, China)

Abstract：Supervisory Control and Data Acquisition (SCADA) is a major type of industrial control system. The Distributed Network Protocol version 3 (DNP3) is a standard SCADA protocol designed to facilitate communication in substations and nodes. DNP3-SA is a mechanism to ensure that end to end communication security provided in substations in DNP3 protocol. This paper presents a formal model of DNP3-SA using Colored Petri Nets (CPN). Our DNP3-SA CPN model is capable of testing and verifying various attack scenarios: modification, replay and spoofing. Using the model has revealed a previously unidentified flaw in the DNP3-SA protocol that an attacker can launch a successful attack on an outstation without possessing the pre-shared keys by replaying a previously authenticated command with arbitrary parameters.
Key words：Colored Petri Net (CPN); Supervisory Control and Data Acquisition (SCADA); DNP3-SA

1 引言

工业控制系统（Industrial Control Systems，ICS）是一个国家的重要基础设施，涉及核、石油化工、电力、水利枢纽、铁路、民航及城市供水供气等诸多国计民生领域。随着 2010 年震网病毒[1]的出现，ICS 的安全问题在全球范围得到广泛的关注，各国政府和企业开始研究 ICS 的系统安全、网络安全以及数据安全等安全防护的各个方面。专有通信协议安全是网络安全中普遍关注的问题之一，目前工业控制系统中一直使用较为复杂的专有通信协议，如 Modbus、DNP3、ICCP、OPC 等，这些通信协议完成了 ICS 的信息交互与数据采集、命令发布与执行、业务监控与管理等诸多重要功能[2]。随着信息化与工业化深度融合及物联网的快速发展，这些已运行数十年的协议安全问题日益突出[3]。

根据 CNNVD 中国国家漏洞库的报告[4]，2014 年发现的工控系统相关漏洞数量超过 2005 年至 2013 年所有年份披露工控相关漏洞数量的总和，大量针对工业控制系统的漏洞攻击和黑客工具在互联网上广泛流传，工业控制系统的信息安全面临严峻挑战。

监控与数据采集系统 (Supervisory Control And Data Acquisition, SCADA)是以计算机为基础的生产

基金项目： 国家自然科学基金(61472448)
作者简介：汪润（1991－），男（汉），安徽，博士研究生。
通信作者：赵磊（1985－），副教授。Email: leizhao@whu.edu.cn。

过程控制与调度自动化系统，广泛应用于电力、冶金、石油、化工、燃气、铁路等领域的数据采集与控制等诸多领域[5][6][7]，能够准确的收集系统运行状态，帮助快速诊断系统故障状态等。

分布式网络协议（Distributed Network Protocol，DNP）是一种应用于自动化组件之间的通讯协议，SCADA 系统常使用 DNP 协议与主机、RTU 及 IED 进行通讯。DNP 协议的提出主要为了解决 SCADA 系统中协议混杂、没有公认标准的问题。DNP3 协议以 DNP 协议规范为基础，适用于要求高度安全的数据通信领域。在安全性方面，DNP3 协议能够加载一种用于保证端到端通信安全的安全认证机制 DNP3-SA[8][9]。

为了验证与分析 DNP3-SA 认证机制的安全性，本文利用 CPN（Colored Petri Net，有色 Petri 网）对 DNP3-SA 协议进行形式化建模，并使用有色 Petri 网建模工具（CPN Tools）进行状态空间分析，实际验证“篡改”、“重放”和“嗅探”等三种攻击场景。实验验证发现 DNP3-SA 协议存在安全隐患，攻击者可以利用该安全隐患重放已有认证信息，在未预先共享密钥的条件下利用该安全隐患可以访问网络中的 DNP3 设备。

2 DNP3 协议概述

2.1 DNP3 协议简介

DNP3 协议[10][11]通过“请求—响应”的方式完成消息的交互。每一个请求和响应数据包的帧包括，应用程序控制域（Application Control Field，AC），功能码（Function Code，FC）和对象头（Object Header，OH）。其中，AC 用来控制给定的传输片段是否以正确的顺序接受；FC 用来指定请求或响应的行为；OH 用于补充信息，通常关联一个需要创建完整 DNP3 消息的 DNP3 对象。

