一种基于攻击序列求解的安全协议验证新算法

韩 进 谢俊元
(南京大学计算机软件新技术国家重点实验室 南京 210093)

摘 要 基于完美加密机制前提及D-Y攻击者模型,指出注入攻击是协议攻击者实现攻击目标的必要手段。分析了注入攻击及其形成的攻击序列的性质,并基于此提出了搜索攻击序列的算法,基于该算法实现了对安全协议的验证。提出和证明了该方法对于规则安全协议的搜索是可终止的,并通过实验实现了NS公钥协议的验证。实验结果表明,与OFMC等同类安全协议验证工具相比,该算法不仅能实现安全协议验证自动化,而且由于规则安全协议验证的可终止性,使得本算法更具实用性。

关键词 安全协议验证,攻击序列求解,自动化
中图法分类号 TP393.08 **文献标识码** A

New Security Protocol Verification Approach Based on Attack Sequence Solving

HAN Jin XIE Jun-yuan
(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

Abstract With the premises of the prefect encryption mechanism and the D-Y attacker model, it concluded that the inject attack is the necessarily method for attackers to realize their aims. In this paper, the attributes of inject attack and attack sequences which come from inject attacks were analyzed. Based on those conclusions, it presented an algorithm to determine whether there is an attack sequence in a security protocol. And an new security protocol automatic verification approach was brought up based on this algorithm. It was also proved that the algorithm can be terminated in the verification process for a regular security protocol. In the paper, the NSPK was verified by the algorithm. The experimental results show that compared with other security protocol verification tools, as OFMC, the algorithm can not only realize security protocol automatic verification, but also more practicability for it can be terminated in regular protocol verification process.

Keywords Security protocol verification, Attack sequence solving, Automatic

安全协议是信息安全的基石。安全协议能否实现其设计的安全目标是每个用户与协议设计者所关心的问题,因此安全协议分析与验证一直是信息安全研究领域的热点。当前,安全协议的验证与分析取得了一些重要的研究成果,如D-Y攻击者模型[1]、BAN逻辑类[2]、SPI演算[3,4]、Multiset Rewriting[5]、Strand space[6]、类型系统验证方法[7]等。这些基于方法实现安全协议分析与验证的过程十分复杂,难以实现自动化分析与校验。因此安全协议的自动化验证与分析成为当前研究工作的重点。

本文首先基于完美加密机制前提及协议攻击者的行为分析,指出注入攻击是协议攻击者实现攻击目标的必要手段。协议攻击者为了实现其攻击目标,可能会采取多次注入攻击行为,从而构成了一个攻击路线。本文分析了注入攻击及注入攻击序列的性质,提出一种搜索攻击路线的方法。文中证明了此搜索方法对于规则安全协议是可以终止的,并可实现对NS公钥协议的验证。分析表明,与OFMC[7]等安全协议验证工具相比,本文的验证方法不仅能实现安全协议验证自动化,而且因方法对于规则安全协议验证的可终止性,使得本文方法更具实用性。

本文第1节是注入攻击对于实现安全协议攻击目标的重要性分析;第2节是安全协议的建模与协议数据表述;第3节是注入攻击、注入攻击路线性质的分析;第4节是本文的攻击序列搜索算法;第5节是本文对于NS公钥协议的验证实验与讨论;最后是本文的总结。

1 注入攻击:实现攻击目标的必要手段

安全协议验证分析的目标是验证目标安全协议能否实现其预期的安全性能。安全协议的安全性能与目标有多种,认证性与秘密性是最为基本的两个安全目标。

秘密性(secrecy):一个保守秘密 M 的协议,即使与攻击者存在交互时,也不会将秘密 M 作为消息公开发布,或发布任何可被攻击者利用而使其能得到 M 的任何消息。

到稿日期:2009-11-20 返修日期:2010-01-08 本文受国家自然科学基金(60503021,60721002,60875038),江苏省高新技术计划(BG2007038)资助。

韩 进(1974—),博士生,CCF会员,主要研究方向为网络安全、安全协议分析与设计等,E-mail:hjhaohj@126.com;谢俊元(1961—),教授,博士生导师,主要研究方向为智能系统、智能信息处理等。

