doi:10.11959/j.issn.1000-436x.2015308

改进的安全协议自适应分析算法

杨京[1,2]，范丹[2,3]，张玉清[1,2]

（1.西安电子科技大学 综合业务网理论及关键技术国家重点实验室，陕西 西安 710071；
2. 中国科学院大学 国家计算机网络入侵防范中心，北京 100190；
3. 中国科学院 信息工程研究所 信息安全国家重点实验室，北京 100093）

摘 要：提出一种改进的安全协议自适应分析算法，即修正学习算法 L_a^*，解决部分教师缺乏经验的问题，并将字符集扩展为大字符集。对提出的修正学习算法，进行了正确性证明和复杂度分析。该修正学习算法将有助于提高安全协议自适应模型检测的效率、降低分析和设计成本、缓解状态空间爆炸并增强协议本身对环境和各种攻击手段的防御能力。

关键词：安全协议；自适应模型检测；学习算法；缺乏经验的教师；符号自动机

中图分类号：TP393.08 **文献标识码**：A

Adjusted automata learning algorithm for security protocol adaptive model checking

YANG Jing[1,2], FAN Dan[2,3], ZHANG Yu-qing[1,2]

(1.Information Security Research Center of State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China;
2. National Computer Network Intrusion Protection Center, University of Chinese Academy of Sciences, Beijing 100190, China;
3. State Key Laboratory of Information Security Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093,China)

Abstract: Modifications and improvements for adjusted automata learning algorithms, which were enabled by recent developments. Adjusted automata learning algorithm L_a^* is correct and efficient. It will be helpful to improve adaptive model checking efficiency, reduce the cost, solve state space explosion problem and resist many kinds of attack methods for security protocols.

Key words: security protocol; adaptive model checking; learning algorithm; inexperienced teacher; symbolic automata

1 引言

目前，安全协议的形式化分析方法大致可以分为 3 类：形式逻辑方法、模型检测方法和定理证明方法[1]。其中，模型检测方法也称作状态空间搜索法。模型检测技术采用形式化方法精确地证明一个系统能够按照预定目标正确工作。模型检测的基本思路是把安全协议看成一个分布式系统，而每个主体执行协议的过程构成局部状态，所有的局部状态则构成系统的全局状态，每个主体的消息接收和发送动作都会引起局部状态的改变，进而引起全局状态的改变，因此需要在系统可达的每个全局状态，检查其安全属性是否满足文献[2]。对于安全协议分析来讲，模型检测技术已被证明是一条非常成功的途径。首先，这种方法的自动化程度高，在验证过程中不需要用户参与；其次，如果协议存在缺陷，能够自动产生反例。但因为容易产生状态空间爆炸问题，所以模型检测方法不能用于比较复杂的安全协议分析。

虽然存在以上缺点，但由于模型检测方法能够实现自动化，这种方法得到很多研究人员的关注，并做

收稿日期：2015-10-24

基金项目：核高基 1-4“开源操作系统内核分析和安全性评估”基金资助项目（2012ZX01039-004-65）；信息安全国家重点实验室开放基金资助项目（2014-12）

Foundation Items: The Core Electronic Devices, High-end Generic Chips and Basic Software 1-4 “Analysis and Security Assessment of the Open Source Operating System Kernel” (2012ZX01039-004-65); The Open Project Program of the State Key Laboratory of Information Security(2014-12)

了许多相关工作。Clark 等[3]通过使用偏序归约技术削减验证过程的状态空间；Shmatikov 等[4]通过规定在合法主题可以发送消息时，攻击者就不发送消息，来实现削减状态空间；Broadfoot 等[5]通过把可信第三方的通信顺序进程合并到攻击者进程来削减状态空间；Hui 和 Gavin [6]先把复杂协议转换成简单协议（简单协议能够体现复杂协议的安全缺陷），然后再利用模型检测方法来分析复杂的安全协议；Stoller[7]提出一种决定攻击一个协议时所需协议实例的上界的方法；Roscoe 等[8]通过使用数据独立技术，给出实现攻击有用的随机数数目的上界。

鉴于普通模型检测方法对于存在漏洞的协议，需人为修正或重新设计，这种处理方式不仅效率低下，而且严重依赖于经验，容易引入新的漏洞。安全协议自适应模型检测（adaptive model checking）方法[9]借鉴可以实时更新模型的自适应技术将安全协议的设计过程与分析过程统一起来，根据反例自动修正初始模型，并以此改进和优化安全协议。与现有的密码协议分析和验证方法相比，安全协议自适应模型检测方法最大的优势在于，当验证出协议漏洞或需求发生变化时，可以对协议初始模型进行自适应的更新。

安全协议自适应模型检测框架如图 1 所示。首先对安全协议（密码协议）建模，即编码实现，利用模型检测工具(model checker)对其进行模型检测，检测结果若为安全（safe）的，则终止（terminate）；否则返回反例（counter-example），可给出图形化结果（attack graph），也可调用 exe 文件和 cmd 窗口，将结果保存为 xml 文件，再对 xml 文件进行解析和处理，提取出反例。把反例转换成字符串并将其输入自动机学习算法工具学习，以修正安全协议初始模型。模型检测工具使用 Scyther 或者 AVISPA/Span、spin、nusmv 等。在该框架中学习算法起着非常重要的作用，它使安全协议的模型检测过程形成一个闭环。因此，学习算法的选取显得尤为重要。

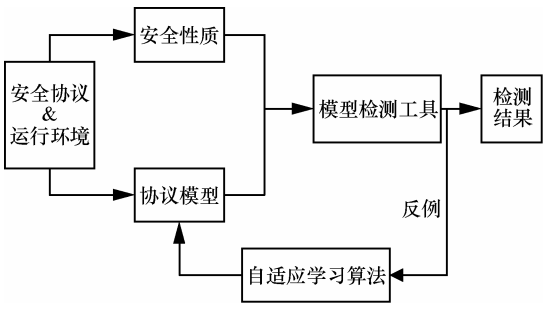

图 1　安全协议自适应模型检测框架

自动机学习算法相关文献提出了许多种学习算法，例如机器学习相关的蚁群优化算法[10]、遗传算法、基于 SAT 求解器的算法、IVAP 算法[11]。新的算法层出不穷，而安全协议自适应模型检测中最实用的学习算法是基于 L*学习算法[12]的一些算法，将自动机学习算法用于修正协议初始模型。本文介绍了 L*学习算法及其相关算法，描述了自动机如何通过反例来学习目标语言。基于 L*学习算法的一系列算法包括推断确定型自动机（如文献[13]中的 Lm+算法等）和非确定型自动机（文献[14]中的 NL*，文献[15]中的 L*nm 等）的多种算法。