DNP3 是一种适用于要求高度安全、中等速率和中等吞吐量的数据通信领域的分布式网络通信协议。DNP3 的设计基于 OSI 七层协议模型中的三层（物理层、数据链路层、应用层），被称之为增强协议结构（EPA）。物理层定义一个普通的 RS-232 或 RS-485 接口；数据链路层定义信息格式；应用层定义规约的功能性。另外，DNP3 还特别使用了一个伪传输层（DNP3 传输层），用于长帧信息的处理。

2.2 DNP3 协议的安全威胁

DNP3 主要运行在主控站和从设备之间，例如 RTU、IED 和控制站之间。DNP3 协议提供了对数据的分片、重组、数据校验、链路控制、优先级等服务。因此，它有一定的可靠性，这种可靠性可以用来对抗恶劣环境中产生的电磁干扰、元件老化等信号失真现象。现有的一些研究报告指出在通信过程中很容易对 DNP3 协议的数据包进行截取、监听和修改，对通信网络的安全性造成了一定的威胁。目前，针对 NDP3 的主要的安全威胁包括：

（1）中间人攻击：它是一种“间接的”入侵攻击，可以在通信的双方毫不知情的情况拦截正常的网络通信数据进行数据“嗅探”和“篡改”等。

在主机和客户机进行通信时，入侵者接入被攻击的工业控制系统网络，在主、客户机毫不知情的情况下，拦截网络上正在传输的数据，获得当前网络上的设备地址，向网内的某个合法设备发送错误报文，进而导致系统工作异常。

（2）拒绝服务攻击：DNP3 增加了主动上传信息的工作模式，这种模式同时也带来了安全隐患，即客户机可以在没有主机允许的情况下可以向其发送数据，从而增加发生拒绝服务攻击的几率。

在拒绝服务攻击中，入侵者进入网络，通过拦截并监听正常报文获取主机的地址，然后充当客户机发送大量非请求报文，使主机忙于应付这些无意义的报文，而无法处理其他正常数据，造成系统瘫痪。

（3）窃听：DNP3 地址和命令都采用明文传输，容易被捕获和解析。窃听攻击虽然不会对系统本身造成危害，但是会造成严重的数据泄露问题。

为应对上述安全威胁，DNP3 提供了一种名为安全认证（Secure Authentication，SA）的保障机制，即 DNP3-SA，它是一种工作在应用层的认证机制。该机制主要通过添加认证等安全机制，来应对一些恶意程序对协议的攻击，从而确保系统端到端通信的安全。DNP3-SA 是单向认证的，它有两种控制模式：“单程”（One-Pass）和“双程”（Two-Pass）。两种模式均基于密钥散列消息认证码（Keyed-Hash Message Authentication Code，HMAC）实现。双程认证被称为“质询-响应”或非侵略“质询-响应”模式（Non-Aggressive Challenge Response，**NACR**），而单程则称为侵略模式（Aggressive Mode，**AGM**）。需要注意的是，在进行单程操作以前，需至少进行一次双程操作。这是由于单程操作需要利用双程操作的某些关系组件。在实际操作时，DNP3-SA 会确保请求，特别是一些“关键请求”送达。当信息中包含强制代码时，该请求和响应被判定为是重要的。强制性代码往