认证性(authentication):一个满足认证性属性的加密协议,能使参与协议会话的参与者确认参与会话的其它参与者的身份与各自声名的身份一致。

在安全协议验证研究中,D-Y 攻击者模型及完美加密机制是目前这方面大多数研究者的研究基础。其中 D-Y 攻击者模型界定了安全协议攻击者的行为能力;而完美加密机制则假定安全协议使用的具体加密机制是完善的,从而使安全协议验证工作不再考虑安全协议使用的具体加密机制的安全性。分析 D-Y 模型中定义的协议攻击者的行为能力,可以看出其行为能力主要分为两类,数据操作行为与网络行为。

数据操作行为:D-Y 攻击者与普通的协议参与者一样,具备随机数据生成、加密/解密、组合与拆分数据的行为;

网络行为:通过网络窃听、拦截协议会话数据、向协议会话发送其掌握的会话数据,即注入攻击。

在以上动作中,数据操作行为可视为 D-Y 攻击者内部行为,而窃听、拦截与注入会话数据则可视为 D-Y 攻击者可观测的网络行为。D-Y 攻击者实现其攻击目标时不一定会使用到以上所有动作,但在完美加密机制前提条件下,D-Y 攻击者实现其对安全协议的攻击目标,必然要采用注入攻击行为。

从 D-Y 攻击者模型可知,若 D-Y 攻击者不使用注入攻击,则其能采用的行为有加/解密、组合/折分、生成随机数、窃听/拦截会话数据,分析如下。

秘密性:由安全协议设计原则可知,秘密数据不能公开在网络上传送,也即秘密数据必然要加密后才传送。由完美加密机制假设,攻击者即使窃听了包含有秘密数据的加密消息,也无法破解该数据,因此无法破坏安全协议的秘密性。

认证性:在 D-Y 攻击者不具备注入攻击行为能力的情况下,攻击者对网络数据只能窃听、拦截,因而不可能使被攻击的协议参与者对与其交互的参与者身份产生错误判断。

由以上分析可知,注入攻击是 D-Y 攻击者实现其攻击目标的必要手段。但攻击者要实现注入攻击,前提是必须提供能满足协议会话规则要求的会话数据。而该数据可能来自于多个途径,如公开协议会话数据,窃听于其它协议会话,自生成的,通过对已掌握的数据加以操作获得的。虽然有多种途径,但由网络外部观察攻击者行为可知 D-Y 攻击者实现其攻击目标必然要实现一次或多次注入攻击、网络窃听行为,最终达到其攻击目标,而这些外部行为在时序上构成了一条攻击行为序列。如果一个安全协议中存在有基于 D-Y 攻击者模型的攻击方法,则由上述可知,其必然存在有一条对应的攻击序列,因此对于安全协议的验证可转换成在该协议会话空间中搜索这样的攻击路线的问题。

2 安全协议建模及协议数据类型描述

本文扩展 MSR 模型[4],实现对安全协议的建模。本文的安全协议建模为

$$P=L\overset{+}{\cup}R\overset{+}{\cup}A_1\overset{+}{\cup}\cdots\overset{+}{\cup}A_n \tag{1}$$

式中,P 代表良构的安全协议模型,L,R,A_i 分别代表安全协议的初始化规则集、角色生成规则集以及协议角色规则集。本文主要关注的是协议角色规则集。与 MSR 模型不同的是本文采用以下形式描述角色规则:

$$Rule:state_i.match(N\sigma,M)\rightarrow state_j.\sigma'.M' \tag{2}$$

式中,*state* 代表协议会话中角色所处的状态,*match* 谓词表示协议角色执行的匹配动作,M' 代表规则执行后发送的数据。*match* 谓词中 M 代表接收的会话数据,$N\sigma$ 代表协议角色在当前状态下期望接收的会话数据模式,其中 N 为会话数据类型,σ 为替代集,定义为 $\sigma=\{\langle x,d\rangle\}$。替代集将 N 中的变量映射成具体的实际会话数据,以下本文使用 $dom(\sigma)$ 表示 σ 的变量集合。*match* 谓词表述了协议角色在接收到会话数据后对会话数据的验证过程。显然,只有接收到的会话数据与期望模式相匹配,协议角色才会执行规则,将会话状态切换到下一状态,发送对应的回复数据,并更新替代集。