考虑实际场景：复杂的安全协议发送者、接收者、入侵者、加解密、认证等状态繁多；在许多应用场景中，一个老师可能并不能回答某些成员资格询问，因为要学习的对象不是完全具象的，是不可观察的或者要回答这些提问所付出的代价是非常高昂的等，所以这些询问可能是悬而未决的。然而，当前的学习算法显然不能满足日益复杂的安全协议，有必要对其进行改进。

本文提出一种改进的安全协议自适应分析算法，即修正学习算法 L_a^*，对 L*学习算法做了 2 方面的修正，一方面字母表可以是大字符集（无穷的或不可枚举的），引入符号自动机，并将观察表扩展为符号观察表；另一方面对老师做了修正，将最小胜任教师(MAT，minimal adequate teacher)推广至缺乏经验的老师（inexperienced teacher）。修正学习算法将有助于提高安全协议自适应模型检测的效率、降低分析和设计成本、缓解状态空间爆炸并增强协议本身对环境和各种攻击手段的自适应防御能力。

2　预备知识

2.1　确定型有穷接受器

首先介绍的自动机是操作确定的有穷接受器，这种类型的自动机特点是没有临时存储。因为输入文件不可改写，所以有穷自动机在计算过程中所能记住的东西是相当有限的。通过把控制部件放在某一个特殊状态上，自动机可以把有限的信息保存到这个控制部件中。但是由于这种特殊状态是有限的，有穷自动机只能处理这种情况，即每一时刻可以存储的信息都是有严格限制的[16]。下面给出精确的形式化定义。

定义 1 确定型有穷接受器（DFA, deterministic finite accepter）是一个五元组 $M=(Q, \Sigma, \delta, q_0, F)$ 其中，Q 是内部状态（internal states）的有限集合，Σ 是符号的有限集合，叫做输入字母表（input alphabets），δ：$Q\times\Sigma\to Q$ 是一个全函数，叫转移函数（transition function），$q_0\in Q$ 是初态，$F\subseteq Q$ 是终态集合。

每个有穷自动机都接受某种语言，如果考察所有可能的有穷自动机，就获得了一个和这些自动机相关的语言集合[17]。能够被确定型有穷接受器接受的语言族是正则语言。

2.2 3 个值的确定型有穷自动机

对确定型有穷接受器（DFA）进行扩展，将其状态集合进行扩充，这对后面的讨论尤为重要，参见第 4 节。下面给出 3 个值的确定型有穷自动机（3DFA）的形式化定义[18]。一个 3DFA 的示例如图 2 所示。

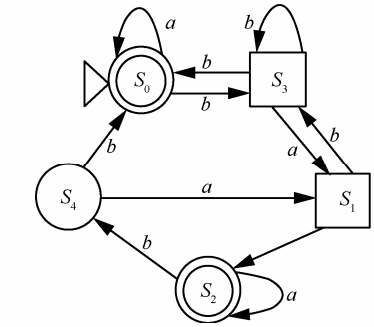

图 2 一个 3DFA 的示例

定义 2 3DFA 是一个 7 元组$(\Sigma, S, s_0, \delta, Acc, Rej, Don't)$，其中 Σ, S, s_0 和 δ 的定义与 DFA 中的相同。将 S 分割成 3 个不相交的集合 Acc，Rej 和 $Don't$。其中，Acc 是接受状态的集合，Rej 是拒绝状态的集合，$Don't$ 是不确定状态的集合。

对于一个 3DFA C = $(\Sigma, S, s_0, \delta, Acc, Rej, Don't)$，

如果 $\delta(s_0, u) \in Acc$，则字符串 u 被接受；

如果 $\delta(s_0, u) \in Rej$，则字符串 u 被拒绝；

如果 $\delta(s_0, u) \in Don't$，则字符串 u 可能被接受也可能被拒绝。

C+表示 DFA $(\Sigma, S, s_0, \delta, Acc\cup Don't)$，这样所有的不确定状态都变成了接受状态。

C−表示 DFA $(\Sigma, S, s_0, \delta, Acc)$，这样所有的不确定状态都变成了拒绝状态。

由此可得出这样的结论，$L(C-)$是 C 中可接受的字符串的集合，$L(C+)$是 C 中拒绝状态集合的补集。

当且仅当一个确定性有穷自动机 A 接受所有 C 接受的字符串并拒绝所有 C 拒绝的字符串时，称 A 与 C 相一致。也就是说，A 接受 $L(C-)$ 中的所有字符串，并拒绝 $L(C+)$ 中的所有字符串，即 $L(C-)\subseteq L(A) \subseteq L(C+)$。$C$ 的一个与其相一致的最小化的有穷自动机 A 是所有与 C 相一致的自动机中状态数目最少的。

图 3 给出了一个与 C（3DFA）相一致的有穷自动机 A。图中矩形框表示全集，即 Σ^*中所有字符串，深色阴影表示 $L(C-)$，深色阴影与浅色阴影的交集代表 $L(C+)$。与 C 相一致的 A 接受 $L(C-)$中所有字符串，拒绝所有不在 $L(C+)$中的字符串。