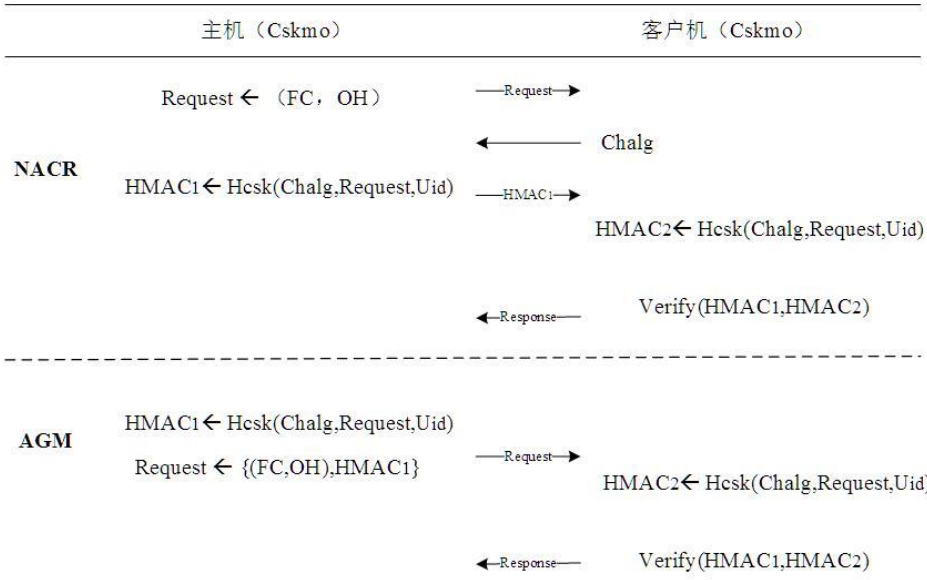

图 1 DNP3-SA 的一般行为（NACR 和 AGM）

往设置相应的参数或调整设定点来执行控制功能。各基站利用其收到的强制性代表来检验它所接收的其它设备的身份。

图 1 通过消息序列表（Message Sequence Chart，MSC）刻画了 DNP3-SA 在两种模式下的行为。主机和客户机代表了通信实体，NACR 代表非侵略质询响应模式，AGM 代表侵略性模式，Csk_{mo} 是控制会话密钥，它由长期密钥 L_K 维护，L_K 在实体之间分发。会话密钥用于认证控制方向上传输的数据。FC，OH 和 IIN 分别代表了功能码、目标头部和内部指示数据（包含请求或回应）。发送特定的请求会得到相应的响应，收到错误的响应则意味着认证失败。*Chlg* 代表质询消息，它包含质询序列号（Challenge Sequence Number，CSQ），消息认证码（Message Authentication Code，MAC），操作码（*H*）和当前标识数据（*N*）。每质询一次，则 *Sn* 增加 *i*。$HMAC_1$ 和 $HMAC_2$ 代表由主机和客户机产生的 *HMAC* 标志。最后，*UID* 代表一个用户认证号，它与通信方相关。

在实验验证过程中，考虑到工业控制系统的特定应用场景，实时的系统安全测试和安全监控实现困难。本文中构建工业控制系统的仿真环境，利用仿真系统研究工业控制系统的动态行为，度量行为复杂性，并进行可生存性测试，验证本文方法的有效性。

3 基于有色 Petri 网的形式化分析

3.1 安全协议的形式化分析

安全协议的形式化分析与验证是一个复杂的过程，一般都通过形式化方法对安全协议进行描述与建模[12][13][14]。协议的非形式化描述，尤其是自然语言描述转变成形式化说明的过程可能会有错误发生，将直接导致后续分析的不确定性。同时，一些形式化方法的提出都是从个别安全协议出发来设计规范的术语，在面临新的协议形式时，需要对原有规范加以扩展，导致形式化术语存在通用性差的问题。所以在进行安全协议的形式化描述与建模之前，需要采用通用的安全协议形式化说明加以规范。

CPN 是一种广泛使用的高级 Petri 网，在多数据类型、复杂交互和并发行为方面具有很强的表达能力。同时，CPN 引入了数据类型和数据操作的概念，增强了 Petri 网的表达能力，更适合交互行为、并发及同步行为较多的软硬件系统的形式化建模。CPN 支持层次化抽象，能够将复杂的系统行为建模在不同的抽象层次上，并支持高效的模型分析与确认。

CPN 非常适合对同步通信的系统进行建模与验证[15][16]，CPN 具有层次建模和标记语言的特性，适用于安全协议的验证与分析。

为了使得通用安全协议说明更加适合于 Petri 网方法，抽象出通用安全协议的基本要素与 Petri 网要素的对应关系，如表 1 所示。

在利用 CPN 进行协议形式化分析时，需要借助 CPN 的安全协议描述和 CPN Tools[17]的扩展，建立 CPN 安全协议操作函数库，再结合实体模型中类似面向对象中类、派生对象概念，利用通用和规范的方法建模，构建一种基于 CPN 的通用安全协议形式化分析语言[18]，基于 CPN 的安全协议形式化分析语

