# 公开追踪和CCA2安全的叛逆者追踪方案

王青龙[1,2] 张少博[1]
(长安大学信息工程学院 西安 710064)[1]
(北京交通大学通信与信息系统北京市重点实验室 北京 100044)[2]

**摘 要** 提出一种新的k-resilient公钥叛逆者追踪方案。方案的追踪方式为公开黑盒追踪。假设DDH问题为困难问题,则方案能被证明是自适应选择密文攻击安全的,并且在撤销不超过$k$个叛逆者条件下仍然是自适应选择密文攻击安全的。与同类方案相比,该方案没有使用一次性消息认证码假设,并且有效降低了追踪时的计算复杂性。另外,方案满足非对称性。

**关键词** 叛逆者追踪,广播加密,消息认证码,可证明安全性,非对称性
**中图法分类号** TP309 **文献标识码** A

## Public Traceable Traitor Tracing Scheme Secure against CCA2

WANG Qing-long[1,2] ZHANG Shao-bo[1]
(School of Information Engineering Chang'an University, Xi'an 710064, China)[1]
(Key Laboratory of Communication and Information System, Beijing Jiaotong University, Beijing 100044, China)

**Abstract** A new k-resilient public-key traitor tracing scheme was proposed. A traitor in this scheme can be traced by the way of public black-box tracing mean. Meanwhile, our scheme is provably secure against adaptive chosen ciphertext attack assume that DDH problem is difficulty. Our scheme remains CCA2-secure when not more than $k$ traitors are revoked. Compared with similar schemes, our scheme does not use one-time message authentication code assumption and reduces the tracing complexity greatly. Further, our scheme satisfies asymmetry.

**Keywords** Traitor tracing, Broadcast encryption, MAC, Provable security, Asymmetry

## 1 引言

叛逆者追踪研究内容包括抗共谋、非对称、撤销性、追踪性、安全性等多个方面,一个追踪方案一般只是解决其中一个或多个方面,很难有一个方案可以解决所有问题。本文主要针对安全性和追踪性,针对其它方面的研究方案见文献[1-9]。文献[10]首次提出了可公开追踪的叛逆者追踪方案,但方案不具有撤销性,安全性也仅对被动攻击者是语义安全的。文献[11]指出文献[10]的公开追踪性只对两个用户的系统有效,并提出一个完全公开追踪的方案。文献[11]同样不具有撤销性,其安全性针对被动型敌手是RCCA(Replayable chosen ciphertext attack)安全的。文献[12-16]提出了几个能够抵抗自适应选择密文攻击(adaptive chosen cipher text attack,简称CCA2)的k-resilient公钥叛逆者追踪方案。文献[12,16]虽然满足非对称性,但是不满足撤销性,两个方案的追踪方式均为非黑盒追踪,也就是需要打开盗版解码器才能够实施追踪过程,其追踪复杂性为$O((C_N^k)k^2)$($N$为系统中用户数量,$k$为共谋门限);文献[13]方案在没有撤销叛逆者时是CCA2安全的,一旦有一个叛逆者被撤销,由于该撤销者可以对广播的密文数据进行锻造(malleability),其不再满足CCA2安全性[14],因此严格来说该方案也不具有撤销性;文献[14]借助于更强的假设(除了DDH,decision diffie-hellman假设,还有一次性MAC-message authentication code假设)提出一种撤销情况下CCA2安全的追踪方案,其主要方法是在解密前后各有一次测试;文献[15]在文献[14]的基础上取消了一次性MAC假设,提出一种改进的CCA2安全的追踪方案,但是与其它方案不同的是,文献[15]中的解密算法没有测试步骤,因此对输入的密文无论合法还是非法,解密Oracle都会输出一个值,但文献[15]在原解密过程没有测试步骤的情况下,在其证明过程GAME3中增加了一个测试步骤$D_{2^{-1}}$是否合理并没有给出解释。文献[13-15]的追踪过程都属于非公开黑盒追踪,也就是只有DS才能够实施黑盒追踪,几个方案的追踪复杂性都是$O(C_N^k)$。

本文同样提出一种自适应选择密文攻击安全的k-resilient公钥叛逆者追踪方案,方案满足撤销性和非对称性。方案的追踪方式为公开黑盒追踪,即任何人都可以对收缴的盗版解码器进行追踪,并且追踪复杂性只有$O(N)$;同时,我们的方案在撤销情况下也是CCA2安全的,构造上借鉴了文献

