MORUS 算法的抗碰撞性分析

关 杰 施泰荣* 李俊志 张 沛
(解放军信息工程大学 郑州 450001)

摘 要：MORUS 算法是 CAESAR 竞赛第 3 轮的候选认证加密算法之一，该文评估了 MORUS-640-128 算法对碰撞攻击的安全性。由碰撞关系确定一系列非线性方程，采用分块分析的方法，从非线性方程中找到消息字差分间的信息泄漏规律，首次给出了算法在两步后发生碰撞的必要条件集，确定了输入差分的字分布情况。在此基础上，将碰撞的必要条件转化成伪布尔函数最优化问题，利用混合整数规划模型进行求解。实验结果显示算法发生碰撞时，输入差的汉明重量至少为 28，其碰撞概率小于 2^{-140}，得到了比文献[7]更紧致的概率上界(原为 2^{-130})。结果表明 MORUS-640-128 算法具备良好的抗碰撞攻击能力。

关键词：CAESAR 竞赛；MORUS 算法；碰撞攻击；差分重量下界
中图分类号： TN918.1 **文献标识码**：A **文章编号**：1009-5896(2017)07-1704-07
DOI: 10.11999/JEIT161185

Analysis of MORUS Against Collision Attack

GUAN Jie SHI Tairong LI Junzhi ZHANG Pei
(*The PLA Information Engineering University*, *Zhengzhou* 450001, *China*)

Abstract: MORUS is an authenticated stream cipher, which is selected is third-round candidate of the ongoing CAESAR competition. In this work, the security of MORUS-640-128 against collision attack is evaluated. The partition method is utilized to find the information leakage between the word differences of message in the nonlinear function determined by the collision. The necessary conditions of collision after two steps are proposed for the first time. The distribution of input difference is determined. Furthermore, necessary conditions are turned into Pseudo-Boolean optimization problems. With the usage of mixed integer programming, it is found that the weight of message difference must be higher than 28 with the collision probability less than 2^{-140}, which is a better upper bound than ref. [7] 2^{-130}. The result shows that MORUS-640-128 has a good performance on resistance against collision attack.

Key words: CAESAR competition; MORUS; Collision attack; Lower bound of difference weight

1 引言

认证加密算法[1]指能同时实现数据加密和真实性认证功能的算法，已广泛应用于各种网络安全系统中。随着移动互联网的快速发展和大数据时代的来临，人们迫切需要寻找到更安全、高效实用的认证加密算法。2013 年在美国标准技术研究所(NIST)专门资助下，国际密码协会(IACR)发起了 CAESAR 竞赛[2]。竞赛共征集到 57 个算法，经过 3 轮评选，截止目前有 15 个候选算法进入第 3 轮评审，最终的优胜算法将于 2017 年 12 月公布。随着 CAESAR 竞赛的推动，对认证加密算法的安全性分析成为密码学界的研究热点。进入第 3 轮评选的候选算法无论是在安全性还是在实现性能上都有着出色的表现，吸引了众多学者的关注[3–6]。

MORUS 算法是由 Wu 等人[7]设计的一类基于流密码的认证加密算法，凭借其独特的设计思路和优越的性能，已顺利进入 CAESAR 竞赛第 3 轮评选。根据内部状态和密钥长度的不同，设计者推荐了 3 个 MORUS 子算法：MORUS-640-128, MORUS-1280-128 和 MORUS-1280-256。针对 MORUS 算法的安全性分析，文献[8]提出了在 Nonce 重用条件下的区分攻击；文献[9]对算法初始化阶段的混乱与扩散性质进行了研究。

碰撞攻击[10–12](collision attack)是针对消息认证码的常用攻击手段，其基本方法在关联数据处理(加密)阶段的某一步由关联数据(明文)引入合适的差分，若干轮迭代后差分抵消。内部状态碰撞意味着生成的认证标签相同，从而达到伪造认证标签的

收稿日期：2016-11-03；改回日期：2017-03-06；网络出版：2017-04-25
*通信作者：施泰荣 strwanzi@163.com
基金项目：国家自然科学基金(61572516, 61272041, 61272488)
Foundation Items: The National Natural Science Foundation of China (61572516, 61272041, 61272488)

目的。为了评估 MORUS-640-128 算法抵抗伪造攻击的能力，设计者假设算法在更新 2 步、3 步或 3 步以上发生碰撞，分析了上述 3 种情况下碰撞发生的概率。其中，当算法在 2 步更新后发生碰撞时，设计者通过穷举搜索寻找可能的输入差分，结果显示满足条件的输入差分重量至少为 26 bit，经过分析此时的碰撞概率小于 2^{-128}。当算法更新 3 步或 3 步以上时，用类似的方法得到碰撞概率同样小于 2^{-128}，因此可以认为 128 bit 认证标签是安全的。

本文对 MORUS-640-128 算法的认证阶段进行了安全性评估。首先列出更新两步后内部状态发生碰撞需要满足的非线性方程，采用分块分析的方法从非线性方程中找到确定的信息泄漏规律，给出两步更新后发生碰撞的必要条件集。在此基础上证明了若两步更新后内部状态发生碰撞，输入差分必须分布在 4 个 32 bit 上。最后将必要条件转化为伪布尔最优化问题[13]，利用混合整数规划模型[14]以及 SCIP 求解器[15]求解得到两步发生碰撞输入差分重量的下界为 28 bit，碰撞概率小于 2^{-140}。结果表明 MORUS 算法抗碰撞攻击。

本文的组织结构如下：第 2 节中给出了 MORUS 的描述；第 3 节介绍了本文需要的相关知识和内部状态差的表示；第 4 节给出两步更新发生碰撞的必要条件集；第 5 节给出碰撞关于字差分的必要条件；第 6 节给出发生碰撞输入差重量的下界；第 7 节总结全文。

2　MORUS 算法简介

MORUS-640-128 算法的密钥规模为 128 bit，初始向量规模为 128 bit，认证码的长度小于或等于 128 bit。在本文中，我们仅对 MORUS-640-128 算法的关联数据处理阶段与加密阶段的安全性进行分析。下面首先进行符号说明，再给出 MORUS 描述。若无特别声明，本文中“MORUS”均指 MORUS-640-128 算法。