以下本文将使用 $Agent_i$ 代表协议会话中的诚实参与者,X 代表攻击者,S 代表协议的第三方服务器,X_Know 代表攻击者掌握的会话数据集。P 代表安全协议,Sid 代表一次协议会话,R_i 代表角色,r_i 代表规则,$R_i:r_j$ 表示角色 R 的第 j 个规则。

安全协议中使用的会话数据主要分为 2 类:一类是原子数据,如 Nonce,Key,Name 等;另一类则为复合数据,主要是协议参与者与攻击者针对原子数据使用数据操作生成的会话数据。本文针对安全协议中使用的原子数据的属性引入以下 4 种数据集合:长期数据集合、临时会话数据集合、公开数据集合、私有数据集合,分别记作 Ld,Sd,Pub,Pri。

显然有 $Ld\cap Sd=\emptyset,Public\cap Private=\emptyset$

除此之外,以上数据集合存在有交集。安全协议设计中临时会话数据具有重要的意义,因此临时会话是参与者验证与判别会话数据所属协议会话的必要手段。如果一个协议会话中传递的会话数据不包括临时会话数据,则该协议的参与者必将在会话数据处理上陷入混乱。由此可得出以下结论:

针对安全协议规则左端参与匹配的 $N\sigma$ 模式,本文给出规则安全协议的定义。以下主要针对规则安全协议的攻击路线性质及其搜索算法进行研究。

定义 1 若安全协议 P 中任一规则 r,其左端参与匹配的 $N\sigma$ 模式中要么临时会话数据个数为空,要么包含的临时会话数据有且仅有一个,则 P 为规则安全协议。

3 注入攻击路线相关性质分析

如上所述,安全协议的攻击者为了实现其攻击目的,必然要实施注入攻击。但如本文第 3 节所述,安全协议规则中具有对接收会话数据进行验证的 *match* 谓词,显然为了使被注入的会话接收攻击注入的会话数据,攻击者提供的会话数据一定要与被注入协议会话规则所定义的 *match* 谓词相成功匹配。而该数据只能由 D-Y 攻击者从以下两个途径中获得:

从 X_Know 数据集中,由数据操作方式求出;

通过对被注入协议会话窃听所得的数据集,生成注入数据,注入到其它协议会话中,再窃听其它协议会话生成的数据,添加到 X_Know 数据集,并使用前一条途径,以期获得待求解的目标数据。

值得注意的是,当攻击者向其它协议会话注入数据后,又可能会使其它被注入的会话执行受阻,因为攻击者干扰了它们的正常执行。攻击者为了获得目标数据,则又须提供会话数据,使其它被注入的会话能顺利执行。攻击者若无法从其 X_Know 数据集中直接求解获得这些数据,则其需要重新采用上述两个途径来获得,从而形成了对目标数据的递归求解过程。

因此，攻击者为了实现最终的注入目标，必须通过上述的两条途径来反复求解最终所需的注入数据。从网络行为观察来看，攻击者的注入行为与窃听行为可以构成一系列的攻击行为序列，以下称之为攻击序列 L。D-Y 攻击者向不同的协议会话注入可能求解出目标会话数据的会话数据，并对被注入的协议会话进行窃听。在攻击者这一系列针对求解目标的攻击行为的形成过程中不可能有循环出现，因为一个会话的 $R:r$ 规则被触发后，不会再次触发。针对攻击序列 L 及规则安全协议有以下结论。

