第 33 卷 第 1 期 计 算 机 工 程 2007 年 1 月
Vol.33 No.1 Computer Engineering January 2007
·博士论文· 文章编号：1000—3428(2007)01—0022—03 文献标识 中图分类号：TP311

COMP128 算法分析中关键问题研究

汪 涛，杨义先，孙 勇
(北京邮电大学网络与交换技术国家重点实验室，北京 100876)

摘 要：COMP128 算法是 GSM 协议采用的认证算法。该文分析了该算法攻击过程中的一个关键问题，在证明该问题为 NP 难题后，用贪婪算法给出了实用的较优解，这个结果比已知最好的攻击软件采用的值有所优化。

关键词：COMP128；密码分析；NP 难题；贪婪算法

Research on Key Problem in Cryptanalysis of COMP128

WANG Tao, YANG Yixian, SUN Yong
(State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

【Abstract】 One key problem in the cryptanalysis of GSM authentication algorithm COMP128 is identified and analyzed. After proving the NP-hardness of this problem, an approximate but practical solution is presented, which can improve the attack efficiency of the best attack software ever known.

【Key words】 COMP128; Cryptanalysis; NP-hardness; Greedy algorithm

1 概述

COMP128-1[1]算法是GSM协议中SIM卡和基站之间认证和密钥协商采用的算法。SIM卡和鉴权中心共享一个 16B密钥KI。SIM卡根据基站发起的 16B挑战与KI进行Hash运算，并将结果返回给基站进行认证和建立会话密钥。在这个挑战-应答的认证模式中，KI扮演了关键的作用。2000 年Goldberg等人在网上发布了COMP128-1 的攻击方法，通过向SIM卡发送一系列精心挑选的挑战并观察应答中是否有Hash碰撞，从而分析出KI。2002 年，Gemplus公司提出Strong-KI方案来抵抗 Goldberg 等人的攻击。其原理如下：16BKI 中的第(1,9),(2,10),(3,11),…,(8,16)这 8 对双字节数据有 256× 256 即 65 536 种可能的取值。其中有 769 对特殊的双字节永远不会产生Goldberg攻击中的碰撞。因此该子空间的双字节构成的KI可以有效地抵抗上述攻击算法，这就是Strong-Ki概念的来源。然而 2003 年，Kaljevic声称实现了对Strong-KI的攻击，并在软件SIM Scan[2]中实现。Kaljevic没有公开攻击的过程，我们通过端口监控软件Port Monitor分析该工具的攻击记录，完整再现了此攻击算法。

假设W为所有双字节数据的全集，其势为|W|=65 536。用 S={ $s_1,s_2,\ldots,s_{769}$}来表示这 769 个双字节子空间。Strong-Ki攻击算法利用了 2-R部分碰撞对(2-Round Partial Collision Pair, 2-RPCP)的概念：假设B=(b_1，b_2)∈W为一个双字节对，如果存在另外两个双字节对X=(x_1，x_2)∈W和Y=(y_1，y_2)∈W，使得B、X两对数据进行COMP128-1 前 2 小轮运算的 4B结果与B、Y两对数据同样计算后的结果中刚好有 3 个字节相同，那么X和Y构成B的一个 2-RPCP。任意一个s_i∈S都有 12 到 39 个 2-RPCP。例如S中的一个元素(00，0B)，第 1 个 2-RPCP 为 (03，D6)和(8E，40)，第 2 个 2-RPCP为(06,22)和(4D,89),…它一共有 26 个 2-RPCP。攻击步骤如下：

(1)对每个s_i∈S，计其所有 2-RPCP集合为 2-RPCP$_i$= {(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}), …, (X_{ij}, Y_{ij})}，此为该Strong-KI的碰撞表。注意其中X_{ik}和Y_{ik}都是双字节，即X_{ik}∈W，Y_{ik}∈W，且X_{ik}和Y_{ik}共同构成s_i的第k个 2-RPCP。在PC上只需要几分钟就可构造出所有Strong-KI的碰撞表。

(2)假设存在W的一个子集C，使得对每一个s_i∈S，都存在至少一个 2-RPCP (X_i, Y_i)，使得X_i ∈C且Y_i ∈C，称C 盖S。

(3)随机选择 14B。保持该 14B 固定，以 C 中的每个元素作为挑战的第 1、第 9 字节，构成一组共|C|个挑战发给 SIM 卡，分别记录其应答；再生成 14B，类似构成新的一组共|C|个挑战发给 SIM 卡，分别记录其应答。如此循环，每一组都有 1/256 的机率出现组内碰撞。一旦检测到组内碰撞，可反查碰撞表，推算出 Strong-KI。

