软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2023,34(8):3708−3725 [doi: 10.13328/j.cnki.jos.006617]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一个切换认证的 5G 鉴权协议及其形式化分析*

刘逸冰, 周 刚

(中国人民解放军信息工程大学 数据与目标工程学院, 河南 郑州 450002)

通信作者: 刘逸冰, E-mail: 18121095425@163.com

摘 要: 随着移动通信的发展, 迎来了第 5 代移动通信技术 (5G). 5G 认证与密钥协商 (5G authentication and key agreement, 5G-AKA) 协议的提出主要是为了实现用户和服务网络的双向鉴权. 然而, 最近的研究认为其可能会遭受信息破译和消息重放攻击. 同时, 发现当前 5G-AKA 的一些变种不能满足协议的无连接性. 针对上述缺陷, 提出一个改进方案: SM-AKA. SM-AKA 由两个并行子协议组成, 通过巧妙的模式切换使更加轻量的子协议 (GUTI 子模块) 被频繁采用, 而另一个子协议 (SUPI 子模块) 则主要用于异常发生时的鉴权. 依据这种机制, 它不仅实现用户和归属网之间的高效认证, 还能提升鉴权的稳定性. 此外, 变量的新鲜性也得到有效维持, 可以防止消息的重放, 而严格的加解密方式进一步提升协议的安全性. 最后, 对 SM-AKA 展开完整的评估, 通过形式建模、攻击假定和 Tamarin 推导, 证明该方案可以达到鉴权和隐私目标, 而理论分析部分也论证了协议性能上的优势.

关键词: 5G 鉴权; 认证协议; 形式化分析; 移动网络

中图法分类号: TP311

中文引用格式: 刘逸冰, 周刚. 一个切换认证的5G鉴权协议及其形式化分析. 软件学报, 2023, 34(8): 3708–3725. http://www.jos.org.cn/1000-9825/6617.htm

英文引用格式: Liu YB, Zhou G. 5G Authentication Protocol Based on Sub-mode Switching Operation and Its Formal Analysis. Ruan Jian Xue Bao/Journal of Software, 2023, 34(8): 3708–3725 (in Chinese). http://www.jos.org.cn/1000-9825/6617.htm

5G Authentication Protocol Based on Sub-mode Switching Operation and Its Formal Analysis

LIU Yi-Bing, ZHOU Gang

(School of Data and Target Engineering, PLA Information Engineering University, Zhengzhou 450002, China)

Abstract: With the development of the Internet, the 5th generation (5G) of mobile communication technology emerges. The 5G authentication and key agreement (5G-AKA) protocol is proposed mainly to achieve two-way authentication between users and service networks. However, recent research suggests that the protocol may be subject to information deciphering and message replay attacks. At the same time, it is found that some variants of the current 5G-AKA cannot satisfy the protocol's unlinkability. Therefore, in response to these shortcomings, this study proposes an improvement plan called SM-AKA. SM-AKA is composed of two parallel sub-protocols in a novel way. In addition, through the flexible mode switching, lightweight sub-protocols (GUTI submodule) are frequently adopted, and the other sub-protocol (SUPI submodule) is used to deal with abnormalities caused by authentication. Therefore, this mechanism not only realizes the efficient authentication between users and networks but also improves the stability of the protocol. Furthermore, the freshness of variables has been effectively maintained to prevent the replay of messages, and strict encryption and decryption methods have further strengthened the security of the protocol. Finally, the study carries out a complete evaluation of SM-AKA. Through formal modeling, attack assumptions, and Tamarin derivation, it is proved that the plan can achieve the authentication and privacy goals, and the theoretical analysis has demonstrated the performance advantage of the protocol.

Key words: 5G authentication; authentication protocol; formal analysis; mobile network

---

* 本文由"形式化方法与应用"专题特约编辑陈立前副教授、孙猛教授推荐.
收稿时间: 2021-09-05; 修改时间: 2021-10-14; 采用时间: 2022-01-10; jos 在线出版时间: 2022-03-24
CNKI 网络首发时间: 2023-02-23

1 引 言

随着通信技术的发展, 移动设备已经广泛普及, 比如手机、智能手表等. 据统计, 目前已有 50 亿人都是移动通信用户, 这是一个庞大的群体[1]. 他们通过多种多样的设备来享受移动网络提供的丰富服务, 并被国际移动电信组织 (3rd generation partnership projec, 3GPP) 制定的安全机制所保护. 在漫长的发展历程中, 移动通信技术已经走过了最初的 2G、惊喜的 3G 和成熟的 4G 时代, 今天正处于 5G 时代, 即第 5 代移动网络.

每一代移动网络的发展都离不开安全这一主题, 都会形成一系列的安全机制, 而其中一个极其重要的机制是用户设备 (user equipment, UE) 和归属网 (home network, HN) 之间的双向认证. 事实上, 3G 已经存在较为稳定的认证方法, 即认证与密钥协商机制 (authentication and key agreement, AKA). 到了 4G, AKA 方法得到进一步发展, 演变为更加成熟和安全的 EPS-AKA 和 EAP-AKA 协议. 现如今, 经 3GPP 在前人基础上长达数年的研究, 5G 的鉴权与认证协议最终形成[2]. 该协议的目标是在包含通用用户身份模块 (universal subscriber identity module, USIM) 的设备和归属网络之间提供一个双向鉴权机制, 以使双方在信息读取、网络接入等权限问题上达成一致. 在确保隐私安全的情况下, 用户可以正常的享用移动通信服务. 为了实现这一目标, AKA 协议会在认证成功之后, 赋予 UE 和 HN 一个共享会话密钥以保障后续双方的信息通信. 经过 3GPP 组织反复的商榷, 最终确立的 5G 鉴权协议有两种, 即 5G-AKA 和 EPS-AKA′. 事实上, 它们的设计目标是一致的, 而其中的差异也只关乎法律因素[3]. 因此, 就本文聚焦的隐私安全来说, 它们没有不同, 在此, 本文主要选取 5G-AKA 协议展开研究和讨论.

近些年, 针对认证协议的研究取得了巨大的进展, 业界已经发现了一些关于 5G-AKA 协议的脆弱性, 如: 不同类型的失败消息的可区分性可能会使鉴权结果泄露、序列号的不完善加密可能会导致其被破解以及 SUPI 的非对称加密可能会带来较高计算复杂度. 同时, 协议分析逐渐大量采用形式化分析方法[4−6]. 一些研究者通过形式化手段分析 5G-AKA, 发现了协议中的一些缺陷, 如 SUPI 在 UE 侧的不可靠存储、会话密钥的弱私密性等[7,8]. 针对当前 5G 认证协议存在的缺陷, 一些安全研究人员提出了改进方案. Koutsos 等人[9]提出了具有两阶段子协议的 AKA$^+$方案, 其最大的特色是设计了两个子协议相互补充的机制, 其总体运行模式如图 1. 它采用新颖的交替运行实现了认证协议所有的功能, 两个子协议互为补充, 分别应对不同的情况, 是一个较为优秀的改进方案. AKA$^+$协议会依据 V-GUTI 的值来决定选用哪个子模块, 而鉴权的结果, 即成功或者失败, 又会反过来影响 V-GUTI 的值. 作者还证明了该协议不仅可以很好地抵抗现有攻击, 还取得了较高的平均认证效率.

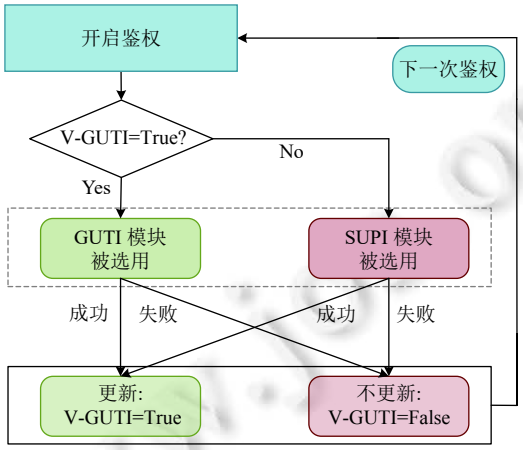

图 1 AKA$^+$运行机制

通过分析 AKA$^+$方案独特的工作模式, 如图 1, 我们发现了其存在一个脆弱点. 在该协议中, 用户永久身份标识 (subscription permanent identifier, SUPI) 和全局唯一临时标识 (globally unique temporary identity, GUTI) 子协议传输的消息结构和组成不同, 因而, 我们在实际观测中可以识别认证网正在运行何种子协议. 据此, 我们设计了一种连接性攻击, 总体过程如图 2. 具体来说, 当需要判断一个隐式 UE$_x$ 与目标 UE$_0$ 的关系时, 攻击者首先监测 UE$_x$

和 UE_0 一次成功的 AKA$^+$认证. 根据协议规定, UE_0 和 UE_x 的 GUTI 标识符 (V-GUTI) 都会被设置为 True. 接下来, 攻击者让 UE_0 再进行一次 AKA$^+$认证, 并通过消息截断的方式使其认证失败, 此时 UE_0 的 GUTI 会被消耗, 即 V-GUTI 变为 False. 最后, UE_0 和 UE_x 再进行一次 AKA$^+$认证. 通过观察传输的消息类型, 攻击者能够判断是哪个子协议在运行. 作为结果, 有两种情况.

- 如果 UE_x 运行的是 SUPI 子协议, 则有 UE_0=UE_x, 即未知的 UE_x 即是目标 UE_0.
- 如果 UE_x 运行的是 GUTI 子协议, 则有 UE_0≠UE_x, 即未知的 UE_x 不是目标 UE_0.

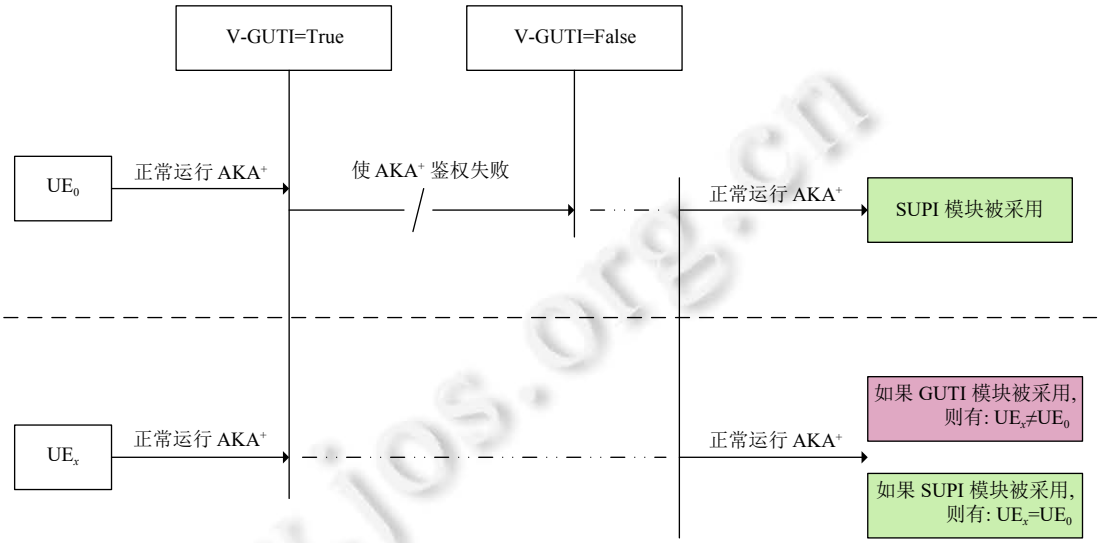

图 2 针对 AKA$^+$协议的连接性攻击

因此, 攻击者可以推断出 UE_x 和目标 UE_0 之间的关系, 这违背了协议的无连接性要求, 即攻击者通过对协议的攻击, 能够探知多个隐式目标之间的关系, 如判断两次跟踪的未知 UE 是否为同一 UE 等.

