(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 110601818 A
(43)申请公布日 2019.12.20

(21)申请号 201910910931.8

(22)申请日 2019.09.25

(71)申请人 东华大学
地址 201600 上海市松江区人民北路2999号

(72)发明人 李玮 李嘉耀 蔡天培 汪梦林 李华婷 李悦 曹珊

(74)专利代理机构 上海申汇专利代理有限公司 31001
代理人 徐俊

(51)Int.Cl.
H04L 9/06(2006.01)
H04L 9/00(2006.01)

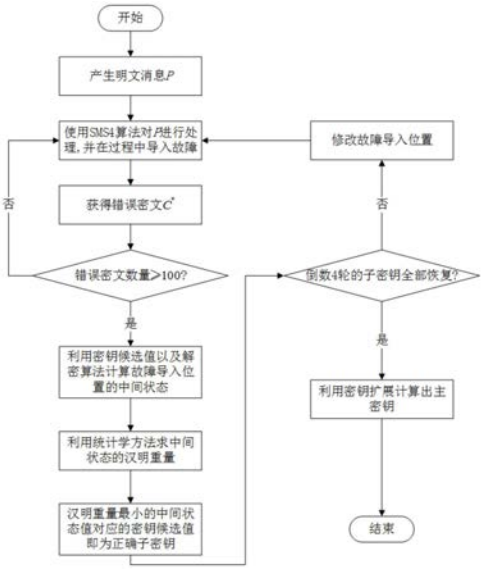
权利要求书2页 说明书6页 附图4页

(54)发明名称

一种检测SMS4密码算法抵御统计故障攻击的方法

(57)摘要

本发明提供了本发明提供了一种检测SMS4密码算法抵御统计故障攻击的方法，首先通过SMS4算法对明文消息进行处理，在此阶段，仅需要控制一种实验环境：在算法对明文消息处理的过程中，使用某些物理手段，对处理过程进行干扰，诱导其产生故障，获得错误的输出，记为C^*。通过解密算法以及统计学的方法，计算汉明重量，来测评SMS4密码算法对统计故障攻击的抵御能力。然后通过判断所导入的故障的有效性，进而恢复出密钥。本发明提供的方法具有简单、快捷、准确且易于实现等特点，对检测SMS4密码算法抵御统计故障攻击的能力提供了良好的分析依据。



CN 110601818 A

1.一种检测SMS4密码算法抵御统计故障攻击的方法，其特征在于，包括以下步骤：

步骤1、随机生成一条明文消息P，消息长度为128比特；

步骤2：利用SMS4密码算法对明文P进行加密，并在加密过程中导入随机单字节故障，得到错误密文集合，记为C^*，其中，SMS4在加密过程的分组长度和密钥长度均为128比特，加密过程中，共需要31轮迭代，每一轮迭代利用可逆的合成置换T进行变换，并将第31轮迭代结果字节翻转作为输出；

步骤3：重复步骤1至步骤3，直到获取足够数量的有效错误密文集合C^*；

步骤4：根据有效故障传播路径确定第31轮迭代受到故障影响的子密钥k_{31}以此确认密钥候选值的取值范围，得到子密钥k_{31}候选值集合SK，再按照下列步骤，确认受到故障影响的子密钥k_{31}的所有字节的取值：

步骤401、将子密钥候选值集合SK和有效错误密文集合C^*作笛卡尔积，则有：$SK\times C^*$，将$SK\times C^*$中的每个元素代入下列公式：

$$\begin{aligned}
&\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)\\
&=\left(x_{8}^{29}x_{9}^{29}x_{10}^{29}x_{11}^{29}\right)\\
&=\left(x_{4}^{30}x_{5}^{30}x_{6}^{30}x_{7}^{30}\right)\\
&=\left(x_{0}^{31}x_{1}^{31}x_{2}^{31}x_{3}^{31}\right)\\
&=\left(x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}\right)\oplus\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<2\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<10\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<18\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<24\right)
\end{aligned}$$

式中，$C^*=x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}$，$x_i^j$表示第j轮的中间状态的第i个字节；$\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)$表示将第32轮中间状态的前12字节按照每4字节为一组分类后与sk进行异或计算，并代入SMS4的加解密算法所使用的非线性变换τ；sk表示穷举过程中所使用的子密钥候选值，以此计算出每个子密钥候选值与所有错误密文对应的中间状态值$\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)$，并按照子密钥候选值分类，使得每个子密钥候选值均对应一组中间状态值；

步骤402、利用统计学的方法，求出每一组中间状态值中的x_{12}^{28}的汉明重量；

步骤403、挑选出汉明重量值最小的一组中间状态值，其对应的子密钥候选值即为正确的子密钥的部分比特；

利用步骤401至步骤403对密钥候选值作穷举从而有效缩小密钥的搜索空间

缩小密钥的搜索空间，并在穷举密钥的所有可能取值后，利用统计学的方法，求得子密钥部分比特的正确值；

步骤5：将故障的导入的轮数提前一轮，改变故障的导入位置，重复步骤1至步骤4，继续求第30轮迭代受到故障影响的子密钥k_{30}的所有比特位，直至按照步骤1至步骤4的方法求得