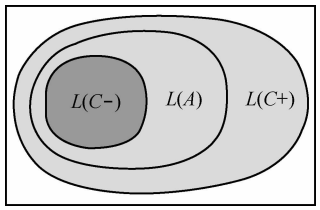

图 3 与 C (3DFA)相一致的有穷自动机 A

给定 2 个不相交的正则语言 $L1$ 和 $L2$，一个有穷自动机（DFA），A 满足 $L1 \subseteq L(A)$ 和 $L(A)\cap L2=\varnothing$。A 接受 $L1$ 中的所有字符串，拒绝 $L2$ 中的所有字符串。即 $L1 \subseteq L(A) \subseteq L2$。当且仅当 A 是一个区分 DFA 时，称一个确定型有穷状态机 A 区分 $L1$ 和 $L2$。如果一个 DFA 在所有可以区分 $L1$ 和 $L2$ 的有穷自动机中是拥有状态数目最少的有穷自动机，则称它为最小化的区分 DFA。区分 $L1$ 和 $L2$ 的 DFA 如图 4 所示。

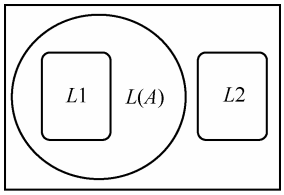

图 4 区分 $L1$ 和 $L2$ 的有穷自动机 A

3 L*学习算法

3.1 算法背景

问题：从一个集合的成员和非成员中识别一个未知的正则语言集合。这个正则语言集合由一个最小胜任教师给出，该老师可以回答关于集合的成员资格询问，并且能够判断一个假定的集合是否等价于未知集合（目标语言集），如果不等价则给出一

个反例。反例是正确集合（目标语言集）与假设集合的对称差。

基于 JFLAP 的 DFA 工具[19]实现了 L*学习算法。L*算法在多项式时间（关于最少状态数目的 DFA 和由老师提供的最大长度的反例）内向最小胜任教师正确学习正则语言。在随机化背景下，老师测试假设的能力可由随机采样预言机替代。

3.2　算法描述

下面详细描述 L*学习算法。

学习者 L*维护一个观察表(S, E, T)，其中 S 和 E 分别代表前缀封闭和后缀封闭的集合。S 是状态集合，E 是区分状态的集合。T 是一个函数，它把 $(S\cup S\Sigma)E$ 中的元素字符串映射到{0, 1}。对于 $s\in(S\cup S\Sigma)E$，如果 $se\in U$，则 $T(s, e)=1$；否则 $T(s, e)=0$。L*学习算法具体步骤如下。

1) 初始化：将 S 和 E 初始化为空串 λ，对 λ 和字母表 A 中的每一个字母进行成员资格询问，构建初始化的观察表(S, E, T)。

2) 完备性和一致性校验：如果(S, E, T)不一致，则需找到 S 中的 $s1$ 和 $s2$，$a\in A$，$e\in E$，使 $row(s1)=row(s2)$，且 $T(s1ae)\neq T(s2ae)$，将 ae 添加至 E 中。通过成员资格询问扩展 T 至$(S\cup SA)$。

如果(S, E, T)不完备，则需找到 $s1\in S$, $s2\in S$ 使对于任意 $a\in A$, $row(s1a)$与 $row(s)$都不同，并将 $s1a$ 添加至 S 中。通过成员资格询问扩展 T 至$(S\cup SA)$。

3) 一旦(S, E, T)是完备且一致的，就可以构造状态机 M，并进行等价询问。如果老师返回一个反例 ce，则将该反例及其所有前缀添加至 S 中。并用成员资格询问扩展 T 至$(S\cup SA)$。然后再返回步骤 2）做一致性和完备性校验，满足一致性和完备性之后再做步骤 3）的等价询问，以此循环往复，直至返回 $M=L(U)$方才停止，并输出 M。

4　修正学习算法 L_a^*

由于安全协议的发送者、接收者、入侵者以及加密、解密、认证等状态繁多，在许多应用场景中，老师可能并不能回答某些成员资格询问，因此本文提出一种改进的安全协议自适应分析算法，即修正学习算法 L_a^*，运用符号自动机，并把教师定义为缺乏经验的老师，学习大字符集。本节还对修正学习算法进行正确性证明和复杂度分析，以及实例分析。

4.1　缺乏经验老师

因为要学习的对象可能不是完全具体的，是不可观察的或者要回答这些提问所付出的代价是非常高昂的等，所以这些询问可能是悬而未决的。最小胜任教师显然不能满足这种实际需要，下面引入缺乏经验的老师(inexperienced teacher)[20]，并向缺乏经验的老师学习，从而获得所需的知识。

定义 3　缺乏经验的教师。一个缺乏经验的教师有权访问 2 个不相交的语言集 $L1$，$L2\subseteq\Sigma^*$，并回答下面 2 类询问。

1) 成员资格询问：如果给定的字符串在 $L1$ 中，老师返回真(+)；如果给定的字符串在 $L2$ 中，老师返回假(−)；如果给定的字符串既不在 $L1$ 中，又不在 $L2$ 中，老师返回不确定(?)。

2) 包含关系询问：老师解决下面 4 种类型的语言集合包含问题，①$L1\subseteq L(Ai)$，②$L(Ai)\subseteq L1$，③$(\frac{\Sigma^*}{L2})\subseteq L(Ai)$，④$L(Ai)\subseteq(\frac{\Sigma^*}{L2})$。其中，$Ai$ 是一个假定的符号自动机。如果包含关系满足，老师返回真；否则老师返回假并给出一个反例。

正如定义 3 所述，给定一个缺乏经验的老师，学习者的任务是用成员资格询问和等价询问获得一个可行的符号自动机。即学习者的任务是找到一个接受 $L1$ 的超集并拒绝 $L2$ 的符号自动机。

$L1$，$L2\subseteq\Sigma^*$，是 2 个不相交的语言集。分 3 种情况讨论。

1) 如果 $L1$ 和 $L2$ 都是正则语言，那么一定存在一个可行的 DFA，例如精确接受 $L1$ 的 DFA。

2) 如果 $L1$ 和 $L2$ 都是确定性上下文无关语言，那么这个判定性问题是否可判定是未知的。

3) 如果 $L1$ 和 $L2$ 都是非确定性上下文无关语言，那么这个判定性问题是不可判定的。

后者可以通过一个简单的归约来判定一个（非确定性）上下文无关语言是否为正则语言。

4.2　符号学习算法

Oded Maler 和 Irini Eleftheria Mens[21]定义了一种大字符集（无限集或不可列集合）上的广义模型，过渡状态由字母表有限分割的元素（子集）标记。然后把从实例中学习正则语言的有穷自动机扩展至符号自动机，改进了学习算法。