到稿日期:2010-03-17 返修日期:2010-06-23 本文受国家自然科学基金(60773175),中央高校基本科研业务费专项基金(CHD2009JC146),长安大学基础研究支持计划专项基金资助。

王青龙(1970-),男,博士,讲师,主要研究方向为密码学与网络安全,E-mail:qlwang@chd.edu.cn;张少博(1974-),男,博士,讲师,主要研究方向为计算机网络服务质量和拥塞控制。

[14]前后分别进行一次测试的方法，但是没有使用一次性MAC假设。方案的安全性是建立在DDH问题为困难问题假设之上的。

## 2 方案组成

### 2.1 参数设置

$q$ 和 $p$ 为两个大素数，且 $q|p-1$。$g$ 是 $Z_p$ 上阶为 $q$ 的本原元。DS在 $Z_q$ 上选取 $f_1(x)=\sum_{i=0}^{k-1}a_i x^i$，$a,b,c_1,c_2,c_1',c_2'$，$b'$，$f_2(x)=\sum_{i=0}^{k-1}b_i x^i$。计算 $c=g^{\alpha_1}g^{k_2} \bmod p$，$d=g^{\alpha'_1}g^{k'_2} \bmod p$。设 $f_1'(x)=a^{-1}(f_1(x)-b)$，$f(x,y)=f_1'(x)+b'y$，DS将 $q$ 和 $p$ 以及公开钥 $e=(g^{f_1(x_0)},g^{ab'},x_0,(x_1,g^{f_1(x_1)}),\cdots,(x_k,g^{f_1(x_k)}),g^a,g^b,c,d,H)$ 以及用于追踪的信息 $T=((x_1,g^{f_2(x_1)}),\cdots,(x_k,g^{f_2(x_k)}))$ 予以公开，其中 $H$ 为一抗碰撞Hash算法，秘密保留 $f_1(x)$，$f_2(x)$ 和 $a,b,c_1,c_2,c_1',c_2',b'$。如无明确指出，本方案的算术运算都是模 $p$ 运算。

### 2.2 注册过程

当用户 $u$ 欲加入系统时，DS选取一个没有使用过的 $i\in_R Z_q\backslash\{x_0,x_1,\cdots,x_k\}$，由 $(i,f_1(i)),(x_1,f_2(x_1))\cdots(x_k,f_2(x_k))$ 这 $k+1$ 个点可得一 $k$ 次多项式 $f_i(x)$，DS计算并记录 $text_i=i||u||g^{f_i(x_0)}$（如 $f_i(x_0)$ 与已有记录中的对应项相等，则重新选取 $i$）。DS再计算 $f_1'(i)=a^{-1}(f_1(i)-b) \bmod q$，得到多项式 $f(i,y)=f_1'(i)+b'y \bmod q$。用户秘密选取 $\alpha_i\in_R Z_q^*$，通过使用OPE[16]协议，得到 $f(i,\alpha_i)$。协议完成后DS发送 $(i,c_1,c_2,c_1',c_2')$ 给用户，用户 $u$ 得到自己的解密钥 $d_u=(i,\alpha_i,c_1,c_2,c_1',c_2',f(i,\alpha_i))$。协议完成后DS将记录 $text_i$ 予以公开。

### 2.3 加密广播

设 $m\in Z_q$ 为待广播信息，DS选取 $s\in_R Z_q^*$ 作为会话密钥，生成对应的广播分组 $(sg^{rf_1(x_0)},g^{rab'},x_0,(x_1,g^{rf_1(x_1)}),\cdots,(x_k,g^{rf_1(x_k)}),g^{ra},g^{rb},v,v',s\oplus m)=(Head,s\oplus m)$，其中 $r\in_R Z_q$，$v=c^r d^{r\alpha}$，$v'=\nu^s$。$\alpha=H(sg^{rf_1(x_0)},g^{rab'},x_0,(x_1,g^{rf_1(x_1)})\cdots(x_k,g^{rf_1(x_k)}),g^{ra},g^{rb})$。