由上述分析可见，|C|是影响攻击效率的关键因素。如何构造势最小且 盖S的C，是提高攻击效率的关键。最直接的方法是对每个s_i，随机从 2-RPCP$_i$选一对，然后将所有结果合并在一起，这样得到的|C|大约在 2× 769 个。但是注意到不同s_i的碰撞对之间有部分重复，故需要寻找一种算法来优化C，使得|C|尽可能小。

2 问题描述

为了直观，用图来重新描述这个问题。用 65 536 个顶点代表W，以 769 种颜色代表S。对s_i的 2-RPCP集合{(X_{i1}, Y_{i1}), (X_{i2}, Y_{i2}), …, (X_{ij}, Y_{ij})}，其中X_{ik}∈W且Y_{ik}∈W,为两个顶点，用一条s_i染色的边连接。

输入 无向图 G 由 65 536 个点和一些边组成，不存在顶点到自身的边，不同顶点间允许重边。现有 769 种颜色对 G 的每条边染色。每种颜色都存在至少一条边。

输出 G 的包含了全部 769 种颜色的子图 Sub(G)，要求

基金项目：国家自然科学基金资助项目(60372094)
作者简介：汪 涛(1978－)，男，博士生，主研方向：信息与网络安全；杨义先，教授、博导；孙 勇，博士生
收稿日期：2006-03-10 **E-mail**：9bit@163.com

其顶点数尽可能小。

如图 1 所示，假设 G 有 4 种颜色，7 个顶点，满足条件的最小子图为虚线框中的 4 个节点和 4 条边。

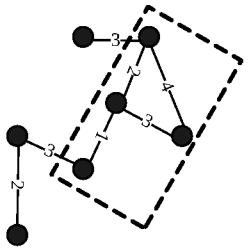

图 1 问题图形化描述

图论中有很多类似的寻找最优解问题，如货郎担货问题(Traveling Salesman Problem, TSP)，最小支撑树问题，最短路问题，中国邮递员问题等。我们的问题涉及 盖的概念，与这些问题并不一致，无法直接采用这些问题的解决方法。经过分析，发现该问题和著名的集合 盖问题[3]相似。(m, n) 集合 盖问题描述如下：

输入 全集U={1,2,…,n}的子集的集合S={s_1, s_2,…,s_m}。

输出 S的子集T={t_1, t_2,…,t_k}，使 $\cup_i^k t_i = U$ ，且k最小。

图 2 中m=5, n=9，U为 9 个顶点，S={s_1, s_2, s_3, s_4, s_5}，可以用 3 个子集s_1、s_2、s_5完全 盖U。当|s_i|都不超过 2 时，集合 盖问题有多项式时间内最优算法，否则为NP难题。我们的问题也是NP难题。

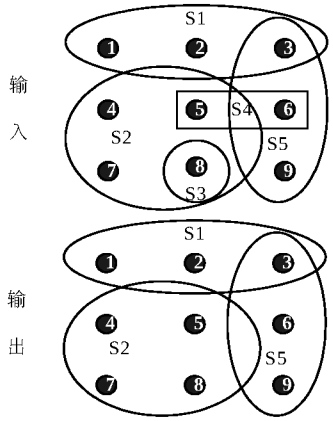

图 2 集合覆盖问题输入输出

证明 对于(m, n) 集合 盖问题，构造一个图形G，含有一个特殊顶点O。然后创建m个顶点分别对应S的m个子集s_1, s_2,…,s_m。顶点s_i和O之间有|s_i|条重边，分别根据s_i的元素对其染色。总共需要n种颜色。图 3 即图 2 中集合 盖问题的图形化描述。

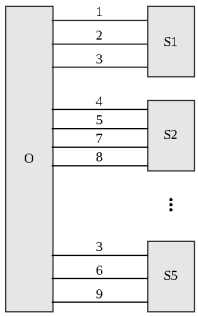

图 3 集合覆盖问题和本问题等效证明

需要在这幅图中寻找一个子图，其包含了所有颜色。显然这样一个子图必定包括O点，且只要存在这样一个子图，就对应了集合 盖问题的一个解。点数最少的子图对应集合 盖问题的最优解。由此证明了我们的问题至少和集合 盖问题一样难，即为NP难题。对于NP难题，一般有 3 种分析方法[3]：(1)配合适当的搜索和剪枝策略，穷举所有可能的组合，寻找最优解。(2)将问题划分为多个满足无后效性(马尔科夫性)的步骤，采用动态规划[4]寻找最优解。(3)采用近似算法得到可接受的较优解。

目前所有已知的解决NP难题的最优算法都随着问题规模的增长而呈指数增长。当问题规模不太大时，可以采用方法 1 或方法 2。如果问题规模较大，则只能采用方法 3。图G有 769 种颜色，共有 19 154 条边。如果每种颜色选择一条边(以及该边的两个顶点)来构成子树，可能的组合超过 12^{769}= 7×10^{829}。直接采用方法 1 或方法 2 都是不现实的。

