安全协议设计与分析

第十一讲：基于定理证明的分析方法

李晖 网络空间安全学院

lihuill@bupt.edu.cn

基于定理证明的分析方法

- 基于逻辑推理的安全协议分析方法问题：
  - **存在规则不完善、 语义不精确**；后面出现的类BAN逻辑过于复杂
- 基于模型检测的安全协议分析方法问题：
  - **状态空间爆炸**；为避免这一点需要对协议进行高度抽象或简化
- 基于定理证明的安全协议分析方法
  - 采用事件序列（迹）来描述安全协议（**来自模型检测中的CSP**）
  - 从消息和已有知识通过公理推出新的结论（**来自逻辑推理**）
  - **基于定理证明器对迹的演算**来看某个定理（目标）是否成立
  - 优点：
    - 可以处理无限状态空间，可以给出安全协议的正确性证明
    - 在较低的抽象层次上进行建模，然后证明安全属性，而不是对主体的信念或知识进行建模

基于定理证明的分析方法

- 主要的方法
  - **Paulson归纳法**
    - 采用了模型检测方法对事件的描述方法
    - 采用与BAN逻辑类似基于信仰推理的方法进行推理
  - Schneider阶函数
  - 串空间
  - 重写逼近法

内容提纲

- **Paulson归纳法思路**
- Paulson归纳法简介
  - 主体和消息
  - 事件和迹
  - 主体知识
  - 操作符
- Paulson归纳法的自动化理论
- Paulson归纳法协议分析示例

Lawrence C. Paulson FRS

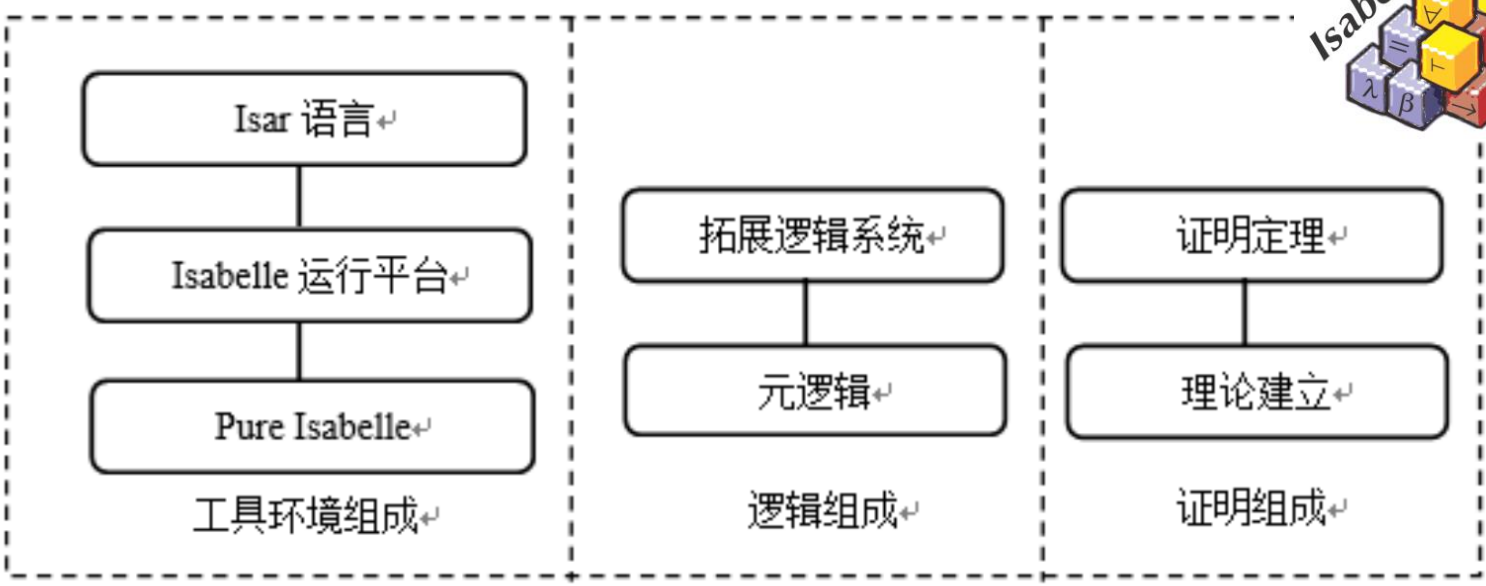
https://www.cl.cam.ac.uk/~lp15/

Paulson归纳法思路

- 将协议形式化为所有可能的**通信事件序列(迹)**的集合。
- 协议模型：
  - 主体并不知道消息的真正发送者，并且可能会转发一些他不知道的内容；
  - 窃听者会知道一些私钥并可以进行消息的加、解密及伪造等。
- 利用**定理证明器Isabelle**，在迹上通过**归纳的方法**来证明协议的安全属性。

Paulson归纳法思路（Isabelle定理证明器）

- 是一种支持高阶逻辑(Higher Order Logic, 简称HOL) 的交互式通用定理证明器。
- 剑桥大学与**慕尼黑理工大学**于1986年共同开发完成
- 逐渐发展成为通用的定理证明辅助工具，支持形式化数学公式，完成对数学公式的逻辑演算。
- 广泛用于数学公式和计算机协议的形式化验证