不可否认的是, AKA$^+$协议的工作模式是新颖的, 在运行效率上具有很大的优势. 其总是倾向于采用速度更快的 GUTI 子协议, 而 SUPI 子协议也在为使用高效的 GUTI 模块服务. 受此启发, 也鉴于上述指出的脆弱点, 我们提出了具有两阶段平行子模块的 SM-AKA 协议. 该协议与 AKA$^+$有类似的工作模式, 即包含一个 SUPI 子模块和一个 GUTI 子模块. 不同的是, 我们令两个子模块的鉴权变量具有相同的组成和结构, 从而确保子协议不可区分. 该协议包含 3 个认证阶段, 分别是: UE 向 HN 发送身份消息, 开启鉴权流程; HN 向 UE 发送结果, 同时更新用于后续认证的变量; UE 完成最后的信息检验, 决定是否更新 GUTI. 此外, 我们对提出的 SM-AKA 协议实施了全面的形式化分析, 并得出较有价值的结论. 本文中我们做了如下贡献.

(1) 提出一个改进的认证方案, 即 SM-AKA. 它不仅可以提供稳定可靠的交互认证, 还可以有效阻止所有已知的针对 5G-AKA 及其一些修正版本的攻击.

(2) 在认证协议中仅用到一个随机数和两次消息传输, 并通过新颖的交替运行模式更多的选用轻便的子模块, 这给协议带来了较高的运行效率.

(3) 对 SM-AKA 协议开展一次精细的形式化分析, 完成了对其各项属性的评估. 同时, 完整的分析证明过程也给形式化验证的发展提供一个经典示范.

本文第 2、3 节描述了 5G-AKA 的协议细节及当前研究进展. 第 4 节介绍 SM-AKA 协议的鉴权流程. 第 5 节对提出的改进方案展开了形式化验证和讨论. 最后, 在第 6 节中给出结论.

2 相关工作

随着 5G 时代的到来, 研究人员在 5G 鉴权协议上投入了大量的精力, 取得了可观的进展. 在此, 我们主要从脆

弱性挖掘、协议改进和形式化分析 3 个方面阐述相关研究.

目前, 业界已经发现了鉴权协议的一些脆弱点. Arapinis 等人[10]利用变量的结构缺陷, 提出了一种连接性攻击, 尽管这种攻击的设计初衷是针对 4G 通信的, 但其对 5G-AKA 依然适用[9]. 具体来说, 由于协议在认证失败后, 会产生两种错误消息, 这两种消息的长度和组成不同, 因而易于区分. 攻击者利用这一缺陷能够判断两个未知 UE 之间的关系, 从而打破协议的无连接性. 攻击能够顺利实现的另一个原因是消息通过空口传输, 攻击者很容易在空气中捕捉到它. 在鉴权中, 空口传输是不可避免的, 因而只能通过改进消息设计阻止可能的攻击. 聚焦于序列号的加密机制, Borgaonkar 等人[11]利用变量的不完备加密, 巧妙地设计了一种信息破译攻击. 攻击者主要通过收集大量鉴权失败响应发挥破译算法的作用. 具体来说, 攻击者首先令 UE 和 HN 之间反复进行鉴权, 并通过信息截断等手段, 让鉴权结果交替出现成功和失败. 而在每次获取失败响应时, 攻击者都会通过重放过时的消息以确保结果是重同步类型的, 所谓“重放”是指把同一消息重复发送超过两次. 经过大量的重复操作, 攻击者可以收集到多条重新步类型的失败响应. 然后依据异或计算的特点, 攻击者采用迭代的方式逐位计算出序列号的二进制编码. 最后攻击者可以根据破译的编码在一段时间内的变化推断出用户的隐私, 如通信活动、地理位置等.

Arapinis 等人[10]提出了 Fixed-AKA 协议, 以应对可能发生的针对消息类型的攻击. 该协议通过公钥加密的方式令两种失败消息具有相同的外部组成, 从而使攻击者无法仅通过观测消息的长度和结构就获悉消息的类型, 故可以阻断攻击行为的进行. 通过研究, Fouque 等人[12]发现上述协议在变量的新鲜度维持上存在缺陷 (新鲜度是指消息的唯一性、时效性、不可重用性, 即确保消息是唯一生成, 同时融入时效信息且不能被重用), 以致其容易遭受一种重放攻击. 于是, 新的改进协议 PRIV-AKA 被提出. 该协议通过使服务侧的序列号只有在收到用户侧的确认消息后才会增加. 这维持了鉴权变量的新鲜性, 避免消息重放. 同时, PRIV-AKA 吸收 Fixed-AKA 的优势, 使鉴权失败消息的类型不可区分. 然而, Koutsos 等人[9]发现, PRIV-AKA 无法满足协议的无连接性要求, 攻击者能够轻松的区分目标用户和未知用户之间的关系. 于是, 他们提出了 AKA$^+$协议, 通过对 SUPI 和 GUTI 的交叉运用, 以及对子协议优先级的设定, 很好地满足了 UE 和 HN 之间双向认证需求. 同时, AKA$^+$每成功运行一次都会为下一次的鉴权生成 GUTI, 这使得相对高效的 GUTI 子协议可以被更多的使用. 此外, Braeken 等人[13]提出了一个两阶段鉴权协议. 它设计了一个身份注册阶段, 通过把 SUPI 和私钥整合为动态的身份变量来实现信息的加密传输. 并且, 该协议仅使用了 2 次空口传输, 具有很高的认证效率. 该协议着重于满足协议在效率和安全方面的需求, 是一个较为优秀的改进方案. 聚焦于提升鉴权效率, Gharsallah 等人[14]提出了 SEL-AKA 协议. 协议去除了认证对全局公钥体系的依赖, 取而代之的是, 用一个新提出的参数发挥相应的作用. 通过其在用户端和服务端的共享, 该参数完成了信息在双端的交换. 分析表明, SEL-AKA 很大程度上简化了流程和计算, 满足了鉴权在功能和效率方面的要求. 另外, Braeken 等人[15]在空口消息中, 创新的使用随机数来取代序列号, 以减少协议所需的通信步骤的数量. 而 Han 等人[16]则采用了一种新的模式, 把移动边缘计算用于鉴权, 大大丰富了用于变量检验的算力, 可以明显降低认证延迟.

形式化分析作为一种强有力的属性验证方法, 出现在大量 5G 鉴权协议的研究工作中, 用于评估其安全有效性. 具体来说, Basin 等人[7]使用 Tamarin 证明器[17]展开协议的安全属性验证, 其把参与鉴权的终端归为 3 个实体, 即 UE、服务网络和 HN, 然后通过 Tamarin 的规则描述协议内容, 通过引理给出隐私目标. 最终, 分析结果显示, 5G-AKA 无法满足无连接性要求以及可能遭受一些特定情况下的攻击. 类似的, Cremers 等人[8]也使用 Tamarin 对 5G-AKA 展开分析, 并得到了很有价值的结论. 不同的是, 后者进一步细化协议的参与者, 引入分别执行用户管理和认证管理的单元, 使协议分析更加精细. 当然, 定义的规则和引理也更加复杂. 另外, Edris 等人[18]采用 ProVerif 工具[19]对 5G-AKA 展开形式化验证. 除此之外, 也有一些针对 5G-AKA 协议变种的分析案例. 如 Gharsallah 等人[14]结合 AVISPA[20]和 SPAN[21]对其提出的 SEL-AKA 协议展开安全属性评估, 并得到一些很有价值的结论. Braeken 等人[13]使用 RUBIN 逻辑[22]检验其提出的协议, 验证了协议拥有的优良特性. Koutsos 等人[9]使用 BANA-COMON 逻辑[23]评估其提出的 AKA$^+$协议, 有力地证明了协议能够满足其声明的若干要求.

在本文, 我们针对 5G-AKA 和 AKA$^+$协议的缺陷, 提出改进方案. 相较于现有研究, 该方案采取一种交替运行的模式, 不仅可以抵御已知的攻击, 还能够大大提高鉴权的平均效率. 同时, 改进方案减少通信步骤, 降低资源消耗. 而通过形式化验证, 我们证明了该方案可以满足协议的包括无连接性在内的诸多要求, 具有很强的安全特性.

此外, 由于认证过程中使用较多的异或操作, 为了更真实的建模 SM-AKA 协议, 我们选用功能强大且支持异或操作的 Tamarin 形式化工具.

3 预知识

本节将阐述我们聚焦的 5G-AKA 协议, 同时介绍参与认证的实体.

3.1 实体描述

认证协议主要涉及 3 个实体: UE、服务网络 (service network, SN) 和归属网络 (home network, HN), 其中:

(1) UE 是指获取通信服务的载体, 例如移动电话. 每个 UE 都包含一张 USIM 卡, 它存储一组用于鉴权的变量: 用户永久身份标识 (SUPI)、全局唯一临时身份标识 (GUTI)、GUTI 标识符 (V-GUTI)、用户密钥 k、UE 侧的序列号 SQN_U 和公钥 pk_N.

(2) SN 控制着基站, 用于给用户提供移动网络接口, 鉴权消息可以通过 UE 和 SN 之间的无线频道连接实现空口传输.

(3) HN 是指用户注册网络, 存储着用户的身份信息. 它通过维护一个数据库参与鉴权, 存储的变量包括: SUPI*、GUTI*、V-GUTI*、k^*、HN 侧的序列号 SQN_H 和私钥 sk_N.

