第33卷 第12期 计 算 机 工 程 2007年6月
Vol.33 No.12 Computer Engineering June 2007
·博士论文· 文章编号：1000—3428(2007)12—0001—03 文献标识码：A 中图分类号：TP309

基于图元的事件图生成算法

王焕宝[1]，张佑生[2]
(1. 安徽建筑工业学院数理系，合肥 230022；2. 合肥工业大学计算机与信息学院，合肥 230009)

摘 要：与单轮运行情形不同，多轮并发运行的密码协议存在更为复杂的安全性问题。并发运行密码协议的形式化分析对象包括密码协议的多轮并发运行和多个密码协议的并发运行两种情形，且二者具有统一的形式化模型。基于扩展的串空间模型和Spi演算理论，提出用于并发运行密码协议安全属性验证的事件图模型。图元是事件图的构造单元，它满足消息事件之间的通信关系和前驱关系约束以及消息语句的新鲜性约束。定义消息事件之间、图元之间以及消息事件和图元之间的前缀、组合和选择运算，并给出事件图生成算法。
关键词：密码协议；事件图；形式化模型；图元

Meta-graph-based Algorithm for Composing Event Graphs

WANG Huanbao[1], ZHANG Yousheng[2]
(1. Department of Maths & Physics, Anhui Institute of Architecture, Hefei 230022;
2. School of Computer Science & Information Technology, Hefei University of Technology, Hefei 230009)

【Abstract】 There are lots of complex security problems in the cryptographic protocols in lots of runs, which are not different from one run. The objects of formal analysis for cryptographic protocols in a concurrent way include the two ones: a cryptographic protocol in lots of runs and cryptographic protocols in a concurrent way. The above-mentioned objects are provided with the unified formal model. The event graph model is introduced based on the extended model of strand space and Spi calculus. It is a formal model for proving the safety properties of the cryptographic protocols in a concurrent way correctly. The meta-graph is a unit for composing event graphs. It meets all the restrictions in the communication and causal predecessor relation between events, and in the freshness of terms. The prefix, parallel composition, and choice operation among events and meta-graphs respectively are defined, and the meta-graph-based algorithms for composing event graphs are proposed.
【Key words】 Cryptographic protocol; Event graph; Formal model; Meta-graph

密码协议，又称安全协议，它是网络通信和应用必不可少的组件之一。协议验证是协议设计与实现中最为关键的环节之一，它可以通过协议分析和协议综合两种途径来实现。两者相比，协议分析则更为有效和实用。近年来，研究人员相继提出多种用于密码协议分析的形式化建模理论及其安全属性验证方法，如串空间模型[1]、Spi演算理论[2]以及Paulson归纳法等。

本文将扩展的串空间模型[3]和Spi演算理论相结合，提出一种新的用于并发运行密码协议安全属性验证的事件图 模型。

1 事件图及其事件之间的约束

处于并发运行状态下密码协议的主体可扮演多个角色，记之为主体角色。主体角色的行为集是由发送消息和接收消息动作构成的多重集。将主体角色完成一次发送或接收消息动作记为由其触发的一个消息事件。消息事件之间存在通信关系和先后次序关系。以下用事件图描述事件集合及其事件之间的联系。

定义 1 对于所有参与安全协议运行的主体角色在安全协议完整并发运行中触发的消息事件的全体N及其消息事件之间的通信关系和次序关系$\rightarrow\cup\Rightarrow$，对应的安全协议的（消息）事件图记为$G=<N, \rightarrow\cup\Rightarrow>$。

定义 2 若g是由事件图G分割得到的局部(子图)，记为$g=<N_g, (\rightarrow\cup\Rightarrow)_g>\subseteq G$，那么，它一定满足以下的分割约束：

(1)若$n\in N_g$，且$term(n)\neq uns_term(n)$，则惟一存在m，使得$m\rightarrow_g n$；

(2)若$n\in N_g$，且$m\Rightarrow n$，则$m\Rightarrow_g n$；

(3) 若m、$n\in N_g$，则存在由$\rightarrow_g$和$\Rightarrow_g$构成的集合$E_g\subseteq(\rightarrow\cup\Rightarrow)_g$，使得消息事件$m$和$n$可由$E_g$关联。

符号说明：记消息事件n的语句(term)为$term(n)$。若消息事件为主体角色发送消息，则其语句记为$+uns_term(n)$；若消息事件为主体角色接收消息，则其语句记为$-uns_term(n)$。