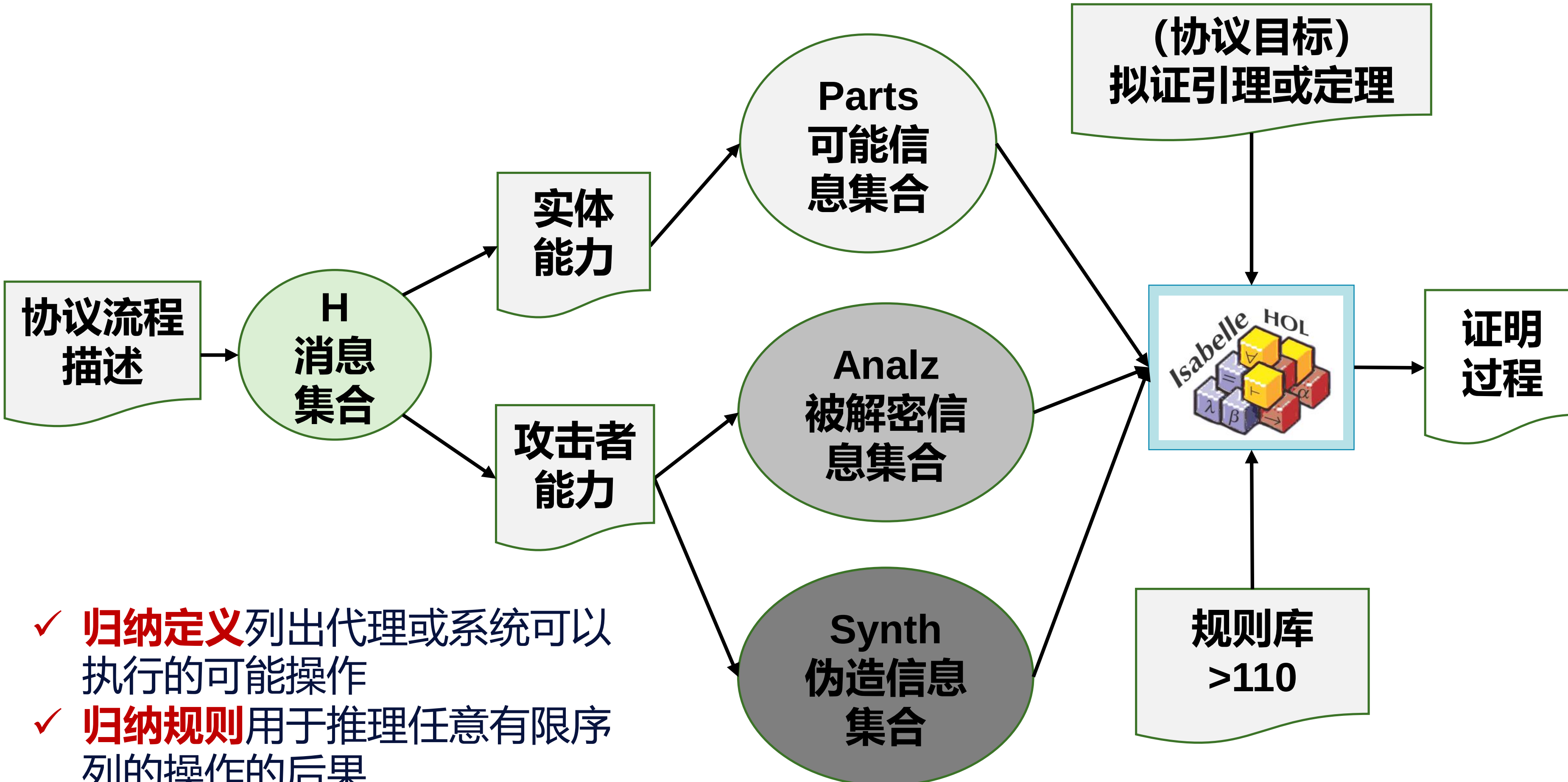
补充：什么是高阶逻辑

- 高阶逻辑（Higher-Order Logic, HOL）是一种形式逻辑系统，它允许量化和操作谓词、函数以及其他逻辑对象。与一阶逻辑（First-Order Logic, FOL）形成对比，后者仅允许量化个体变量。
- 在高阶逻辑中，我们可以有不同类型的变量：
- **个体变量**：代表具体的对象。
- **函数变量**：代表从个体到个体的函数。
- **谓词变量**：代表性质或关系。
- 示例："对于所有集合 S，如果 S 中的所有元素都是偶数，则 S 的和也是偶数"。可以表示成：

$$\forall S(\forall x(x \in S \to x \text{ is even}) \to \sum_{x \in S} x \text{ is even}$$

这里，S 是一个集合变量，x 是一个个体变量。

Paulson归纳法思路



- ✓ **归纳定义**列出代理或系统可以执行的可能操作
- ✓ **归纳规则**用于推理任意有限序列的操作的后果
- ✓ **归纳**已被广泛用于指定编程语言的语义

内容提纲

- Paulson归纳法思路
- **Paulson归纳法简介**
  - 主体和消息
  - 事件和迹
  - 主体知识
  - 操作符
- Paulson归纳法的自动化理论
- Paulson归纳法协议分析示例

Lawrence C. Paulson FRS

https://www.cl.cam.ac.uk/~lp15/

Paulson归纳法简介

1．主体和消息

- **主体定义**：**服务器**Server（S）、**友好主体**Friend *i*（*i*为自然数）和**攻击者**Spy，即

  datatype agent = Server | Friend nat | Spy

- 在Paulson归纳法中，消息项包括：
  - 主体标识：A，B，…；
  - 随机数：N_a，N_b，…；
  - 密钥：K_a，K_b，K_{ab}，…；
  - 复合消息：{X,X' }；
  - 哈希消息：Hash X；
  - 加密消息：Crypt K X。

Paulson归纳法简介

➢**消息项的定义:**

```
datatype msg = Agent    agent
             | Number  nat          (可猜测的)
             | Nonce   nat          (不可猜测的)
             | Key     key
             | Mpair   msg msg
             | Hash    msg
             | Crypt   key msg
```

➢假设不同的数据类型是不相交的；同样 $\mathsf{S} \neq \mathsf{Friend}\, i$, $\mathsf{S} \neq \mathsf{Spy}$,

➢Hash是不会产生碰撞的： $\mathsf{Hash}\, X = \mathsf{Hash}\, X'$ only if $X = X'$.