### 2.4 解密算法

用户收到广播的分组数据 $(Head,M)$ 后，先使用分组头 $Head=(S,F_0,x_0,(x_1,F_1),\cdots,(x_k,F_k),F_a,F_b,v,v')$ 验证等式 $v=F_a^{c_1+c_1'\alpha}F_b^{c_2+c_2'\alpha}$（测试1）是否成立，其中 $\alpha=H(S,F_0,x_0,(x_1,F_1),\cdots,(x_k,F_k),F_a,F_b)$。若不成立，则拒绝解密，输出⊥后停止；若成立，则执行解密算法如下：

Step1 计算

$F_a^{f(i,\alpha_i)}F_b/(F_0)^{\alpha_i}=g^{ra(f_1'(i)+b'\alpha_i)}g^{rb}/(F_0)^{\alpha_i}=g^{rf_1(i)}$；

Step2 由 $(i,g^{rf_1(i)}),(x_1,g^{rf_1(x_1)}),\cdots,(x_k,g^{rf_1(x_k)})$ 插值求得 $g^{rf_1(x_0)}$；

Step3 计算 $s=sg^{rf_1(x_0)}/g^{rf_1(x_0)}$；

Step4 验证 $v'=\nu^s$（测试2）是否成立。若成立，继续；若不成立，输出⊥后停止。

Step5 计算 $m=s\oplus M$，输出明文信息。

事实上，如果 $Head$ 为一合法密文分组头，则有

$v=F_a^{c_1+c_1'\alpha}F_b^{c_2+c_2'\alpha}=g^{ra(c_1+c_1'\alpha)}g^{rb(c_2+c_2'\alpha)}=(g^{rac_1}g^{rbc_2})(g^{raac_1'}g^{rbac_2'})=c^r d^{r\alpha}$ 以及 $v'=\nu^s$

即合法密文分组头都可以通过验证过程，通不过验证的分组头都被认为是非法密文分组头。

### 2.5 追踪算法

由下述定理2可知，当参与共谋的叛逆者数量不超过 $k$ 时，共谋者不能得到另一个不同的解密钥。因此，假设缴获的盗版解码器中包含的解密钥为某个叛逆者（不妨设为 $u$）自己的解密钥 $d_u=(i,\alpha_i,c_1,c_2,c_1',c_2',f(i,\alpha_i))$。

利用公开的信息 $T$ 和 $text$ 进行黑盒追踪过程，对 $t$ 从1到 $N$，DS分别往盗版解码器中输入分组：

$(sg^{rf_t(x_0)},g^{rab'},x_0,(x_1,g^{rf_2(x_1)}),\cdots,(x_k,g^{rf_2(x_k)}),g^{ra},g^{rb},v,v',s\oplus m)=(S,F_0,x_0,(x_1,F_1),\cdots,(x_k,F_k),F_a,F_b,v,v',M)$

其中各参数和计算方法与上述加密算法相同。如盗版解码器输出与 $m$ 相同，则对应的 $t$ 为叛逆者。

事实上，当 $t=i$ 时，输入的分组数据为 $(sg^{rf_i(x_0)},g^{rab'},x_0,(x_1,g^{rf_2(x_1)}),\cdots,(x_k,g^{rf_2(x_k)}),g^{ra},g^{rb},v,v',s\oplus m)$，盗版解码器按照正常解密算法计算：

Step1 验证 $v=F_a^{c_1+c_1'\alpha}F_b^{c_2+c_2'\alpha}$ 成立；

Step2 计算 $F_a^{f(i,\alpha_i)}F_b/(F_0)^{\alpha_i}=g^{ra(f_1'(i)+b'\alpha_i)}g^{rb}/(F_0)^{\alpha_i}=g^{r(af_1'(i)+b)}g^{rab'\alpha_i}/(g^{rab'})^{\alpha_i}=g^{rf_1(i)}$；

Step3 由 $(i,g^{rf_1(i)}),\cdots,(x_k,g^{rf_2(x_k)})$ 插值求得 $g^{rf_i(x_0)}$；

Step4 计算 $s=sg^{rf_i(x_0)}/g^{rf_i(x_0)}$；

Step5 验证 $v'=\nu^s$ 成立；

Step6 输出信息 $m=s\oplus M$；

即输入与叛逆者对应的分组数据后，盗版解码器能够输出正确的明文信息。如果输入与其他用户 $t\neq i$ 对应的分组数据，则此时解密过程的Step4计算值为 $sg^{rf_t(x_0)}/g^{rf_i(x_0)}\neq s$，因此其不能通过Step5的验证，盗版解码器输出⊥。