在鉴权协议中, 通常 UE 是客户端, HN 是服务端, 二者共享 SUPI、GUTI、k 和 V-GUTI. 但为了方便表述, 我们用“*”来区分这些变量在双侧的表示. SN 负责二者鉴权消息的双向转发, 即将 UE 的信息传输到 HN 和 HN 的消息传输到 UE. 这样做的原因是 UE 和 HN 的地理位置不会总是很近, 因此需要一个中介来确保通信可达. 事实上, SN 和 HN 之间通过可信信道通信, 可以确保信息的安全稳定传输. 因此, 为了简化, 我们把 HN 和 SN 归为一个实体, 并用 HN 表示 (这种做法已经被一些研究所采纳, 如 Koutsos 等人[18]和 Braeken 等人[8]的工作. 当然, 也有一些工作把 HN 和 SN 看作是单独实体, 如 Basin 等人[6]和 Cremers 等人[9]的研究).

3.2 5G-AKA 协议

这里描述 3GPP 组织制定的 5G-AKA 协议, 该协议包含 3 个必备的通信阶段和一个非必须的重同步阶段.

(1) UE 向 HN 发送初始化消息

UE 通过向 HN 发送自己的身份信息开启一个鉴权会话. 事实上, 代表身份的变量为恒定不变的 SUPI, 然而为了防止由于 SUPI 泄露而遭受一些外部攻击, 该变量通常会被密文传输. 实际上, UE 会引进一个非对称加密算法, 同时结合一个新鲜的随机数 r 和公钥 pk_N, 把 SUPI 转化为隐藏身份标识符 SUCI:

$$\text{SUCI} = \{\text{SUPI}\}^r_{pk_N}.$$

而 SUCI 是临时的, 只能被使用一次, 因而可以确保身份信息不被公开. 接下来, SUCI 被作为身份变量发送给 HN.

(2) HN 生成并转发检验消息给 UE

HN 一旦收到 UE 的身份信息, 会开启一个鉴权会话. 对于 SUCI 消息, HN 会结合私钥 SQN_H 使用非对称解密算法解析 SUCI. 依据获得的 SUPI, HN 会在用户注册数据库中检索出 SQN_H 和 k. 然后, 两个哈希函数利用一个新生成的随机数 r_m 把 SQN_H 和 k^*融入检验变量 MAC^* 和 $CONC^*$ 中, 即:

$$MAC^* = f_1(k^*, \langle SQN_H, r_m\rangle),\ \ CONC^* = SQN_H \oplus f_5(k^*, r_m),$$

其中, $\oplus$ 表示异或操作, f_1 和 f_5 为二元哈希函数. 最后 HN 把完整的用于查验的消息 S^* 发送给 UE, 其中:

$$S^* = \langle CONC^*, MAC^*, r_m\rangle.$$

(3) UE 执行检验并把结果反馈给 HN

UE 收到消息 S^* 后, 首先解析出 HN 的序列号 $SQN^*_H = CONC^* \oplus f_5(k, r_m)$, 然后计算出 UE 侧的身份变量 MAC= $f_1(k, \langle SQN^*_H, r_m\rangle)$, 最后进行身份和新鲜度的检验, 即① $MAC^* == MAC$, ② $SQN_U < SQN^*_H$. 其中, ==表示等价判断符号. 作为结果, 有 3 种情况.

● 如果①=True 且②=True, 则鉴权成功, UE 会给 HN 反馈消息 $X = f_2(k, r_m)$, 并把其自身的序列号 SQN_U 设置为 SQN^*_H.

● 如果①=True 且②=False, 则新鲜度检验失败, 此时, UE 首先生成变量 $CONC$ 和 $MACS$, 即:

$$CONC = SQN_U \oplus f_5^*(k, r_m),\ \ MACS = f_1^*(k, \langle SQN_U, r_m\rangle),$$

然后给 HN 返回重同步消息 $S = \langle CONC, MACS\rangle$.

● 对于其他情况, UE 直接给 HN 返回鉴权失败“Failure”消息.

响应发出后, UE 侧完成认证流程并结束当前会话.

(4) HN 侧进行重同步响应

当收到重同步响应消息时, HN 从消息中恢复出 UE 的序列号 SQN_U^*, 并生成 HN 侧身份变量 $MACS^*$:

$$SQN_U^* = CONC \oplus f_5^*(k^*, r_m),\ MACS^* = f_1^*(k^*, \langle SQN_U^*, r_m\rangle).$$

然后, 检验 $MACS^* == MACS$ 是否成立. 如果通过, 则 HN 更新自身的序列号, 即 $SQN_H = SQN_U^* + 1$. 这可以使 SQN_H 的值回归到正常范围, 避免因 SQN_H 的意外增加 (比如遭受攻击) 而导致新鲜度检验永久性失败. 如果不通过, HN 不采取任何操作.

4 SM-AKA 协议

我们在这里描述 SM-AKA 协议的详细设计, 它在 5G-AKA 及其一些修正版本[9,13]的基础上, 针对 2 种失败消息可区分、非对称加密方式计算复杂等缺陷改进. 依据 GUTI 是否可用, SM-AKA 设计了 2 种运作机制, 分别为 GUTI 模式和 SUPI 模式. 每种模式都包含 2 次鉴权变量通信和 1 次信息检验, 总共 3 个步骤. 2 种模式的差别主要体现在身份变量的选取, 集中于第 1、2 步骤, 而第 3 个步骤完全相同.

在第 3.1 节中, 我们介绍参与 5G 鉴权的实体所包含的基础变量. 其中的 V-GUTI 是区分 GUTI 和 SUPI 模式的依据. 这 2 种模式交替运行, 其选用机理为: HN 侧默认选用 GUTI 模式, 从认证变量 M 中提取 P 并在数据库中检索对应身份信息的鉴权参数, 若存在且 V-GUTI 为 True, 则 SM-AKA 会继续运行 GUTI 模式. 若检索不存在, 则跳转进入 SUPI 模式. 在 UE 侧, V-GUTI 发挥作用后总会被置为 False. 2 种模式都包含完整的流程, 可以独立完成认证. 每当鉴权成功时, UE 侧都会把 V-GUTI 置为 True, 以便下次认证能够选用 GUTI 模式.

4.1 GUTI 模式

本节详细描述协议的 GUTI 模式, 其包含 3 个步骤, 具体如图 3. 值得注意的是, 只有当 V-GUTI 为 True 时, GUTI 模式才会被选用.

(1) UE 向 HN 通信阶段

一旦鉴权会话开启, UE 就着眼于认证变量的生成. 首先, UE 会产生一个随机数 r, 并通过一个哈希函数把其转化为密文 a, 以便直接在空气中传输:

$$a = f_2(k, \mathrm{GUTI}) \oplus r.$$

紧接着, 结合时敏信息 r 和 GUTI, 用于身份检验的变量 MAC 被生成:

$$MAC = f_1(\mathrm{GUTI}, r) \oplus SQN_U.$$

UE 在此模式下将采用 GUTI 作为其身份的标志 P, 即 P=GUTI. 从而, 完整的认证变量可被获得并用 M 来表示:

$$M = (P, MAC, a).$$

最后, M 将被发送给 HN. 发出后, 序列号 SQN_U 的值增加 1, 同时 V-GUTI 被设置为 False 以防止 GUTI 被重放.

(2) HN 向 UE 通信阶段

HN 在接收到认证变量后, 首先依据明文传输的 GUTI 在用户数据库中检索, 以获得用户密钥 k^*和序列号 SQN_H. 同时, HN 侧的 V-GUTI 被更改 False, 以表示 GUTI 不再可用. 然后, HN 从 M 中恢复出被加密传输的随机数, 即:

$$r^* = \pi_3(M) \oplus f_2(k^*, \pi_1(M)).$$

进一步, UE 侧的序列号会由于随机数的解析成功被重载:

$$SQN_U^* = f_1(\pi_1(M), r^*) \oplus \pi_2(M).$$

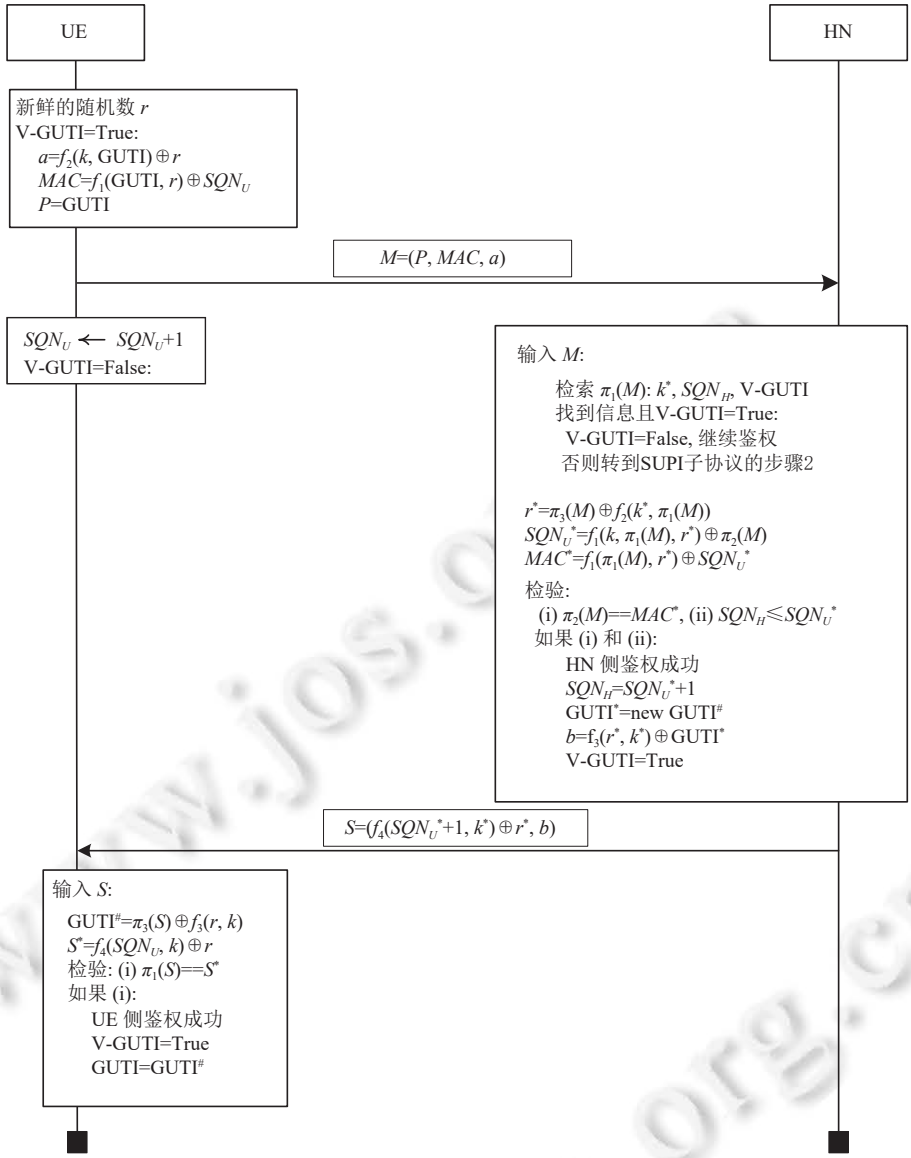

图 3 GUTI 子协议具体流程

接下来, HN 生成它的身份检验变量 MAC^*:

$$MAC^* = f_1(\pi_1(M), r^*) \oplus SQN_U^*.$$

因此, 鉴权协议规定的两个检验可以被实施, 即:

$$\text{(i) } MAC^* == \pi_2(M), \quad \text{(ii) } SQN_U^* \geqslant SQN_H.$$

对于结果, 如果检验 (i) 和 (ii) 均通过, 则代表 HN 侧鉴权成功. 此时, 新的临时身份标识符 GUTI^*被生成, 并被加密为中间变量 b:

$$b = f_3(r^*, k^*) \oplus \text{GUTI}^*.$$

并且, SQN_H 的值被更新为 $SQN_U + 1$, V-GUTI 被重置为 True. 最后, HN 进一步把认证变量补充完整, 并形成 S:

$$S = (f_4(SQN_U^* + 1, k^*) \oplus r^*, b),$$

S 被发送给 UE, 以用于开展最后的确认阶段.