可将同一主体角色触发的所有消息事件按其发生的先后次序排列形成主体角色序列。若主体角色为i，通信事件在主体角色序列上的位置为j，那么，该消息事件记为(i, j)。以下命题给出消息事件之间的通信关系约束。

命题 1 对于消息事件$(i, j), (r, h)\in N_g$，若$(i, j)\rightarrow_g (r, h)$，则$\nexists((r, h)\rightarrow_g (i, j))$；对于消息事件$(i, j_1), (i, j_2)\in N_g$，$\nexists((i, j_1)\rightarrow_g (i, j_2))$，或者$\nexists((i, j_2)\rightarrow_g (i, j_1))$。

证明 由事件之间通信关系可得。

对于事件$(i, j)\in N_g$，其语句$uns_term((i, j))$可表示为其分量的连接运算形式$uns_term((i, j))=join(a_1, a_2, \ldots, a_m)$，其中，集合$\{a_1, a_2, \ldots, a_m\}$中包含的时新子句集为$S_{fresh}^{(i, j)}$。以下命题

基金项目：国家自然科学基金资助项目(50379003)；高等学校博士学科点专项科研基金资助项目(20050359012)；安徽省自然科学基金资助项目(03042207)
作者简介：王焕宝(1963－)，男，博士、副教授，主研方向：并发理论，安全协议分析与设计；张佑生，教授、博导
收稿日期：2006-06-24 **E-mail：**wanghuanbao@yahoo.com.cn

及其推论刻画出事件之间的先后次序关系约束和消息语句的新鲜性约束。

命题 2 已知安全协议完整并发运行生成的事件图$G=<N_G, (\Rightarrow\cup\rightarrow)_G>$。若$(i, j), (r, h)\in N_G$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}$的任一元素均惟一源发于消息事件$(i, j)$，则$(i, j)\preccurlyeq_G(r, h)$，其中$\preccurlyeq_G=(\Rightarrow\cup\rightarrow)_G^*$是$(\Rightarrow\cup\rightarrow)_G$的自反、传递闭包。

证明 利用反证法证明。假设$(i, j)\not\preccurlyeq_G(r, h)$，因此，$(i, j)\not\Rightarrow_G^*(r, h)$。$\exists(r', h')\in N_G\wedge term((r', h'))\neq uns_term((r', h'))$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\sqsubset S_{fresh}^{(r', h')}$，且由时新子句的新鲜性推知，$(r', h')\Rightarrow_G^*(r, h)$。其中，$S_{fresh}^{(r', h')}$是消息事件$(r', h')$的无符号语句包含的时新子句集。

此外，事件图 G 是一个最大的局部 g，它满足：

(1)无环性；

(2)事件之间的通信约束条件，即对于消息事件$(x, y)\in N_G$，且$term((x, y))\neq uns_term((x, y))$，则惟一存在$(x', y')$，使得$(x', y')\rightarrow_G(x, y)$。

因此，对于消息事件$(r_1, h_1)\in N_G$，且$term((r_1, h_1))=uns_term((r_1, h_1))$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\subseteq S_{fresh}^{(r_1, h_1)}$，且对于任一消息事件$(r_2, h_2)\in N_G\wedge term((r_2, h_2))\neq uns_term((r_2, h_2))$，如果$(r_2, h_2)\Rightarrow_G^*(r_1, h_1)$，那么，$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\not\subseteq S_{fresh}^{(r_2, h_2)}$。其中，$S_{fresh}^{(r_1, h_1)}$和$S_{fresh}^{(r_2, h_2)}$分别是事件$(r_1, h_1)$和$(r_2, h_2)$的无符号语句包含的时新子句集。

对于消息事件(i, j)，$(i, j)\not\Rightarrow_G^*(r_1, h_1)$，否则，$(i, j)\preccurlyeq_G(r, h)$。由此可得，事件$(i, j)$和$(r_1, h_1)$都同属于事件图$G$，那么，这与$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}$中时新子句的新鲜性矛盾。因此，$(i, j)\preccurlyeq_G(r, h)$。其中，如果对于$(r, h)\wedge term((r, h))=uns_term((r, h))$，且$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\subseteq S_{fresh}^{(r,h)}$，那么，$(i, j)\preccurlyeq_G(r, h)\wedge(r, h)\preccurlyeq_G(i, j)$，从而，$(i, j)=(r, h)$。