➢采用强密码算法，即：

$$\mathsf{Crypt}\, K X = \mathsf{Crypt}\, K' X' \Longrightarrow K = K' \wedge X = X'.$$

Paulson归纳法简介

2．事件和迹

- 事件的**迹**是对网络进行过的操作序列集合。一个**事件**是构成迹的单个操作。
- 用列表结构对**事件定义：**

datatype event = Says agent agent msg

| Notes agent msg

例如：Says A B X： 表示“A发送消息X给B”

Says A’ B X ：表示“B接收X”

notes A X：表示“A内部存储(知道)了X”

- 定义**Oops**事件
  - Oops将密钥泄露给攻击者：以便分析会话密钥意外丢失时的**影响**。
- **迹**： ev#evs 表示用ev事件对迹evs进行扩展

Paulson归纳法简介

3. 主体知识

- **主体**的状态可由其初始知识及由事件序列可获得的知识来表示。
  - **诚实主体** 能够读出发送给他们自己的消息，同时也知道她所发送的消息。
  - **攻击者** ✓ 可以获得网络中的所有通信；✓ 发送伪造的所有欺骗性消息；✓ 知道被攻破实体的内部信息；✓ 可能掌握了某个主体的密钥。

- 例（Otway-Rees协议)
  - 主体A和服务器长期共享密钥 shrK A
  - 攻击者Spy和服务器的共享密钥为 shrK Spy
  - 用**initState**函数来定义**主体初始知识**，例如：

$$\mathsf{initState\,S} \stackrel{\text{def}}{=} \text{all long-term keys}$$

$$\mathsf{initState}(\mathsf{Friend}\ i) \stackrel{\text{def}}{=} \{\mathsf{Key}(\mathsf{shrK}(\mathsf{Friend}\ i))\}$$

$$\mathsf{initState\,Spy} \stackrel{\text{def}}{=} \{\mathsf{Key}(\mathsf{shrK}(A)) \mid A \in \mathsf{bad}\}$$

Paulson归纳法简介

4．操作符

假设H是一个实体的消息集，包括该实体所有的初始知识和迹中所有发送消息的历史记录。

- **parts操作**
  - H中所有可公开获得的信息+加密消息体
- **analz操作**
  - 所有可被解密的信息
- **synth操作**
  - 所有伪造的信息

内容提纲

- Paulson归纳法思路
- Paulson归纳法简介
  - 主体和消息
  - 事件和迹
  - 主体知识
  - 操作符
- **Paulson归纳法的自动化理论**
- Paulson归纳法协议分析示例

Lawrence C. Paulson FRS

https://www.cl.cam.ac.uk/~lp15/

Paulson归纳法的自动化理论

1．操作符定义

Isabelle对parts的语法描述为

```
consts parts :: "msg set => msg set"
inductive "parts H"
  intros
    Inj [intro]: "X ∈ H ==> X ∈ parts H"
    Fst: "{|X,Y|} ∈ parts H ==> X ∈ parts H"
    Snd: "{|X,Y|} ∈ parts H ==> Y ∈ parts H"
    Body: "Crypt K X ∈ parts H ==> X ∈ parts H"
```

Paulson归纳法的自动化理论

1．操作符定义

Isabelle对analz的语法描述为

consts **analz** :: msg set => msg set

inductive “analz H”

intros

Inj “X∈ H => X∈analz H”

Fst “{|X,Y|}∈analz H => X ∈ analz H”

Snd “{|X,Y|}∈analz H => Y ∈ analz H”

Decrypt “[|Crypt K X|∈analz H; Key(invkey K) ∈analz H|] => X∈analz H”

Paulson归纳法的自动化理论

1．操作符定义

Isabelle对synth的语法描述为

consts **synth** :: "msg set => msg set"

inductive "synth H"

intros

Inj [intro]: "X ∈ H ==> X ∈ synth H"

Agent [intro]: "Agent agt ∈ synth H"

Number [intro]: "Number n ∈ synth H"

Hash [intro]: "X∈synth H ==> Hash X∈synth H"

MPair [intro]: "[|X∈synth H; Y∈synth H|] ==> {|X,Y|}∈synth H"

Crypt [intro]: "[|X∈synth H; Key(K)∈H|] ==> Crypt K X∈synth H"

Paulson归纳法的自动化理论

2.操作符规则

parts H规则：

$$\frac{X \in H}{X \in \text{parts } H} \qquad \frac{\text{Crypt } K X \in \text{parts } H}{X \in \text{parts } H}$$

$$\frac{\{X, Y\} \in \text{parts } H}{X \in \text{parts } H} \qquad \frac{\{X, Y\} \in \text{parts } H}{Y \in \text{parts } H}$$

analz H规则：

$$\frac{\mathrm{X} \in \mathrm{H}}{\mathrm{X} \in \text{analz H}}; \frac{\text{Crypt K X} \in \text{analz H}, \mathrm{K}^{-1} \in \text{analz H}}{\mathrm{X} \in \text{analz H}};$$

$$\frac{\{\mathrm{X}, \mathrm{Y}\} \in \text{analz H}}{\mathrm{X} \in \text{analz H}}; \frac{\{\mathrm{X}, \mathrm{Y}\} \in \text{analz H}}{\mathrm{Y} \in \text{analz H}}$$

synth H规则： Agent A $\in$synth H; Number N$\in$synth H;

$$\frac{\mathrm{X} \in \mathrm{H}}{\mathrm{X} \in \text{synth H}}; \frac{\mathrm{X} \in \text{synth H}}{\text{Hash X} \in \text{synth H}};$$

$$\frac{\mathrm{X} \in \text{synth H}, \mathrm{Y} \in \text{synth H}}{\{\mathrm{X}, \mathrm{Y}\} \in \text{synth H}}; \frac{\mathrm{X} \in \text{synth H}, \mathrm{K} \in \mathrm{H}}{\text{Crypt K X} \in \text{synth H}}$$