(3) UE 信息查验阶段

这一阶段在 UE 侧完成, 是鉴权的最后一步. 首先, UE 根据 S 的组成以及自身的基础变量, 重载出被密文传输的 GUTI$^{\#}$:

$$\text{GUTI}^{\#} = \pi_3(S) \oplus f_3(r,k).$$

然后 UE 的确认变量 S^*被计算:

$$S^* = f_4(SQN_U, k) \oplus r.$$

紧接着, 检验 (i) $S^* == \pi_1(S)$ 被实行. 如果检验通过, 则 UE 侧鉴权成功. 此时, V-GUTI 的值被置为 True, 并且, GUTI 被更新为 GUTI$^{\#}$, 以便在下一次鉴权中使用.

至此, GUTI 模式下的 SM-AKA 协议流程结束, 特别的, 只有当 UE 和 HN 两侧的鉴权都成功时, 整个过程才为认证成功.

4.2 SUPI 模式

我们在此细致描述协议的 SUPI 模式, 其只有在 V-GUTI 为 False 时才被选用. 图 4 刻画了 SUPI 模式的具体流程, 与 GUTI 模式类似, 它也通过 3 个步骤来完成鉴权, 且每个步骤都有相同的地方. 本节仅描述不同之处, 而其余部分可参考上述 GUTI 模式的实现.

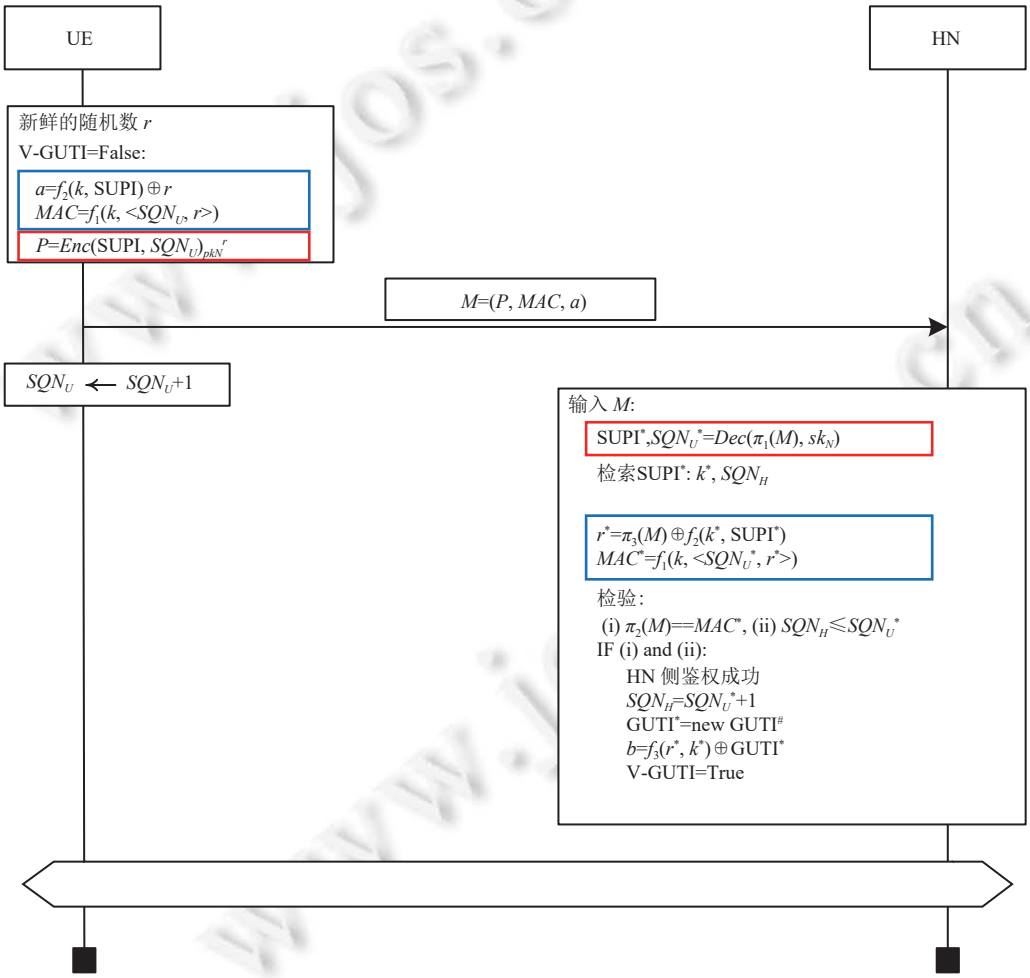

图 4 SUPI 子协议具体流程

(1) SUPI 的加密传输

由于 GUTI 不再可用, 因而 UE 只能通过加密 SUPI 来完成身份信息的传递. 具体来说, UE 首先结合随机数 r 和序列号, 采用一个基于公钥的非对称加密算法把 SUPI 化为隐藏变量 P:

$$P = Enc(\text{SUPI}, SQN_U)^r_{pk_N},$$

其中, P 代替 GUTI 充当身份变量进行传输.

从 GUTI 模式跳转到 SUPI 模式后, HN 选用基于私钥的非对称解密算法把 SUPI 和 UE 的序列号从变量 P 中恢复:

$$\text{SUPI}^*, SQN^*_U = Dec(\pi_1(M), sk_N).$$

接下来, HN 在用户数据库中检索 SUPI, 以获取用户密钥 k^*和 SQN_H. 此部分已在图 4 中用红色方框标出.

(2) 身份检验变量

在此模式下, UE 侧身份检验变量的生成不再依赖 GUTI, 而是基于带有时效特征的序列号 SQN_U 和随机数 r, 以及带有身份特征的密钥 k, 即:

$$MAC = f_1(k, \langle SQN_U, r\rangle),$$

而在鉴权过程中, UE 通过加密的方式把 r 传输给 HN, 以便生成对应的 MAC 变量. r 被隐藏为 a 的公式如下:

$$a = f_2(k, \text{SUPI}) \oplus r.$$

而在 HN 侧, 它首先需要恢复出 r, 可通过下式实现:

$$r^* = \pi_3(M) \oplus f_2(k^*, \text{SUPI}^*),$$

其中, $\pi_3(M)$指的是上述的隐藏变量 a. 进而, 身份检验变量被计算:

$$MAC^* = f_1(k^*, \langle SQN^*_U, r^*\rangle).$$

通过此, HN 可以正常的进行两个检验. 此部分已在图 4 中用蓝色方框标出.

5 形式化分析

在本节, 我们形式化地评估 SM-AKA 协议的功能性和安全性. 我们主要从形式建模、证明目标与策略、攻击模型的设定等方面展开介绍. 此外, 对于鉴权协议的属性规范, 我们也做了必要的描述.

5.1 形式化简介

形式化分析主要通过把协议用规范化语言描述, 然后通过逻辑定理推导, 检验其是否满足某一方面的要求. 这个过程主要包括协议的流程和属性描述、协议的推导验证 2 个部分.

流程属性建模主要是把鉴权实体的信息生成、通信交互、变量处理等描述规范化, 同时, 还包含对攻击模型能力的模拟. 形式化描述语言基于数学模型, 克服自然语言或者程序语言描述的不精确性. 实际上, 由于形式化系统层出不穷, 因而吻合系统自身特点而构造的语言也多种多样, 这使得推理工具之间的兼容性很差. 因而, 业界往往会选择利用一些通用性较强的描述语言. 协议验证是对格式化后的属性实施推理, 判断各项属性是否达到预期目标, 得出对协议性能评估的结论. 它基于鉴权中有序的变量生成、传递、转化以及销毁的过程, 通过把属性要求设定为推理终点, 施行一系列的逻辑引导, 检验属性能够被满足. 随着移动网络技术的快速发展, 形式化验证已经变得十分重要, 成为协议研究必不可少的一环.

形式化分析通过推导证明完成目标的验证, 主要是检查逻辑上的缺陷, 对协议的安全隐私特性展开评估. 而对于性能效率等属性, 形式化难以应对, 因而, SM-AKA 的性能分析主要通过理论探讨完成. 形式化分析应用于 5G 鉴权, 最大的难点就是如何把协议的表达方法从自然语言转化为逻辑语言, 使其能够输入到证明工具, 进行数学推理. 事实上, 协议流程通常是复杂的, 表述方式通常是多样化的. 而逻辑语言具有高精准性的特点, 即使存在一点差异, 也会对协议的含义造成很大偏差. 因此, 准确无误地把自然语言形式转化为逻辑表达是很具挑战性的. 同时, 属性要求以文本形式给出, 没有具象化阐述, 如何按照其含义精准的抽取出具体指标, 把其转化为标准的逻辑形式也是一大难点. 由于形式化语言丰富多样, 且高度凝练抽象, 因而需要花费大量精力完成标准化建模. 此外, 协议的分

析需要模拟数量庞大的攻击可能, 通过多重搜索算法推理证明, 以检验攻击是否存在. 这一过程对逻辑推导、攻击解析提出了很高的要求, 需要经过复杂的循环演算才能检验协议的属性是否被满足. 为了应对这些困难, 本文主要采用 Spthy 语言建模协议流程和属性, 采用 Tamarin 证明器进行自动化的推导验证. 在本文中, 多重搜索算法主要指的是: Tamarin 首先考虑协议所有的可行运行情况, 建立完整的状态空间, 然后依据一定的搜索方式在空间中查找符合特定条件的状态, 以期发现可能存在的协议漏洞.