推论 1 已知安全协议完整并发运行生成的事件图$G=<N_G, (\Rightarrow\cup\rightarrow)_G>$。若$(i, j), (r, h)\in N_G$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}$的任一元素均惟一源发于事件$(i, j)$，且$\exists(r', h')\in N_G\wedge term((r', h'))\neq uns_term((r', h'))$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\subseteq S_{fresh}^{(r', h')}$，则$(r', h')\Rightarrow_G^*(r, h)$。其中$S_{fresh}^{(r', h')}$是消息事件$(r', h')$的无符号语句包含的时新子句集，$\Rightarrow_G^*$是$\Rightarrow_G$的自反、传递闭包。

推论 2 已知安全协议完整并发运行生成的事件图$G=<N_G, (\Rightarrow\cup\rightarrow)_G>$。若语句$a$惟一源发于事件$(i, j)\in N_G$，且$\exists(r, h)\in N_G\wedge term((r, h))\neq uns_term((r, h))$，使其与事件$(i, j)$相关，亦即$a\sqsubset i_j\wedge a\sqsubset r_h$，且当主体角色集$R=\{i, r\}$时，那么，$i=r\wedge j<h$。

限于篇幅，略去推论 1 和推论 2 的证明。

2 图元

定义 3 设$\rightarrow_D\subseteq\rightarrow$，$\Rightarrow_D\subseteq\Rightarrow$，且$D=<N_D, (\rightarrow_D\cup\Rightarrow_D)>=<N_D, (\rightarrow\cup\Rightarrow)_D>$是安全协议完整并发运行形成的事件图$G=<N_G, (\Rightarrow\cup\rightarrow)_G>$的有限非循环子图，记为图元$D$，则：

(1)若$(r, h)\in N_D$，且$term((r, h))\neq uns_term((r, h))$，则惟一存在事件$(i, j)$，使得$(i, j)\rightarrow_D(r, h)$；

(2)若$(i, j)\Rightarrow_G(i, j+1)$，且消息事件$(i, j+1)\in N_D$，则$(i, j)\Rightarrow_D(i, j+1)$；

(3)若$(i, j), (r, h)\in N_D$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}$的任一元素均惟一源发于消息事件$(i, j)$，那么：1)$(i, j)\preccurlyeq_D(r, h)$，或 2)$\exists(r', h')\in N_D$，且$term((r', h'))\neq uns_term((r', h'))$，使得$S_{fresh}^{(i,j)}\cap S_{fresh}^{(r,h)}\subseteq S_{fresh}^{(r', h')}$，且$(r', h')\not\preccurlyeq_D(i, j)\wedge(r', h')\Rightarrow_D^*(r, h)$。其中，$S_{fresh}^{(i,j)}$，$S_{fresh}^{(r,h)}$和$S_{fresh}^{(r', h')}$分别是事件$(i, j)$，$(r, h)$和$(r', h')$的无符号语句包含的时新子句集。此外，$\Rightarrow_D^*$是$\Rightarrow_D$的自反、传递闭包。

(4)若$(i, j), (r, h)\in N_D$，则存在由$\rightarrow_D$和$\Rightarrow_D$构成的集合$E_D\subseteq(\rightarrow\cup\Rightarrow)_G$，使得事件$(i, j)$和$(r, h)$可由$E_D$关联。

上述定义的图元为事件图构造单元，它满足消息事件之间的通信关系和前驱关系约束以及消息语句的新鲜性约束。以下定义前缀、组合和选择算子用于消息事件之间、图元之间以及消息事件和图元之间的合成运算。

3 算子

定义 4 对于图元$B=<N_B, (\rightarrow_B\cup\Rightarrow_B)>$以及消息事件$n\notin N_B$，且$\alpha=term(n)$，如果$\exists p=n\mapsto m$[4]，使得$m\in N_B$，那么，称$\alpha.B$为消息事件$n$的触发行为$\alpha$作用于$B$的前缀(顺序)运算。其中，符号“.”为前缀(顺序)算子。

命题 3 前缀(顺序)运算$\alpha.B$为图元。其中，$B=<N_B, (\rightarrow_B\cup\Rightarrow_B)>$为图元，$\alpha$为事件$n\notin N_B$的触发行为。

证明 由上述前缀算子和图元的定义可知，通过添加关系(通信关系或前驱关系)$E_\alpha=(\rightarrow_\alpha\cup\Rightarrow_\alpha^*)$，可将由触发行为$\alpha=\{$主体发送消息，主体接收消息$\}$触发的主体角色消息事件$n$合成到已知的图元$B$中，形成$B^+=<N_B\cup\{n\}, (\rightarrow_B\cup\Rightarrow_B)\cup E_\alpha>=<N_B^+, (\rightarrow_B\cup\Rightarrow_B)^+>$。其中，$B^+$为图元，$N_B^+=N_B\cup\{n\}$，且$(\rightarrow_B\cup\Rightarrow_B)^+$满足：