Paulson归纳法的自动化理论

3．操作符定理

（1）单调性定理

如果G⊆H，则parts G⊆parts H，analz G ⊆analz H，synthG ⊆synth H

（2）幂等律

parts(parts H) = parts H

analz(analz H) = analz H

synth(synth H)= synth H

（3）等价定理

parts(analz H)=parts H，analz(parts H)=parts H，

parts(synth H)=parts H∪synth H，

analz(synth H)=analz H∪synth H

注意：synth(parts H)和synth(analz H)是不可约的。

Paulson归纳法的自动化理论

3．操作符定理

$$\{X, Y\} \in \text{synth}(\text{analz } H) \Leftrightarrow X \in \text{synth}(\text{analz } H) \wedge Y \in \text{synth}(\text{analz } H)$$

更一般地，下面的定理给出了攻击者可以发送消息的范围：

$$\frac{X \in \text{synth}(\text{analz } H)}{\text{parts}(\{X\} \cup H) \subseteq \text{synth}(\text{analz } H) \cup \text{parts } H}$$

其中，H表示迹上发送的所有消息的集合。该规则可消除对消息X的伪造，得到了parts({X} ∪ H)的上限。

Paulson归纳法的自动化理论

4．符号求值的重写规则

1）parts重写规则

parts $\phi = \phi$

parts({Agent **A**} ∪**H**)= {Agent **A**}∪ parts **H**

parts({Nonce **N**}∪ **H**)= {Nonce **N**} ∪ parts **H**

parts({Key **K**}∪ **H**)= {Key **K**} ∪ parts **H**

parts({**X,Y**}∪**H**)= {**X,Y**} ∪ parts {{**X**}∪ {**Y**}∪ **H**}

parts({Hash **X**}∪ **H**)= {Hash **X**}∪ parts **H**

parts({Crypt **K X**} ∪ **H**)= {Crypt **K X**}∪ parts ({**X**} ∪ **H**)

Paulson归纳法的自动化理论

4．符号求值的重写规则

2) analz重写规则

➢ 解密**H**中消息的密钥集合的定义：

$$\text{keysFor } H \stackrel{\text{def}}{=} \{K^{-1} \mid \exists X.\ \text{Crypt } K\, X \in H\}$$

➢ 解密时不需要的密钥，则可以将其移出**analz H**集合

$$\frac{K \notin \text{keysFor}(\text{analz } H)}{\text{analz}(\{\text{Key } K\} \cup H) = \{\text{Key } K\} \cup (\text{analz } H)}$$

➢ 加密信息分析的重写规则

若有解密密钥，即可以分析出X，则把X添加到analz中

$$\text{analz}(\{\text{Crypt K X}\} \cup \text{H}) = \begin{cases} \{\text{Crypt K X}\} \cup (\text{analz}(\boxed{\{X\}} \cup \text{H})) & \text{K}^{-1} \in \text{analz H} \\ \{\text{Crypt K X}\} \cup (\text{analz H}) & \text{其他} \end{cases}$$

➢ 与幂等性相关的重写规则

$$\frac{X \in \text{analz } H}{\text{analz}(\{X\} \cup H) = \text{analz } H}$$

∈

Paulson归纳法的自动化理论

4．符号求值的重写规则

3） synth重写规则

- 对于synth的符号求值不可能（因其结果无限），也不必要；
- 只需对X∈synth H的假设进行简化。在考虑一个消息是否可能被伪造时会用到该假设。由于在归纳定义时将随机数及密钥视为不可猜测的，因此有

$$\text{Nonce } N \in \text{synth } H \Rightarrow \text{Nonce } N \in H$$

$$\text{Key } K \in \text{synth } H \Rightarrow \text{Key } K \in H$$

- 如果Crypt K X∈synth H，则Crypt K X ∈H，或者X ∈synth H 且K∈H。
- Hash X ∈ synth H和{X,Y}∈synth H有类似的规则成立。

Isabelle中内置了类似的parts, analz, synth 和 keysFor定理有110个

Paulson归纳法的自动化理论

5．归纳法

Paulson归纳法是指归纳地定义出可能的迹的集合。

- **自然数集合N的归纳定义**为 $0 \in \mathbf{N}$ and $n \in \mathbf{N} \Longrightarrow \mathsf{Suc}\, n \in \mathbf{N}$
- **对集合N上的归纳推理可定义**为：对于集合N，要证明P(n)对所有的自然数都成立，只需证明P(0)，以及对所有的 $x \in \mathbf{N}$，有 $P(x) \Longrightarrow P(\mathsf{Suc}\, x)$
- **迹的集合**上的归纳原理为：如果P在生成迹的所有规则下都成立，那么*P(evs)*对所有迹都成立。
  - 首先必须证明P[ ]成立；
  - 对于其他规则，必须证明

$$P(evs) \Longrightarrow P(ev\#evs)$$

其中*ev*包括了新消息，*ev#evs*表示用事件*ev*对*evs*扩展后得到的迹。

内容提纲

- Paulson归纳法思路
- Paulson归纳法简介
  - 主体和消息
  - 事件和迹
  - 主体知识
  - 操作符
- Paulson归纳法的自动化理论
- **Paulson归纳法协议分析示例**