2.1 符号说明

S^i 为第 i 步的内部状态；S_k^i 为第 i 步更新第 k 轮的内部状态，$S_k^i=(S_{k,0}^i,S_{k,1}^i,S_{k,2}^i,S_{k,3}^i,S_{k,4}^i)$，$0\le k\le 4$；$S_{k,l}^i$ 为第 i 步第 k 轮内部状态的第 l 块 128 bit 分组，$S_{k,j}^i=(S_{k,j,1}^i,S_{k,j,2}^i,S_{k,j,3}^i,S_{k,j,4}^i),0\le j\le 4$；$S_{k,l,j}^i$ 为第 i 步第 k 轮内部状态第 l 分组的第 j 个 32 bit，$1\le l\le 4$，$0\le j\le 4$；0^n 为 n 长全 0 比特串；1^n 为 n 长全 1 比特串；$\lceil x\rceil$ 为不小于 x 的最小的整数；& 为按位与运算；$<<<$ 为循环左移；$>>>$ 为循环右移；$\mathrm{Rotl}(x,n)$ 为将 128 bit 长的 x 分成 4 块 32 bit，每块循环左移 n bit；$\mathrm{Rotl}^{-1}(x,n)$ 为和 $\mathrm{Rotl}(x,n)$ 相对，将 128 bit 长的 x 分成 4 块 32 bit，每块循环右移 n bit；$\mathrm{W}(\alpha)$ 为 α 的汉明重量，即 α 二进制表示中比特 1 的个数；$\mathrm{WB}(\Delta m_t)$ 为消息差分 $\Delta m_t=(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$ 中非零字的个数，其中 $\alpha_i(1\le i\le 4)$ 表示 32 bit，称 $\mathrm{WB}(\Delta m_t)$ 为字差分的重量。

2.2 MORUS 算法状态更新函数

在 MORUS 算法的每步更新中，都有 5 轮相似运算来更新内部状态 S，其中消息分组 m_i 用于更新第 2 轮至第 5 轮，但没有更新第 1 轮。下面给出 MORUS 算法的状态更新函数 $S^{i+1}=\mathrm{StateUpdate}(S^i,m_i)$：

第 1 轮：

$$S_{1,0}^i=\mathrm{Rotl}\left(S_{0,0}^i\oplus\left(S_{0,1}^i\&S_{0,2}^i\right)\oplus S_{0,3}^i,b_0\right)$$
$$S_{1,3}^i=S_{0,3}^i<<<w_0$$
$$S_{1,1}^i=S_{0,1}^i$$
$$S_{1,2}^i=S_{0,2}^i$$
$$S_{1,4}^i=S_{0,4}^i$$

第 2 轮至第 5 轮：For $k=1$ to 4，

$$S_{(k+1)\bmod 5,k}^i=\mathrm{Rotl}\left(S_{k,k}^i\oplus\left(S_{k,(k+1)\bmod 5}^i\ \&\ S_{k,(k+2)\bmod 5}^i\right)\oplus S_{k,(k+3)\bmod 5}^i\oplus m_i,b_k\right)$$
$$S_{(k+1)\bmod 5,(k+3)\bmod 5}^i=S_{k,(k+3)\bmod 5}^i<<<w_k$$
$$S_{(k+1)\bmod 5,(k+1)\bmod 5}^i=S_{k,(k+1)\bmod 5}^i$$
$$S_{(k+1)\bmod 5,(k+2)\bmod 5}^i=S_{k,(k+2)\bmod 5}^i$$
$$S_{(k+1)\bmod 5,(k+4)\bmod 5}^i=S_{k,(k+4)\bmod 5}^i$$

S^{i+1} 生成：For $k=0$ to 4，$S_{0,k}^{i+1}=S_{0,k}^i$。

函数中使用的移位常数如表 1 所示。

表 1　MORUS 算法中的移位常数

Rotl 移位常数	循环左移常数
$b_0=5$	$w_0=32$
$b_1=31$	$w_1=64$
$b_2=7$	$w_2=96$
$b_3=22$	$w_3=64$
$b_4=13$	$w_4=32$

2.3 MORUS 算法的关联数据处理阶段

设关联数据的长度为 adlen，令 $u=\left\lceil \mathrm{adlen}/128\right\rceil$，按照如下步骤处理关联数据：

For I=0 to $u-1$，S^i=StateUpdate(S^i, AD_i^{128})。

2.4 MORUS 算法的加密阶段

在算法的加密阶段，16 Byte 的明文分组 P_i 用于更新内部状态，P_i 与密钥流进行异或运算得到密文 C_i。设明文消息的长度为 msglen，令 $v=\lceil \mathrm{msglen}/128\rceil$，进行加密及状态更新操作如下：

For $i=0$ to $v-1$,

$C_i = P_i \oplus S_0^{u+i} \oplus (S_1^{u+i} <<< 96) \oplus (S_2^{u+i} \,\&\, S_3^{u+i})$;
$S^{u+i+1} = \mathrm{StateUpdate}(S^{u+i}, P_i)$。

3 准备工作

3.1 相关知识

MORUS 算法的碰撞时，内部状态与消息差分可以表示为一系列非线性方程，为从非线性方程中找到消息字差分间的信息泄漏规律，我们需要介绍一个新的定义以及一些性质。

定义 1 设 $A^1, A^2, \cdots, A^m, B \in \{0,1\}^n$，若存在 $X^1, X^2, \cdots, X^m \in \{0,1\}^n$，使得等式

$$\left(X^1 \,\&\, A^1\right) \oplus \left(X^2 \,\&\, A^2\right) \oplus \cdots \oplus \left(X^m \,\&\, A^m\right) \oplus B = 0$$

成立，称 A^i 和 B 满足关系 $B \rhd (A^1 * A^2 * \cdots * A^m)$。

这里“$*$”只表示各 A^i 之间的并列连接，关系“$\rhd$”有以下定理和性质成立。

定理 1 设 $A^1, A^2, \cdots, A^m, B \in \{0,1\}^n, A_j^i, B_j$ 分别是 A^i, B 的第 j 比特，$1 \le i \le m$，$1 \le j \le n$。若 $B \rhd (A^1 * A^2 * \cdots * A^m)$，则当 $B_j = 1$ 时，至少存在一个 A^i，满足 $A_j^i = 1$。