定义 4　符号自动机。确定性符号自动机是一个七元组 $A=(c, \Sigma', \psi, Q, \delta, q_0, F)$，其中，$-\Sigma$ 是输

入字符集；–Σ'是一个有限字符集, 可分解为 $\Sigma = Uq \in Q\Sigma q$；–$\psi$= $\{\psi q: q \in Q\}$ 是一个满射函数族，$\psi q: \Sigma \to \Sigma q$；–$Q$ 是一个有限的状态集；–$\delta: Q \times \Sigma \to Q$ 是一个部分转移函数，可分解为一族全函数 $\delta q: \{q\} \times \Sigma q \to Q$；–$q0$ 是初始状态，F 是接受状态（终态）集合。

定义 5 符号观察表。符号观察表是一个八元组 $A = (\Sigma, \Sigma', S, R, \psi, E, f, \mu)$，其中，–$\Sigma$ 是符号的集合，叫做字母表（字符集）；–E 是 Σ^*的子集合；–$f: (S \cup R)E \to \{0, 1\}$是符号分类函数。

下面用一种直观的表述方式来描述符号学习算法。假设字符集是有序的，且 a_0 表示它的最小值。假设老师总是提供关于长度和 Σ^*上字典序最小的反例。这个算法与 L^*算法非常类似，只是在处理反例时有所不同。

先对符号观察表 $T = (\Sigma, \Sigma', S, R, \psi, E, f, \mu)$初始化，其中，$\Sigma=\{a_0\}$,$S=\{\lambda\}$，$R = \{a_0\}$, $E = \{\lambda\}$，和 $\mu(a_0)=\{a_0\}$。无论何时，若 T 不完备，对于任意 $s \in S$，一定存在某个 $r \in R$ 使 $fr \neq fs$。为了使观察表完备，要把 r 从 R 移动至 S 并把 $r'=ra$ 添加到 R 中，其中，a 是一个新的字符。

当得到一个反例 w 时，对其进行分解，即 $w=uav$，其中，u 是最大前缀，且 $u \in S \cup R$，满足 $u \in \mu(u)$，分 2 种情况讨论。

1) 若 $u \in S$。反例表明 ua 被错误的假设为$[ua]$的一部分，而且 a 被错误的假设为$[a]$的一部分，有以下两种情况。

①若存在某个 $a' \neq a$，使由符号自动机分类的 $ua'v$ 与 uav 的分类相一致，要把 a 及所有比 a 大的字符从$[a]$移动至$[a']$中，此时没有新状态产生。

②如果没有上述字符，要生成一个新的 a'，$[a']=\{b \in [a]: b \geqslant a\}$，并把 $[a]$-$[a']$ 更新为 $[a]$，$\mu(ua')=\mu(u)a$，把 ua'添加到 R 中。

2) 若 $u \in R$。反例表明 $g(u)=s \in S$，由于该观察表是最简化的，有对于其他任意 $s' \in S$，$fu \neq fs'$。因为 w 是最短的反例，所以 sav 的分类是正确的。这样可以推断出 u 是一个新状态并将其添加至 S 中。为了区分 u 和 s，要把 av 及其后缀添加到 E 中。根据 a 区分下面 2 种情况。

①如果 $a= a_0$ 是 Σ 的最小元素，则把字符 a 添加至 Σ 中，字符 ua 添加至 R 中。

②如果如果 $a \neq a_0$，则把 2 个字符 a 及 a'都添加至 Σ 中，其中，$[a]=\{b: b<a\}$，$[a']=\{b: b \geqslant a\}$，$\mu(ua)=\mu(u)a_0$，$\mu(ua')=\mu(u)a$，将 ua 和 ua'添加至 R 中。

4.3 修正学习算法 L_a^*

基于以上对学习算法的讨论和缺乏经验老师的描述，以及安全协议状态繁多可能造成状态空间爆炸的问题，本文提出一种修正学习算法 L_a^*。该算法可用于大字符集的场景，并结合经验不足的老师可能无法回答某些提问，对之前的算法进行改进。

修正学习算法 L_a^*是一种主动学习算法，它可以识别正则语言。假设一个老师可以回答 2 类询问：成员资格询问和包含关系询问。

修正学习算法流程如图 5 所示。

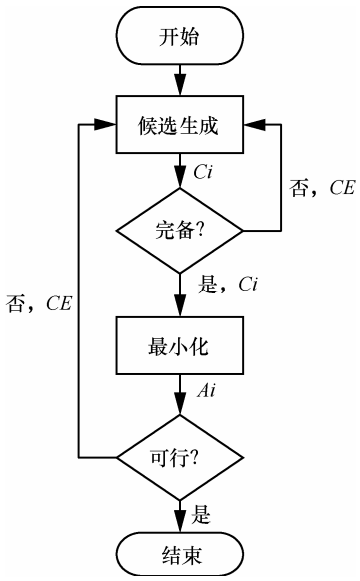

图 5 L_a^*学习算法流程

1) 候选生成步骤由候选生成器操作，运用扩展的 L*算法（多了一项不确定状态）以及符号学习算法，产生一系列以 3DFA C 为目标的候选自动机（3DFA）。候选生成器通过成员资格询问来构造观察表。如表 1 和图 6 所示，举例说明了一个观察表和与之相对应的 3DFA（方框表示不确定状态）[18]。根据观察表产生一个 Ci（3DFA），如果这个 Ci（3DFA）是不完备的或不一致的，候选生成器从反例中提取出区分字符串来扩展观察表，然后产生一个猜测的 3DFA。N 是完备且一致的 3DFA 的大小，m 是由集合包含关系返回的最长反例的长度。候选生成器生成一个完备且一致的 3DFA 所需要的成员资格询问数目为 $O(n^2+n \log m)$，至多产生 $n-1$ 个不正确的 3DFA。