5.2 属性要求

3GPP 在移动通信标准文档的 R16 版本中对 5G 鉴权做出了属性要求, 为我们的形式化分析设定了明确的检验方向. 其以文本形式给出, 我们在此介绍, 具体如下.

(1) 认证属性. 协议应通过双端实体 (此处指 UE 和 HN) 的一系列通信活动完成双向认证, 确保正常用户能够顺利接入服务网, 且提供服务的是合法服务网; 同时, 相同的会话钥不应该被生成两次, 即要确保其唯一性, 防止废弃的变量被非法重用.

(2) 保密属性. 协议应保证参与认证的永久性变量不被泄露, 包括密钥 k、私钥 sk_N、序列号 (SQN_U, SQN_H) 和永久身份标识符 SUPI. 而临时产生的鉴权变量, 如 GUTI、SUCI 等, 可以明文传输. 一些临时变量的公开可能会使协议遭受攻击, 因而需要融入对完整认证流程的考虑.

(3) 隐私属性. 协议应具备很好的防攻击能力, 保证用户隐私安全, 如通信行为、通信习惯、地理位置等, 这些信息都应具有不可追踪性. 同时, 协议应满足无连接性, 即让攻击者无法区分多个隐式目标之间的关系, 如判断两次追踪的 UE 是否为同一个 UE.

(4) 性能属性. 在确保基本的认证和安全属性不受影响下, 协议应尽量简洁, 提升认证效率. 同时, 协议应尽可能地使用更少的变量和更低的资源. 协议应具备较强的抗干扰性, 确保鉴权能够长久稳定.

5.3 形式化工具

在推理验证协议属性的过程中, 我们不可避免的需要用到一些自动证明机. 随着形式化的不断发展, 业界已经开发出了各具特色的工具, 主要包括: AVISPA、ProVerif、Maude-NPA、Scyther、Tamarin 等.

AVISPA[20]的初衷是为了克服复杂协议的描述和证明的难题. 它提供了面向对象的标准化语言, 并整合了 4 种经典的证明器, 不仅可以发现协议的脆弱点, 还能够验证具有无限会话特性的协议. ProVerif[19]通过若干预定义的规则判定事件是否会发生, 从而实现对协议属性的验证. 它解决了状态爆炸的难题, 可以模拟协议无限次数的执行. Maude-NPA[24]在 2009 年被提出, 支持多种运算: 加解密、交换律、结合律等, 并且具备无限会话模型, 不仅能够完成逆向的漏洞挖掘, 还能正向的验证协议的安全性. Scyther[25]对于具有无限轮数的协议可以给出具体的终止, 并且能够同时分析多个协议. 它在脆弱性查找、实体功能实现以及功能性证明发挥着重要作用. Tamarin[17]利用 Scyther 的逆向搜索算法, 提供两种操作机制: 全自动分析和交互分析模式. 它基于集写规则和逻辑引理描述协议流程, 可以分析复杂的信息流, 支持加解密、异或等多种代数运算, 不仅支持 Dolev-Yao 模型, 还支持强安全模型. 为了证明和分析 SM-AKA 协议, 我们选择使用 Tamarin 证明器[17]. 它是最先进的协议验证工具, 已经被大量用于分析基于密钥的通信协议, 如 TESLA 协议、YubiKey 协议等. 它支持状态化的协议, 可以实现很大程度的自动化, 能够建模等价属性, 这些是验证无连接性要求的必要特征. 同时, 它可以模拟多种攻击方式. Tamarin 支持众多的加密方法和代数运算, 尤其是鉴权协议严重依赖的异或运算. 综合来看, Tamarin 完美融合了完成 5G 鉴权分析所需的全部特性.

Tamarin 基于 Spthy 语言建模, 完成协议流程的描述, 利用时序逻辑和消息耦合, 完成协议属性的描述. Spthy 是一种程序语言, 可把协议以编程语言的形式表达, 适用于计算机的编译运行, 可以用于协议的标准化建模. 在 Tamarin 中, Spthy 语言首先被用于协议的程序化表达, 将自然语言文本描述的协议转化为计算机可编译、计算的逻辑形式. 在使用 Spthy 语言表述协议后, Tamarin 工具根据内置的搜索算法对协议属性进行证明. 因此, Tamarin 主要借助 Spthy 语言的标准化建模完成协议的形式化分析, 基于 Spthy 的标准化建模是采用 Tamarin 对协议进行形式化分析的第 1 步. Tamarin 主要定义了两类对象: 多重集写 *rule* 和属性 *lemma*, 其分别服务于协议的流程规范

和属性描述. 前者主要把多个阶段性事实组合在一起以模拟当前通信实体的状态, 通过事实的消耗与演变, 实现实体状态的转变; 而后者使用数学逻辑语言把特定的阶段性事实组合在一起, 以时间顺序和全局限制精确定义各项属性, 并通过搜索算法验证协议是否满足各方面的要求. 此外, Tamarin 通过一些内置的函数实现加解密运算、异或运算等, 我们在建模过程中不需要显式的写出计算过程, 而是以一种黑盒的形式实现所需的运算. 同时, 对于变量的生成、时效性、新鲜性等, 它也通过一些内置符号给出了约束. 对于检测到的攻击, Tamarin 以攻击流程图的形式给出, 明确刻画攻击的参与实体、实现方式、协议薄弱点等要素. Tamarin 是基于网页的工具, 以客户端-服务器的模式运行. 在全自动模式下, 用户通过客户端将基于 Spthy 语言的协议脚本上传至服务端, 随后, 服务端解析 Spthy 脚本, 自动化的逐一证明脚本中定义的属性 *lemma*, 并暂存结论. 在完成所有的属性证明后, 服务端将结果以攻击流程图的形式返回客户端, 同时给出形式化分析的总结. 如果 Spthy 脚本中属性全部得到验证, 则 Tamarin 仅返回分析结论, 而不返回任何流程图.

5.4 攻击模型

在评估鉴权协议之前, 我们首先对攻击者能力作出假设, 以便模拟可能发生的攻击.

SM-AKA 协议包括 3 个通信实体: UE、SN 和 HN. 事实上, SN 和 HN 之间维护着一个安全信道, 可以提供封闭可靠的信息传输, 因此在鉴权协议的研究中, 它们被归纳为一个实体. 进一步的, 我们可以假设 SN 和 HN 之间的通信不存在攻击, 即在其中传输的消息均可以顺利通过且不会发生任何安全问题. 而 UE 和 HN 之间通过无线网络基站通信, 在这种场景下, 信息暴露在空气中, 攻击者仅通过简单的监听操作就可以接触到完整的信息内容, 因此这种通信被认为是不可靠传输. 在设计协议的过程中, 我们即要实现通信实体间的信息交互, 又要考虑到传输消息的加密性. 因此, 对于 SM-AKA 这样一个新协议来说, UE 和 HN 之间的安全性研究至为重要. 我们主要对此展开形式化分析.

结合上述, 我们在 UE 和 HN 之间引入一个标准的 Dolev-Yao 攻击模型[26]. 具体来说, 我们假设攻击者知悉 SM-AKA 协议的完整流程, 可以建立鉴权会话. 在 DY 模型中, 攻击者控制着整个通信网络, 可以窃听、拦截、伪造、注入、延迟和删除通信实体间被交换的消息, 同时可以进行公开的运算, 如异或、哈希等. 另外, 在现实中, 攻击者很难获取到存储在终端的用户信息. 因而, 我们假设攻击者只能接触到空口消息, 而无法知晓 SUPI、用户密钥等存储在实体中的信息. 对于序列号、随机数等鉴权信息, 由于其数位比较大, 因而攻击者无法通过盲猜知晓其具体数值. 除了对已有的通信设施发挥作用, 攻击者还可以伪造基站, 因此, 攻击者可以重放捕获的消息, 实施一些重放攻击等. 同时, 攻击者也可以创造虚拟用户, 故攻击者可以对服务网发起非法鉴权会话等.

5.5 SM-AKA 协议的形式建模

我们使用两种类型的规则描述 SM-AKA 的流程, 即注册 *rule* 和通信 *rule*. 协议的属性要求通过两种类型的引理来实现, 即功能 *lemma* 和安全 *lemma*, 其中, 前者对应于认证属性, 后者对应于保密和隐私属性, 而性能属性主要通过理论分析进行验证. 此外, 攻击模型的能力也通过多集重写规则表达, 即攻击 *rule*. 而攻击尝试则通过引理实现, 即攻击 *lemma*. 下面我们结合相应的例子详细说明每个对象的具体作用.

(1) 流程定义

流程的形式化主要由两类规则完成. 注册 *rule* 主要用于生成实体终端, 并赋予其相应的基础身份认证信息. 涉及的实体有两个, 即 UE 和 HN, 它们分别对应一个注册 *rule*. 我们在此以 UE 的 *rule* 为例, 具体实现如下:

rule init_UE:

[*Fr*(~*guti*), *Fr*(~*k*), *Fr*(~*sqn*)]

--[*Register*(~*guti*)]->

[!*State_0_UE*(~*guti*, ~*k*,~*sqn*)]

其中, *Register*(~) 为活动事实, 是该 *rule* 的标签. 该规则首先初始化 UE, 给其配备在第 4.1 节中规定的各项基本信息 (紫色部分), 然后通过 *Register*(~) 触发, 获得初始事实 (蓝色部分). 该事实是鉴权的起始状态, 在后续的认证步骤中被转化为代表下一阶段状态的事实. 上述创建了一个合法的 UE, 这是协议建模的第 1 步. 实际上, 该 *rule* 还包含

一些 Tamarin 的辅助证明项, 但为聚焦于协议流程, 同时便于理解, 我们仅描述其主干部分, 本文后续展示的 *rule* 也是如此.

通信 *rule* 是流程建模的主体部分, 主要实现信息的接收、处理和发送功能. 通信规则通常包含一些加解密、异或等操作, 在 Tamarin 中, 这些运算主要通过一些内置的函数来实现. 我们以第 4.2 节定义的通信流程为例:

rule ue_to_hn_GUTI:

let

P = ~*guti*

$a = f_2$(~*k*, ~*supi*, ~a_n ~b_n, y_n)

MAC= f_1(~*guti*, *r*) XOR ~*sqn*

M= <*P*, *MAC*, *a*>

SqnJ = *SqnUE*+'1'

v-guti = '0'

in

[!*State_0_UE*(~*guti*,~*k*,~*sqn*),
Sqn_UE (-*guti*, *SqnUE*, ~*sqn*, *count*),
Fr(~*r*)]
--[*UE_Send_HN*(~*guti*)]->
[*State_1_UE*(~*guti*, *SqnJ* , ~*k*, ~*sqn*, ~r_n, *v-guti*),
Sqn_UE(~*guti*, *SqnJ* , ~*sqn*, *count*+'1'),
Out(*M*)]

其中, XOR 为异或运算, f_1, f_2 为哈希函数, *Out*(*x*) 表示发出信息 *x*. 该规则首先经过一系列的信息集成和演变, 生成鉴权所需的变量 $M = (P, MAC, a)$, 然后基于 UE 的初始状态 (紫色部分), 通过一个 *UE_Send_HN*(~) 活动事实触发流程, 完成 3 个子活动: 发出消息, 由 *Out*(*S*) 函数完成; 更新 UE 状态, 由 *State_0_UE* 转化为 *State_1_UE*; 更新序列号和 V-GUTI 的值 (蓝色部分). 至此, 协议的第一阶段描述完毕, 而剩余部分也将通过这种事实转换和变量更新的方式完成.