(1)若$\exists m\in N_B$，使得$n\rightarrow_\alpha m$，且$\rightarrow_\alpha\in\rightarrow_B$，那么，$N_B^+=N_B\cup\{n\}\wedge(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)$；若$\exists m\in N_B$，使得$n\rightarrow_\alpha m$，且$\rightarrow_\alpha\notin\rightarrow_B$，那么，$N_B^+=N_B\cup\{n\}\wedge(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)\cup\{\rightarrow_\alpha\}$。若$\exists n'$，使得$n'\Rightarrow n$，那么，$n'\Rightarrow_\alpha n$，且$(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)\cup\{\rightarrow_\alpha, \Rightarrow_\alpha\}$。

(2)若$\exists m\in N_B$，使得$n\Rightarrow_\alpha m$，且$\Rightarrow_\alpha\in\Rightarrow_B$，那么，$N_B^+=N_B\cup\{n\}\wedge(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)$；若$\exists m\in N_B$，使得$n\Rightarrow_\alpha m$，且$\Rightarrow_\alpha\notin\Rightarrow_B$，那么，$N_B^+=N_B\cup\{n\}\wedge(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)\cup\{\Rightarrow_\alpha\}$。若$\exists n'$，使得$n'\Rightarrow n$，那么，$n'\Rightarrow_\alpha^{(1)}n$，且$(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)\cup\{\Rightarrow_\alpha^{(1)}, \Rightarrow_\alpha\}$；若$term(n)\neq uns_term(n)$，且$\exists n''$，使得$n''\rightarrow n$，那么，$n''\rightarrow_\alpha n$，且$(\rightarrow_B\cup\Rightarrow_B)^+=(\rightarrow_B\cup\Rightarrow_B)\cup\{\rightarrow_\alpha, \Rightarrow_\alpha^{(1)}, \Rightarrow_\alpha\}$。

定义 5 对于图元$B=<N_B, (\rightarrow_B\cup\Rightarrow_B)>$和$C=<N_C, (\rightarrow_C\cup\Rightarrow_C)>$，如果图元$B$(进程)与图元$C$(进程)并发运行，且$\exists p=m\mapsto n$，使得$m\in N_B$，$n\in N_C$，那么，称$B|C$为组合(并发)运算。其中，符号“|”为组合(并发)算子。

命题 4 组合(并发)运算$B|C$为图元。其中，$B=<N_B, (\rightarrow_B\cup\Rightarrow_B)>$为图元，且$C=<N_C, (\rightarrow_C\cup\Rightarrow_C)>$为图元。

证明 由上述图元及其组合运算的定义可知，通过添加关系$E_\delta=(\rightarrow_\delta\cup\Rightarrow_\delta)$以及事件$N_\delta$，可将图元$B$和图元$C$连接起来，组合成$F=<N_B\cup N_C\cup N_\delta, (\rightarrow_B\cup\Rightarrow_B)\cup(\rightarrow_C\cup\Rightarrow_C)\cup E_\delta>=<N_F, (\rightarrow_F\cup\Rightarrow_F)>$。其中，$F$为图元，且$(\rightarrow_F\cup\Rightarrow_F)$满足：

(1)若$\exists m\in N_B$，使得$m\in N_C$，即图元B和图元C之间共享消息事件，那么，$(\rightarrow_F\cup\Rightarrow_F)=(\rightarrow_B\cup\Rightarrow_B)\cup(\rightarrow_C\cup\Rightarrow_C)$；特别地，对于$N_B\subseteq N_C$或$N_C\subseteq N_B$，那么，$(\rightarrow_B\cup\Rightarrow_B)\subseteq(\rightarrow_C\cup\Rightarrow_C)\vee(\rightarrow_C\cup\Rightarrow_C)\subseteq(\rightarrow_B\cup\Rightarrow_B)$，因此，$(\rightarrow_F\cup\Rightarrow_F)=(\rightarrow_C\cup\Rightarrow_C)\vee(\rightarrow_F\cup\Rightarrow_F)=(\rightarrow_B\cup\Rightarrow_B)$。