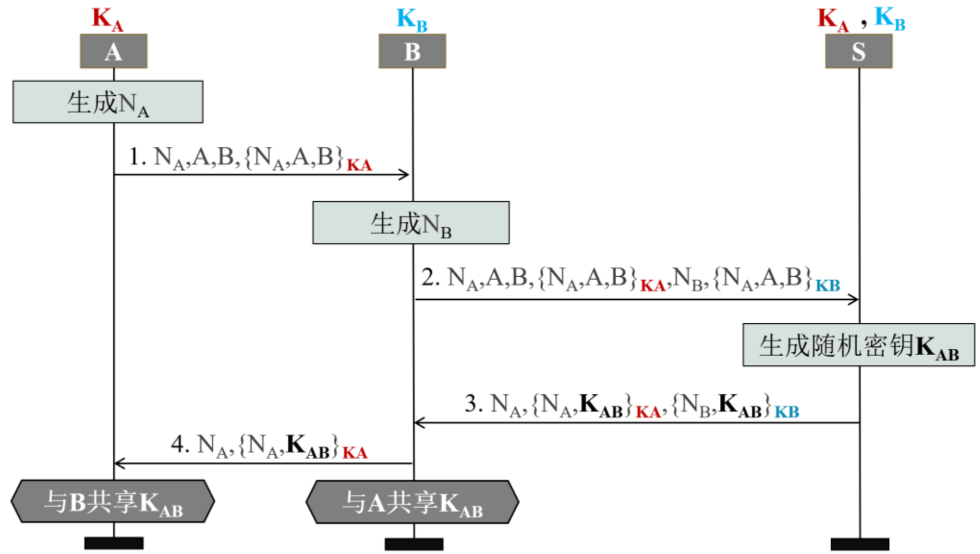
Lawrence C. Paulson FRS

https://www.cl.cam.ac.uk/~lp15/

Paulson归纳法协议分析示例

1．协议模型

以**变型的Otway-Rees协议**为例：



协议前提：KA为A与S共享的密钥；

KB为B与S共享的密钥；

协议流程：

（1） $A \rightarrow B$: $N_A,A,B,\{N_A,A,B\}_{KA}$。

（2） $B \rightarrow S$: $N_A,A,B,\{N_A,A,B\}_{KA}$,N_B, $\{N_A,A,B\}_{KB}$。

（3） $S \rightarrow B$: $N_A,\{N_A,\mathbf{K_{AB}}\}_{KA}$,$\{N_B,\mathbf{K_{AB}}\}_{KB}$。

（4） $B \rightarrow A$: $N_A,\{N_A,\mathbf{K_{AB}}\}_{KA}$。

协议目标：A和B之间建立会话密钥$\mathbf{K_{AB}}$。

Paulson归纳法协议分析示例

- 协议建模：将协议步骤表示为利用新的事件对迹所做的扩展
- 设服务器为常量S，A、B为主体名称（包括攻击者）：

(1) 如果*evs*是一个迹，则*evs*可被拓展为：Says A B $\{N_A, A, B, \{N_A, A, B\}_{K_A}\}$

(2) 如果*evs*是一个迹，且*evs*中有事件：Says A’ B {Na, A, B, X }

则*evs*可被拓展为：Says B S $\{N_A, A, B, X, N_B, \{N_A, A, B\}_{K_B}\}$

(3) 如果*evs*是一个迹，且*evs*中有事件：

Says B’ S $\{N_A, A, B, \{N_A, A, B\}_{K_A}, N_B, \{N_A, A, B\}_{K_B}\}$

则*evs*可被拓展为：Says S B $\{N_A, \{N_A, K_{AB}\}_{K_A}, \{N_B, K_{AB}\}_{K_B}\}$

(4) 如果*evs*是一个迹，且*evs*中有事件：Says B S $\{N_A, A, B, X', N_B, \{N_A, A, B\}_{K_B}\}$

Says S’ B $\{N_A, X, \{N_B, K\}_{KB}\}$

则*evs*可被拓展为：Says B A $\{N_A, X\}$

Paulson归纳法协议分析示例

Otway-Rees协议的Isabelle定理证明器的协议描述：

Nil [ ] ∈ **otway**

常量otway表示协议的迹的集合

Fake [| **evs** ∈ **otway**; B ≠ Spy; X ∈ synth (analz (**spies evs**)) |]

=> Says Spy B X # evs ∈ otway

攻击者spy可获得的消息项的集合

OR1 [| evs1 ∈ otway; A ≠ B; B ≠ Server; Nonce NA ∉ used evs1 |]

说明evs是一条已存在的迹

=> Says A B {|Nonce NA, Agent A, Agent B,

Crypt (shrK A) {|Nonce NA, Agent A, Agent B|} |}

evs1 ∈ otway

OR2 [| evs2 ∈ otway; B ≠ Server; Nonce NB ∉ used evs2;

Says A' B {|Nonce NA, Agent A, Agent B, X|} ∈ **set evs2** |]

=> Says B Server

表示迹evs上事件的集合

{|Nonce NA, Agent A, Agent B, X, Nonce NB,

Crypt (shrK B) {|Nonce NA, Agent A, Agent B|}|}

evs2 ∈ otway

Paulson归纳法协议分析示例

OR3 [| evs3 otway; B ≠ Server; Key KAB ∉ used evs3;