命题 1 对于安全协议 P，D-Y 攻击者求解目标数据形成的攻击序列 L 中包含的每一个会话都处于受阻状态。

证明：由上所述的攻击者求解目标数据的过程直接可得。

定理 1 针对规则安全协议 P，D-Y 攻击者求解目标数据形成的攻击序列 L 为树结构。

证明：由规则安全协议的定义及命题 1 分析可得以下结论：

由规则安全协议定义可得，2 个不同的会话数据不会同时注入到攻击序列中同一协议会话的同一规则中。设这 2 个不同的会话数据分别为 $D0$，$D1$。

若 $D0$，$D1$ 的注入能求解出目标数据，显然 $D0$，$D1$ 中必然包含目标数据中的临时会话数据。

由规则安全协议定义，其任一规则的会话数据模式中只含有一个临时会话数据。若注入 $D0$，$D1$ 能求解出目标数据，则要么 $D0$ 包含该数据，或 $D1$ 包含该数据，或 $D0$，$D1$ 都包含该数据。显然此时使用 $D0$，$D1$ 单独注入该协议会话，都可以获得目标数据。

由此可知，攻击序列 L 不会出现两个会话数据注入到同一协议会话的同一规则中，因此 L 结构中不会出现环结构。

由命题 1 可知，L 结构中会话都处于受阻状态。一旦有受阻会话求解出目标数据，则由该受阻会话出发形成的求解步骤都不需要在 L 结构中继续保留，也即从 L 结构中删除。因此，L 结构中保留的会话一定都处于受阻状态。而攻击者求解目标数据时，选择被注入的会话只能针对 L 结构外的会话。由此可知，L 结构中不会出现各协议会话数据之间的相互注入，而形成注入攻击序列的环结构。

又由协议规则的执行可知，一旦一个协议会话在某个规则被注入后，就不可能再次注入该协议会话的同一规则，因为协议会话的状态已切换成下一状态。由此可知，L 结构中不会出现循环结构。

综上所述，可知本定理成立。

由 D-Y 攻击者的求解途径的第二条可知，在发动每一次注入攻击之前，攻击者都需要准备用于求解目标数据的会话数据集，以及选择待注入的会话。针对这两方面的数据集，本文采用$\langle N\sigma,\Omega\rangle$的二元组表示，其中 $N\sigma$ 表示待匹配的会话数据模式，Ω 则代表向其它会话注入的数据集。针对该数据集有以下结论。

命题 2 $\langle N\sigma,\Omega\rangle$中若 $N\sigma$ 包含临时会话数据不为空，则 Ω 的数据集只需包含 $N\sigma$ 所属的协议会话生成的数据。

证明：由临时会话数据性质可知，其只生成于 $N\sigma$ 所对应的会话，因此 Ω 数据集中只需包含 $N\sigma$ 所属的协议会话生成的数据即可。

当攻击者准备好$\langle N\sigma,\Omega\rangle$的求解数据集 Ω 后，由于安全协议会话空间中包含无限的会话，而同一个数据集可以注入到多个会话中，因此对于攻击者选择待注入协议会话的问题，本文先给出以下相似会话定义，并对相似会话有以下定理。

定义 2 安全协议 P 的两个会话 Sid，Sid'中，若 $R=R'$，则称 Sid，Sid'为相似协议会话，记作 $Sid\cong Sid'$。其中 R，R' 分别为 Sid，Sid'会话角色生成阶段规则。

定义 2 表明具有相似会话关系的协议会话其参与会话的参与者集合相等，且同一参与者在这两个会话中担任同一角色。针对相似会话的注入，有以下结论。

定理 2 针对规则安全协议的攻击序列 L 中，攻击者求解$\langle N\sigma,\Omega\rangle$时，不会将数据 D，$D\in\Omega$ 注入到 Sid，Sid'同一规则中，其中 $Sid\cong Sid'$。

证明：由上述定理 1 中的论点及相似会话性质分析可得：

在定理 1 的证明中，得到了攻击者在求解目标数据时，选择被注入的会话只能针对 L 结构外的会话这一结论点。由此可知，D 既然能注入到 Sid，Sid'同一规则中，则显然该规则的会话数据模式包含的临时会话数据为空。

设 D 注入到 Sid，Sid'生成的会话数据 $D0$，$D1$，由相似会话性质可得，$D0$，$D1$ 中除了在 Sid，Sid'中生成的临时会话数据之外，数据类型、长期会话数据，包括由 D 注入引入的临时会话数据必然完全相等。