表 1　　**符号观察**

T	λ	b
λ	−	?
b	?	?
ba	−	+
bab	+	+
a	−	?
bb	?	?
baa	−	+
$baba$	−	?
$babb$	+	+

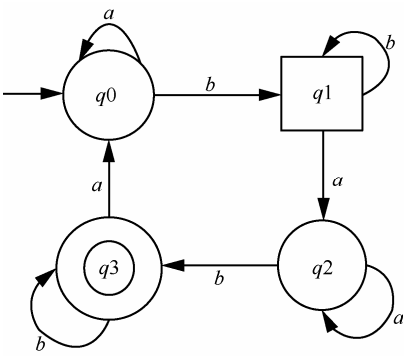

图 6　与表 1 相对应的 3DFA（方框表示不确定状态）

2) 完备性校验步骤：检验 *Ci* 关于 *L*1 和 *L*2 是否完备，如果 *Ci* 不完备，会返回给候选生成器一个反例来精化下一个猜测。否则，*Ci* 是完备的，至下一步，计算与 *Ci* 相一致的 *Ai*。修正学习算法通过计算与 *Ci* 相一致的最小化的符号自动机来找到分割 *L*1 和 *L*2 的最小化的符号自动机。为了确保所有分割 *L*1 和 *L*2 与 *Ci* 相一致的最小化的符号自动机都被考虑到，所以修正学习算法要检验 *Ci* 是否完备。检验完备性可以归约为检验 $L(Ci-) \subseteq L1$ 和 $\frac{\Sigma^*}{L2} \subseteq L(Ci+)$是否成立，这些可以通过集合包含关系来得到。

3) 最小化符号自动机步骤：完备性校验完成后，下一步是计算一个与 *Ci* 相一致最小化的符号自动机。本文把该问题归约为不完全具体化的有限状态机的最小化问题，可参照文献[22]。修正学习算法将 3DFA *Ci* 转换成一个不完全具体化的符号自动机 *M*。引入文献[21]中的算法，得到一个与 *M* 相一致的最小化的符号自动机 *Mi*。最后再把 *Mi* 转换成一个符号自动机 *Ai*。也可参照文献[20]中的最小化方法，运用 CSP 求解器、SAT 求解器或者 SMT 求解器。其中，CSP 可以转化为等价的 SAT。

4) 可行性检验：修正学习算法在计算出来与 *Ci* 相一致的最小化的符号自动机 *Ai* 之后，要通过集合包含关系询问 $L1 \subseteq L(Ai)$和 $L(Ai) \subseteq \frac{\Sigma^*}{L2}$ 来验证 *Ai* 是否区分开了 *L*1 和 *L*2。如果满足了上述集合包含关系，则输出正确结果，即最小化的符号自动机。否则将反例输入第 1 步，以此循环往复，直至正确结果输出方才停止。

4.4　正确性证明和复杂度分析

下面的定理表明了修正学习算法的正确性。

定理 1　修正学习算法会终止并输出一个最小的可以区分 *L*1 和 *L*2 的符号自动机。

定理证明过程可参考文献[18]。

引理 1　假设 *Bi* 是可以接受正则语言 *Li* 的最小化的符号自动机，其中 *i*=1，2。最小化的接受 *L*1 中所有字符串并拒绝 *L*2 中所有字符串的 *C*（3DFA）的大小不超过$|B1| \times |B2|$ 。

定理 2　假设 *Bi* 是可以接受正则语言 *Li* 的最小化的符号自动机，其中 *i*=1，2。修正学习算法学习区分 *L*1 和 *L*2 的最小化的符号自动机所需要的成员资格询问数目至多为 $O((|B1| \times |B2|)^2 + (|B1| \times |B2|) \log m)$，包含关系询问数目至多为 $4(|B1| \times |B2|) - 1$。其中，*m* 为老师返回的最长反例的长度。

修正学习算法所需要的成员资格询问数目至多为 $O((|B1| \times |B2|)^2 + (|B1| \times |B2|) \log m)$，包含关系询问数目至多为 $4(|B1| \times |B2|) - 1$。其中，*m* 为老师返回的最长反例的长度。

证明　假设 *C* 是接受 *L*1 中所有字符串并拒绝 *L*2 中所有字符串的最小化的 3DFA。候选生成器所需要的成员资格询问数目至多为 $O((|C|^2 \times |C| \log m)$，提出错误假设的 3DFA 数目至多为$(|C| - 1)$。根据引理，*C*（3DFA）的大小不超过$|B1| \times |B2|$。由此可得，在最坏的情况下，修正学习算法学习一个最小化的符号自动机所需要的成员资格询问数目至多为 $O((|B1| \times |B2|)^2 + (|B1| \times |B2|) \log m)$，包含关系询问数目至多为$(4(|B1| \times |B2|) - 1)$（对于每一个假设的 3DFA，修正学习算法需要 2 次包含关系询问来检验完备性是否满足，以及 2 次包含关系询问来检验可行性是否满足）。其中，*m* 为老师返回的最长反例的长度。