证明 假设对所有的 $A^i, 1 \le i \le m$，$A_j^i = 0$。由 $B \rhd (A^1 * A^2 * \cdots * A^m)$ 的定义可知，存在 $X^1, X^2, \cdots, X^m \in \{0,1\}^n$ 使得等式 $(X^1 \,\&\, A^1) \oplus (X^2 \,\&\, A^2) \oplus \cdots \oplus (X^m \,\&\, A^m) \oplus B = 0$ 成立。将 $A_j^i = 0$ 代入等式得 $B_j = 0$，与 $B_j = 1$ 矛盾，因此假设不成立，至少存在一个 A^i，满足 $A_j^i = 1$。 证毕

定理 2 设 $A^1, A^2, \cdots, A^m, B \in \{0,1\}^n, A_j^i, B_j$ 分别是 A^i, B 的第 j 比特 $1 \le i \le m$，$1 \le j \le n$。若存在 $j, 1 \le j \le n$ 使得 $B_j = 1$，并且对所有 $A^i, 1 \le i \le m$ 均有 $A_j^i = 0$，则 $B \rhd (A^1 * A^2 * \cdots * A^m)$ 不成立。

性质 1 对任意的 $A, B \in \{0,1\}^n$，均有 $A \,\&\, B \rhd (A * B)$ 成立。

证明 对任意的 A, B，取 $X^1 = B, X^2 = 0$，则 $A \,\&\, B = (X^1 \,\&\, A) \oplus (X^2 \,\&\, B) = (B \,\&\, A) \oplus (0 \,\&\, B)$ 成立，按照定义 1，得到 $A \,\&\, B \rhd (A * B)$。 证毕

性质 2 对任意的 $A, B, C \in \{0,1\}^n$，$(A \,\&\, B) \oplus C \rhd (A * B)$ 等价于 $C \rhd (A * B)$。

证明 必要性：由 $(A \,\&\, B) \oplus C \rhd (A * B)$ 可知，存在 $Y^1, Y^2 \in \{0,1\}^n$，使得等式 $(A \,\&\, B) \oplus C = (Y^1 \,\&\, A) \oplus (Y^2 \,\&\, B)$ 成立。将 $A \,\&\, B$ 移至等式右边得到 $C = (Y^1 \,\&\, A) \oplus (Y^2 \,\&\, B) \oplus (A \,\&\, B)$。令 $X^1 = Y^1, X^2 = Y^2 \oplus A$ 得到 $C = (X^1 \,\&\, A) \oplus (X^2 \,\&\, B)$，根据定义 1 可知 $C \rhd (A * B)$。

充分性：由 $C \rhd (A * B)$ 可知，存在 $Z^1, Z^2 \in \{0,1\}^n$ 使得 $C = Z^1 \,\&\, A \oplus Z^2 \,\&\, B$。由性质 1 可知 $A \,\&\, B \rhd (A * B)$，即存在 $Y^1, Y^2 \in \{0,1\}^n$ 使得 $A \,\&\, B = (Y^1 \,\&\, A) \oplus (Y^2 \,\&\, B)$。将 $C = (Z^1 \,\&\, A) \oplus (Z^2 \,\&\, B)$ 与 $A \,\&\, B = (Y^1 \,\&\, A) \oplus (Y^2 \,\&\, B)$ 等式两边分别进行异或，得到

$$C \oplus A \,\&\, B = (Z^1 \,\&\, A) \oplus (Z^2 \,\&\, B) \oplus (Y^1 \,\&\, A) \oplus (Y^2 \,\&\, B)$$

令 $X^1 = Z^1 \oplus Y^1, X^2 = Z^2 \oplus Y^2$，再根据定义 1 得到 $(A \,\&\, B) \oplus C \rhd (A * B)$。 证毕

性质 3 (传递性) 对任意的 $A, B, C \in \{0,1\}^n$，若 $A \rhd (B * C)$，$B \rhd C$，则 $A \rhd C$。

证明 由 $A \rhd (B * C)$，$B \rhd C$ 得到存在 $Y^1, Y^2, Z^1 \in \{0,1\}^n$，使得 $A = (Y^1 \,\&\, B) \oplus (Y^2 \,\&\, C)$，$B = Z^1 \,\&\, C$，将 $B = Z^1 \,\&\, C$ 代入 $A = (Y^1 \,\&\, B) \oplus (Y^2 \,\&\, C)$，得到 $A = (Y^1 \,\&\, Z^1 \,\&\, C) \oplus (Y^2 \,\&\, C) = (Y^1 \,\&\, Z^1 \oplus Y^2) \,\&\, C$，令 $X^1 = Y^1 \,\&\, Z^1 \oplus Y^2$，则 $A = X^1 \,\&\, C$，根据定义 1 得到 $A \rhd C$。 证毕

3.2 内部状态差表示

假设在 MORUS 算法关联数据处理或加密阶段的第 t 时刻引入消息差分 $\Delta m_t (\Delta m_t \ne 0)$，此时内部状态差 $\Delta S^t = 0$。根据对 MORUS 状态更新函数的差分分析，容易得到第 t+1 时刻内部状态各分组的差分表示。

$$\Delta S_{0,0}^{t+1} = 0$$

$$\Delta S_{0,1}^{t+1} = \mathrm{Rotl}(\Delta m_t, 31) <<< 64$$

$$\Delta S_{0,2}^{t+1} = \mathrm{Rotl}(\Delta m_t, 7) <<< 32$$

$$\Delta S_{0,3}^{t+1} = \mathrm{Rotl}(\mathrm{Rotl}(\Delta m_t, 31) \oplus \Delta m_t, 22) <<< 32$$

$$\Delta S_{0,4}^{t+1} = \mathrm{Rotl}(S_{0,0}^{t+1} \,\&\, \Delta S_{0,1}^{t+1} \oplus (\Delta S_{0,2}^{t+1} >>> 32) \oplus \Delta m_t, 13) <<< 64$$