(2)若$\exists m\in N_B$，且$\exists n\in N_C$，使得$m\rightarrow_\delta n\vee m\Rightarrow_\delta n$，那么，$(\rightarrow_F\cup\Rightarrow_F)=(\rightarrow_B\cup\Rightarrow_B)\cup(\rightarrow_C\cup\Rightarrow_C)\cup\{\rightarrowtail_\delta\}$，其中，符号“$\rightarrowtail_\delta$”表示“$\rightarrow_\delta$”或“$\Rightarrow_\delta$”。

(3)若$\exists m\in N_B$，且$\exists n\in N_C$，使得$p=m\mapsto n$，其中，$p_1=m$，$l(p)=n$，那么，$N_\delta=\{p_i\mid i=2, 3, \ldots, |p|-1\}$，且$E_\delta=\{p_i\rightarrowtail_\delta p_{i+1}\mid i=1,$

2, …, | p|-1}。其中，符号“$\rightarrowtail_{\delta}$”表示“$\rightarrow_{\delta}$”或“$\Rightarrow_{\delta}$”。对于与消息事件{p_i | i=2, 3, …, | p|-1}关联的关系，且尚未被并运算至集合E_{δ}，那么，将满足图元定义中前两个条件之一的关系并 入E_{δ}。

(4)遍历图元B和图元C的消息事件域N_B和N_C，按照上述3种情况计算出N_{δ}和E_{δ}。

定义 6 对于图元B=＜N_B, ($\rightarrow_B \cup \Rightarrow_B$)＞和$C$=＜$N_C$, ($\rightarrow_C \cup \Rightarrow_C$)＞，如果仅图元$B$(进程)和图元$C$(进程)中之一运行，且$\nexists\ p = m \mapsto n$，使得$m \in N_B$，$n \in N_C$，那么，称$B$+$C$为选择运算。其中，符号“+”为选择算子。

4 事件图生成算法

生成并发运行安全协议的事件图有两种方法：(1)由“事件对”扩张生成的方法；(2)由“主体角色消息事件序列”扩张生成的方法。事件对和主体角色消息事件序列都是图元，可以利用它们来合成事件图。

算法 1 基于事件对扩张的事件图生成算法

对于安全协议(并发)运行形成的消息事件空间(集合)N和触发消息事件的施主(主体)空间(集合)R，对应的(消息)事件图 G 的生成算法记述如下。

Step1 初始化消息事件对集合$S_D \equiv \varnothing$。遍历消息事件集合N，将其中的消息事件依次顺序合成为消息事件对(图元)D_i，并记入集合S_D中，即$D_i \in S_D$，S_D中的消息事件元素构成的消息事件集合为N_D，那么，消息事件集合N中不能合成为消息事件对的消息事件集合为$N \setminus N_D$。

Step2 遍历消息事件对集合S_D，将S_D中的图元D_i并发合成为图元D，记消息事件对集合S_D中不能并发合成为图元D的消息事件对集合为ΔS_D。

Step3 遍历消息事件集合$N \setminus N_D$，将其中的消息事件依次顺序作用于由Step 2得到的图元D，记顺序合成后的图元为G，并记不能作用于D的消息事件集合记为$\Delta(N \setminus N_D)$。

Step4 反复多次使用Step 2 的算法将ΔS_D中消息事件合成更大的图元$D_j \in \Delta S_D$。注意，ΔS_D的元素与Step 2 中的ΔS_D的元素不同。

Step5 将由Step 3 中得到的图元G选择合成由Step 4 得到的图元$D_j \in \Delta S_D$，记之为$G + \sum D_j$。

Step6 选择$G + \sum D_j$中的图元作为分析对象(安全协议)的事件图模型。由于集合$\Delta(N \setminus N_D)$中的消息事件均为独立事件，不能构成任何安全协议的模型，因此将其舍弃。

Step7 算法结束。

算法 2 基于事件序列扩张的事件图生成算法

对于安全协议(并发)运行形成的消息事件空间(集合)N和触发消息事件的施主(主体)空间(集合)R，对应的(消息)事件图G的生成算法记述如下。

Step1 初始化消息事件序列集合$S_D \equiv \varnothing$。遍历消息事件集合N，将其中的消息事件按主体角色依次顺序合成为消息事件序列(图元)D_i，并记入集合S_D中，即$D_i \in S_D$。

Step2 遍历消息事件序列集合S_D，将S_D中的图元D_i并发合成为图元D，记消息事件序列集合S_D中不能并发合成为图元D的消息事件序列集合为ΔS_D。

Step3 反复多次使用Step 2 的算法将ΔS_D中消息事件序列合成更大的图元$D_j \in \Delta S_D$。注意，ΔS_D的元素与Step 2 中的ΔS_D的元素不同。