4.5 学习算法实例分析

目标语言集合为：$L=\{au:b\leqslant a<c,u\in\Sigma^*\}$。

运用修正学习算法对 L 进行学习，可得如下的观察表（如表 2~表 6 所示）和自动机模型（如图 7~图 10 所示）。

表 2 观察表 T_0

T_0	λ
λ	0
a_0	0

表 3 观察表 T_1

T_1	λ
λ	0
a_0	0
a_1	1

表 4 观察表 T_2

T_2	λ
λ	0
a_1	1
a_0	0
a_1a_2	1

表 5 观察表 T_3

T_3	λ
λ	0
a_0	0
a_1	1
a_0a_4	0
a_1a_2	1
a_3	0

表 6 观察表 T_4

T_4	λ	b
λ	0	1
a_0	0	0
a_1	1	1
a_0a_4	0	0
a_1a_2	1	1
a_3	0	0

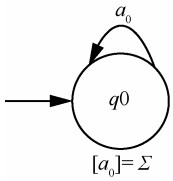

图 7 观察表 T_0 对应的自动机

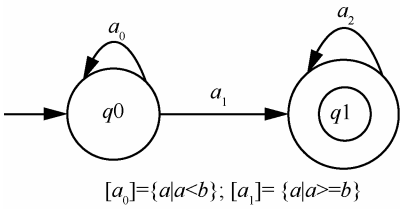

图 8 观察表 T_2 对应的自动机

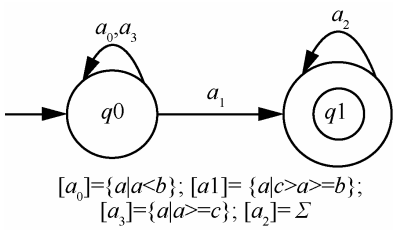

图 9 观察表 T_3 对应的自动机

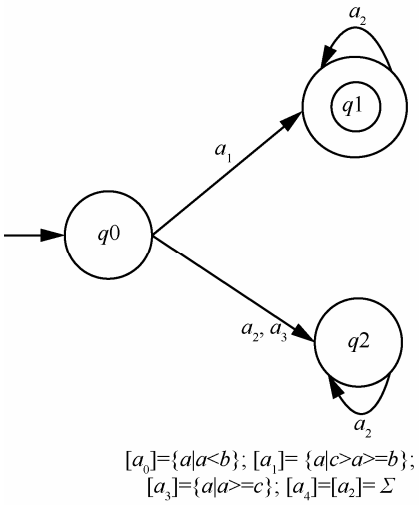

图 10 观察表 T_4 对应的自动机

可以得出 L_a^*算法比 L*算法效率高，例如：当 $\Sigma=\{1,\cdots,100\}$，$b=20$，$c=50$ 时，L*学习算法需要大约 400 次询问，而 L_a^*算法需要的询问次数却不超过 10 次。因为符号算法不受字母表的大小影响，而是由所划分的区间来决定。

4.6 安全协议实例分析

Needham-Schroeder 公钥协议（NSPK 协议）[23] 按功能分为获取公开密钥和双方身份认证 2 部分。这里研究其身份认证部分。

首先使用 Scyther[24]工具对经典的 NS 认证协议进行模型检测，然后运用修正学习算法对攻击图（即反例）进行学习，最后得到改进的 NS 协议，即 NSL 协议[25]。

4.6.1 Needham-Schroeder 协议分析

1) 协议描述（protocol description）：

```
protocol ns3(I,R)
{
    role I
    {
        fresh ni: Nonce;
        var nr: Nonce;

        send_1(I,R, {ni,I}pk(R) );
        recv_2(R,I, {ni,nr}pk(I) );
        claim(I,Running,R,ni,nr);
        send_3(I,R, {nr}pk(R) );

        claim(I,Secret,ni);
        claim(I,Secret,nr);
        claim(I,Alive);
        claim(I,Weakagree);
        claim(I,Commit,R,ni,nr);
        claim(I,Niagree);
        claim(I,Nisynch);
    }

    role R
    {
        var ni: Nonce;
        fresh nr: Nonce;

        recv_1(I,R, {ni,I}pk(R) );
        claim(R,Running,I,ni,nr);
        send_2(R,I, {ni,nr}pk(I) );
        recv_3(I,R, {nr}pk(R) );
        claim(R,Secret,ni);
        claim(R,Secret,nr);
        claim(R,Alive);
        claim(R,Weakagree);
        claim(R,Commit,I,ni,nr);
        claim(R,Niagree);
        claim(R,Nisynch);
    }
}
```

2) 检测结果（scyther results:autoverify）如图 11 所示。

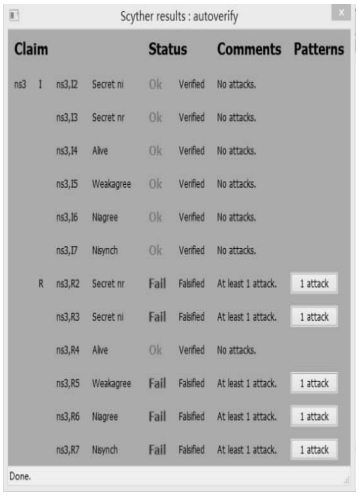

图 11　Scyther 工具对 NS 协议的检测结果

3）攻击图如图 12 所示。

4) 反例学习及协议改进

运用安全协议自适应学习算法对反例（攻击路径或不安全的路径）进行学习，通过循环往复的执行学习算法，将反例的前缀和后缀分别添加到观察表的前缀集合和后缀集合中，直至输出一个满足完备性和一致性的观察表所对应的自动机为止。最后通过协议模型逆向指导安全协议的优化和改进。如 NS 协议存在遗漏身份的攻击，当在协议执行的第 2)步骤加入发送者 *B* 的身份，即可修复该漏洞。下面介绍 NS 协议的改进协议 NSL 协议。

4.6.2 Needham-Schroeder-Lowe 协议

```
protocol nsl3(I,R)
{
    role I
    {
        fresh ni: Nonce;
        var nr: Nonce;

        send_1(I,R, {ni,I}pk(R) );
        recv_2(R,I, {ni,nr,R}pk(I) );
        claim(I,Running,R,ni,nr);
        send_3(I,R, {nr}pk(R) );
        claim(I,Secret,ni);
```