Says B' Server

{|Nonce NA, Agent A, Agent B,

Crypt (shrK A) {|Nonce NA, Agent A, Agent B|},

Nonce NB,

Crypt (shrK B) {|Nonce NA, Agent A, Agent B|}|}

set evs3 |]

=> Says Server B

{|Nonce NA,

Crypt (shrK A) {|Nonce NA, Key KAB|},

Crypt (shrK B) {|Nonce NB, Key KAB|}|}

evs3 ∈ otway

Paulson归纳法协议分析示例

OR4 [| evs4∈ otway; A ≠B;

Says B Server {|Nonce NA, Agent A, Agent B, X',Nonce NB,

Crypt (shrK B) {|Nonce NA, Agent A, Agent B|}|}

∈ set evs4;

Says S' B {|Nonce NA, X, Crypt (shrK B) {|Nonce NB, Key K|}|}

∈ set evs4 |]

=> Says B A {|Nonce NA, X|} # evs4 ∈ otway

Oops [| evso ∈ otway; B ≠Spy;

Says Server B {|Nonce NA, X, Crypt (shrK B) {|Nonce NB, Key K|}|}

∈ set evso |]

=> Notes Spy {|Nonce NA, Nonce NB, Key K|} # evso ∈ otway

Paulson归纳法协议分析示例

2. 可能性属性的证明

- **可能性属性的含义**，确保从协议的第一步到最后一步中的消息格式都是符合协议规范的。
- **保证协议的形式化描述正确**。
- **Otway-Rees协议的可能性属性**：对于主体A、B（A、B不是同一主体且不是服务器S），存在密钥K、随机数N及一条迹，使得最后一条消息能够被发送。
- 该属性可通过顺序连接所有协议规则，并说明所有的前提条件都可满足来证明。

```
lemma "[| A ≠ Server; Key K ∉ used []; K
   ==> ∃ N. ∃ evs ∈ ds_lowe.
       Says A B (Crypt K {| Nonce N, Nonc
```

```
apply (cut_tac SesKeyLife_LB)
apply (intro exI bexI)
apply (rule_tac [2] ds_lowe.Nil [THEN ds_lowe.DS1,
                    THEN ds_lowe.DS2, THEN ds_lowe.DS3,
                    THEN ds_lowe.DS4, THEN ds_lowe.DS5])
apply (possibility, simp add:  used_Cons)
done
```

Paulson归纳法协议分析示例

3．转发引理的证明

- **lemma-OR2：**（主体转发他们并不知道的消息项）
  - 例如OR2，有 $$\frac{\text{Says } A' \ B \{N, \text{Agent } A, \text{Agent } B, X\} \in \text{set evs}}{X \in \text{analz(spies evs)}}$$
  - 证明过程：入侵者可以得到整条消息; 因为X是明文传输的，因此，analz可以得到它。
  - 当B响应并转发该消息时，入侵者得不到任何新的信息。
- **lemma-Oops：**（主体转发解密后的消息项）
  - 引理：$$\frac{\text{Says } S \ B \{N_A, X, \text{Crypt } K' \{N_B, K\}\} \in \text{set evs}}{K \in \text{parts(spies evs)}}$$
  - 用parts而不是analz表述，表明parts(spies evs) 没有引入新密钥
  - 证明：应用Oops规则（假设攻击者知道了加密密钥K'），则K已经在parts(spies evs)中

Paulson归纳法协议分析示例

4. 常规引理的证明

➢形如 $X \in parts(spies\ evs) \to \cdots$ 的引理，关注消息中出现X所造成的结果。

➢对Otway-Rees协议，用常规引理表明秘密密钥保持秘密性，即如果evs ∈ otway，那么就有

$$Key(shrK\ A) \in parts(\mathbf{spies}\ evs) \Leftrightarrow A \in bad$$

➢证明过程：

1. 应用归纳为协议的每个步骤及Nil、Fake和Oops等规则产生情况语句；
2. 对转发消息的各个步骤应用相应的转发引理，必要时可以利用 $\mathbf{analz\ H} \subseteq \mathbf{parts\ H}$ 来得到关于parts的结论；
3. 用标准的自动策略来证明Nil情形；
4. 简化所有的其它情形。

Paulson归纳法协议分析示例

5．常规引理的证明

➢在Isabelle定理证明器中，用户可以定义一种策略来执行上述任务，并得到所有的子目标：

表示应用 **otway** 集合上的归纳规则

```
(*Regularity lemmas*)
lemma Spy_see_shrK [simp]:
    "evs ∈ otway ==>
        (Key (shrK A) ∈ parts (spies evs)) = (A ∈ bad)"

apply (erule otway.induct, force)
apply (drule_tac [4] OR2_msg_in_parts_spies)
apply simp_all
apply blast+
done
```

Force为参数，表示强制应用该规则

Paulson归纳法协议分析示例

5．单一性定理的证明

- 新鲜的会话密钥和随机数唯一地标识了该条消息。
- 分为两种情况：
  - 关于会话密钥的唯一性定理：

S为本次密钥生成的随机密钥

$$\exists B'\ Na'\ Nb'\ X'.\quad \forall B\ Na\ Nb\ X.$$
$$\mathsf{Says\ S}\ B\ \{\!|Na, X, \mathsf{Crypt}(\mathsf{shrK}\ B)\{\!|Nb, \boxed{K}|\!\}|\!\} \in \mathsf{set}\ evs$$
$$\longrightarrow B = B' \wedge Na = Na' \wedge Nb = Nb' \wedge X = X'.$$

  - 对于随机数的唯一性定理：

$$\exists B'.\forall B.\ \mathsf{Crypt}(\mathsf{shrK}\ A)\{\!|Na, \mathsf{Agent}\ A, \mathsf{Agent}\ B|\!\} \in \mathsf{parts}(\mathsf{spies}\ evs)$$
$$\longrightarrow B = B'.$$