第31、30、29、28轮迭代受到故障影响的子密钥{k_{31},k_{30},k_{29},k_{28}}的所有比特,根据SMS4的密钥扩展,逆推出主密钥K,公式如下:

$$\begin{cases} k_i = k_{i+4} \oplus T'\left(k_{i+1} \oplus k_{i+2} \oplus k_{i+3} \oplus ck_i\right), i \in \{31,30,\cdots,0\} \\ MK_i = k_i \oplus FK_i, i \in \{3,2,1,0\} \\ K = MK_0 MK_1 MK_2 MK_3 \end{cases}$$

式中,T′表示合成置换T的逆,ck_i表示密钥扩展使用的32个固定参数,FK_i表示密钥扩展系统参数。

2.如权利要求1所述的一种检测SMS4密码算法抵御统计故障攻击的方法,其特征在于,所述步骤2中,对于故障的导入位置,说明如下:

当单子节故障在除第25、26、27、28轮外被导入时,被视为无效故障;

当单子节故障的导入位置在第25、26、27、28轮的第0、第1、第2、第3字节时,导入的故障为无效故障;

当单子节故障的导入位置在第28轮除第0、第1、第2、第3字节外时,受到故障影响的子密钥为第31轮子密钥的所有比特;

当单子节故障的导入位置在第27轮除第0、第1、第2、第3字节外时,受到故障影响的子密钥为第30轮子密钥的所有比特;

当单子节故障的导入位置在第26轮除第0、第1、第2、第3字节外时,受到故障影响的子密钥为第29轮子密钥的所有比特;

当单子节故障的导入位置在第25轮除第0、第1、第2、第3字节外时,受到故障影响的子密钥为第28轮子密钥的所有比特。

3.如权利要求1所述的一种检测SMS4密码算法抵御统计故障攻击的方法,其特征在于,步骤2中,利用SMS4密码算法对明文P进行加密的过程中,使用外部物理设备改变周围物理环境,使得SMS4密码算法受到干扰,诱导SMS4在运行过程中产生故障,从而得到错误输出。

一种检测SMS4密码算法抵御统计故障攻击的方法

技术领域

[0001] 本发明涉及一种检测SMS4分组密码算法抵御统计故障攻击的方法，属于信息安全技术领域。本发明可用于评估出SMS4分组密码算法抵御统计故障攻击的能力，主要应用于测评封装了SMS4分组密码算法的产品的安全性。

背景技术

[0002] 随着信息技术的飞速发展，技术人员在设计互联网的信息交互方式时，消息的完整性以及保密性必须由一种既安全又可靠的密码算法来保证，正因为如此，密码算法的安全性问题一直受到国内外学者的重视。

[0003] SMS4是一种在国内广泛使用的WAPI无线网络标准中使用的密码算法，于2012年被国家商用密码管理局确定为国家密码行业标准，在我国密码行业中有着极其重要的位置。SMS4密码算法的分组长度和密钥长度均为128比特，加解密算法采用32轮非平衡Feistel迭代结构。由于SMS4算法的结构特点，其不得不面对统计故障攻击的威胁。

[0004] 在密码分析学中，统计故障攻击属于唯密文故障攻击的一种，其针对分组密码的结构和轮函数的特性，将故障攻击和统计学的分析方法相结合，是攻击者能力要求最弱，也是目前唯一一种只需要知道密文就可以计算出加密密钥的攻击方法。通过多次实验，在执行算法时导入故障，并分析故障对密文的影响，利用计算中间状态的汉明重量的方法，求出子密钥的可能取值，并最终通过密钥扩展算法，恢复出主密钥。目前还没有公开的报告评估SMS4密码算法抵御统计故障攻击的能力，为封装了SMS4算法的产品带来了安全隐患。。

发明内容

[0005] 本发明要解决的技术问题是：如何对SMS4分组密码算法抵御统计故障分析的能力进行评估。

[0006] 为了解决上述技术问题，本发明的技术方案是提供了一种检测SMS4密码算法抵御统计故障攻击的方法，其特征在于，包括以下步骤：

[0007] 步骤1、随机生成一条明文消息P，消息长度为128比特；

[0008] 步骤2：利用SMS4密码算法对明文P进行加密，并在加密过程中导入随机单字节故障，得到错误密文集合，记为C^*，其中，SMS4在加密过程的分组长度和密钥长度均为128比特，加密过程中，共需要31轮迭代，每一轮迭代利用可逆的合成置换T进行变换，并将第31轮迭代结果字节翻转作为输出；

[0009] 步骤3：重复步骤1至步骤3，直到获取足够数量的有效错误密文集合C^*；

[0010] 步骤4：根据有效故障传播路径确定第31轮迭代受到故障影响的子密钥k_{31}以此确认密钥候选值的取值范围，得到子密钥k_{31}候选值集合SK，再按照下列步骤，确认受到故障影响的子密钥k_{31}的所有字节的取值：