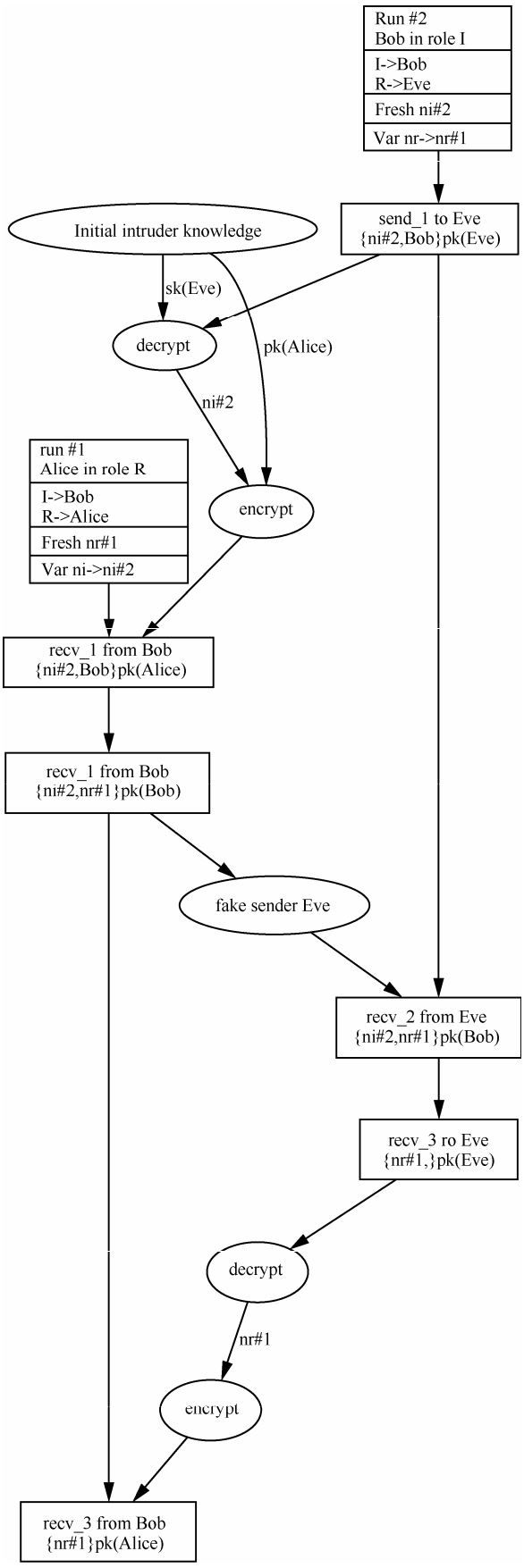

图 12　基于 Scyther 工具的 NS 协议攻击

```
        claim(I,Secret,nr);
        claim(I,Alive);
        claim(I,Weakagree);
        claim(I,Commit,R,ni,nr);
        claim(I,Niagree);
        claim(I,Nisynch);
    }

    role R
    {
        var ni: Nonce;
        fresh nr: Nonce;

        recv_1(I,R, {ni,I}pk(R) );
        claim(R,Running,I,ni,nr);
        send_2(R,I, {ni,nr,R}pk(I) );
        recv_3(I,R, {nr}pk(R) );

        claim(R,Secret,ni);
        claim(R,Secret,nr);
        claim(R,Alive);
        claim(R,Weakagree);
        claim(R,Commit,I,ni,nr);
        claim(R,Niagree);
        claim(R,Nisynch);
    }
}
```

检测结果如图 13 所示。

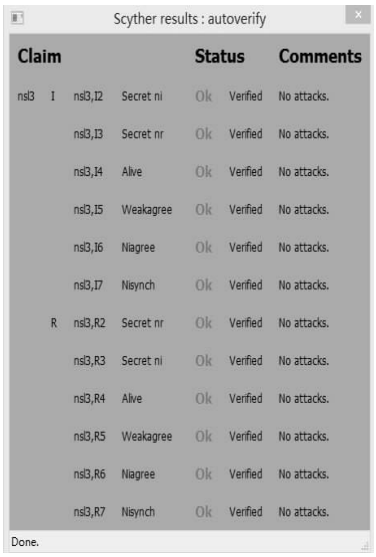

图 13　基于 Scyther 工具的 NS 协议模型检测结果

5　相关工作

当前学术界在自动机学习算法方面已经有很多相关工作。在此简要介绍并与本文的工作进行比较，具体见表 7。

表 7　　相关工作比较

算法	主动学习	大字符集	Inexperienced Teacher	状态机类型
L^*	√	X	X	DFA
L_M^*	√	X	X	DFA
L^{sep}	√	√	X	DFA
L_M^+	√	X	√	DFA
L_{NM}^*	√	X	X	NFA
*CL*1 ,*L*2	√	X	√	DFA
NL*	√	X	X	NFA
RPNI	X	X	X	DFA
DeLeTe2	X	X	X	DFA
L_a^*	√	√	√	DFA

注：表中√表示适用，X 表示不适用。La* 为本文算法。

现有的自动机学习算法从方法上主要分为主动(active)学习算法和被动（passive）学习算法。主动学习算法也称作在线（online）学习算法，被动学习算法也称作离线（offline）学习算法。

被动学习算法得到正反例和负反例的固定集合，包括接受和拒绝的字符串。学习算法提供一个接受正字符串拒绝负字符串的(最小化的)自动机。Oncina 和 Garcia 提出一种被动学习算法 RPNI，该算法是一种推断非必需最小化的 DFA 的有效算法。Denis 等提出一种离线学习算法 DeLeTe2，该算法是在 RPNI 算法的基础上提出来的，文献[8]中阐述领该算法的扩展算法以及学习的 NFA 的离线学习算法。

主动学习算法有可能提出更深的询问，某字符串是否在目标语言中。经典的主动学习算法 Angluin's L*算法基于成员资格询问和等价询问学习一个最小化的 DFA。以 L*为基础，后续又出现多种主动学习算法。Benedikt 等提出 NL*算法，用来学习 NFA 或 RFSA。Muzammil Shahbaz 和 Roland Groz 提出 L_M^*和 L_M^+2 种算法，可以推断出 Mealy 自动机，前者是对 L*的改进，后者在前者的基础上增加了处理反例的新方法。Pacharoen 等提出 L_{NM}^*算法，可以推断可观察的不确定的有穷状态机。Chen 等提出一种主动学习算法 L^{sep}，可以学习并产生区分 DFA。 Martin Leucker 和 Daniel Neider 提出学习正则语言的学习算法 C_{L_1,L_2}，可以在缺乏经验的老师情形下学习，但此时字符集依然是有限的。