注：为了下文表述方便，这里提前对 ΔS_3^{t+1} 和 ΔS_4^{t+1} 进行了移位操作。

由 ΔS^{t+1} 我们给出第 t+2 时刻的差分 ΔS^{t+2}，内部状态发生碰撞当且仅当 $\Delta S_i^{t+2} = 0 (0 \le i \le 4)$。由于我们只关注第 2 步更新后各状态分组的差分值是否为 0，更新函数最外层的 $\mathrm{Rotl}(x, n)$ 移位变换并不影响差分值，因此我们省去 $\mathrm{Rotl}(x, n)$ 运算，得到两步更新后各状态分组的输出差如下：

$$\Delta S_{0,0}^{t+2} = \left(S_{0,1}^{t+1} \,\&\, \Delta S_{0,2}^{t+1}\right) \oplus \left(\Delta S_{0,1}^{t+1} \,\&\, S_{0,2}^{t+1}\right) \oplus \left(\Delta S_{0,1}^{t+1} \,\&\, \Delta S_{0,2}^{t+1}\right) \oplus \left(\Delta S_{0,3}^{t+1} >>> 32\right)$$

$$\Delta S_{0,1}^{t+2} = \Delta S_{0,1}^{t+1} \oplus \left(S_{0,2}^{t+1} \,\&\, \Delta S_{0,3}^{t+1}\right) \oplus \left(\Delta S_{0,2}^{t+1} \,\&\, S_{0,3}^{t+1}\right) \oplus \left(\Delta S_{0,2}^{t+1} \,\&\, \Delta S_{0,3}^{t+1}\right) \oplus \left(\Delta S_{0,4}^{t+1} >>> 64\right) \oplus \Delta m_{t+1}$$

$$\Delta S_{0,2}^{t+2} = \Delta S_{0,2}^{t+1} \oplus \left(S_{0,3}^{t+1} \,\&\, \Delta S_{0,4}^{t+1}\right) \oplus \left(\Delta S_{0,3}^{t+1} \,\&\, S_{0,4}^{t+1}\right) \oplus \left(\Delta S_{0,3}^{t+1} \,\&\, \Delta S_{0,4}^{t+1}\right) \oplus \Delta m_{t+1}$$

$$\Delta S_{0,3}^{t+2} = \Delta S_{0,3}^{t+1} \oplus \left(\Delta S_{0,4}^{t+1} \,\&\, S_{0,0}^{t+2}\right) \oplus \Delta m_{t+1}$$

$$\Delta S_{0,4}^{t+2} = \Delta S_{0,4}^{t+1} \oplus \Delta m_{t+1}$$

需要说明的是，这里我们在 $\Delta S^{t+2}=0$ 已经满足的前提下，由第 1 步的输出差表示出第 2 步更新后各状态分组的差分形式，以分析 Δm_t 需满足的条件。

4 MORUS 算法两步更新发生碰撞的必要条件集

通过观察 MORUS 算法的状态更新函数，发现各步运算均以 32 bit 为单位进行，且字间移位数均为 32 的整数倍，故我们采用分块的思想，将每个状态分组分成 4 个 32 bit，以字为单位来分析。根据 3.2 节中的内部状态差给出引入消息差分 Δm_t 后的内部状态值的具体表示，列出产生碰撞需满足的非线性方程。

令 $\Delta m_t=(\alpha_1,\alpha_2,\alpha_3,\alpha_4)$，此时可以得到第 $t+1$ 时刻各轮运算的状态差分如下：

$$\Delta S_{0,1}^{t+1}=\text{Rotl}(\Delta m_t,31)<<<64=\left(\alpha_3^{'},\alpha_4^{'},\alpha_1^{'},\alpha_2^{'}\right)$$

其中，$\alpha_i^{'}=\alpha_i<<<31, i=1,2,3,4$。

$$\Delta S_{0,2}^{t+1}=\text{Rotl}(\Delta m_t,7)<<<32=\left(\alpha_2^{''},\alpha_3^{''},\alpha_4^{''},\alpha_1^{''}\right)$$

其中，$\alpha_i^{''}=\alpha_i<<<7, i=1,2,3,4$。

$$\begin{aligned}\Delta S_{0,3}^{t+1}&=\text{Rotl}(\text{Rotl}(\Delta m_t,31)\oplus\Delta m_t,22)<<<32\\&=(\beta_2,\beta_3,\beta_4,\beta_1)\end{aligned}$$

其中，$\beta_i=\left(\alpha_i\oplus\alpha_i^{'}\right)<<<22, i=1,2,3,4$。

$$\begin{aligned}\Delta S_{0,4}^{t+1}&=\text{Rotl}\left(S_{0,0}^{t+1}\ \&\ \Delta S_{0,1}^{t+1}\oplus\left(\Delta S_{0,2}^{t+1}>>>32\right)\right.\\&\quad\left.\oplus\Delta m_t,13\right)<<<64\\&=\Big(\text{Rotl}\left(S_{0,3}^{t+1}\ \&\ \alpha_1^{'}\oplus\alpha_3^{''}\oplus\alpha_3,13\right),\\&\quad\text{Rotl},\left(S_{0,4}^{t+1}\ \&\ \alpha_2^{'}\oplus\alpha_4^{''}\oplus\alpha_4,13\right)\\&\quad\text{Rotl}\left(S_{0,1}^{t+1}\ \&\ \alpha_3^{'}\oplus\alpha_1^{'}\oplus\alpha_1,13\right),\\&\quad\text{Rotl}\left(S_{0,2}^{t+1}\ \&\ \alpha_4^{'}\oplus\alpha_2^{''}\oplus\alpha_2,13\right)\Big)\end{aligned}$$

由 $\Delta S_{0,4}^{t+2}=\Delta S_{0,4}^{t+1}\oplus\Delta m_{t+1}$，$\Delta S_{0,4}^{t+2}=0$ 可得 $\Delta m_{t+1}=\Delta S_{0,4}^{t+1}$。

如果在下一步发生碰撞，则 ΔS^{t+2} 的各 128 bit 分组必须全部为 0，由于 $\Delta S_{0,4}^{t+1}=\Delta m_{t+1}$ 已经成立，故只需考虑 $\Delta S_{0,0}^{t+2}$, $\Delta S_{0,1}^{t+2}$, $\Delta S_{0,2}^{t+2}$, $\Delta S_{0,3}^{t+2}$ 产生碰撞的必要条件。下面逐个进行分析：