[0011] 步骤401、将子密钥候选值集合SK和有效错误密文集合C^*作笛卡尔积，则有：$SK\times C^*$，将$SK\times C^*$中的每个元素代入下列公式：

$$
\begin{aligned}
&\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)\\
&=\left(x_{8}^{29}x_{9}^{29}x_{10}^{29}x_{11}^{29}\right)\\
&=\left(x_{4}^{30}x_{5}^{30}x_{6}^{30}x_{7}^{30}\right)\\
&=\left(x_{0}^{31}x_{1}^{31}x_{2}^{31}x_{3}^{31}\right)\\
&=\left(x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}\right)\oplus\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<2\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<10\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<18\right)\\
&\oplus\left(\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<24\right)
\end{aligned}
\tag{0012}
$$

[0013] 式中，$C^{*}=x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}$，$x_i^j$表示第j轮的中间状态的第i个字节；$\tau\left(\left(x_{0}^{32}x_{1}^{32}x_{2}^{32}x_{3}^{32}\right)\oplus\left(x_{4}^{32}x_{5}^{32}x_{6}^{32}x_{7}^{32}\right)\oplus\left(x_{8}^{32}x_{9}^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)$表示将第32轮中间状态的前12字节按照每4字节为一组分类后与sk进行异或计算，并代入SMS4的加解密算法所使用的非线性变换τ；sk表示穷举过程中所使用的子密钥候选值，以此计算出每个子密钥候选值与所有错误密文对应的中间状态值$\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)$，并按照子密钥候选值分类，使得每个子密钥候选值均对应一组中间状态值；

[0014] 步骤402、利用统计学的方法，求出每一组中间状态值中的x_{12}^{28}的汉明重量；

[0015] 步骤403、挑选出汉明重量值最小的一组中间状态值，其对应的子密钥候选值即为正确的子密钥的部分比特；

[0016] 利用步骤401至步骤403对密钥候选值作穷举从而有效缩小密钥的搜索空间缩小密钥的搜索空间，并在穷举密钥的所有可能取值后，利用统计学的方法，求得子密钥部分比特的正确值；

[0017] 步骤5：将故障的导入的轮数提前一轮，改变故障的导入位置，重复步骤1至步骤4，继续求第30轮迭代受到故障影响的子密钥k_{30}的所有比特位，直至按照步骤1至步骤4的方法求得第31、30、29、28轮迭代受到故障影响的子密钥$\{k_{31},k_{30},k_{29},k_{28}\}$的所有比特，根据SMS4的密钥扩展，逆推出主密钥K，公式如下：

$$
\begin{cases}
k_i=k_{i+4}\oplus T'\left(k_{i+1}\oplus k_{i+2}\oplus k_{i+3}\oplus ck_i\right), i\in\{31,30,\cdots,0\}\\
MK_i=k_i\oplus FK_i, i\in\{3,2,1,0\}\\
K=MK_0MK_1MK_2MK_3
\end{cases}
\tag{0018}
$$

[0019] 式中，T′表示合成置换T的逆，ck_i表示密钥扩展使用的32个固定参数，FK_i表示密钥扩展系统参数。

[0020] 优选地，所述步骤2中，对于故障的导入位置，说明如下：

[0021] 当单子节故障在除第25、26、27、28轮外被导入时，被视为无效故障；

[0022] 当单子节故障的导入位置在第25、26、27、28轮的第0、第1、第2、第3字节时，导入的故障为无效故障；

[0023] 当单子节故障的导入位置在第28轮除第0、第1、第2、第3字节外时，受到故障影响的子密钥为第31轮子密钥的所有比特；

[0024] 当单子节故障的导入位置在第27轮除第0、第1、第2、第3字节外时，受到故障影响的子密钥为第30轮子密钥的所有比特；

[0025] 当单子节故障的导入位置在第26轮除第0、第1、第2、第3字节外时，受到故障影响的子密钥为第29轮子密钥的所有比特；

[0026] 当单子节故障的导入位置在第25轮除第0、第1、第2、第3字节外时，受到故障影响的子密钥为第28轮子密钥的所有比特。

[0027] 优选地，步骤2中，利用SMS4密码算法对明文P进行加密的过程中，使用外部物理设备改变周围物理环境，使得SMS4密码算法受到干扰，诱导SMS4在运行过程中产生故障，从而得到错误输出。

[0028] 本发明提供了一种检测SMS4密码算法抵御统计故障攻击的方法，首先通过SMS4算法对明文消息进行处理，在此阶段，仅需要控制一种实验环境：在算法对明文消息处理的过程中，使用某些物理手段，对处理过程进行干扰，诱导其产生故障，获得错误的输出，记为C^*。通过解密算法以及统计学的方法，计算汉明重量，来测评SMS4密码算法对统计故障攻击的抵御能力。然后通过判断所导入的故障的有效性，进而恢复出密钥。