Paulson归纳法协议分析示例

6. 会话密钥泄露定理的证明

- 会话密钥泄露定理

  对任意迹evs ∈ otway，

  $$K \in \mathsf{analz}(\mathcal{K} \cup \mathsf{spies}\, evs)) \iff K \in \mathcal{K} \vee K \in \mathsf{analz}(\mathsf{spies}\, evs)$$

  其中$\mathcal{K}$为任意的一个会话密钥集合。

- 其证明非常困难，基本思路为：

  1. 应用归纳法；
  2. 对于协议每条消息的转发引理，若结论是用analz表述的，则应用相应的转发引理；
  3. 简化所有情形，运用重写规则计算analz。

Paulson归纳法协议分析示例

7. 会话密钥秘密性定理的证明

➢定理表述如下：

设evs∈ otway，且A、B b~~ad，假设服务~~器给A、B分发了会话密钥K，

即

$$\mathsf{Says}\ \mathsf{S}\ B\ \{Na, \mathsf{Crypt}(\mathsf{shrK}\ A)\{Na, K\}, \mathsf{Crypt}(\mathsf{shrK}\ B)\{Nb, K\}\} \in \mathsf{set}\ evs$$

再假设该密钥没有在包含相同随机数的Oops事件中丢失，即

$$\text{Notes Spy}\{N_A, N_B, K\} \notin \text{set evs}$$

那么可以得到K $\notin$ analz(spies evs)，即攻击者永远也得不到该密钥。

➢ 定理的证明：略

- ➢ 证明过程的脚本由**7**个命令组成，超过**4000**个步骤

Paulson归纳法协议分析示例

8. 认证性定理的证明

- 上述Otway-Rees协议不安全，攻击过程如下：

$$
\begin{aligned}
&1. \quad A \rightarrow C_B : Na, A, B, \{Na, A, B\}_{Ka} \\
&1'. \quad C \rightarrow A : Nc, C, A, \{Nc, C, A\}_{Kc} \\
&2'. \quad A \rightarrow C_S : Nc, C, A, \{Nc, C, A\}_{Kc}, Na', \{Nc, C, A\}_{Ka} \\
&2''. \; C_A \rightarrow S : Nc, C, A, \{Nc, C, A\}_{Kc}, Na, \{Nc, C, A\}_{Ka} \\
&3'. \quad S \rightarrow C_A : Nc, \{Nc, Kca\}_{Kc}, \{Na, Kca\}_{Ka} \\
&4. \quad C_B \rightarrow A : Na, \{Na, Kca\}_{Ka}
\end{aligned}
$$

- 产生攻击的原因是*Nb*明文传输，正确的协议如下：

$$
\begin{aligned}
&1.\ A \rightarrow B : Na, A, B, \{Na, A, B\}_{Ka} \\
&2.\ B \rightarrow S : Na, A, B, \{Na, A, B\}_{Ka}, \{Na, \boxed{Nb,} A, B\}_{Kb} \\
&3.\ S \rightarrow B : Na, \{Na, Kab\}_{Ka}, \{Nb, Kab\}_{Kb} \\
&4.\ B \rightarrow A : Na, \{Na, Kab\}_{Ka}
\end{aligned}
$$

Paulson归纳法协议分析示例

8．认证性定理的证明

➢**B的认证性定理：**

如果一个迹中包含有形如

$$Says\ S'\ B\ \{N_A,X,Crypt(shrK\ B)\{N_B,Key\ K\}\}$$

的事件，并且B未被攻破，且发送过M2:

$$Says\ B\ S\ \{NA,Agent\ A,Agent\ B,X',NB,$$
$$Crypt(shrK\ B)\{NA,NB,Agent\ A,Agent\ B\}\}$$

则说明S已发送了步骤3的一个正确数据，则完成对B的认证。

（B在收到M3后，可以验证NB的新鲜性，从而保证K是一个可以用来和A通信的good key。）

Paulson归纳法协议分析示例

8．认证性定理的证明

- 其主要前提是B的认证证据已经出现，

$$\mathsf{Crypt}(\mathsf{shrK}\ B)\{\!|Nb, \mathsf{Key}\ K|\!\} \in \mathsf{parts}(\mathsf{spies}\ evs),$$

- 它的证明很复杂，需要几个辅助引理：
  - 如果消息2的加密部分出现，则实际发送了M2。
  - *nonce Nb*唯一识别了M2加密部分的其他组件。
  - 在协议的两次运行中，nonce不能同时用作Na和Nb。

  $A \notin$ Bad，则下面信息不能同时在parts(spies evs)中

$$\mathsf{Crypt}(\mathsf{shrK}\ A)\{\!|Na, \mathsf{Agent}\ A, \mathsf{Agent}\ B|\!\}$$
$$\mathsf{Crypt}(\mathsf{shrK}\ A)\{\!|Na', Na, \mathsf{Agent}\ A', \mathsf{Agent}\ A|\!\}$$

Paulson归纳法小结

- 该方法灵活而通用。
- 将协议形式化为所有可能“迹”的集合，而“迹”是协议的**通信事件序列**。
- Paulson协议模型包含了攻击者及密钥丢失等情形。
- Paulson归纳法**利用定理证明器Isabelle**，在迹上通过归纳的方法来证明协议的安全属性。

小结

- 模型检测
  - 自动化程度高
  - 但只能处理有限状态
  - 不能指出协议为什么会遭受攻击
- 定理证明
  - 分析更为详细，更深入了解协议应对不同情况的原因
  - 自动化程度不高，需要人工参与