$\Delta S_{0,0}^{t+2}=0$ 的必要条件分析：

$$\begin{aligned}\Delta S_{0,0}^{t+2}&=\left(S_{0,1}^{t+1}\ \&\ \Delta S_{0,2}^{t+1}\right)\oplus\left(\Delta S_{0,1}^{t+1}\ \&\ S_{0,2}^{t+1}\right)\\&\quad\oplus\left(\Delta S_{0,1}^{t+1}\ \&\ \Delta S_{0,2}^{t+1}\right)\\&\quad\oplus\left(\Delta S_{0,3}^{t+1}>>>32\right)=0\end{aligned}\tag{1}$$

式(1)中，各异或项表示如下：

$$\begin{aligned}S_{0,1}^{t+1}\ \&\ \Delta S_{0,2}^{t+1}&=\left(S_{0,1,1}^{t+1}\ \&\ \alpha_2^{''},S_{0,1,2}^{t+1}\ \&\ \alpha_3^{''},\right.\\&\quad\left.S_{0,1,3}^{t+1}\ \&\ \alpha_4^{''},S_{0,1,4}^{t+1}\ \&\ \alpha_1^{''}\right)\\\Delta S_{0,1}^{t+1}\ \&\ S_{0,2}^{t+1}&=\left(S_{0,2,1}^{t+1}\ \&\ \alpha_3^{'},S_{0,2,2}^{t+1}\ \&\ \alpha_4^{'},\right.\\&\quad\left.S_{0,2,3}^{t+1}\ \&\ \alpha_1^{'},S_{0,2,4}^{t+1}\ \&\ \alpha_2^{'}\right)\end{aligned}$$

$$\begin{aligned}\Delta S_{0,1}^{t+1}\ \&\ \Delta S_{0,2}^{t+1}&=\left(\alpha_2^{'}\ \&\ \alpha_2^{''},\alpha_4^{'}\ \&\ \alpha_3^{''},\right.\\&\quad\left.\alpha_1^{'}\ \&\ \alpha_4^{''},\alpha_2^{'}\ \&\ \alpha_1^{''}\right)\\\Delta S_{0,3}^{t+1}>>>32&=(\beta_1,\beta_2,\beta_3,\beta_4)\end{aligned}$$

整体产生碰撞等价于各 32 bit 产生碰撞，将各异或项中对应位置的 32 bit 进行异或，结果如下：

$$\left(S_{0,1,1}^{t+1}\ \&\ \alpha_2^{''}\right)\oplus\left(S_{0,2,1}^{t+1}\ \&\ \alpha_3^{'}\right)\oplus\left(\alpha_3^{'}\ \&\ \alpha_2^{''}\right)\oplus\beta_1=0\tag{2}$$

$$\left(S_{0,1,2}^{t+1}\ \&\ \alpha_3^{''}\right)\oplus\left(S_{0,22}^{t+1}\ \&\ \alpha_4^{'}\right)\oplus\left(\alpha_4^{''}\ \&\ \alpha_3^{''}\right)\oplus\beta_2=0\tag{3}$$

$$\left(S_{0,1,3}^{t+1}\ \&\ \alpha_4^{''}\right)\oplus\left(S_{0,2,3}^{t+1}\ \&\ \alpha_1^{'}\right)\oplus\left(\alpha_1^{'}\ \&\ \alpha_4^{''}\right)\oplus\beta_3=0\tag{4}$$

$$\left(S_{0,1,4}^{t+1}\ \&\ \alpha_1^{''}\right)\oplus\left(S_{0,2,4}^{t+1}\ \&\ \alpha_2^{'}\right)\oplus\left(\alpha_2^{'}\ \&\ \alpha_1^{''}\right)\oplus\beta_4=0\tag{5}$$

根据定义 1，抽取出式(2)，式(3)，式(4)，式(5)中的“$\triangleright$”关系，得到各 32 bit 产生碰撞时差分的必要条件：

$$\begin{aligned}\beta_1\oplus\left(\alpha_3^{'}\ \&\ \alpha_2^{''}\right)&\triangleright\left(\alpha_2^{''}*\alpha_3^{'}\right)\\\beta_2\oplus\left(\alpha_4^{'}\ \&\ \alpha_3^{''}\right)&\triangleright\left(\alpha_3^{''}*\alpha_4^{'}\right)\\\beta_3\oplus\left(\alpha_1^{'}\ \&\ \alpha_4^{''}\right)&\triangleright\left(\alpha_4^{''}*\alpha_1^{'}\right)\\\beta_4\oplus\left(\alpha_2^{'}\ \&\ \alpha_1^{''}\right)&\triangleright\left(\alpha_1^{''}*\alpha_2^{'}\right)\end{aligned}$$

类似地，我们分别得到 $\Delta S_{0,1}^{t+2}$, $\Delta S_{0,2}^{t+2}$, $\Delta S_{0,3}^{t+2}$ 的碰撞必要条件，共得到两步更新产生碰撞的 16 个条件，反复运用性质 1，性质 2 和性质 3 将必要条件进行简化，再编号分类，得到如表 2 中 16 个最简必要条件。

结论 1 对于 MORUS 算法，表 2 中的 16 个约束条件是其两步更新发生内部状态碰撞的必要条件集合。

5 MORUS 算法碰撞关于字差分的必要条件

本节中，我们根据字差分重量 $\text{WB}(\Delta m_t)$ 的不同，分情况讨论是否符合结论 1 中碰撞的必要条件，从而确定字差分的重量及分布。

5.1 假设 $\text{WB}(\Delta m_t)=1$

此时消息差分仅有一个 32 bit 不为 0，不妨设 $\Delta m_t=(\alpha_1,0,0,0)$, $\alpha_1\neq0$。由 $\alpha_i^{''}=\alpha_i<<<7$ 可知 $\alpha_1^{''}\neq0$。由 $\alpha_2=\alpha_3=\alpha_4=0$，得到 $\text{Rotl}(\alpha_2^{''}\oplus\alpha_2,13)=0$, $\text{Rotl}(\alpha_4^{'},13)=0$。利用定理 1，可知这与表 2 中条件(9) $\alpha_1^{''}\triangleright(\text{Rotl}(\alpha_4^{'},13)*\text{Rotl}(\alpha_2^{''}\oplus\alpha_2,13))$ 产生矛盾。类似地，$\Delta m_t=(0,\alpha_2,0,0)$, $\Delta m_t=(0,0,0,\alpha_4)$, $\Delta m_t=(0,0,0,\alpha_4)$ 时分别与条件(10)-(12)矛盾，故假设不成立。