因此若 $D0$ 能满足 $N\sigma$，则 $D1$ 也能满足，反之亦然。

综上所述，可知本定理成立。

由定理 2 可得，攻击者求解$\langle N\sigma,\Omega\rangle$时，由 $N\sigma$ 出发求解形成的求解序列 L 为树状结构，则$\langle N\sigma,\Omega\rangle$所对应的会话为 L 树的根结点，而该树的叶子结点则为攻击者正在求解的目标数据所对应的协议会话。由根结点出发至叶子结点形成一条路径，针对该路径，本文有定理 3。

定理 3 针对规则安全协议的攻击序列 L 中包含的路径不会出现 Sid，Sid'，$Sid\cong Sid'$受阻于同一状态。

证明：由于 Sid，Sid'位于同一路径，必然在位置上有前后之分。设 Sid 在前，Sid'在后，则有以下分析。

设针对 Sid 的求解有二元组$\langle N\sigma,\Omega\rangle$，对于 Sid'求解有$\langle N\sigma',\Omega'\rangle$，显然有定理 1 中攻击者在求解目标数据时，选择被注入的会话只能针对 L 结构外的会话这一结论点。由相似会话性质，可得若有一个数据 $D0\in\Omega$。注入到会话 $Sid0$ 的某一规则中，则 Ω'中必然有一个数据 $D1$，可以注入到 $Sid1$ 的同一规则中，且 $Sid0\cong Sid1$。

又由相似会话性质，若 $Sid0$ 被注入后能顺利结束，则 $Sid1$ 被注入后也必然能顺利结束。否则若 $Sid0$ 受阻于某一状态，则 $Sid1$ 必受阻于同一状态。

由上述可得，若针对$\langle N\sigma,\Omega\rangle$的求解必须由$\langle N\sigma',\Omega'\rangle$的求解成功为前提条件，则针对$\langle N\sigma',\Omega'\rangle$的求解必然也要以$\langle N\sigma'',\Omega''\rangle$的求解成功为前提。而$\langle N\sigma'',\Omega''\rangle$对应的会话 Sid''，有 $Sid''\cong Sid'$，且 Sid''与 Sid'受阻于同一状态。

由此可知，针对$\langle N\sigma,\Omega\rangle$的求解路径会形成无穷递归，因此$\langle N\sigma,\Omega\rangle$的求解不可能以$\langle N\sigma',\Omega'\rangle$的求解成功为前提。

因而在求解路径中对$\langle N\sigma',\Omega'\rangle$的求解是不必要的。

综上所述，可得本命题成立。

4 基于攻击路线搜索的安全协议验证算法

通过上述对注入攻击序列 L 的性质分析，本文提出针对

规则安全协议的基于攻击路线搜索的安全协议验证算法如下。设有一规则安全协议 P，含有 N 个角色(除服务器 S 外)，初始化攻击者的知识空间 X_Know，及注入序列 L 为空。给出 N 个不同参与者集合{Agent$_i$，$1\leqslant i\leqslant n$}，攻击者 X 生成一个协议会话，以下记为 S，算法分为 3 个阶段进行。

第一阶段　1)初始化攻击者知识空间 X_Know，使 $X\_Know=Msg\cup Public$；

2)分析协议规则 P 的某一规则 $R:r$，若其 $N\sigma$ 为空，则 X_Know 必然有消息数据 N 可注入该规则，则针对该规则注入 N，并将该注入动作添加入注入序列 L 中，然后观察被注入会话 Sid 的执行情况。

第二阶段　1)若 Sid 执行完毕，则返回 L，分析注入攻击对协议安全属性的破坏情况；

2)若 Sid 受阻，则生成求解二元组$\langle N\sigma,\Omega\rangle$，针对 Ω 中每个可注入的数据注入到 L 结构外的可注入协议会话中。这些会话要么不具备相似会话关系，要么注入到相似会话的不同规则中，将所有注入攻击添加到 L 中；