3 问题分析

先尝试缩小问题的规模。G 中存在 5 774 条孤立边，这些边与 G 的其他部分不是连通的。选择某条孤立边就相当于该颜色占用了 2 个子图顶点数，这是任意一种颜色消耗顶点数的最大值，即对任意一种颜色而言，孤立边一定不是最优的选择。除去这些孤立边后依然能得到最优解。处理后总边数由 19 154 减少到 13 380。下面的讨论都基于此裁剪过的图，记为 G'=(V,E)。

G'的规模依然很大，不适合用穷举或动态规划算法来解决。对于此类问题，实践中大多采用近似最优算法，例如贪婪算法、模拟退火、遗传算法、人工神经网络等。

问题的目标是找到顶点数最小同时又包含了各种颜色子图。这启发我们尽量利用那些度数(与某顶点相连的边数即位该顶点的度数)最高的顶点，因为这些顶点连接了多种颜色。统计 G'中所有顶点的度数，记为 d(v)(v∈V)。算法描述如下：

(1)选边策略：最终生成的子图需要从各种颜色的边中至少选择一条，同时尽可能地减少顶点总数。假设一条边e连接的顶点为u, v，那么记f(e)=d(u)+d(v)为该边的评估值。对所有e∈E，计算f(e)并取值最大的边E_{max}作为此轮首选，记该边的颜色为已 盖。

(2)选择E_{max}的同时也选择了该边的两个顶点。由于f(e)是基于利用多边共用一个顶点的原理来减少最终顶点的总数，因此为了让E_{max}符合f(e)，必须同时选择E_{max}两个顶点相连的所有边。将上述所有选择的边和顶点合并到结果子图中，然后继续新一轮的统计和选边，直到最终子图 盖各种颜色。

(3)同一时刻可能有多条e∈E，使得f(e)=f(E_{max})，可随机选择一个E_{max}。运行随机算法多次后，取其中的最优值。这是随机算法的思想，在模拟退火、遗传算法等近似算法中经常采用。

该算法在 Pentium 4，2.4GHzPC 上运行 1 000 次约 10min，运行最佳结果为一个 986 顶点的子图。

4 算法改进

上述算法仅仅是估计算法，对 G 没有直观的认识，且边数和顶点数过多，编程作图显示出来的效果也不太满意。对于 G 的一些关键属性，比如是否有环？是否有重边？也还没有结论。下面将考察 G 的这些属性，并尝试通过对 G 的进一步了解来改进算法。

图形的环检测(Cycle Detection)在很多领域都有应用，如在大型系统架构，程序编译，工作安排等实际应用中，当一

个系统的多个模块间存在复杂的依赖关系时，需要尽可能检测排除环形的依赖，否则可能会产生不良后果。考察是否有环和重边的方法是利用深度优先搜索(Depth First Search, DFS)遍历图形，如果遍历过程中遇到一个已经遍历过的边，此图必存在环或重边。

利用Boost C++库来实现环检测。Boost是一个准标准库，相当于STL的延续和扩充。Boost的图形库[6]包含了大量的图论算法的实现，如：DFS，Dijkstra最短路算法，Prim最小支撑树算法等。我们参考了Boost示范程序中提供的一个检查有机分子中是否有原子环的实现，利用其无向图DFS包(undirected_dfs)实现深度优先算法。实现的关键是提供一个back_edge回调方法。Boost遍历时每遇到一个已经遍历过的边就会调用我们的back_edge方法来打印出边的信息。最终检测结果是：G'中不存在环和重边。即G'是由多棵树组成的森林。显然，G'的子图Sub(G')也是由多棵树组成的森林。

根据 Euler 公式，树的顶点数=边数+1。假设 Sub(G')中一棵树包含了 n 条边，那么有 n+1 个顶点。由于该 n 条边颜色不一定各不相同，因此一棵树要 盖 n 种颜色，至少需要 n+1 个顶点。假设 Sub(G')有 k 棵树，那么其顶点数>颜色数+k。可见，在颜色数一定的情况下，需要尽可能地减少 k 的大小，即尽可能地优先选择 G'中的最大树来构成 Sub(G')。对任意一棵树 T，记其包含的不同颜色总数为 C，顶点数为 V，那么记 f(T)=(V-C)/C 为该树的消耗评估函数。改良后的算法如下：

(1)选边策略：G'是由多棵树构成的森林。计算每棵树的消耗函数 f，把 f 最小的树的所有顶点和边加入到结果子图中。

(2)更新算法：每当选出一棵树添加到结果子图后，该树盖的颜色相应地被添加到子图中。后续的挑选策略不该再考虑这些颜色，需要将剩余图形中这些颜色染色的边删除，更新剩余图形后再进行选边，直到 Sub(G') 盖了所有颜色。