通过上述, 我们用形式语言刻画了认证流程, 完成了协议的规则建模.

(2) 属性定义

我们将形式化的描述 SM-AKA 的认证属性、保密和隐私属性, 其分别通过功能性 *lemma*、安全性 *lemma* 实现.

功能性 *lemma* 通过给出一系列的可执行路径把既定的规则按照不同方式组合在一起, 检验协议是否满足功能要求. 执行路径都是基于协议的逻辑运行顺序而产生, 而一个引理往往仅表达局部的逻辑顺序. 下面, 我们给出一个 UE 的功能性 *lemma*:

lemma executability_ue_to_hn:

"(*All guti #i.UE_Send_HN*(*guti*)@*i* ==>
(*Ex #j. Register*(*guti*)@*j* & *j*<*i*)"

该 *lemma* 表示, 如果有活动 *UE_Send_HN*(*guti*) 在时间点 *i* 发生, 则在此之前必有 *Register*(*guti*) 活动发生. 这是一条针对第 2 节所述流程的执行路径, 即先 *init_UE rule*, 然后 *ue_to_hn_GUTI rule*. 这种执行顺序表示在鉴权过程中, 如果 UE 发出一条认证消息, 则这个 UE 必定经过注册阶段并被赋予基础身份变量. 这是协议正常实现双向鉴权功能的一环, 即要求终端必须是合法生成的, 且必须在参与鉴权之前完成这一过程.

安全性 *lemma* 主要是在正常轨迹的基础上, 把可能存在的情况考虑进来, 以探索协议的运行状况. 为了模拟一些情景, *lemma* 做了特定的假设. 衡量这些假设的破坏性可以评估协议的安全属性, 以密钥 *k* 的保护为例:

lemma ue_to_hn_attack:

"*All guti #i.UE_Send_HN*(*guti*)@*i* ==>
(*Ex #j.Register*(*guti*)@*j* &*j*<*i*)
| (*Ex X k #r.Rev*(*X*, *k*)@*r* & *Honest*(*X*)@*i*)"

上式除包含一个正常 *lemma* 之外 (紫色部分), 还包含一个信息泄露假设, 即密钥发生泄露 (蓝色部分). 该引理表明, 如果在时间点 *i*, 活动 *UE_Send_HN*(*guti*) 被触发, 除了可以推断出一个合法 UE 被注册 (紫色部分), 还可能是由于一个攻击者 *X* 在时间点 *i* 获取 *k* 并据此伪造一个非法 UE. 而这个伪造的 UE 由于具备合法的密钥, 可以正常的参与认证, 这揭示密钥的泄露会给认证体系带来巨大的安全隐患.

我们还围绕着这两种类型定义其他的安全性 *lemma*, 通过这些 *lemma* 来准确刻画协议属性, 为后续推理提供结束节点.

(3) 攻击能力定义

本节将攻击者的能力形式化, 以便将攻击行为整合到搜索证明中, 这通过攻击规则和攻击引理实现.

攻击 *rule* 客观陈述攻击者的能力, 表示其可以接触到什么层次的信息. 它也给出每个基本信息的暴露标准, 规范哪些变量是公开的, 哪些是私密的. 这里我们以 UE 的密钥 *k* 为例:

```
rule reveal_UE_k:
    [!State_0_UE (~guti, ~k, ~sqn)]
        --[Rev (~guti, <'k', ~k>)]->
    [Out(~k)]
```

其中, *Out*(*x*) 表示发出消息 *x*. *Out* 函数发出的所有消息都被广播到空气中, 攻击者有能力捕捉到它. 该 *rule* 表示在 UE 的初始状态 (紫色部分), 如果有 *Rev*(*x*) 活动, 密钥 *k* 将被发出 (蓝色部分). 这意味着 *k* 将从私有变为公开, 将被攻击者捕获. 通常我们会将该 *rule* 视为一种假设, 以探索密钥泄漏时协议的安全状态. 该 *rule* 规定了攻击者获取密钥 *k* 的能力, 这种情况在现实中是很难实现的, 但我们希望能够模拟这种情形以查找潜在的脆弱点. 另外, 我们也形式化的定义其他的能力, 用于模拟可能的攻击方式.

攻击 *lemma* 基于预定义的 *rule* 发起攻击活动, 我们通过评估其可行性, 界定协议的安全性能, 以下式为例:

```
lemma reveal_sqn:
    "(All X K sqn #i.Rev(X, k)@i & Honest(X)@i)==>
        (Ex#j. Rev(X, sqn)@j & j>i)"
```

上式表明, 当密钥 *k* 在时间点 *i* 被泄露, 之后可能存在一个时间点 *j* 使得序列号 SQN_U 也被泄漏. 这描述一个在已知 *k* 的情况下, 攻击者试图获取 SQN_U 的活动. 值得注意的是, 这里引用一个针对序列号的攻击 *rule* (紫色部分), 其并没有在本文列出, 但其形式与 *reveal_UE_k rule* 类似. 证明器将通过判断该 *lemma* 是否为真来给出分析结论. 如果该 *lemma* 成立, 我们可以断定序列号的安全性高度依赖于密钥. 一旦密钥被公开, 序列号也将不会是私密的.

5.6 证明目标与证明策略

通过建模, 我们已经形式化的描述协议的流程、属性和攻击者的能力. 在本节, 我们将使用 Tamarin 证明器验证其属性定理. 我们主要介绍证明过程、证明目标以及采取的策略.

基于 Tamarin 的形式化分析主要分为协议建模、协议分析和得出结论 3 个阶段. 在协议建模阶段, 我们利用 Spthy 语言对协议流程程序化建模, 得到分析脚本, 以计算机能够理解的形式表述协议. 在第 2 阶段, Tamarin 基于 Spthy 脚本展开自动化推理, 利用自身的搜索算法查找协议潜在的漏洞并构造攻击方式, Tamarin 的推理界面如图 5 所示. 在第 3 阶段, Tamarin 给出结论, 输出结果. 如果协议存在漏洞, 则结果包含一系列的攻击流程图, 反之, 则直接给出“属性已完成验证”的结论.

我们在第 5.2 节罗列了 3GPP 规定的 5G 鉴权要求, 目标就是评估 SM-AKA 协议能否满足这些属性. 就建模过程而言, 上述的 *rule* 主要提供一些陈述实体状态的事实, 是 *lemma* 的骨干组成部分, 并为其服务. 而 *lemma* 是协议属性的载体, 是真正需要被证明的部分. 按照既定的变量、时间点和活动, 我们可以对其逻辑推导, 以达到对属性评估的目的. 因此, Tamarin 的证明目标就是以预定义的 *lemma* 为基础, 生成一个容纳所有可能状态的集合, 然后通过内置算法在状态空间搜索, 通过协议的模拟运行查找潜在的脆弱点. 同时, 它也会使用攻击模型对协议实施试探性攻击, 并通过攻击的结果来得出结论. 可以发现, Tamarin 主要通过判断 *lemma* 描述的内容是否成立发挥作用, 因为 *lemma* 已经对属性要求做出了精确的定义. 我们的形式化目标就是通过推演标准化描述以间接完成属性验证.

在 SM-AKA 的建模中, 我们构建了 27 个 *rule*, 其中 18 个用于协议流程, 9 个用于攻击模型能力的描述. 我们构建了 15 个引理, 其中用于功能和安全属性的个数分别是 6 和 9. 考虑到庞大的状态空间, 我们没有采用 Tamarin 的人机交互模式, 而是使用全自动模式开展对 *lemma* 的证明. 全自动模式的优势是操作简单, 不需要大量的人力, 这很适用于本协议的分析. 而产生的缺点就是状态空间不受人为引导, 会生成一些不切实际的协议状态, 从而带来不必要的逻辑推理. 受到之前工作[7,8]的启发, 我们添加了一些辅助 *lemma* 来粗略地指明推导方向, 限制 Tamarin

假定的协议状态, 以缩小搜索空间. 事实上, 一次性验证所有属性所需计算资源较大, 受限于实验条件, 我们把全部属性划分为若干份, 采取分阶段证明的策略. 我们在一台配有 AMD 3900X 的 12 线程处理器和 16 GB 内存的电脑上展开实验, 经过累计约 12 h 的自动分析, 完成了 SM-AKA 的形式化评估.

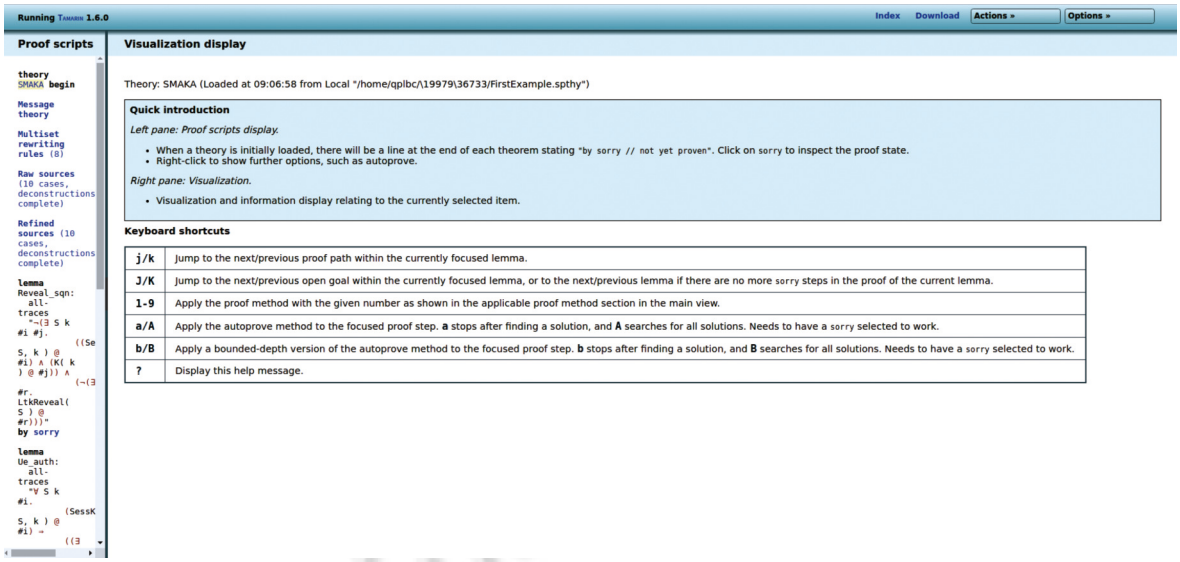

图 5 Tamarin 的推理界面

5.7 形式化分析结论

经过 Spthy 语言建模和 Tamarin 工具证明, 我们完成了 SM-AKA 协议的形式化分析. 图 6 展示了部分属性的分析结果, 可以发现, 图中 *lemma* 的证明结果均为“verified”, 即检验通过.