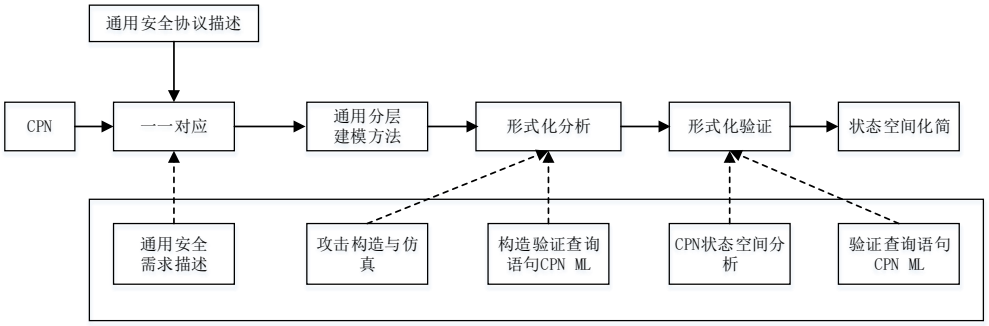

图 2 基于 CPN 的安全协议形式化分析

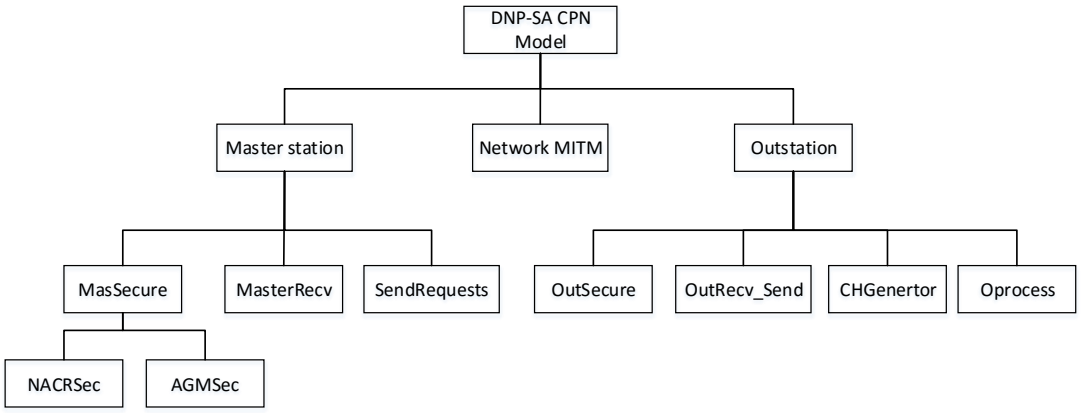

图 3 页面关系图

言工作流程图如图 2 所示，首先利用 CPN 对通用安全协议进行描述，采用分层建模方法对协议进行形式化建模，通过形式化验证和状态空间化简，实现协议的形式化分析。

4 基于 CPN 的 DNP3-SA 行为建模

4.1 层次建模设计

利用 CPN 对 DNP3-SA 协议进行层次化建模主要分为三层，分别是顶层、中间层和底层，见图 3 页面关系图所示。其中顶层和中间层是为了简化模型的复杂性，底层描述的是模型的细节信息。

在顶层主要是主机（Master Station）和客户机（Outstation），以及 Master Station 与 Outstation 进行通信的中间网络（Network）。在 Master Station 部分有三个不同的详细描述页面分别是主机安全控制页面（MasSecure）、主机消息接受页面（MasterRecv）和发送请求页面（SendRequests），在 Outstation 部分有四个不同的详细描述页面分别是客户机安全控制页面（OutSecure）、客户机消息发送页面（OutRecv_Send）、消息生成页面（CHGenerator）和消息处理页面（Oprocess）。

中间层页面是顶层页面图中的细化，主要包括八个不同的替代变迁。其中 Master Station 中的 SendRequests、MasterRecv 和 MasSecure 表示的是主机中三个不同的行为。

NETWORK MITM 表示的是网络或其他的活动例如中间人攻击等，详细描述网络和中间人攻击引发的攻击行为。主要分为三部分，分别是顶部、中间部分和底部。顶部表示主机和外部通信利用 *SendRq* 发送 NACR 请求，变迁 *Connect AGM* 表示的是从主机通过库所 *SendAGMRq* 发往外部的 AGM 请求。设