[0029] 本发明提供的方法具有简单、快捷、准确且易于实现等特点，对检测SMS4密码算法抵御统计故障攻击的能力提供了良好的分析依据。

附图说明

[0030] 图1为本实施例提供的检测SMS4算法抵御统计故障攻击的方法流程图；

[0031] 图2为SMS4密码算法中导入故障后的故障传播路径图；

[0032] 图3为SMS4密码算法的加密流程图；

[0033] 图4为本实施例方案的实验环境示意图。

具体实施方式

[0034] 下面结合具体实施例，进一步阐述本发明。应理解，这些实施例仅用于说明本发明而不用于限制本发明的范围。此外应理解，在阅读了本发明讲授的内容之后，本领域技术人员可以对本发明作各种改动或修改，这些等价形式同样落于本申请所附权利要求书所限定的范围。

[0035] 本发明中所使用的基本符号说明如下：

[0036] $\oplus$:异或运算；

[0037] ||:连接运算，串a与串b的连接表示为a||b，也可以表示为ab；

[0038] M:明文消息；

[0039] C^*:使用SMS4算法处理明文消息M并导入故障后所得到的错误密文；

[0040] $|C^*|$:错误密文长度；

[0041] x_i^j：第j轮的中间状态值的第i个字节，其中0≤j≤31，0≤i≤15且i、j为整数；

[0042] X^j:第j轮的中间状态值，其中 $X^j = x_0^j x_1^j \cdots x_{15}^j$；

[0043] K:加密时使用的主密钥,且K＝$MK_0MK_1MK_2MK_3$;

[0044] k_i:对K使用密钥扩展产生的第i个子密钥,0≤i≤31;

[0045] ck_i:密钥扩展使用的32个固定参数,具体可参考SMS4标准文档;

[0046] FK_i:密钥扩展系统参数,FK_0＝A3B1BAC6,FK_1＝A3B1BAC6,FK_2＝A3B1BAC6,FK_3＝A3B1BAC6;

[0047] sk:穷举过程中所使用的子密钥候选值。

[0048] 使用SMS4密码算法使用同一个密钥对同一个明文消息进行处理时,通过改变实验环境(正常情况与受到时钟、电压、湿度、辐射、压力、光和涡电流等物理因素影响的情况),攻击者可以获得一个错误输出,并根据此错误输出的推断出关键信息。攻击者可以在SMS4算法的执行过程中,诱导其产生随机故障,但并不清楚故障导入的具体位置。由此可见,获取故障导入的位置尤为重要,也因此想要从错误输出中获取重要信息,则必须保证导入故障的位置是有效的,否则,攻击者不能够从错误输出中获取到关键信息。

[0049] 图1为本发明提供的检测SMS4密码算法抵御统计故障攻击的方法的流程图。所述的检测SMS4算法抵御统计故障攻击的方法包括如下步骤:

[0050] 步骤1:随机生成一条明文消息P,消息长度为128比特;

[0051] 步骤2:利用SMS4密码算法对明文P进行加密,并在加密过程中导入随机单字节故障,得到错误密文,记为C^*;

[0052] 步骤3:重复步骤1至步骤2,直到获取足够数量的有效错误密文C^*;

[0053] 步骤4:根据有效故障的传播路径,可以缩小密钥的搜索空间,并在穷举密钥的所有可能取值后,利用统计学的方法,求得子密钥部分比特的正确值;

[0054] 步骤5:重复上述步骤,直到恢复第31、30、29、28轮子密钥的所有比特,并利用密钥扩展计算主密钥。

[0055] 针对步骤2,使用SMS4密码算法处理明文消息M的过程中,为了保障实验的正确性,需要对实验环境作出控制,具体操作如下:

[0056] (1)输入消息P,在算法对P进行加密的过程中,使用外部物理设备改变周围物理环境,使得SMS4密码算法受到干扰,诱导SMS4在运行过程中产生故障,从而得到错误输出,结果记为C^*。

[0057] 其中,步骤(1)通过改变周边环境诱导SMS4算法产生故障的方法有:改变时钟、电压、湿度、辐射、压力、光和涡电流等;

[0058] 针对步骤4,对C^*的统计故障分析的原理如下:

[0059] SMS4是一种轻量级分组密码算法,由国内密码学者提出,其主旨在于可应用在低资源、低消耗的嵌入式系统(如银行卡、手机SIM卡等等)当中,以确保其安全性。SMS4在加/解密过程的分组长度和密钥长度均为128比特,加/解密过程中,共需要31轮迭代,每一轮迭代利用可逆的合成置换T进行变换,并将第31轮迭代结果字节翻转作为输出。其中,T(•)＝L(τ(•)),由非线性变换τ和线性变换L复合而成;非线性变换τ为四个并排的S盒,其输出作为L的输入,且$L(x)=x\oplus(x<<<2)\oplus(x<<<10)\oplus(x<<<18)\oplus(x<<<24)$。