Step4 将由Step 2 中得到的图元D选择合成由Step 3 得到的图元$D_j \in \Delta S_D$，记之为$D + \sum D_j$。

Step5 选择$D + \sum D_j$中的图元作为分析对象(安全协议)的事件图模型。

Step6 算法结束。

Kerberos 协议为第三方认证协议，其主体为客户机 C、Kerberos 服务器 KS 和票据服务器 TGS 和应用服务器 S。协议完整一轮运行主体间交互的消息事件及其语句如下：

$uns\_term((C, 1))=uns\_term((KS, 1))=C \cdot TGS$，

$uns\_term((C, 2))=uns\_term((KS, 2))=encr(K_{C, TGS}, K_{C, TGS}) \cdot encr(K_{TGS}, T_{C, TGS})$，

$uns\_term((C, 3))=uns\_term((TGS, 1))=encr(K_{C, TGS}, A_{C, TGS}) \cdot encr(K_{TGS}, T_{C, TGS})$，

$uns\_term((C, 4))=uns\_term((TGS, 2))=C \cdot S \cdot encr(K_{C, TGS}, K_{C, S}) \cdot encr(K_S, T_{C, S})$，

$uns\_term((C, 5))=uns\_term((S, 1))=encr(K_{C, S}, A_{C, S}) \cdot encr(K_S, T_{C, S})$。

其中，$T_{C, TGS}=C \cdot a_C \cdot t_V \cdot K_{C, TGS}$，$A_{C, TGS}=C \cdot t \cdot k$，$T_{C, S}=C \cdot a_C \cdot t_V \cdot K_{C, S}$，$A_{C, S}=C \cdot a_C \cdot t$。符号含义如下：$T_{X, Y}$票据，$A_{X, Y}$鉴别码，$a_C$地址，$t$时间标记，$t_V$票据的有效起止时间，$k$备用密码，$K_{X, Y}$共享密钥，$K_X$私钥。

由算法 1 或算法 2，得到 Kerberos 协议在单轮运行状态下生成的事件图：

D_{Kerb}=＜N_{Kerb}, ($\Rightarrow_{Kerb} \cup \rightarrow_{Kerb}$)＞=＜$N_{Kerb}$, $(\Rightarrow \cup \rightarrow)_{Kerb}$＞，其中，

N_{Kerb}={(C, 1), (C, 2), (C, 3), (C, 4), (C, 5), (KS, 1), (KS, 2), (TGS, 1), (TGS, 2), (S, 1)};

$\Rightarrow_{Kerb}$={(C, 1)⇒(C, 2), (C, 2)⇒(C, 3), (C, 3)⇒(C, 4), (C, 4)⇒(C, 5), (KS, 1)⇒(KS, 2), (TGS, 1)⇒(TGS, 2)};

$\rightarrow_{Kerb}$={(C, 1)→(KS, 1), (KS, 2)→(C, 2), (C, 3)→(TGS, 1), (TGS, 2)→(C, 4), (C, 5)→(S, 1)}。

当Kerberos协议处于多轮并发运行状态下时，利用算法 2 并发合成得到事件图。将Kerberos协议在单轮运行状态下生成的事件图D_{Kerb}改记为$D_{Kerb}^{(w)}$。当Kerberos协议并发运行的次数为W时，其事件图可以由$D_{Kerb}^{(W)}$={$D_{Kerb}^{(w)}$, $1 \le w \le W$}中的图元合成得到。

5 结束语

基于Spi演算理论[2]和扩展的串空间模型[3]，提出用于描述并发运行密码协议的事件图模型。通过定义图元以及前缀、组合和选择算子，给出事件图的生成算法。它是对现有串空间理论[1]和Spi演算理论的深化和扩展。下一步将给出事件图的互模拟等价验证，以便将事件图模型用于密码协议安全属性的验证。

参考文献

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

2 Abadi M, Gordon A D. A Calculus for Cryptographic Protocols: The Spi Calculus[C]//Proceedings of the 4th ACM Conference on Computer and Communications Security. 1997: 36-47.

3 Wang Huanbao, Zhang Yousheng, Li Yuan. Modeling for Security Verification of a Cryptographic Protocol with MAC Payload[C]//Proc. of ICIC'05. Springer-Verlag, 2005: 538-547.

4 Guttman J D, Fábrega F J T. Protocol Independence Through Disjoint Encryption[C]//Proceedings of the 13th Computer Society Foundation Workshop. IEEE Computer Security Press, 2000-07.