由上可知，追踪一个叛逆者最多只需要 $N$ 次输入，因此方案的追踪复杂性为 $O(N)$。又由于追踪的信息都是公开的，任何人都可以用来进行追踪，因此方案满足公开追踪性。

### 2.6 撤销算法

假设 $u$ 为叛逆者，其解密钥为 $d_u=(i,\alpha_i,c_1,c_2,c_1',c_2',f(i,\alpha_i))$，则DS将公钥中包含的 $((x_1,g^{f_1(x_1)}),\cdots,(x_k,g^{f_1(x_k)}))$ 项中的没有被替换过的某项（例如 $(x_1,g^{f_1(x_1)})$）用 $(i,g^{f_1(i)})$ 替换即可。这样，在解密时得不到插值计算所需的 $k+1$ 个点，所以 $u$ 无法由分组头获得正确的会话，也就无法得到广播的明文信息。

当不超过 $k$ 个数量的叛逆者被撤销后，因为撤销者对新广播的密文分组进行成功解密的概率是可忽略的，所以撤销后的叛逆者将密文分组锻造为另一个可以通过测试2验证的不同密文分组的概率是可忽略的，因此方案在撤销叛逆者前后安全性是相同的。

## 3 安全性分析

自适应选择密文攻击模型

自适应选择密文攻击包括三个阶段：

密钥生成 $(e,d_i)\leftarrow G(1^k)$。

**定理1** 假设DDH问题为困难问题，则上述叛逆者追踪方案使用的加密算法对于选择密文攻击是安全的。注：以下的证明过程我们忽略测试2，因为测试2的存在只能增强而不会降低方案的安全性，因此在测试1下证明是CCA2安全

的方案，加上测试 2 后仍是 CCA2 安全的。

证明：设四元组$(g_1, g_2, u_1, u_2)$为待判断 DDH 问题。又设敌手 $A$ 在进行选择密文攻击时使用的算法为 $A_1$ 和 $A_2$，其中 $A_1$ 的输入参数为对应的公开钥，输出为两个会话密钥 $s_0$，$s_1$。DS 随机选取一个会话密钥 $s_\beta, \beta \in \{0,1\}$，生成一个分组头并输入 $A_2$。$A$ 询问足够多的选择密文后，从 $A_2$ 输出 $s_\beta$。假设 $A$ 能够以不可忽略优势 $Adv_A^{cca}(\lambda) = \Pr[(\beta^* = \beta) - 1/2] = \varepsilon$ 攻破该追踪方案，则 DS 可构造一个模拟真实攻击的试验，使得经过足够多的试验后 DS 可以同样不可忽略概率区分 DDH 问题。试验由一个模拟器 $S$ 和敌手 $A$ 组成，$S$ 包括一个加密 oracle 和一个解密 oracle。试验过程为输入待判断四元组给 $S$，利用此四元组 $S$ 构造一个追踪方案并和 $A$ 一起完成对此方案的选择密文攻击，攻击完成后 $S$ 输出一个试验结果。

具体试验过程为（与追踪有关部分此处忽略）：

1. 输入待判断四元组$(g_1, g_2, u_1, u_2)$给 $S$。

2. 密钥建立：$S$ 在 $Z_q$ 上选择 $f_1(x) = \sum_{i=0}^{k} a_i x^i$，$f_1'(x) = \sum_{t=0}^{k} r_t x^t$ 和 $\omega, j, b', \alpha_j, c_1, c_2, c_1', c_2'$，令 $g = g_1$，$g_2 = g_1^a = g^a$，$f(x) = f_1(x) + a\omega \bmod q$，$f_1'(j) = a^{-1}(f_1(j) - b) \bmod q$，$g^b = g^{f_1(j)} / g_2^{f_1'(j)}$，$f(x,y) = f_1'(x) + b'y + \omega$，（此处不知道 $b$，但是可以得到 $g^b$）。计算 $c = g^{\alpha_2} g^{k_1}$，$d = g^{\alpha_2'} g^{k_1'}$。$S$ 生成公开钥 $e = (g^{f_1(x_0)} g_2^{\omega}, g_2^{b'}, x_0, (x_1, g^{f_1(x_1)} g_2^{\omega}), \cdots, (x_k, g^{f_1(x_k)} g_2^{\omega}), g^a, g^b, c, d, H)$。