5.2 假设 $\text{WB}(\Delta m_t)=2$

此时分两种情况进行讨论。

情况 1 假设 $\Delta m_t=(\alpha_1,\alpha_2,0,0)(\alpha_1\neq0,\alpha_2\neq0,\alpha_3=\alpha_4=0)$。

根据条件(2) $\beta_2\triangleright(\alpha_3^{''}*\alpha_4^{'})$，由 $\alpha_3=\alpha_4=0$，可知 $\beta_2=0$，又 $\alpha_2\neq0$，可知 $\alpha_2=1^{32}$，从而得到

表 2 MORUS 算法两步更新产生碰撞的差分必要条件

内部状态差	碰撞必要条件
$\Delta S_{0,0}^{t+2}$	(1) $\beta_1 \rhd (\alpha_2^{''} * \alpha_3^{'})$
	(2) $\beta_2 \rhd (\alpha_3^{''} * \alpha_4^{'})$
	(3) $\beta_3 \rhd (\alpha_4^{''} * \alpha_1^{'})$
	(4) $\beta_4 \rhd (\alpha_1^{''} * \alpha_2^{'})$
$\Delta S_{0,1}^{t+2}$	(5) $\alpha_1^{'} \oplus \mathrm{Rotl}(\alpha_3^{''} \oplus \alpha_3,13) \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_3^{'},13) * \mathrm{Rotl}(\alpha_1^{''} \oplus \alpha_1,13))$
	(6) $\alpha_2^{'} \oplus \mathrm{Rotl}(\alpha_4^{''} \oplus \alpha_4,13) \rhd (\mathrm{Rotl}(\alpha_2^{'},13) * \mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13))$
	(7) $\alpha_3^{'} \oplus \mathrm{Rotl}(\alpha_1^{''} \oplus \alpha_1,13) \rhd (\mathrm{Rotl}(\alpha_3^{'},13) * \mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_3^{''} \oplus \alpha_3,13))$
	(8) $\alpha_4^{'} \oplus \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13) \rhd (\mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{'},13) * \mathrm{Rotl}(\alpha_4^{''} \oplus \alpha_4,13))$
$\Delta S_{0,2}^{t+2}$	(9) $\alpha_1^{''} \rhd (\mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13))$
	(10) $\alpha_2^{''} \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_3^{''} \oplus \alpha_3,13))$
	(11) $\alpha_3^{''} \rhd (\mathrm{Rotl}(\alpha_2^{'},13) * \mathrm{Rotl}(\alpha_4^{''} \oplus \alpha_4,13))$
	(12) $\alpha_4^{''} \rhd (\mathrm{Rotl}(\alpha_3^{'},13) * \mathrm{Rotl}(\alpha_1^{''} \oplus \alpha_1,13))$
$\Delta S_{0,3}^{t+2}$	(13) $\beta_1 \rhd (\mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13))$
	(14) $\beta_2 \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_3^{''} \oplus \alpha_3,13))$
	(15) $\beta_3 \rhd (\mathrm{Rotl}(\alpha_2^{'},13) * \mathrm{Rotl}(\alpha_4^{''} \oplus \alpha_4,13))$
	(16) $\beta_4 \rhd (\mathrm{Rotl}(\alpha_3^{'},13) * \mathrm{Rotl}(\alpha_1^{''} \oplus \alpha_1,13))$

$\mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13)=0$；又 $\mathrm{Rotl}(\alpha_4^{'},13)=0$，而 $\alpha_1^{''} \neq 0$，$\mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13)=0$，根据定理 1 可知这与条件 (9) $\alpha_1^{''} \rhd (\mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13))$ 产生矛盾。

$\Delta m_t=(0,\alpha_2,\alpha_3,0), \Delta m_t=(0,0,\alpha_3,\alpha_4), \Delta m_t=(\alpha_1,0,0,\alpha_4)$ 时类似，均不成立。

情况 2 假设 $\Delta m_t=(\alpha_1,0,\alpha_3,0), (\alpha_1 \neq 0, \alpha_3 \neq 0, \alpha_2=\alpha_4=0)$。

由 $\alpha_1 \neq 0, \alpha_3 \neq 0$，$\alpha_2=\alpha_4=0$ 易知 $\alpha_1^{''} \neq 0$，$\mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13)=0$。根据定理 1，这与表 2 中条件 (9) $\alpha_1^{''} \rhd (\mathrm{Rotl}(\alpha_4^{'},13) * \mathrm{Rotl}(\alpha_2^{''} \oplus \alpha_2,13))$ 产生矛盾。

$\Delta m_t=(0,\alpha_2,0,\alpha_4)(\alpha_2 \neq 0, \alpha_4 \neq 0)$ 时类似，与条件(10)产生矛盾。故假设不成立。

5.3 假设 $\mathrm{WB}(\Delta m_t)=3$

此时输入差 $\Delta m_t=(\alpha_1,\alpha_2,\alpha_3,0)(\alpha_i \neq 0, i=1,2,3)$。我们对结论 1 中的条件进行整理，令 $\alpha_4=0$，反复利用性质 1、性质 2 和性质 3 进行化简，然后按照块间的关系对碰撞必要条件进行分类并重新标号，得到 $\mathrm{WB}(\Delta m_t)=3$ 时产生碰撞消息差分的 13 个必要条件，如表 3 所示

根据由(5) $\beta_2 \rhd \alpha_3^{''}$ (6) $\alpha_3^{''} \rhd \mathrm{Rotl}(\alpha_2^{'},13)$，通过性质 3(传递性)可以得到 $\beta_2 \rhd \mathrm{Rotl}(\alpha_2^{'},13)$。