表 1 CPN 元素与安全协议基本元素对照表

安全协议要素	CPN 元素
原子消息 （实体名称、密钥、随机数…）	简单颜色集
复合消息 （原子消息经过加密或连接）	复合颜色集
集合消息 （一组原子消息或者复合消息）	颜色集列表
密码操作（加密、解密等）	查询函数
协议运行环境	联合颜色集
时间	时间颜色集

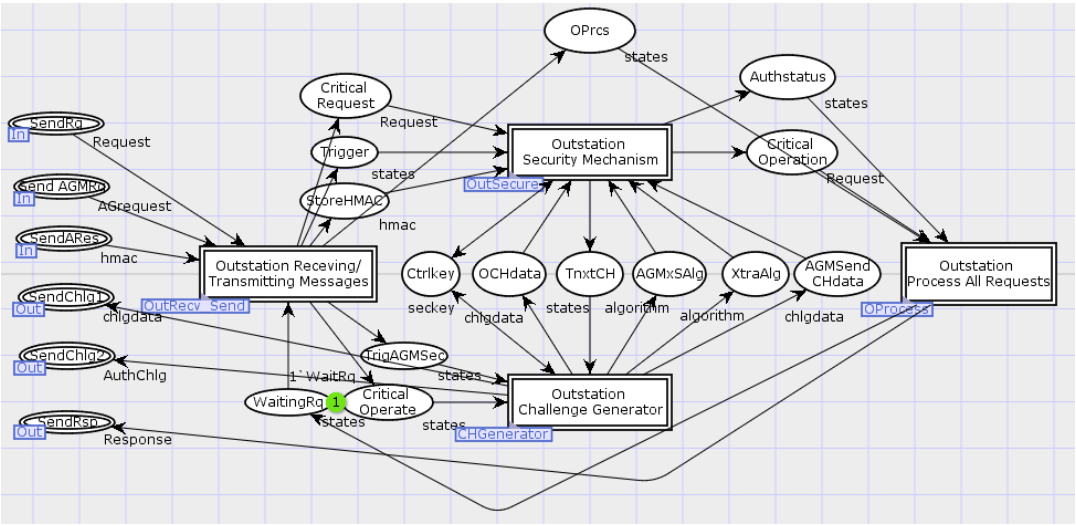

图 4 OutStation CPN 模型

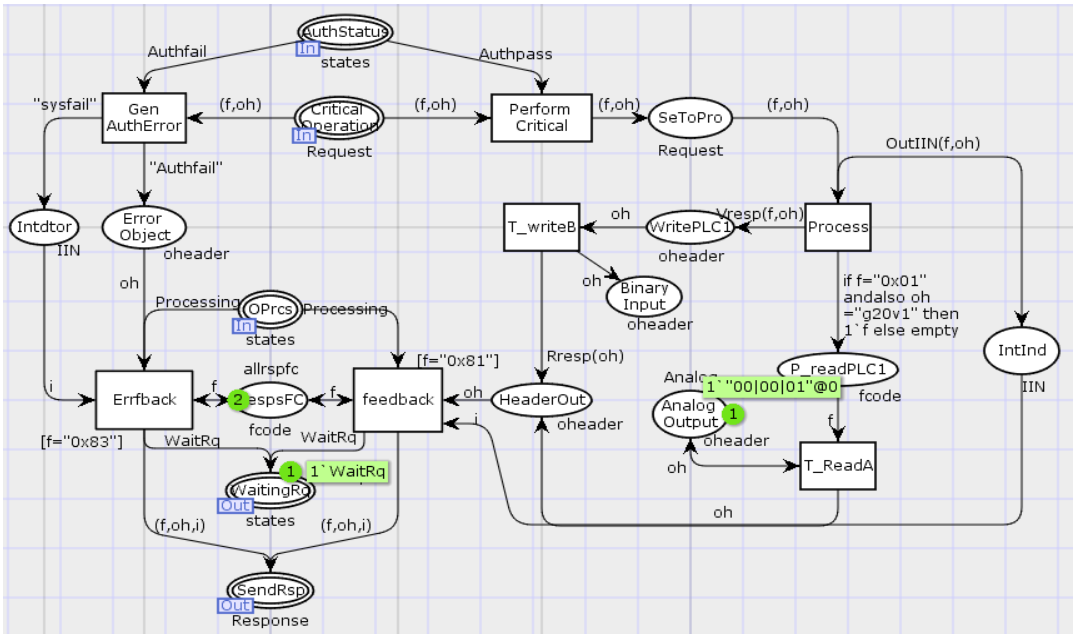

图 5 OProcess CPN 模型图

置 MITM 中的参数变量可以让攻击者加载恶意的活动，例如修改、重放请求和嗅探身份等。

OutStation 表示与主机通信的活动行为，具体的 CPN 模型见图 4 所示，其中包括子页面有 OutSecure、OutRecv_Send、CHGenerator 和 OProcess 等。子页面 OutSecure 是对客户机安全机制的形式化描述，OutRecv_Send 是客户机的消息接受和处理的形式化描述，CHGenerator 是对客户机消息生成的描述，OProcess 描述请求处理和对主机响应的生成。各子页面的详细描述见 4.3 节客户机行为建模部分。

底层描述了模型具体的细节信息，主要是 Master Station 和 Outstation 部分，总共包括 9 个子页面。

4.2 主机行为建模

主机 Master Station 的 CPN 模型可以细化为 SendRequests、MasterRecv 和 MasSecure 等部分。

SendRequests 是对 Master Station 发送请求的 CPN 描述，包括的变迁（Transition）有 *Inipack* 和 *SendPack* 两个部分，*SendPack* 接收 *Inipack* 的令牌（Token），将其发送并给予相应的反馈给 *Inipack*。该模型中可以发送 NACR 或 AGM 请求，最初的请求组成由库所（Place）*Ready*、*Function Code*、*Object Headers* 和 *LPN* 组合而成，Request 的数据类型是

*product fcode*oheader*，当 *SendPack* 满足发送请求时，*SendPack* 将 Token 分发到不同的输出库所中。