3)窃听被注入会话生成的数据，并添加到 X_Know 中；

4)针对 X_Know 求解目标数据。若求解出目标数据，则返回 L，分析注入攻击对协议安全属性的破坏情况；

5)若求解不出目标数据，则针对以上被注入会话 Sid，遍历 L。若 Sid 所处路径中存在 Sid' 会话，有 $Sid'\subseteq Sid$，且处于同一受阻状态，则删除 Sid 会话。对于保留的会话，递归调用执行本阶段第一步。

第三阶段　1)若第二阶段求解过程中找不到任一新的会话可以注入，则由当前 X_Know 集合中求解可注入到 S 中的会话数据；

2)若没有可注入到 S 中的会话数据，则验证结束，协议 P 无安全漏洞。若有，则将数据注入到 S 中，执行第一阶段。

以上验证过程是基于上述的注入攻击性质及求解序列 L 的性质实现的。验证过程一共要进行两次，分别对应攻击者 X 参与或不参与的协议会话。针对该算法，本文有以下结论。

定理 4　基于攻击路线搜索的安全协议验证算法对于规则安全协议是可终止的。

证明：基于上述求解序列 L 的性质可得以下结论：

由定理 1 可得，针对一求解二元组，攻击者可注入的目标协议会话必然是有限的。因为攻击者不能将一个数据注入到不同的相似会话的同一规则中，而规则安全协议中参与会话的角色是有限的，因此不具备相似会话关系的会话数据是有限的，则协议的规则是有限的，由此本论点成立。

根据定理 1 所得的结论，对于规则安全协议，求解序列 L 为树，因此若本算法不能终止，则结合上一论点可知，序列 L 中必存在一条无限长的求解路径。而由定理 3 可知，求解路径中不可能出现具有相似会话关系的两个协议会话受阻于同一状态，因此序列 L 中不可能出现无限长的求解路径。

综上所述，可知本定理成立。

5　实验与分析

本文使用基于约束的目标求解加密协议验证算法实现了 Needham-Schroeder 公钥协议的验证。NS 协议是出现较早的认证协议，也是一个具有重要影响的加密协议。NS 协议可分为私钥体制和公钥体制两种版本，本文验证的是其公钥协议版本，其非形式化中存在的攻击漏洞如下所示[9]。

$(1)A\rightarrow X:\{Na,A\}\_PKX;(1)'\ X\rightarrow B:\{Na,A\}\_PKB$

$(2)X\rightarrow A:\{Na,Nb\}\_PKA;(2)'\ B\rightarrow X:\{Na,Nb\}\_PKA$

$(3)A\rightarrow X:\{Nb\}\_PKX;(3)'\ X\rightarrow B:\{Nb\}\_PKB$

显然该协议满足本文的规则安全协议定义。针对该协议，本文采用上述算法验证如下。

1)首先利用无攻击者 X 参与会话实施注入攻击。如图 1 所示，本文使用的注入数据来自 A 为发起者、B 为参与者的正常协议会话，注入数据为$\{nonce1,A\}_PKB$，从而形成求解序列。从图中可以看到，求解序列形成的树最后都终止了，但终止的原因不一。其中 Session1，Session4 的终止是因为求解数据集 Ω 中的数据无法再注入到求解序列 L 外的会话中。而 Session3 终止的原因是其路上已有 Session2 与其有相似会话关系，且受阻于同一规则，因此由定理 3，不需再求解而终止。