[0060] 以故障导入在第28轮、第12字节为例,故障的传播路径如图2所示。因此可以判断,受到故障影响的子密钥为k_{31}中的所有字节,即32比特,以此确认密钥候选值的取值范围,再按照下列步骤,确认k_{31}所有字节的取值。

[0061] 首先，将SK和$\mathbb{C}^*$作笛卡尔积，所得结果如下所示：

[0062] $$SK\times\mathbb{C}^*=\left\{\left(sk_0,C_0^*\right),\left(sk_0,C_1^*\right),\cdots,\left(sk_0,C_j^*\right),\left(sk_1,C_0^*\right),\left(sk_1,C_1^*\right),\cdots,\left(sk_1,C_j^*\right),\cdots,\left(sk_i,C_j^*\right)\right\}$$

[0063] 其中，SK为子密钥候选值集合，$\mathbb{C}^*$为步骤2生成的错误密文集合，$SK\times\mathbb{C}^*$的元素是由SK和$\mathbb{C}^*$两个集合元素组成的二元组，并将上述集合中的每个元素代入下列公式：

[0064]
$$\begin{aligned}
&\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)\\
&=\left(x_8^{29}x_9^{29}x_{10}^{29}x_{11}^{29}\right)\\
&=\left(x_4^{30}x_5^{30}x_6^{30}x_7^{30}\right)\\
&=\left(x_0^{31}x_1^{31}x_2^{31}x_3^{31}\right)\\
&=\left(x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}\right)\oplus\tau\left(\left(x_0^{32}x_1^{32}x_2^{32}x_3^{32}\right)\oplus\left(x_4^{32}x_5^{32}x_6^{32}x_7^{32}\right)\oplus\left(x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)\\
&\oplus\left(\tau\left(\left(x_0^{32}x_1^{32}x_2^{32}x_3^{32}\right)\oplus\left(x_4^{32}x_5^{32}x_6^{32}x_7^{32}\right)\oplus\left(x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<2\right)\\
&\oplus\left(\tau\left(\left(x_0^{32}x_1^{32}x_2^{32}x_3^{32}\right)\oplus\left(x_4^{32}x_5^{32}x_6^{32}x_7^{32}\right)\oplus\left(x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<10\right)\\
&\oplus\left(\tau\left(\left(x_0^{32}x_1^{32}x_2^{32}x_3^{32}\right)\oplus\left(x_4^{32}x_5^{32}x_6^{32}x_7^{32}\right)\oplus\left(x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<18\right)\\
&\oplus\left(\tau\left(\left(x_0^{32}x_1^{32}x_2^{32}x_3^{32}\right)\oplus\left(x_4^{32}x_5^{32}x_6^{32}x_7^{32}\right)\oplus\left(x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}\right)\oplus sk\right)<<<24\right)
\end{aligned}$$

[0065] 其中，$C^*=x_{12}^{32}x_{13}^{32}x_{14}^{32}x_{15}^{32}x_8^{32}x_9^{32}x_{10}^{32}x_{11}^{32}x_4^{32}x_5^{32}x_6^{32}x_7^{32}x_0^{32}x_1^{32}x_2^{32}x_3^{32}$，以此计算出每个子密钥候选值与所有错误密文对应的中间状态值$\left(x_{12}^{28}x_{13}^{28}x_{14}^{28}x_{15}^{28}\right)$，并按照子密钥候选值分类，使得每个子密钥候选值均对应一组中间状态值；然后，利用统计学的方法，求出每一组中间状态值中的x_{12}^{28}的汉明重量，即x_{12}^{28}二进制表示的1的个数；最后，挑选出汉明重量值最小的一组中间状态值，其对应的子密钥候选值即为正确的子密钥的部分比特。

[0066] 上述的密钥候选值，一般是指在故障传播的过程中，受故障影响的子密钥的部分或者全部比特所组成的值。利用所述步骤4，可以对密钥候选值作穷举，并有效缩小搜索空间。本例子中，所求的子密钥为k_{31}的全部比特，因此密钥候选值的长度为32比特，即仅需尝试2^{32}次，即可求出子密钥所有字节，而其余的k_{30}、k_{29}和k_{28}，可以通过步骤5，改变故障的导入轮数完成。

[0067] 针对步骤5，当k_i的所有比特均被求出之后，重复步骤1至步骤4，将故障的导入的轮数提前一轮，改变故障的导入位置，继续求k_{i-1}的所有比特位，以此类推，直到按照顺序求出子密钥$\{k_{31},k_{30},k_{29},k_{28}\}$的所有比特位后，根据SMS4的密钥扩展，逆推出主密钥K。其公式如下，其中，T′为上述合成置换T的逆：

[0068]
$$\begin{cases}
k_i=k_{i+4}\oplus T'\left(k_{i+1}\oplus k_{i+2}\oplus k_{i+3}\oplus ck_i\right),i\in\{31,30,\cdots,0\}\\
MK_i=k_i\oplus FK_i,i\in\{3,2,1,0\}\\
K=MK_0MK_1MK_2MK_3
\end{cases}$$