3. $S$ 发送 $e$ 给 $A_1$，$A_1$ 返回两个挑战会话密钥 $s_0, s_1$。

4. $S$ 任选 $s_\beta, \beta \in \{0,1\}$，利用待判断四元组生成质询分组头 $Head_\beta = (s_\beta u_1^{f_1(x_0)} u_2^{\omega}, u_2^{b'}, x_0, (x_1, u_1^{f_1(x_1)} u_2^{\omega}), \cdots, (x_k, u_1^{f_1(x_k)} u_2^{\omega}), F_a, F_b, v, v')$（因为测试 2 不予以考虑，此处的 $v'$ 在证明中将不起作用，只是为了与前面分组头形式上保持一致），其中 $F_a = u_2$，$F_b = u_1^{f_1(j)} / u_2^{f_1'(j)}$，$v' = v^\beta$，$v = (u_1^b)^{c_1 + c_1'\alpha} u_2^{c_2 + c_2'\alpha}$（此处 $u_1^b$ 可以通过 $g^b$ 求得，即有 $u_1^b = u_1^{f_1(j)} / u_2^{f_1'(j)}$）。$\alpha = H(s_\beta u_1^{f_1(x_0)} u_2^{\omega}, u_2^{b'}, x_0, (x_1, u_1^{f_1(x_1)} u_2^{\omega}), \cdots, (x_k, u_1^{f_1(x_k)} u_2^{\omega}), F_a, F_b)$。将 $Head_\beta$ 输入 $A_2$。

5. 敌手 $A$ 收到质询的分组头 $Head_\beta$ 后，继续选择询问加密 Oracle 和解密 Oracle。如果 $A$ 询问加密 Oracle，则加密 Oracle 根据公开钥 $e$ 生成相应密文；如果 $A$ 询问解密 Oracle 的密文分组头为 $Head = (S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b, v, v')$（不能询问 $Head_\beta$），解密 Oracle 首先验证 $v = F_a^{c_2 + c_2'\alpha} (F_b)^{c_1 + c_1'\alpha}$ 是否成立，不成立则输出 $\perp$ 停止，成立则按照上述解密步骤求得一个会话密钥 $s$ 送给 $A$。经过多次询问后，$A_2$ 输出 $s_\beta$。

6. $S$ 输出 1 当且仅当 $\beta' = \beta$。

当$(g_1, g_2, u_1, u_2)$来自于 $D$ 时，由引理 1 知，$A$ 能够以不可忽略优势攻破该追踪方案，则 $S$ 能够以同样的优势 $\varepsilon$ 输出 1；若来自于 $R$，由引理 2，此时 $A$ 没有优势输出正确的 bit 值 $\beta'$。因此，重复足够多次试验后，如果有 $|\Pr(S=1) - \Pr(S=0)| \approx \varepsilon$，则 DS 判断$(g_1, g_2, u_1, u_2)$来自于 $D$；若有 $|\Pr(S=1) - \Pr(S=0)| \approx 0$，则 DS 判断$(g_1, g_2, u_1, u_2)$来自于 $R$。