2)图 1 所示的是求解算法中的第二阶段。此后还要进行第三阶段，最终算法终止，利用无攻击者 X 参与会话中生成的会话数据发动注入攻击，都失败了。

3)再利用攻击者参与的会话中生成的数据实施注入攻击，对 NS 协议进行验证。算法终止后，得到表 1 所列的 4 条成功攻击路径。

表 1　有攻击者 X 参与的会话所生成的数据形成的成功注入求解序列

(1)A→X:{nonce1,A}_PKX (2)'X→A:{nonce1,nonce16}_PKA (3)A→X:{nonce16}_PKX	(1)'X→A:{nonce1,A}_PKA (2)A→X:{nonce1,nonce16}_PKA	(1)A→X:{nonce1,A}_PKX (2)'X→A:{nonce1,nonce18}_PKA (3)A→X:{nonce18}_PKX	(1)'B(X)→A:{nonce1,A}_PKA (2)A→B(X):{nonce1,nonce18}_PKA
(1)A→X:{*nonce1*,A}_PKX (2)'X→A:{nonce1,nonce27}_PKA (3)A→X:{nonce27}_PKX	(1)'X→B:{*nonce1*,A}_PKB (2)B→X:{nonce1,nonce27}_PKA	(1)A→X:{*nonce1*,A}_PKX (2)'X→A:{nonce1,nonce29}_PKA (3)A→X:{nonce29}_PKX	(1)'A(X)→B:{nonce1,A}_PKB (2)B→A(X):{nonce1,nonce29}_PKA

4)针对上述 4 次注入分析可知，对于表 1 左上的注入路径，由于两轮的注入都涉及到攻击者 X，没有意义，因此不予讨论。而对于表 1 左下角与右下角的注入路径，显然就是上面提及的注入攻击方法。而表 1 右上角的注入路径则表明了一种新的攻击，虽然这种攻击不能造成协议在秘密性、认证性方面的破坏，但是它也发现了协议在具体实现时可能存在的漏洞，必须要在实现时对接收的消息所声明的参与者身份进行判断。从这一点来看，这种攻击方法对于实现 NS 加密协议也是有一定意义的。

5)实验的过程与界面如图 2 所示，在无攻击者参与的会话中生成的数据实施注入攻击验证过程中，攻击者掌握的数据量一共有 552 个，而第二次验证中攻击者掌握的数据量为 2292 个。

(下转第 53 页)

[14] Mennie D, Pagurek B. An Architecture to Support Dynamic Composition of Service Components[C]// the Fifth International Worshop on Component-Oriented Programming-WCOP 2000. held in conjunction with ECOOP 2000, Sophia Antipolis, France, June 2000

[15] Tosic V, Mennie D, Pagurek B. On Dynamic Service Composition and Its Applicability to E-Business Software Systems[C]// WOOBS'01(Workshop on Object-Oriented Business Solutions) workshop(at ECOOP 2001). Budapest, Hungary, June 2001

[16] Fujii K, Suda T. Loose Interface: An Extended Interface Definition for Dynamic Service Composition[C]// Proc. of the First Annual Symposium on Autonomous Intelligent Netwrks and Systems. Los Angeles, CA, May 2002

[17] Wang jiacun, Deng Yi, Xu Guang. Reachability analysis of real-time system using Petri nets[J]. IEEE Trans. on system, Man and Cybernetics, 2000, 30(5)

(上接第 35 页)