表 3 $\mathrm{WB}(\Delta m_t)=3$ 时产生碰撞消息差分的必要条件

涉及的字差分	碰撞必要条件
α_2	(1) $\alpha_2^{'} \rhd (\mathrm{Rotl}(\alpha_2^{'},13) * \mathrm{Rotl}(\alpha_2 \oplus \alpha_2^{''},13))$
	(2) $\mathrm{Rotl}(\alpha_2 \oplus \alpha_2^{''},13) \rhd \mathrm{Rotl}(\alpha_2^{'},13)$
α_1 和 α_2	(3) $\alpha_1^{''} \rhd \mathrm{Rotl}(\alpha_2 \oplus \alpha_2^{''},13)$
	(4) $\beta_1 \rhd \mathrm{Rotl}(\alpha_2 \oplus \alpha_2^{''},13)$
α_3 和 α_2	(5) $\beta_2 \rhd \alpha_3^{''}$
	(6) $\alpha_3^{''} \rhd \mathrm{Rotl}(\alpha_2^{'},13)$
	(7) $\beta_3 \rhd \mathrm{Rotl}(\alpha_2^{'},13)$
α_1 和 α_3	(8) $\beta_3 \rhd \alpha_1^{'}$
	(9) $\alpha_3^{'} \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_3^{'} \oplus \alpha_1 \oplus \alpha_1^{''},13) * \mathrm{Rotl}(\alpha_1^{'} \oplus \alpha_3 \oplus \alpha_3^{''},13))$
	(10) $\alpha_1^{'} \rhd (\mathrm{Rotl}(\alpha_1^{'} \oplus \alpha_3 \oplus \alpha_3^{''},13) * \mathrm{Rotl}(\alpha_3^{'} \oplus \alpha_1 \oplus \alpha_1^{''},13))$
α_1, α_2 和 α_3	(11) $\beta_1 \rhd (\alpha_2^{''} * \alpha_3^{'})$
	(12) $\alpha_2^{''} \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_1^{'} \oplus \alpha_3 \oplus \alpha_3^{''},13))$
	(13) $\beta_2 \rhd (\mathrm{Rotl}(\alpha_1^{'},13) * \mathrm{Rotl}(\alpha_1^{'} \oplus \alpha_3 \oplus \alpha_3^{''},13))$

我们设计下面的算法来实现寻找 $\mathrm{WB}(\Delta m_t)=3$ 时产生碰撞所有可能的输入差分，约定 $|A|$ 表示集合 A 中元素个数。算法主要以定理 2 为判断依据来完成每一步的穷举或差分筛选。

算法 1 $\mathrm{WB}(\Delta m_t)=3$ 时可能消息差分搜索算法。

步骤 1 对于所有的 $\alpha_2 \in \{0,1\}^{32}$，执行如下步骤：

步骤 1.1 如果 α_2 满足表 3 中的条件(1)，条件(2)，同时满足 $\beta_2 \rhd \mathrm{Rotl}(\alpha_2^{'},13)$，则将 α_2 存储在集合 A 中；

步骤 1.2 若 $|A|=0$，则返回{0}，终止算法；

步骤 2 对于所有的 $\alpha_1 \in \{0,1\}^{32}$ 和 $\alpha_2 \in A$，执行如下步骤：

步骤 2.1 如果 (α_1,α_2) 满足表 3 中的条件(3)，条件(4)，则将 α_1 存储在集合 B 中；

步骤 2.2 若 $|B|=0$，则返回{0}，终止算法；

步骤 3 对于所有的 $\alpha_3 \in \{0,1\}^{32}$ 和 $\alpha_1 \in B$，执行如下步骤：

步骤 3.1 如果 (α_1,α_3) 满足表格 3 中的条件(5)，条件(6)，条件(7)，则将 α_3 存储在集合 C 中；

步骤 3.2 若 $|C|=0$，则返回{0}，终止算法；

步骤 4 对于所有的 $\alpha_1 \in B$ 和 $\alpha_3 \in C$，执行如下步骤：

步骤 4.1 如果 (α_1,α_3) 满足表 3 中的条件(8)，条件(9)，条件(10)，则将 (α_1,α_3) 存储在集合 D 中；

步骤 4.2　若$|D|=0$，则返回{0}，终止算法；

步骤 5　对于所有的$\alpha_2 \in A$和$(\alpha_1,\alpha_3) \in D$，执行如下步骤：

步骤 5.1　如果$(\alpha_1,\alpha_2,\alpha_3)$满足表 3 中的条件(11)，条件(12)，条件(13)，则将$(\alpha_1,\alpha_2,\alpha_3)$存储在集合$E$中；

步骤 5.2　若$|E|=0$，则返回{0}，终止算法；否则，返回E; // E中元素是最终候选输入差分。

实验发现算法 1 在步骤 2 后终止，$|B|=0$，因此$\mathrm{WB}(\Delta m_t)$=3 时没有符合的输入差，即$\mathrm{WB}(\Delta m_t) \neq 3$。

复杂度分析：算法首先以α_2自身限制条件(包括(1), (2)以及$\beta_2 \triangleright \mathrm{Rotl}(\alpha_2',13)$)为突破口，穷举得到符合条件的$\alpha_2$。实验发现$\alpha_2$只有 6 个候选值，再执行步骤 2 穷举$\alpha_1$，符合条件的$(\alpha_1,\alpha_2)$对数为 0。因此整个算法的实际计算复杂度只有$O(2^{32})$，远小于穷举$(\alpha_1,\alpha_2,\alpha_3)$的复杂度$O(2^{96})$。

经过以上讨论可知$\mathrm{WB}(\Delta m_t)=1$，$\mathrm{WB}(\Delta m_t)=2$，$\mathrm{WB}(\Delta m_t)=3$时不能产生碰撞，因此得到如下结论。

结论 2　对于 MORUS 算法，若在算法第 t 和 t+1 时刻由关联数据(或明文)引入差分，若经过两步更新后内部状态发生碰撞，则第 t 时刻引入的差分Δm_t满足$\mathrm{WB}(\Delta m_t)=4$。

6　内部状态发生碰撞关于比特差分的必要条件

由于内部状态碰撞的概率与输入差分的重量有着直接关系，基于第 5 节得到的内部状态碰撞必要条件，本节着重讨论碰撞时，输入差分汉明重量的下界，进而得到碰撞的概率上界。主要包括以下步骤：

步骤 1　将结论 2 的 16 个碰撞必要条件转化为混合整数规划问题并表示成 OPB 语言，列出伪布尔方程；

步骤 2　设置输入差分的最小重量为目标函数。

步骤 3　使用 SCIP 求解器求解伪布尔方程，找出使得内部状态产生碰撞的输入差分重量的下界。

由实验结果可以得到如下结论：

结论 3　对于 MORUS 算法，若在算法第 t 和 t+1 时刻由关联数据(或明文)引入差分，且经过两步更新后内部状态发生碰撞，则第 t 时刻引入的差分汉明重量至少为 28。