由于更新算法的缘故，每一次选边都会对后续的选择有影响，即算法依然是局部优化的贪婪算法，不能保证总体最优解。上述算法运行后结果为 938。这个结果已经比Kaljevic[1]使用的 946 有所优化。由于Strong-Ki分析过程中平均需要约 128 组，每一组都要用到子图中的所有点，因此该结果共节省约 128× 8 即 1 024 次鉴权。

5 结论

在对 COMP128-1 算法分析的过程中，本文研究了一个影响攻击效率的关键问题。通过图形化的描述，并结合 Boost 图形库，利用贪婪算法得到一个近似最优解。这个结果比已知最好的攻击软件采用的值有所优化。

参考文献

1 A3A8[Z]. http://www.gsm-security.net/papers/a3a8.shtml.

2 SimScan[Z]. http://users.net.yu/~dejan/.

3 Skiena S S. The Algorithm Design Manual[M]. Springer-Verlag, 1997.

4 Bellman R E. Dynamic Programming[M]. Princeton University Press, 1957.

5 Lee Liequan, Siek J G. Andrew Lumsdain[M]. Boost Graph Library, 2000.

（上接第 6 页）

UDP 源端产生的业务量都是服从指数分布的 on-off 源，on 和 off 的平均持续时间分别为 200ms 和 400ms，on 时传输的速率为 30KBps。图 5 是几种算法下的队列长度，可见 RED 在非响应流的作用下更容易使队列值到达 0 值，从而导致更低的链路利用率；非响应流对于 PI 算法的性能影响不大；虽然 Dahlin 算法在其作用下响应时间有所增加，但是仍然优于 PI 算法。

6 结论

主动队列管理是近几年来TCP端到端拥塞控制中研究的一个热点，但已提出的机制大都没有考虑大 RTT 对系统性能的影响。本文基于控制理论中的 Dahlin 算法，给出了一种针对较大 RTT 的 AQM 算法，该算法可以克服由于大的 RTT 对于系统性能的负面影响。然后分析了该算法稳定性和参数选择对系统性能的影响。仿真实验结果表明，基于 Dahlin 算法的 AQM 算法在 RTT 较大的情况下，可以较快 f 使得队列长度收敛到给定值，表现出比较好的瞬态性能和稳态性能；对于不同 RTT 和非响应流的鲁棒性也明显优于 RED 和 PI 算法。因此在大 RTT 网络环境中，本算法可以保证网络工作在较低的分组丢弃率下，同时达到高链路利用率和比较低的延迟抖动。

参考文献

1 Braden B. Recommendations on Queue Management and Congestion Avoidance in the Internet[S]. IETF, RFC2309, 1998.

2 Floyd S, Jacobson V. Random Early Detection Gateways for Congestion Avoidance[J]. IEEE /ACM Transactions on Networking, 1993, 1(4): 397-413.

3 Athuraliya S, Low S H. Optimization Flow Control, II: Random Exponential Marking[EB/OL]. 2000. http://netlab.caltech.edu/pub.html.

4 Kunniyur S, Srikant R. Analysis and Design of an Adaptive Virtual Queue (AVQ) Algorithm for Active Queue Management[J]. ACM Computer Communication Review, 2001, 31(4): 123-134.

5 Misra V, Gong W B, Towsley D. Fluid-based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED[C]// Proceedings of the ACM SIGCOMM Conference,. Stockholm, Sweden: 2000: 151-160.

6 Hollot C V. A Control Theoretic Analysis of RED[C]// Proceedings of INFOCOM Conference, Tel Aviv, Israel, 2000: 1510-1519.

7 Hollot C V, Misra V, Towsley D, et al. On Designing Improved Controllers for AQM Routers Supporting TCP Flows[C]// Proceedings of IEEE INFOCOM, Anchorage, Alaska, USA, 2001: 1726-1734.

8 任丰原，林 闯，刘卫东. IP 网络中的拥塞控制[J]. 计算机学报, 2003, 26(9): 1025-1034.

9 Dahlin E B. Design and Tuning Digital Controllers[J]. Instrument Control Systems, 1968: 41(6): 77-83.

10 Dong Yu, Zhu Xuefeng. Root Locus Analysis of Dahlin Controller[C]// Proceedings of the IEEE International Symposium on Intelligent Control, Hangzhou, China: 2004: 517-522.

11 UCN/ LBL/ VINT. Network Simulator NS2[DB/OL]. http://www-mash. cs.berkeley.edu/ns.

12 何克忠，李 伟. 计算机控制系统[M]. 北京：清华大学出版社, 1998.