MasterRecv 表示的是模型工作流程中的最后一步，Master Station 接受并存储来自 Outstation 的响应，Master Station 可以通过 SendRequest 执行下一步的请求。

MasSecure 包括两个替代变迁组合，分别是 *NACRSec* 和 *AGMSec*，还包括变迁 *Challenge Authenticator*，该模型中所有的变迁均是对 Master Station 的安全行为进行建模，最初的协议行为建模使用的是变迁 *NACRSec*。MasSecure CPN 中包括另外两个替代变迁，其组成的两个子页面如下:

1) Master Station 通过库所 *sendchlg1* 接受 Outstation 的消息，库所 *MPrSeq* 从消息中抽取 CSQ 数据，库所 *Algo* 从消息中获取 HMAC 算法。库所 *Chdata* 发送一个消息的副本给变迁 *SecureNACR* 计算 NACR HMAC 标签，当 *SecureNACR* 满足状态迁移时，*SecureNACR* 计算 HMAC 标签利用库所 *SendARes* 通过网络传送给 Outstation。在模型的右边主要是 AGM 操作中的 CSQ 消息更新的建模。

2）在 AGM 模式中，库所 *AGMRequests* 存放 Master Station 发送给 Outstation 的最初请求数据，库所 *RecentCHM* 存放最近的消息。同 NACR 计算类似，AGM 模式中当 SecAGM 满足状态迁移时根据请求计算 HMAC 标签，然后库所 *AGMPack* 组合原始的请求和计算出的 HMAC 标签，利用库所 *SendAGMRq* 通过网络向 Outstation 发送请求。

4.3 客户机行为建模

在 Outstation 客户机的安全机制 CPN 形式化建模中，Outstation 可以计算 NACR 或 AGM HMAC 标签。库所 *StoreHMAC* 用来接收和存储 HMAC 标签，变迁 *SecureO NACR* 和 *SecureO AGM* 用来计算 HMAC 标签，库所 *OHMAC* 用来存放并用于验证 Outstation 计算出的 HMAC 标签。最终变迁 *Tagcheck* 验证标签，函数 *Verifytags(Mtag,Otag)*用来执行验证过程。