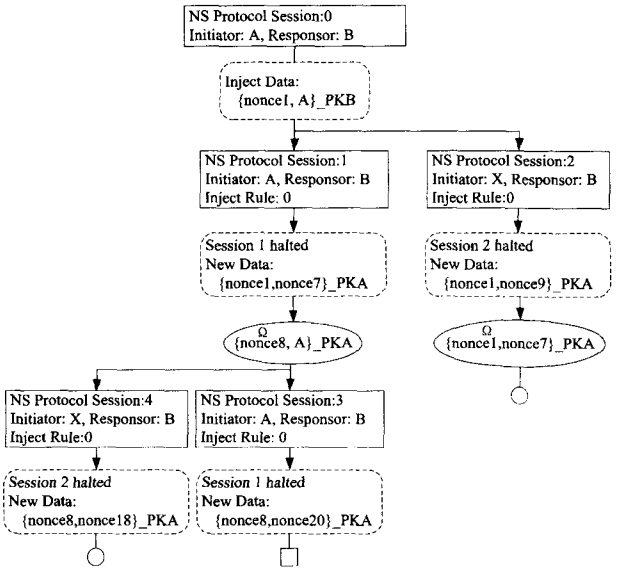

图 1 无攻击者 X 参与的会话所生成的数据形成的注入求解序列

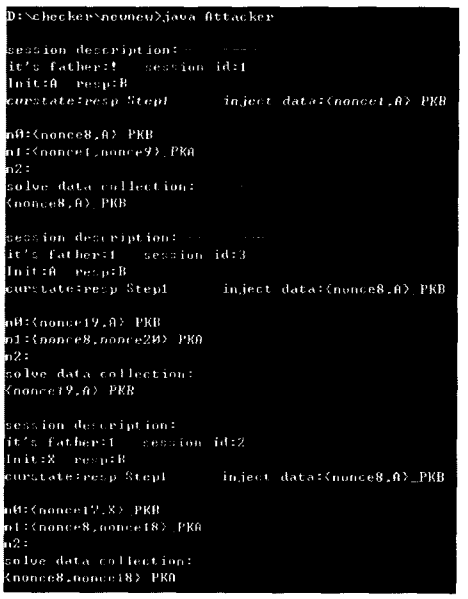

图 2 实验结果

结束语 本文分析了注入攻击是安全协议攻击者实现其攻击目标的必要手段，并基于注入攻击和注入攻击序列的性质，提出了一种基于攻击序列搜索的安全协议验证算法。本文分析了攻击序列的性质，并证明了对于规则安全协议的验证算法的可终止性。

本文的算法与其它同类方法，如模型检测、理论证明等方法相比有以下优点：

基于本方法实现的自动化检验工具对于规则安全协议一定会终止并给出确定结果。其它同类工具，如 OFMC[7] 工具虽然能在发现漏洞时终止，但是用户很难判断其运行状态，不知道是协议没有漏洞还是需要继续等待下去。而其它方法经常会出现无解的情况和不能终止的情况。因此本算法与之相比，其终止性更具实用性。

基于本方法实现的自动化检验工具能给出具体的攻击方法。与理论证明方法不同的是，它们虽然能证明可能存在的安全漏洞，却未必能给出具体攻击方法，如 SPI[3,4]，BAN[2] 等。

基于本方法便于实现并行检验，因为本方法是针对安全协议的每个可实施注入攻击的规则进行的，因此可将检验工作拆分成多个子任务进行验证。

本文算法的缺点在于目前方法只适用于规则安全协议。下一步的工作将对算法进行扩展，使其能实现更多类型的安全协议的验证。

参 考 文 献

[1] Syverson P, Meadows C, Cerversato I. Dolev-Yao is no better than Machiavelli[C]// Degano P, ed. Proceedings of the First Workshop on Issues in the Theory of Security. Geneva, Switzerland, 2000: 87-92

[2] Burrows M, Abadi M, Needham R. A logic of authentication[C]// Proceedings of the Twelfth ACM Symposium on Operating Systems Principles. 1989: 1-13

[3] Abadi M, Gordon A D. A calculus for cryptographic protocols: the spi calculus[C]// Graveman R. ed. Proceedings of the 4th ACM Conference on Computer and Communications Security. Zurich, 1997: 36-47

[4] Abadi M, Gordon A D. Reasoning about cryptographic protocols in the spi calculus[C]// Proceedings of the 8th International Conference on Concurrency Theory. 1997: 59-73

[5] Durgin N A, Lincoln P D, Mitchell J C, et al. Multiset rewriting and the complexity of bounded security protocols[J]. Journal of Computer Security, 2004

[6] Javier F, Herzog J C, Guttman J D. Strand Spaces: Proving Security Protocols Correct[J]. Journal of Computer Security, 1999, 7 (2/3): 191-230

[7] Abadi M, Blanchet B. Secrecy Types for Asymmetric Communication[C]// Honsell F, Miculan M, eds. Proceedings of the 4th International Conference on Foundations of Software Science and Computation Structures. London, UK: Springer-Verlag, 2001: 25-41

[8] Basin D, Modersheim S, Vigano L. An On-the-Fly Model-Checker for Security Protocol Analysis[C]// Proc. Eighth European Symp. on Research in Computer Security. 2003: 253-270

[9] Lowe G. An attack on the Needham-Schroeder public key authentication protocol[J]. Information Processing Etters, 1995 (3): 131-136