**引理 1** 当四元组$(g_1, g_2, u_1, u_2)$来自于 $D$ 时，敌手 $A$ 可以不可忽略优势 $\varepsilon$ 输出满足 $\beta' = \beta$ 的 bit 值 $\beta'$。

证明：根据 $S$ 生成的公开钥中用来隐藏会话密钥的项 $l = g^{f_1(x_0)} g_2^{\omega}$ 可知有 $\log_g^l = f_1(x_0) + a\omega$。因为四元组$(g_1, g_2, u_1, u_2)$来自于 $D$，即满足 $r = \log_{g_1}^{u_1} = \log_{g_2}^{u_2}$，则加密 Oracle 输出的密文分组头 $Head = (S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b, v, v')$ 满足，$F_i = u_1^{f_1(x_1)} u_2^{\omega} = g^{rf(x_1)}$，$1 \leqslant i \leqslant k$，$F_a = g_2^r = g^{ar}$，$F_b = u_1^{f_1(j)} / u_2^{f_1'(j)} = g^{br}$，$F_a^{c_2} F_b^{c_1} = c^r$，$F_a^{c_2'} F_b^{c_1'} = d^r$，$S = s_\beta u_1^{f_1(x_0)} u_2^{\omega} = s_\beta g^{rf(x_0)}$，$v = c^r d^{r\alpha}$，$v' = v^\beta$，其中 $\alpha = H(S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b)$，即 $Head$ 为一合法分组头。敌手 $A$ 对此追踪方案的选择密文攻击过程与对真实追踪方案的选择密文攻击过程完全相同，假设此时 $A$ 可以不可忽略概率 $\varepsilon$ 破解此追踪方案，也就是 $A$ 可以不可忽略概率 $\varepsilon$ 输出满足 $\beta' = \beta$ 的 bit 值 $\beta'$。

**引理 2** 假设解密 Oracle 只对合法密文进行解密（引理 3），则当$(g_1, g_2, u_1, u_2)$来自于 $R$ 时，敌手 $A$ 没有优势输出满足 $\beta' = \beta$ 的 bit 值 $\beta'$。

证明：当$(g_1, g_2, u_1, u_2)$来自于 $R$ 时，也就有 $t_1 = \log_{g_1}^{u_1} \neq \log_{g_2}^{u_2} = t_2$。由 $S$ 生成的公开钥中用来隐藏会话密钥的项 $l = g^{f_1(x_0)} g_2^{\omega}$ 可知有

$$\log_g^l = f_1(x_0) + a\omega \tag{1}$$

对于敌手的加密请求，加密 Oracle 利用公开钥生成的合法密文分组头中用来隐藏会话密钥的项为 $g^{rf_1(x_0)} g_2^{r\omega}$，由此可得与式(1)相关的某个方程 $r\log_g^l = rf_1(x_0) + ra\omega$。但是 $S$ 输出的挑战密文分组头中用来隐藏 $s_\beta$ 的项为 $t = g_1^{t_1 f_1(x_0)} g_2^{t_2 \omega}$，此时有

$$\log_g^t = t_1 f_1(x_0) + t_2 a\omega \tag{2}$$

显然此式与式(1)是独立的，即敌手不能由 $l$ 得到 $t$ 的任何信息。

对于解密 Oracle，因为只解密合法密文，所以由被解密密文分组头中用来隐藏会话密钥 $s$ 的项得到的同样是与式(1)相关的某个方程 $r\log_g^l = rf_1(x_0) + ra\omega$，敌手并不会从此式中得到比式(1)更多的信息。

由上可知，$A$ 不能从挑战密文中得到任何用来隐藏 $s_\beta$ 的项 $t$ 的信息，也就无法得到任何有关 $s_\beta$ 的信息。即敌手此时没有优势输出满足 $\beta' = \beta$ 的 bit 值 $\beta'$。

**引理 3** 当$(g_1, g_2, u_1, u_2)$来自于 $R$ 时，非法密文被解密 Oracle 解密的概率是可忽略的。

证明：由公开钥中的 $c, d$ 敌手可得方程式

$$\log_g^c = ac_2 + bc_1 \tag{3}$$

$$\log_g^d = ac_2' + bc_1' \tag{4}$$

由加密 Oracle 输出的密文分组头 $Head = (S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b, v, v')$ 可得

$$\log_g^v = t_1 bc_1 + t_2 ac_2 + at_1 bc_1' + at_2 ac_2' \tag{5}$$

假设敌手 $A$ 输入一个非法密文（即 $t_1' = \log_g^{u_1'} \neq \log_g^{u_2'} = t_2'$）分组头 $Head' = (S', F_0', x_0, (x_1, F_1'), \cdots, (x_k, F_k'), F_a', F_b', v_0, v_0')$ 给解密 Oracle，考虑三种情况：

CASE1 $\langle S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b \rangle = \langle S', F_0', x_0, (x_1, F_1'), \cdots, (x_k, F_k'), F_a', F_b' \rangle$

因为 $Head \neq Head'$，所以有 $v \neq v_0$。显然非法密文不能通过验证，Oracle 将输出 $\perp$ 停止。

CASE2 $\langle S, F_0, x_0, (x_1, F_1), \cdots, (x_k, F_k), F_a, F_b \rangle \neq \langle S', F_0', x_0, (x_1, F_1'), \cdots, (x_k, F_k'), F_a', F_b' \rangle$，$v = v_0$