CHGenerator 描述的是 Outstation 如何生成消息，当库所 *Critical Operate* 接收请求 Token 时，变迁 *Create Challenge Data* 可以产生一个消息。库所 *RecentGH*、*NextCH1* 和 *NextCH2* 对 AGM 操作中消息进行建模，函数 *PRandom()*用于在消息生成时产生随机数字。

OutRecv_Send 主要由三个部分组成，分别是 Outstation 从 Master Station 通过库所 *sendRq* 接受的 NACR 请求，Outstation 接受从 Master Station 接受的 AGM 请求，Outstation 接受从 Master Station 接受的 HMAC 标签。

OProcess CPN 模型图见图 5 所示，OProcess 描述 Outstation 的请求处理和对 Master Station 响应的

表 2 模型颜色集 Colset 定义

名称	定义	名称	定义
fcode	colset fcode=string timed;(*function code*)	Response	colset Response = product fcode*oheader*IIN;
algorithm	colset algorithm = string;	states	colset states = with Ready |WaitRq|WaitRsp|Authpass|Authfail| Processing|Critical|Trigger|Terminate|SecInit;
limit	colset limit=unit; (*for correcting*)	chlgdata	colset chlgdata = product seq*algorithm*PseudoRand;
seckey	colset seckey = int;	mac	colset mac = record ff:fcode*cdc:chlgdata*uu:userid;
seq	colset seq = int;	hmac	colset hmac = product algorithm*seckey*mac;
IIN	colset IIN = string;	Request	colset Request=product fcode*oheader;
PseudoRand	colset PseudoRand=int;	AGrequest	colset AGrequest=product Request*hmac;
oheader	colset oheader = string timed(*object header components*);	Chalgtag	colset Chalgtag=product algorithm*seckey*chlgdata*userid;

表 3 状态空间分析统计数据

State Space		SCC Graph	
Nodes	83	Nodes	83
Arcs	164	Arcs	164
Sec	0	Sec	0
Status	full		

表 4 Liveness 信息统计数据

Liveness 属性	值
Dead Marking	6 [83,82,81,80,79,...]
Dead Transition Instances	39
Live Transition Instances	None

生成，库所 *Authstatus* 对标签验证状态的建模。当 *Authstatus* 通过验证时，变迁 *Perform Critical* 发送认证的请求处理。

4.4 基于 CPN 的 DNP3-SA 协议模型定义

CPN 是在保证原型 Petri 网系统性质不变的前提下加入颜色集扩展形成的一种高级 Petri 网，引入颜色集能够提高 Petri 网对分布式系统建模的表达能力，本文中使用 CPN 对 DNP3-SA 协议进行形式化建模，其模型中部分颜色集定义如表 2 所示。

5 模型实现与验证

本文中使用 CPN 的状态分析工具对 DNP3-SA 协议进行安全属性验证，利用 CPN Tools 可以支持的计算树逻辑（Computational Tree Logic，CTL）形式化定义认证属性。

在验证中模型认证条件定义为：如果主机可以产生一个有效的 HMAC 标签，且产生的标签和客户机计算的标签相同，这样客户机能够验证主机的身份；因此，如果标签相同认证通过，否则，认证失败。

实验验证中，本文中使用 CPN Tools 对定义的 CTL 进行形式化建模实现，修改 NETWORK MITM 模型中的参数（*Replay* 和 *Spoof* 设置为 *true*），*Tagcount*=1，测试并验证三种攻击场景为修改、重放请求和嗅探。攻击验证主要由状态空间计算、SCC 图计算和分析报告生成等三个步骤。状态空间分析生成的报告，可生存性统计数据如表 3 和表 4 所示。

表 3 是模型的状态空间分析统计，从表中看出有 83 个节点，从表 4 可知有 6 个死标记（Dead Marking），39 个死变迁实例（Dead Transition Instances）。从表 3 和表 4 统计数据看出当 NETWORK MITM 模型中的 *MITM* 参数设置为 *true*，*Tagcount*=1，攻击者可以加载恶意的活动如修改（攻击者可以修改数据包如请求或消息的内容）、重放请求（攻击者可以重放任意来自 Master Station 的消息）和嗅探（攻击者可以伪装成 Master Station 与 Outstation 通信）等，利用状态空间分析研究发现以下隐患：