[0069] 针对上述步骤，实验环境如图4所示，其中，封装有SMS4算法的设备1用来处理输入的消息；设备2是一台计算机，用来产生用于被设备1加密的明文消息以及采集、分析设备1的输出结果；产生故障的设备3用来改变实验的运行环境，使得设备2运行时产生故障，实现故障的导入功能，从而产生错误的输出。

[0070] 利用上述的分析方法,本发明在Intel® Core™ i5CPU 1.4GHz 4GB内存的计算机上,使用IntelliJ IDEA CE开发工具和Java语言编程来模拟故障的导入和消息的处理过程,重复执行2000次,实验结果表明上述检测方法准确无误。该方法为评估SMS4算法的安全性提供了充分的理论依据,而且此方法操作简单,计算结果准确。

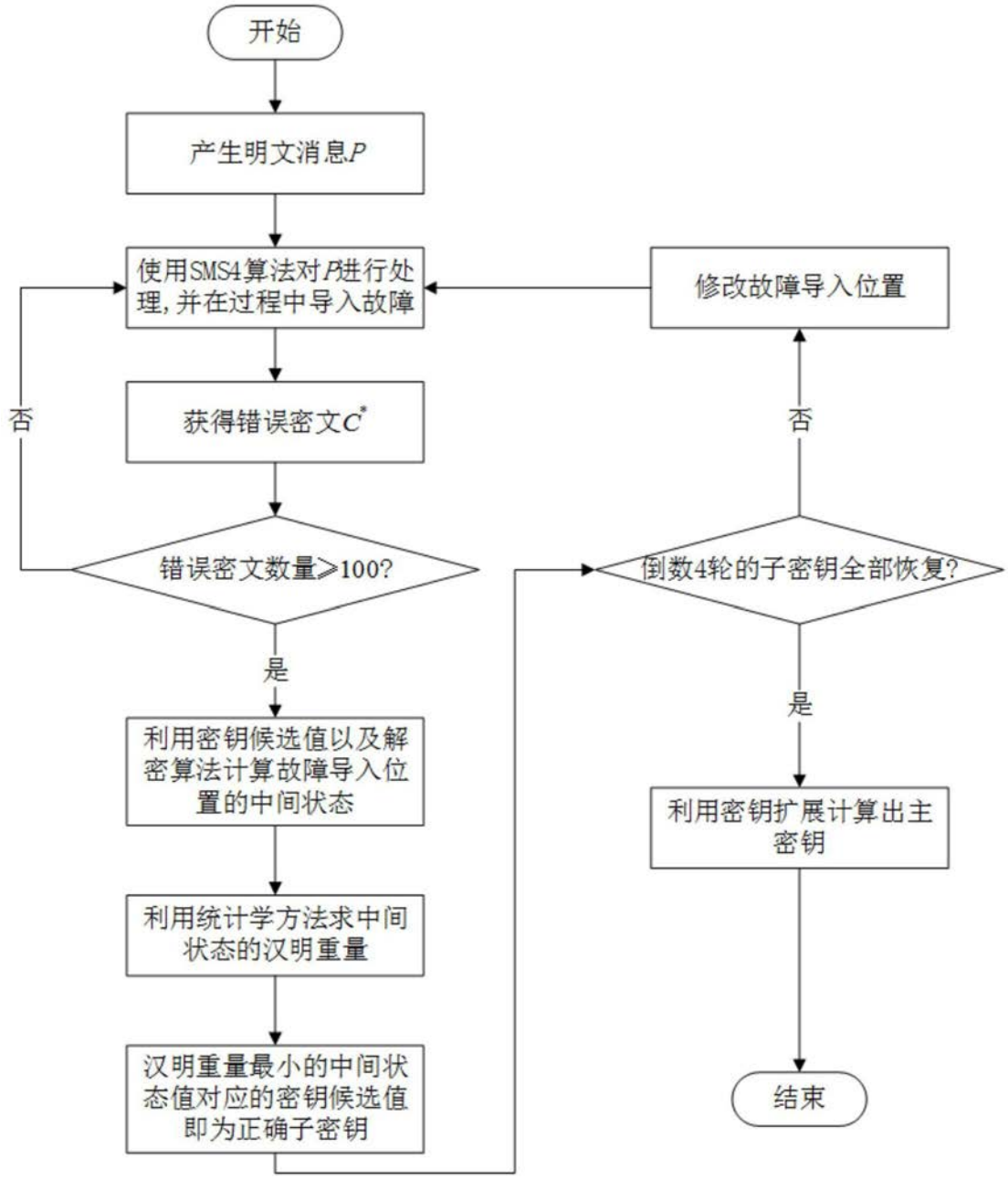

开始
产生明文消息P
使用SMS4算法对P进行处理，并在过程中导入故障
获得错误密文C*
错误密文数量≥100?
否
是
利用密钥候选值以及解密算法计算故障导入位置的中间状态
利用统计学方法求中间状态的汉明重量
汉明重量最小的中间状态值对应的密钥候选值即为正确子密钥
倒数4轮的子密钥全部恢复?
否
修改故障导入位置
是
利用密钥扩展计算出主密钥
结束