模型检测 or 定理证明？

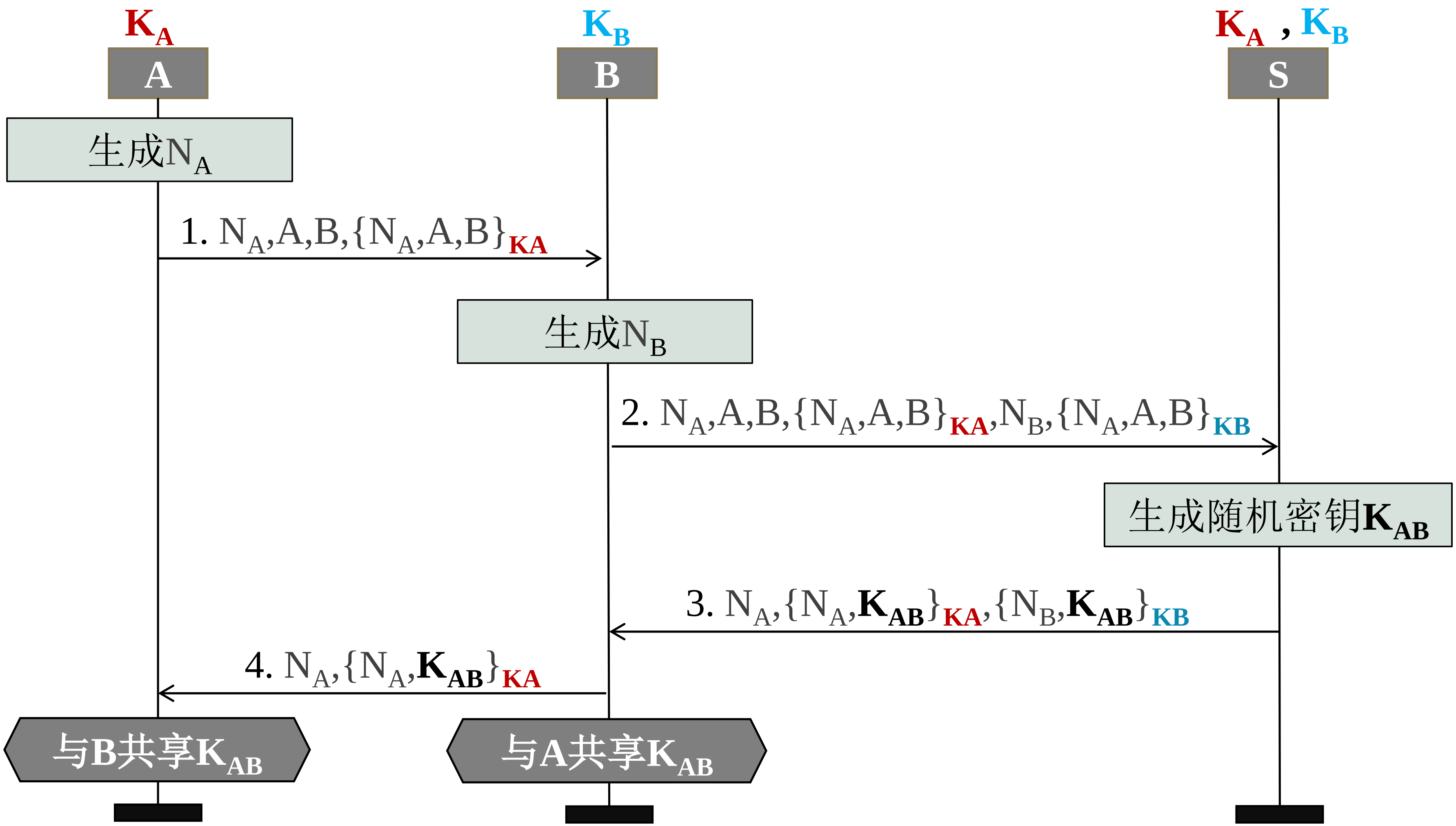
参考文献

- L.C. Paulson. The inductive approach to verifying cryptographic protocols. Journal of Computer Security, 6(1):85–128, 1998.
- Q Wang. Verification of Security Protocols Using A Formal Approach. http://www2.imm.dtu.dk/pubdb/edoc/imm5453.pdf
- Tobias Nipkow. Lawrence C. Paulson. Markus Wenzel. A Proof Assistant for Higher-Order Logic. 2021

谢谢大家！欢迎提问！

OTWAY-REES认证协议

- 目标：Alice和Bob之间建立会话密钥。



Paulson归纳法简介

3. 主体知识

- *函数**spies**建模了攻击者通过迹得到的消息的集合，即*

  spies [ ] $\overset{\text{def}}{=}$initState Spy　/*在空路径上，攻击者只能得到自己的初始状态

  spies ((Says A B X) #evs) $\overset{\text{def}}{=}$ {X} U spies evs

  $$\text{spies ((Notes A X) \#evs)} \overset{\text{def}}{=} \begin{cases} \{X\} \bigcup \text{spies evs} & \text{if } A \in \text{bad} \\ \text{spies evs} & \text{otherwise} \end{cases}$$

- 集合**used** evs形式化描述了新鲜性的概念。该集合包括parts(spies evs)及任意主体的私有消息的parts集合，即

  used [ ] $\overset{\text{def}}{=} \bigcup B.\text{parts(initState B)}$

  used ((Says A B X) # evs) $\overset{\text{def}}{=}$ parts{X} $\bigcup$ used evs

  used ((Notes A X) # evs) $\overset{\text{def}}{=}$ parts{X} $\bigcup$ used evs