综合全部属性的形式化证明结果, 我们有以下结论.

(1) 功能属性得到验证, 主要包括鉴权功能、子模式切换、逻辑衔接、抗外部干扰等. 我们严格按照既定协议建模, 设定认证轨迹, 并依据属性引理推理, 成功证明了协议可以有序执行, 实现 UE 和 HN 之间的双向鉴权. 此外, 依据 V-GUTI 在 SUPI 和 GUTI 模式之间的切换也得到了验证; 在大量多次的模拟运行之后, 协议依然具备很好的鉴权性能.

(2) 安全属性得到验证, 主要包括空口参数的不可破译、用户基础信息的私密、认证变量的防重放、伪基站伪 UE 的鉴别等. 在 Dolev-Yao 攻击模型的设定下, Tamarin 没有给出任一攻击引理的成立条件, 而所定义的安全性引理也得到证明. 这说明 SM-AKA 具备很好的安全性能, 可以抵御所有已知的攻击方式.

(3) 我们通过开启两个形式化的 UE 认证会话, 让两个隐式 UE 分别依据功能性引理所描述的轨迹实施鉴权, 然后尝试用 Tamarin 确定正常状态和遭受攻击状态时两个 UE 的关系, 最终一无所获, 这证明 SM-AKA 可以满足协议的无连接性.

```
==============================================================
summary of summaries:

analyzed: /home/qplbc/桌面/my_protocl

  dummy (all-traces): verified (1 steps)
  sqn_ue_invariance (all-traces): verified (26 steps)
  sqn_ue_src (all-traces): verified (15 steps)
  sqn_hn_src (all-traces): verified (8 steps)
  sqn_ab_init (all-traces): verified (4 steps)
  ue_k_attack (all-traces): verified (4 steps)
  sqn_ue_nodecrease (all-traces): verified (3045 steps)

==============================================================
```

图 6 SM-AKA 协议形式化分析的部分结果

5.8 讨 论

本节我们从流程设计、变量优化等方面对 SM-AKA 协议的安全性和性能展开讨论.

(1) 安全性

首先, SM-AKA 协议通过 SUPI 和 GUTI 两种模式的交替运行实现了双向鉴权. 一方面, HN 通过解析来自 UE 的认证变量, 实施两种检验, 完成用户身份的核查. 另一方面, UE 通过处理来自 HN 的响应, 也实现对服务网是否合法的验证. 同时, 我们借鉴 5G-AKA 协议的一些优点, 把序列号融入变量维持其新鲜性, 可以确保鉴权消息仅一次有效. 这提升认证的时效性, 可以很好地抵御重放攻击.

其次, 鉴于 5G-AKA 协议可能遭受信息破译攻击[7], 我们提出改进措施. 我们在第 2 节已经说明, 这种攻击是由于序列号的不完备加密秘方式造成的. 事实上, 在 5G-AKA 的重同步响应阶段, UE 会发送消息: $CONC = SQN_U \oplus f_5^*(k, r_m)$. 而对于同一个终端 (即秘钥 k 保持不变), 当攻击者通过伪基站向 UE 重演带有 r_m 的消息 $S^* = \langle CONC^*, MAC^*, r_m \rangle$ 时, UE 每次反馈到空口的消息都会包含 $f_5^*(k, r_m)$, 即 $CONC_x = SQN_x \oplus f_5^*(k, r_m)$. 如果捕捉两次 $CONC$ 消息, 并经过一个与或操作, 就会得到 $CONC \oplus CONC_x = SQN_U \oplus SQN_x$. 对于攻击者而言, 其中仅有 SQN_U 和 SQN_x 未知, 但它们有已知关系: $SQN_x = SQN_U + x$, 且 x 已知. 基于此, 攻击者可以逐位破译用户的序列号 SQN_U, 从而非法获取隐私. 而 SM-AKA 去掉重同步响应, 取而代之的是, 它采用两个子模式交替运行的机制. 当需要重同步响应时, SM-AKA 会通过 SUPI 子模式把 UE 和 HN 的各项参数重新置于同步, 以完成协议的修复. 鉴于重同步响应导致序列号破译漏洞[11], 在设计协议的过程中, 我们特别注重基础变量的空口传输. 在加密算法的设计上, 我们把多个变量聚合在一起, 同时融入随机数, 通过哈希算法来把它们集成在一起, 以提升消息的不可再造性. 我们也通过理论计算证明 SM-AKA 采用的加密方式无法被破译, 攻击者难以有所收获. 因此, 尽管 5G-AKA 重同步响应被剔除, 但 SM-AKA 通过增加子模式机制弥补其功能, 还通过优化加密方式阻断其引发的攻击.

此外针对 5G-AKA 协议的连接性攻击[9], SM-AKA 可以完全阻止它. 该攻击实现是因为 $S=\langle CONC, MACS \rangle$ 和“Failure”两种失败消息, 由于它们的长度和组成不同, 故易于区分. 事实上, 5G-AKA 协议在认证失败后, 会发送这两种消息的一种. 基于这个漏洞, 攻击者首先会监听一次目标 UE_0 的成功鉴权, 并捕获 HN 侧的消息 $S^* = \langle CONC^*, MAC^*, r_m \rangle$. 然后在一个新的鉴权会话中, 攻击者会把 S^* 作为鉴权变量发给一个未知 UE_x. 由于该消息已经用过一次, 新鲜度检验必不会通过, 因而最终的鉴权不会成功, 故 UE_x 会返回错误消息. 而攻击者可以推断出消息的类型 t, 如果 t 是身份失败消息 (“Failure”), 则有 $UE_0 \neq UE_x$, 如果 t 是重同步消息 ($S = \langle CONC, MACS \rangle$), 则有 $UE_0 = UE_x$. 据此, 攻击者可以获知 UE_0 和 UE_x 的关系, 这打破了 5G-AKA 协议的无连接性, 而 SM-AKA 可以避免 5G-AKA 中的这种攻击. 不论鉴权成功与否, SM-AKA 均有且只有 M 和 S 两种鉴权消息, 不存在因鉴权结果不同而产生不同消息的情况. 攻击者虽然可以继续从空口中捕捉到鉴权消息, 但已无法区分出它们的不同. 这打破了上述连接性攻击的基础, 加强了对子模式切换的保护, 使攻击无法成功. 相较于 5G-AKA, SM-AKA 通过优化空口消息的设计避免连接性攻击, 大大提升了鉴权的安全性.

类似地, 我们已在前文描述, AKA^+协议也存在连接性攻击. AKA^+协议之所以会遭受连接性攻击, 主要原因是两个子协议的鉴权流程和空口消息结构不同, 易被攻击者区分和利用, 导致用户隐私泄露. 而 SM-AKA 协议提升了对无连接性的保护, 能够完美防范这种攻击. SM-AKA 协议和 AKA^+具有相似的鉴权模式, 即通过两个子协议交互的形式完成认证. 但 SM-AKA 重新设计了作用机制, 使两个子协议的流程 (均为由 UE 向 HE 发送鉴权消息, 由 HN 向 UE 返回鉴权响应, 详见第 4 节) 和变量结构 (均为 $S = (f_4(SQN_U^* + 1, k^*) \oplus r^*, b)$ 和 $M = (P, MAC, a)$, 详见第 4 节) 保持一致, 避免被攻击者区分, 从而阻断上述的连接性攻击. 相较于 AKA^+, SM-AKA 采用更完备的流程和消息发挥作用, 具有更好的安全性能.

(2) 性能

为了进一步探索协议的性能, 我们将 SM-AKA 与原始 5G 认证协议和最新变体进行比较. 考虑到延迟和计算复杂度, 我们选择 4 个对协议性能影响较大的指标. 首先, 通信次数 (TC) 是一个不可忽视的因素, 因为变量的空中传输需要经过发送、传输和接收的过程, 需要很长时间. 然后是非对称加密算法 (AE) 和哈希散列算法 (HA). 文

献 [13] 已经证明, 在计算复杂度方面, 这两种算法远远超过其他算法. 最后是随机数的数量 (RN), 它的生成会消耗大量计算资源. 上述指标越小对协议性能越有利, 我们在表 1 中展示了比较结果.

综合来看, 无论是 GUTI 还是 SUPI 子协议, SM-AKA 在 TC (分别为 2, 2) 和 RN (分别为 1, 1) 两个指标上都具有绝对优势. 而指标 AE 和 HA 的优势不是直观的, 我们首先对比 AE 和 HA 的性能. 为了探索计算消耗, Patonico 等人[27]使用带有 ARM Cortex-M3 32 MHz 时钟, 并同时配备 512 KB 闪存和 32 KB 内存的 Zolertia REmote 传感器作为微控制器进行实验测试. 结果显示, 非对称加解密算法 AE 耗时 T_{AE}=342.39 ms, 而作为对比的哈希运算 (HA) 耗时 T_{HA}=0.03 ms. 它们之间的量级差异是巨大的, 一个 AE 的时间消耗大概是 HA 的 10^4 倍. 显然, 在 HA 差异不明显的情况下, 使用 AE 算法会使协议性能更差. 基于这个前提, 我们可以在必要时增加 HA, 但要减少 AE.