图1

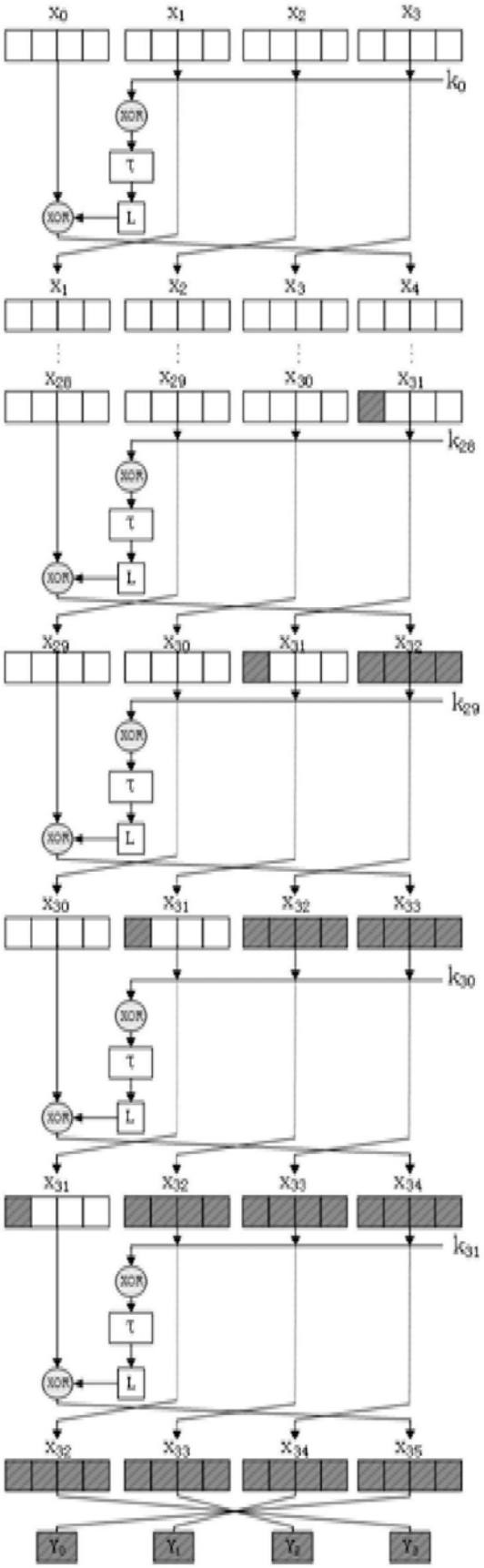

X0
X1
X2
X3
k0
XOR
τ
XOR
L
X1
X2
X3
X4
X28
X29
X30
X31
k28
XOR
τ
XOR
L
X29
X30
X31
X32
k29
XOR
τ
XOR
L
X30
X31
X32
X33
k30
XOR
τ
XOR
L
X31
X32
X33
X34
k31
XOR
τ
XOR
L
X32
X33
X34
X35
Y0
Y1
Y2
Y3