6　结束语

安全协议是保障网络安全的基础。安全协议自适应分析框架[9]通过反例指导协议的修正，其中，学习算法起着至关重要的作用。然而，经典的 L* 学习算法并不能满足日益复杂的安全协议的分析需要，为此，本文提出一种改进的安全协议自适应分析算法，即修正学习算法 La*，解决部分教师缺乏经验的问题，并将字符集扩展为大字符集。将该修正学习算法运用到安全协议自适应模型检测中，可以实现对复杂协议的检测结果（较长反例）的学习。修正学习算法 L_a^*将有助于提高安全协议自适应模型检测的效率、降低分析和设计成本、缓解状态空间爆炸并增强协议本身对环境和各种攻击手段的自适应防御能力。

自动机学习算法仍有待于深入研究，比如对上下文无关语言的学习。上下文无关语言包括正则语言，需要对学习正则语言的修正学习算法做进一步推广和改进。

参考文献：

[1] 王亚弟, 束妮娜, 韩继红, 等. 密码协议形式化分析[M]. 北京: 机械工业出版社, 2006.
WANG Y D, SHU N N,HAN J H. Formal Analysis of Security Protocol[M]. Beijing: China Machine Press, 2006.

[2] WOO T, LAM S S. A semantic model for authentication protocols[A]. Proceedings of IEEE symposium on security and privacy[C]. 1993. 178-194.

[3] CLARKE E M, JHA S, MARRERO W. Partial order reductions for security protocol verification[A]. Tools and Algorithms for the Construction and Analysis of Systems[C]. 2000.

[4] SHAMATIKOV V, STERN U. Efficient finite state analysis for large security protocols[A]. Proceedings of 11th IEEE Computer Security Foundation Workshop[C]. 1998. 106-115.

[5] BROADFOOT P J, ROSCOE A W. Internalising agents in CSP protocol models[A]. Proceedings of Workshop on Issues in the Theory of Security[C]. 2002.

[6] HUI M L, GAVIN L. Fault-preserving simplifying transformations for security protocols[J].Journal of Computer Security, 2001,9(1-2):3-46.

[7] STOLLER S D. A Bound on Attacks on Authentication protocols[M]. Foundations of Information Technology in the Era of Network and Mobile Computing. New Yorks Springer US, 2002.

[8] ROSCOE A W. Proving security protocols with model checkers by data independence techniques[A]. Proceedings of 11th IEEE Computer Security Foundations Workshop[C]. 1998. 84-95.

[9] FAN D, ZHANG Y Q. An adaptive formal modeling and analysis schema for security protocols[A]. Chinese Association for Cryptologic Research[C]. ZhengZhou, China, 2014.

[10] MAZHARI S M, MONSEF H, FALAGHI H. A hybrid heuristic and

learning automata - based algorithm for distribution substations siting, sizing and defining the associated service areas[J]. International Transactions on Electrical Energy Systems, 2014, 24(3): 433-456.

[11] 张孝红. 基于形式化方法的安全协议自动化验证算法的研究[D]. 湖南: 湖南大学, 2010.
ZHANG X H. Automated verification algorithm for security protocols based on formal methods[D]. Hunan: Hunan University, 2010.

[12] ANGLUIN D. Learning regular sets from queries and counterexamples[J]. Information and computation, 1987, 75(2): 87-106.

[13] SHAHBAZ M, GROZ R. Inferring Mealy Machines[M]. FM 2009: Formal Methods. Springer Berlin Heidelberg, 2009. 207-222.

[14] BOLLIG B, HABERMEHL P, KERN C, *et al*. Angluin-style learning of NFA[A]. IJCAI[C]. 2009. 1004-1009.

[15] PACHAROEN W, AOKI T, BHATTARAKOSOL P, *et al*. Active learning of nondeterministic finite state machines[J]. Mathematical Problems in Engineering. 2013. 2013(8):1-11.

[16] LINZ P, *et al*. 形式语言与自动机导论[M]. 北京: 机械工业出版社, 2005.
LINZ P, *et al*. An Introduction to Formal Languages and Automata[M]. Beijing: China Machine Press. 2005.

[17] SALOMAA A. Formal Languages[M].New York: Academic Press. 1973.

[18] CHEN Y F, FARZAN A, CLARKE E M, *et al*. Learning minimal separating DFA's for compositional verification[A]. Tools and Algorithms for the Construction and Analysis of Systems[C]. Springer Berlin Heidelberg, 2009.

[19] ALECHA M, HERMO M. A learning algorithm for deterministic finite automata using JFLAP[J]. Electronic Notes in Theoretical Computer Science, 2009, 248: 47-56.

[20] LEUCKER M, NEIDER D. Learning minimal deterministic automata from inexperienced teachers[A]. Leveraging Applications of Formal Methods, Verification and Validation. Technologies for Mastering Change[C]. Springer Berlin Heidelberg, 2012.

[21] MALER O, MENS I E. Learning regular languages over large alphabets[A]. Tools and Algorithms for the Construction and Analysis of Systems[C]. Springer Berlin Heidelberg, 2014.

[22] PAULL M C, UNGER S H. Minimizing the number of states in incompletely specified sequential switching functions[A]. IRE Transitions on Electronic Computers EC-8[C]. 1959.356–366.

[23] NEEDHAM R M, SCHROEDER M D. Using encryption for authentication in large networks of computers[J]. Communications of the ACM, 1978, 21(12): 993-999.

[24] CREMERS C J F. The scyther tool: verification, falsification, and analysis of security protocols[A]. Computer Aided Verification[C]. Springer Berlin Heidelberg, 2008. 414-418.

[25] LOWE G. Breaking and fixing the needham-schroeder public-key protocol using FDR[A].Tools and Algorithms for the Construction and Analysis of Systems[C]. Springer Berlin Heidelberg, 1996. 147-166.

作者简介：

杨京（1990-），女，河南新乡人，西安电子科技大学硕士生，主要研究方向为网络和协议安全。

范丹（1982-），女，河北定州人，博士，中国科学院大学讲师，主要研究方向为网络和协议安全。

张玉清（1966-），男，陕西宝鸡人，博士，中国科学院大学教授、博士生导师，主要研究方向为网络与信息系统安全。