攻击者可以在不进行密钥共享的情况下，能够在客户机上重放 AGM 消息执行命令；在 AGM 操作中读写值，甚至在 IED 上面初始化应用程序。

6 总结

本文利用 CPN 对 DNP3-SA 协议进行形式化建模，利用 CPN Tools 状态空间分析工具实际验证了“篡改”、“重放”和“嗅探”等三种攻击场景，发现 DNP3-SA 协议存在未知的安全隐患，攻击者可以利用该安全隐患重放已有认证程序，在未共享密钥条件下可以访问网络连接中的 DNP3 设备。基于本文中提出的模型，可以帮助协议设计者分析发现所有可能造成未知异常行为的触发条件，提高协议设计安全性。

7 参考文献

[1] J. P. Farwell and R. Rohozinski. Stuxnet and the Future of Cyber War[J]. Survival, 53(1):23–40, Feb. 2011.

[2] Amoah R, Suriadi S, Camtepe S, et al. Security analysis of the non-aggressive challenge response of the DNP3 protocol using a CPN model[C]//Communications (ICC), 2014 IEEE International Conference on. IEEE, 2014: 827-833.

[3] Gungor V C, Sahin D, Kocak T, et al. A survey on smart grid potential applications and communication requirements[J]. Industrial Informatics, IEEE Transactions on, 2013, 9(1): 28-42.

[4] 中国国家信息安全漏洞库, http://www.cnnvd.org.cn/ China National Vulnerability Database of Information Security, http:// http://www.cnnvd.org.cn/ (in Chinese)

[5] Ancillotti E, Bruno R, Conti M. The role of communication systems in smart grids: Architectures, technical solutions and research challenges[J]. Computer Communications, 2013, 36(17): 1665-1697.

[6] Miller B, Rowe D. A survey SCADA of and critical infrastructure incidents[C]//Proceedings of the 1st Annual conference on Research in information technology. ACM, 2012: 51-56.

[7] Nicholson A, Webber S, Dyer S, et al. SCADA security in the light of Cyber-Warfare[J]. Computers & Security, 2012, 31(4): 418-436.

[8] Gilchrist G. Secure authentication for DNP3[C]//2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century. 2008.

[9] Crain J A, Bratus S. Bolt-on security extensions for industrial control system protocols: A case study of DNP3 SAv5[J]. IEEE Security & Privacy, 2015, 13(3): 74-7

[10] Lee D, Kim H, Kim K, et al. Simulated attack on dnp3 protocol in scada system[C]//Proceedings of the 31th Symposium on Cryptography and Information Security, Kagoshima, Japan. 2014: 21-24.

[11] Yan Y, Qian Y, Sharif H, et al. A survey on smart grid communication infrastructures: Motivations, requirements and challenges [J]. Communications Surveys & Tutorials, IEEE, 2013, 15(1): 5-20.

[12] Bodei C, Buchholtz M, Degano P, et al. Static validation of security protocols[J]. Journal of Computer Security, 2005, 13(3): 347-390.

[13] Bolignano D, Le Métayer D, Loiseaux C. Formal methods in practice: The missing links. A perspective from the security area[M]//Modeling and verification of parallel processes. Springer Berlin Heidelberg, 2001: 169-180.

[14] Floreani D J, Billington J, Dadej A. Designing and verifying a communications gateway using coloured Petri nets and design/CPN™[M]. Springer Berlin Heidelberg, 1996.

[15] Kristensen L M, Petrucci L. An approach to distributed state space exploration for coloured petri nets[C]//International Conference on Application and Theory of Petri Nets. Springer Berlin Heidelberg, 2004: 474-483.

[16] Jensen K, Kristensen L M. Coloured Petri nets: modelling and validation of concurrent systems [M]. Springer Science & Business Media, 2009.

[17] Jensen K, Kristensen L M, Wells L. Coloured Petri Nets and CPN Tools for modelling and validation of concurrent systems[J]. International Journal on Software Tools for Technology Transfer, 2007, 9(3-4): 213-254.

[18] Tritilanunt S, Boyd C, Foo E, et al. Using coloured petri nets to simulate dos-resistant protocols [J]. 2006.