图2

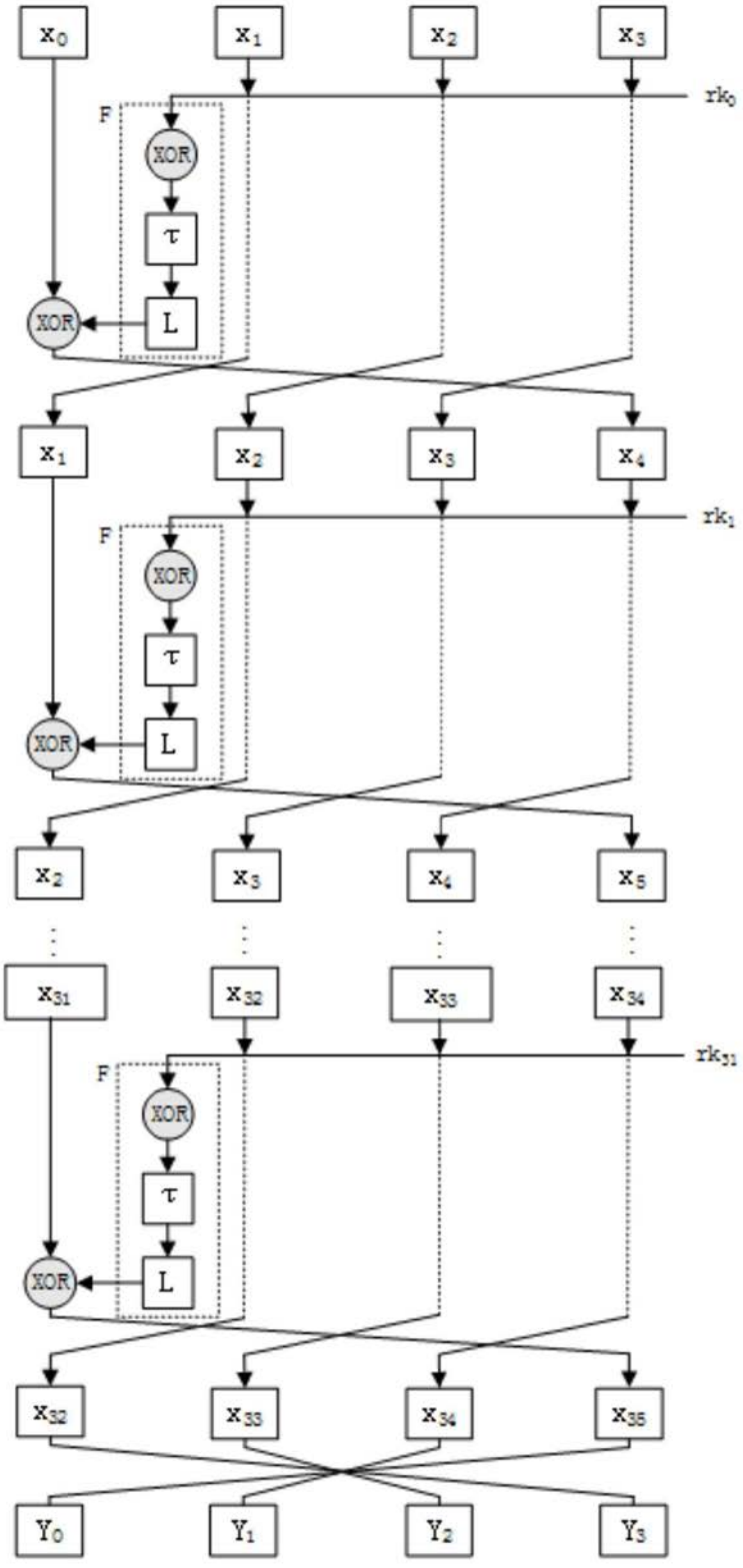

x_0
x_1
x_2
x_3
rk_0
F
XOR
τ
L
XOR
x_1
x_2
x_3
x_4
rk_1
F
XOR
τ
L
XOR
x_2
x_3
x_4
x_5
x_{31}
x_{32}
x_{33}
x_{34}
rk_{31}
F
XOR
τ
L
XOR
x_{32}
x_{33}
x_{34}
x_{35}
Y_0
Y_1
Y_2
Y_3

图3

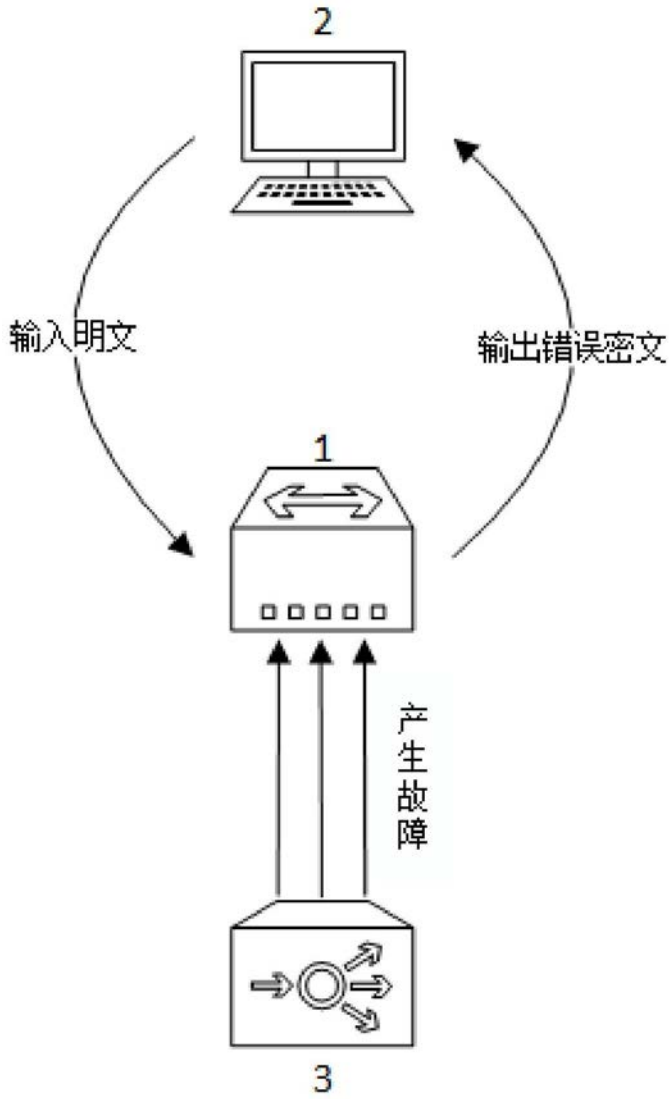


图4