这意味着敌手找到一对碰撞值，与 $H(.)$ 为抗碰撞 Hash 函数矛盾。

CASE3 $\langle S,F_0,x_0,(x_1,F_1),\cdots,(x_k,F_k),F_a,F_b\rangle\neq\langle S',F_0',x_0,(x_1,F_1'),\cdots,(x_k,F_k'),F_a',F_b'\rangle,v\neq v_0$

若此密文能够通过解密 Oracle 验证，则一定有

$$\log_g^{v'}=bt_1'c_1+at_2'c_2+\alpha'bt_1'c_1'+\alpha'at_2'c_2' \tag{6}$$

此时由式(3)－式(6)可得

$$det\begin{pmatrix} a & b & 0 & 0\\ 0 & 0 & a & b\\ at_2 & bt_1 & \alpha at_2 & \alpha bt_1\\ at_2' & bt_1' & \alpha at_2' & \alpha bt_1' \end{pmatrix}=a^2b^2(\alpha-\alpha')(t_1'-t_2')$$

$$(t_1-t_2) \tag{7}$$

因为 $\alpha\neq\alpha',t_1'\neq t_2',t_1\neq t_2$，所以式(7)是不等于 0 的。由式(3)－式(5)敌手可知，$(c_2,c_1,c_2',c_1')$ 位于某个确定的直线 $L$ 上，由式(7)又可知满足式(3)－式(6)的 $(c_2,c_1,c_2',c_1')$ 只有一个。因此，第一次敌手输入一个非法密文被解密 Oracle 接收的概率为 $1/q$。第一次被解密 Oracle 拒绝后，敌手可以排除掉此点，则第二次输入的非法密文被接收的概率为 $1/q-1$。同理，第 $i$ 次输入的非法密文被解密 Oracle 接收的概率为 $1/q-i$。这样，经过 $n$ 次询问后，解密 Oracle 接收一个非法密文的总概率为 $\sum_{i=1}^{n}1/q-i+1\leqslant n/q-n$，因为 $q>>n$，所以此概率为可忽略概率。也就是，解密 Oracle 能够拒绝几乎全部输入的非法密文。

**定理 2** $k$ 个叛逆者通过共谋得到另一个不同解密钥的计算复杂性难度相等于求解以素数阶元为底的离散对数困难问题。

证明：假设 $k$ 个叛逆者能够通过共谋得到另一个不同的解密钥，设其使用的算法为 $A$，则 DS 可以 $A$ 为子程序求解有限域上以素数阶元为底的离散对数困难问题。

设待求离散对数问题为 $y=g^c \bmod p$，其中 $g$ 的阶为素数 $q$。DS 在 $Z_q^*$ 上随机选取 $a,b,c_1,c_2,c'_1,c'_2,b'$ 和 $k$ 组数据 $(i_j,\alpha_j,d_j),1\leqslant j\leqslant k$，计算 $c=g^{\alpha_1}g^{k_2},d=g^{\alpha'_1}g^{k'_2},f_1'(i_j)=(d_j-b'\cdot\alpha_j) \bmod q,f_1(i_j)=af_1'(i_j)+b,1\leqslant j\leqslant k$。令 $g^{f_1(i_0)}=y$，由 $k+1$ 组 $(i_0,g^{f_1(i_0)}),(i_1,g^{f_1(i_1)}),\cdots,(i_k,g^{f_1(i_k)})$，利用 Lagrange 插值 DS 能够得到另外 $k$ 组数据 $(i_{k+1},g^{f_1(i_{k+1})}),\cdots,(i_{2k},g^{f_1(i_{2k})})$，其中 $g^{f_1(i_{k+j})}=\prod_{l=0}^{k}g^{\lambda_j f_1(i_l)}$，$1\leqslant j\leqslant k$。DS 得到公开钥 $e=(g^{f_1(i_0)},g^{ab'},i_0,(i_{k+1},g^{f_1(i_{k+1})}),\cdots,(i_{2k},g^{f_1(i_{2k})}),g^a,g^b,c,d,H)$。DS 发送 $k$ 组数据 $(i_j,\alpha_j,c_1,c_2,c'_1,c'_2,d_j),1\leqslant j\leqslant k$ 给叛逆者作为其解密钥（此处 DS 直接将解密钥发送给叛逆者而不是使用 OPE 协议对定理证明没有影响）。容易验证，叛逆者拥有的解密钥都能够正常解密利用公开钥 $e$ 生成的广播分组。由假设叛逆者使用算法 $A$ 可得到另一个不同的解密钥 $(i_l,\alpha_l,c_1,c_2,c_1',c_2',d_l)$，则 DS 能够得到 $f_1(i_l)=(d_l-b'\cdot\alpha_l) \bmod q$，最后，DS 利用 $k+1$ 组数据 $((l,f_1(l)),(i_1,f_1(i_1)),\cdots,(i_k,f_1(i_k))$ 通过插值求得 $f_1(i_0)=c$。