根据设计报告[7]，若消息差分的活动比特不少于 n 比特，算法更新两步后差分概率至少是2^{-5n}。故我们得到碰撞概率至少为$2^{-28\times5}=2^{-140}<2^{-128}$。表 4 将文献[7]的碰撞分析结果与本文分析结果进行了对比。

表 4 两步更新发生内部状态发生碰撞分析结果对比

来源	分析方法	差分必要集	字差分分布特点	差分重量下界	碰撞概率
文献[7]	实验搜索	–	–	$\mathrm{W}(\Delta m_t)\geq 26$	$\leq 2^{-130}$
本文	理论推导	16 个碰撞必要条件	$\mathrm{WB}(\Delta m_t)=4$	$\mathrm{W}(\Delta m_t)\geq 28$	$\leq 2^{-140}$

7　结束语

本文针对 MORUS 算法认证的安全性，利用分块的思想给出了内部状态碰撞的必要条件，得到确定字差分的重量及分布，即输入差的非零比特必须分布在每一个 32 bit 上。最后将碰撞必要条件集表示成伪布尔最优化问题，实验求解出碰撞输入差分重量的下界为 28 bit，碰撞概率小于2^{-140}，保证了认证标签的安全性，得到了比文献[7](2^{-130})更好的分析结果。我们的工作缩小了寻找可能输入差分的穷举范围，降低了搜索的时间复杂度，为 MORUS 算法认证阶段的安全性评估提供理论支撑。

本文使用的从非线性方程中找到信息泄漏规律的方法有一定普适性，可用于对其他认证加密算法产生碰撞的差分搜索，进一步缩小搜索范围。此外，我们将对 MORUS 算法抵抗其他攻击方法的能力进行评估。

参 考 文 献

[1] BELLARE M and NAMPREMPRE C. Authenticated encryption: Relations among notions and analysis of the generic composition paradigm[J]. *Journal of Cryptology*, 2008, 21(4): 469–491. doi: 10.1007/s00145-008-9026-x.

[2] BERNSTEIN D J. Caesar: Competition for authenticated encryption: Security, applicability, and robustness[OL]. http://competitions.cr.yp.to/index.html, 2015.

[3] DOBRAUNING C, EICHLSEDER M, and MENDEL F. Heuristic tool for linear cryptanalysis with applications to CAESAR candidates[C]. Advances in Cryptology — ASIACRYPT 2015, Auckland, New Zealand, 2015: 490–509. doi: 10.1007/978-3-662-48800-3_20.

[4] DEY P, ROHIT S R, SARKAR S, *et al*. Differential fault analysis on Tiaoxin and AEGIS family of ciphers[C]. Security in Computing and Communications 2016, Jaipur, India, 2016: 74–86. doi: 10.1007/978-981-10-2738-3_7.

[5] PEYRIN T, SIM S, WANG L, *et al*. Cryptanalysis of JAMBU[C]. Fast Software Encryption 2015, Istanbul, Turkey, 2015: 264–281. doi: 10.1007/978-3-662-48116-5_13.

[6] SALAM M, BARTLETT H, PIEPRZYK J, *et al*. Investigating cube attack on the authenticated encryption stream cipher ACORN[C]. Applications and Techniques in Information Security 2016, Cairns, QLD, Australia, 2016: 15–26. doi: 10.1007/978-981-10-2741-3_2.

[7] WU H and HUANG T. The authenticated cipher MORUS (v1)[OL]. http://competitions.cr.yp.to/round2/morusv11.pdf, 2015.

[8] MILEVA A, DIMITROVA V, and VELICHKOV V. Analysis of the authenticated cipher MORUS (v1)[C]. Cryptography and Information Security in the Balkans 2015, Koper, Slovenia, 2015: 45–59. doi: 10.1007/978-3-319-29172-7_4.

[9] 张沛, 关杰, 李俊志, 等. MORUS 算法初始化过程的混乱与扩散性质研究[J]. 密码学报, 2015, 2(6): 536–548. doi: 10.13868/j.cnki.jcr.000100.
ZHANG Pei, GUAN Jie, LI Junzhi, *et al*. Research on the confusion and diffusion properties of the initialization of MORUS[J]. *Journal Cryptologic Research*, 2015, 2(6): 536–548. doi: 10.13868/j.cnki.jcr.000100.

[10] WANG Xiaoyun and YU Hongbo. How to break MD5 and other hash functions[C]. Advances in Cryptology — EUROCRYPT 2005, Aarhus, Denmark, 2005: 19–35. doi: 10.1007/11426639_2.

[11] FUHR T, LEURENT G, and SUDER V. Collision attacks against CAESAR candidates — Forgery and key-recovery against AEZ and Marble[C]. Advances in Cryptology — ASIACRYPT 2015, Auckland, New Zealand, 2015: 510–532. doi: 10.1007/978-3-662-48800-3_21.

[12] PEYRIN T. Collision attack on Grindahl[J]. *Journal of Cryptology*, 2015, 28(4): 879–898. doi: 10.1007/s00145-014-9186-9.

[13] ROUSSEL O and MANQUINHO V. Input/output format and solver requirements for the competitions of pseudo-boolean solvers[OL]. http://www.cril.univ-artois.fr/PB12/format.pdf, 2012.

[14] BERTSIMAS D and WEISMANTEL R. Optimization over Integers[M]. Massachusetts, USA, Dynamic Ideas, 2005: 73–82.

[15] ACHTERBERG T. SCIP: Solving Constraint Integer Programs[J]. *Mathematical Programming Computation*, 2009, 1(1): 1–41. doi: 10.1007/s12532-008-0001-1.

关 杰： 女，1974 年生，教授，博士生导师，研究方向为密码技术与分析.

施泰荣： 女，1992 年生，硕士生，研究方向为对称密码设计与分析.

李俊志： 男，1990 年生，博士生，研究方向为序列密码的设计与分析.

张 沛： 男，1991 年生，硕士，研究方向为对称密码设计与分析.