表 1 SM-AKA 与主流鉴权协议性能比较

协议	通信次数(TC)	非对称加密(AE)	哈希次数(HA)	随机数数量(RN)
5G-AKA[2]	3	是	5/8	2
Novel-AKA[15]	4	是	9	2
SEL-AKA[14]	3	是	7	2
AKA$^+$ (GUTI)[9]	4	否	10	1
AKA$^+$ (SUPI)[9]	4	是	8	1
SM-AKA (GUTI)	**2**	**否**	**9**	**1**
SM-AKA (SUPI)	**2**	**是**	**8**	**1**

表中除了 AKA$^+$外, 其余协议均用到了 AE 算法. 对于 SM-AKA, GUTI 子协议没有用到 AE, 因而其相较于这些协议具备性能优势. 而尽管 SUPI 子协议用到了 AE 算法, 但两个子协议交替运行的机制缓解了 AE 算法给 SM-AKA 性能带来的影响, 这是因为该机制降低了 AE 的平均使用次数. SM-AKA 的两个子协议中, SUPI 模式基于永久身份变量, 并使用 AE, 运行效率相对较低. 而 GUTI 模式选用临时身份变量, 以增加几个 HA 为代价取缔 AE, 大大提高了认证效率. 两种模式的选用取决于 V-GUTI 的值, 在执行过程中, 两种模式认证成功后都会把其置为 True, 以为下一次选择 GUTI 模式做铺垫. 只有在协议运行出现差错时, V-GUTI 的值才会为 False. 这种情况下, 下一次的认证会选择 SUPI 模式. 因此, 在协议运行不出现差错时, GUTI 模式总是会被选用, 而这在实际中是大多数情况. 因而, 综合 GUTI 和 SUPI 两个子协议来看, 每次鉴权 AE 的平均运行次数介于 0–1 之间, 且不会接近于 1. 因此, 在表中 HA 相差不明显的事实下, 我们可以断定 SM-AKA 协议的 HA 和 AE 指标总体优于 AKA$^+$之外的协议.

由于 AKA$^+$协议和 SM-AKA 具有相似的运行机制, 我们可以从子协议的角度评估二者的性能. 由表中数据可知, 对于 GUTI 子协议, SM-AKA 和 AKA$^+$的 AE 和 RN 相同, 而 SM-AKA 的 TC 和 HA 数量要较 AKA$^+$少. 对于 SUPI 子协议, 二者的 AE、HA 和 RN 相同, 而 SM-AKA TC 的数量要较 AKA$^+$少. 因而, 无论是 GUTI 还是 SUPI, SM-AKA 协议的性能都要优于 AKA$^+$. 综合来看, 在 4 个主要关系协议性能的指标上, SM-AKA 从整体上取得了提升, 相较于 5G-AKA 等主流协议, 具有更高的认证效率.

6 结 论

在这篇文章中, 我们针对当前 5G-AKA 及其变种的缺陷, 提出一个新颖的鉴权协议: SM-AKA. 它利用 SUPI 和 GUTI 两个子模式的交替有序运行实现协议的高效认证, 大大降低认证延迟. 同时, 两种模式下信息变量的组成也变得一致, 以规避连接性攻击. 通过形式化评估, 我们证明该协议不仅可以取得 UE 和 HN 之间的双向认证, 还可以抵御当前的已知攻击, 具有优秀的功能和安全特性. 我们还通过对 SM-AKA 的理论分析展示其设计的合理性.

在未来, 我们将改进目前的形式化验证工具, 致力于提升其可视化效果, 以避免黑盒式的自动证明. 同时, 我们也将在 UE、SN 和 HN 这 3 个实体, 甚至 5G 网元的基础上展开协议分析, 以期发现并修复更为深层的脆弱点.

References:

[1] Shayea I, Ergen M, Azmi M H, Çolak SA, Nordin R, Daradkeh YI. Key challenges, drivers and solutions for mobility management in 5G networks: A survey. IEEE Access, 2020, 8: 172534–172552. [doi: 10.1109/ACCESS.2020.3023802]

[2] 3GPP. Security architecture and procedures for 5G system. 3GPP TS 33.501 version 15.2.0 release 15. 3GPP, 2018.

[3] Chow MC, Ma M. A blockchain-enabled 5G authentication scheme against DoS attacks. Journal of Physics: Conf. Series, 2021, 1812(1): 012030. [doi: 10.1088/1742-6596/1812/1/012030]

[4] Haddad Z, Fouda MM, Mahmoud M, Abdallah M. Blockchain-based authentication for 5G networks. In: Proc. of the 2020 IEEE Int'l Conf. on Informatics, IoT, and Enabling Technologies. Doha: IEEE, 2020. 189–194. [doi: 10.1109/ICIoT48696.2020.9089507]

[5] Okumura N, Ogata K, Shinoda Y. Formal analysis of RFC 8120 authentication protocol for HTTP under different assumptions. Journal of Information Security and Applications, 2020, 53: 102529. [doi: 10.1016/j.jisa.2020.102529]

[6] Xiao MH, Li W, Zhong XM, Yang K, Chen J. Formal analysis and improvement on ultralightweight mutual authentication protocols of RFID. Chinese Journal of Electronics, 2019, 28(5): 1025–1032. [doi: 10.1049/cje.2019.06.022]

[7] Basin DA, Dreier J, Hirschi L, Radomirovic S, Sasse R, Stettler V. A formal analysis of 5G authentication. In: Proc. of the 2018 ACM SIGSAC Conf. on Computer and Communications Security. Toronto: ACM, 2018. 1383–1396. [doi: 10.1145/3243734.3243846]

[8] Cremers C, Dehnel-Wild M. Component-based formal analysis of 5G-AKA: Channel assumptions and session confusion. In: Proc. of the 26th Annual Network and Distributed System Security Symp. San Diego: NDSS, 2019. 1–15. [doi: 10.14722/ndss.2019.23394]

[9] Koutsos A. The 5G-AKA authentication protocol privacy. In: Proc. of the 2019 IEEE European Symp. on Security and Privacy. Stockholm: IEEE, 2019. 464–479. [doi: 10.1109/EuroSP.2019.00041]

[10] Arapinis M, Mancini L, Ritter E, Ryan M, Golde N, Redon K, Borgaonkar R. New privacy issues in mobile telephony: Fix and verification. In: Proc. of the 2012 ACM Conf. on Computer and Communications Security. Raleigh: ACM, 2012. 205–216. [doi: 10.1145/2382196.2382221]

[11] Borgaonkar R, Hirschi L, Park S, Shaik A. New privacy threat on 3G, 4G, and upcoming 5G AKA protocols. Proc. on Privacy Enhancing Technologies, 2019, 2019(3): 108–127. [doi: 10.2478/popets-2019-0039]

[12] Fouque PA, Onete C, Richard B. Achieving better privacy for the 3GPP AKA protocol. Proc. on Privacy Enhancing Technologies, 2016, 2016(4): 255–275. [doi: 10.1515/popets-2016-0039]

[13] Braeken A. Symmetric key based 5G AKA authentication protocol satisfying anonymity and unlinkability. Computer Networks, 2020, 181: 107424. [doi: 10.1016/j.comnet.2020.107424]

[14] Gharsallah I, Smaoui S, Zarai F. A secure efficient and lightweight authentication protocol for 5G cellular networks: SEL-AKA. In: Proc. of the 15th Int'l Wireless Communications & Mobile Computing Conf. Tangier: IWCMC, 2019. 1311–1316. [doi: 10.1109/IWCMC.2019.8766448]

[15] Braeken A, Liyanage M, Kumar P, Murphy J. Novel 5G authentication protocol to improve the resistance against active attacks and malicious serving networks. IEEE Access, 2019, 7: 64040–64052. [doi: 10.1109/ACCESS.2019.2914941]

[16] Han KH, Ma MD, Li XH, Feng ZY, Hao JY. An efficient handover authentication mechanism for 5G wireless network. In: Proc. of the 2019 IEEE Wireless Communications and Networking Conf. Marrakesh: IEEE, 2019. 1–8. [doi: 10.1109/WCNC.2019.8885915]

[17] Meier S, Schmidt B, Cremers C, Basin D. The TAMARIN prover for the symbolic analysis of security protocols. In: Sharygina N, Veith H, eds. Computer Aided Verification. Berlin: Springer, 2013. 696–701. [doi: 10.1007/978-3-642-39799-8_48]

[18] Edris EKK, Aiash M, Loo JKK. Formal verification and analysis of primary authentication based on 5G-AKA protocol. In: Proc. of the 7th Int'l Conf. on Software Defined Systems. Paris: IEEE, 2020. 256–261. [doi: 10.1109/SDS49854.2020.9143899]

[19] Blanchet B. Modeling and verifying security protocols with the applied Pi calculus and ProVerif. Foundations and Trends in Privacy and Security, 2016, 1(1–2): 1–135. [doi: 10.1561/3300000004]

[20] Armando A, Basin D, Boichut Y, Chevalier Y, Compagna L, Cuellar J, Drielsma PH, Heám PC, Kouchnarenko O, Mantovani J, Mödersheim S, von Oheimb D, Rusinowitch M, Santiago J, Turuani M, Viganò L, Vigneron L. The AVISPA tool for the automated validation of internet security protocols and applications. In: Etessami K, Rajamani SK, eds. Computer Aided Verification. Berlin: Springer, 2005. 281–285. [doi: 10.1007/11513988_27]

[21] Glouche Y, Genet T, Heen O, Courta O. A security protocol animator tool for AVISPA. 2006. https://www.researchgate.net/publication/228356197_A_security_protocol_animator_tool_for_AVISPA

[22] Rubin AD, Honeyman P. Nonmonotonic cryptographic protocols. In: Proc. of the 1994 Computer Security Foundations Workshop VII. Franconia: IEEE, 1994. 100–116. [doi: 10.1109/CSFW.1994.315943]

[23] Bana G, Comon-Lundh H. A computationally complete symbolic attacker for equivalence properties. In: Proc. of the 2014 ACM SIGSAC Conf. on Computer and Communications Security. Scottsdale: ACM, 2014. 609–620. [doi: 10.1145/2660267.2660276]

[24] Escobar S, Meadows C, Meseguer J. Maude-NPA: Cryptographic protocol analysis modulo equational properties. In: Aldini A, Barthe G, Gorrieri R, eds. Foundations of Security Analysis and Design V: FOSAD 2008/2009 Tutorial Lectures. Berlin: Springer, 2009. 1–50. [doi: 10.1007/978-3-642-03829-7_1]

[25] Cremers CJF. The scyther tool: Verification, falsification, and analysis of security protocols. In: Proc. of the 20th Int'l Conf. on Computer Aided Verification. Princeton: Springer, 2008. 414–418. [doi: 10.1007/978-3-540-70545-1_38]

[26] Dolev D, Yao A. On the security of public key protocols. IEEE Trans. on Information Theory, 1983, 29(2): 198–208. [doi: 10.1109/TIT.1983.1056650]

[27] Patonico S, Braeken A, Steenhaut K. Identity-based and anonymous key agreement protocol for fog computing resistant in the Canetti-Krawczyk security model. Wireless Networks, 2019. [doi: 10.1007/s11276-019-02084-6]

刘逸冰(1996－), 男, 博士生, 主要研究领域为 LTE 通信技术, 5G 通信网络安全, 机器学习.

周刚(1977－), 男, 博士, 教授, 博士生导师, 主要研究领域为移动通信, 大数据, 数据挖掘.