## 4 性能比较

为了对本方案的性能有个直观了解，表 1 给出了本方案和其它 CCA2 安全的方案的比较，结果表明，我们的方案整体上优于其它方案。

表 1 本章方案与同类方案的比较

| | 撤销性 | 追踪方式 | 非对称性 | 追踪复杂性 | 备注 |
|---|---|---|---|---|---|
| 文献[12] | ⊥ | 非黑盒追踪 | √ | | |
| 文献[13] | √ | 非公开黑盒追踪 | ⊥ | $O(C_N^k)$ | 撤销叛逆者后不再 CCA2 安全 |
| 文献[14] | √ | 非公开黑盒追踪 | ⊥ | $O(C_N^k)$ | 使用一次性 MAC 假设 |
| 文献[15] | √ | 非公开黑盒追踪 | ⊥ | $O(C_N^k)$ | |
| 文献[16] | ⊥ | 非黑盒追踪 | √ | | |
| 本方案 | √ | 公开黑盒追踪 | √ | O(N) | |

⊥：不满足对应属性；√：满足对应属性

**结束语** 本文首次提出一种可公开追踪的 CCA2 安全的公钥叛逆者追踪方案。该方案满足撤销性并且在撤销叛逆者后仍然满足 CCA2 安全性。基于 DDH 假设，我们的方案在标准模型下是安全的。与其它同类方案相比较，本方案没有使用一次性 MAC 假设，并且显著降低了追踪复杂性。目前存在的问题是现有满足 CCA2 安全的追踪方案都是 k-resilient 的方案，也就是方案存在共谋门限，设计完全抗共谋的 CCA2 安全的追踪方案还需进一步研究。

## 参 考 文 献


[1] Jin H X, Lotspiech J. Hybrid traitor tracing[C]// IEEE International Conference on multimedia and expo. New York: IEEE Press, 2006: 1329-1332

[2] Kazuto O, Go O. Goichiro H. Trade-off traitor tracing[C]// INDOCRYPT2007. Berlin: Springer press, 2007: 331-340

[3] Lv Xi-xiang, Yang Bo. Efficient Traitor tracing scheme based on NTRU[C] // PDCAT'05. San Jose: IEEE Computer Society Press, 2005: 120-124

[4] Pascal J, Alexandre K, Arjen K. Improving the boneh-franklin traitor tracing scheme[C] // PKC '09. Berlin / Heidelberg: Springer press, 2009: 88-104

[5] Michel A, Alexander W D, John M L, et al. Identity-based traitor tracing[C]// PCK2007. Berlin / Heidelberg: Springer press, 2007: 361-376

[6] Hongxia J, Jeffory B L, Mario B. Traitor tracing for subscription-based systems[C] // SECRYP2006. Portugal: TINSTICC press, 2006: 223-228

[7] Wu Yong-dong, Robert H D. On the security of fully collusion resistant traitor tracing schemes[EB/OL]. http://eprint. iacr. org/2008/450. pdf, 2008-10-24

[8] Aggelos K, Serdar P. Pirate evaluation: how to make most of your traitor keys[C]// Crypt2007. Berlin: Springer press, 2007: 448-465

[9] Duong H P, Reihaneh S N, Ngvu T. Generic construction of hybrid public key traitor tracing with full-public-traceability[C]// ICALP2006. Berlin: Springer Press, 2006: 264-275

[10] Aggelos K, Moti Y. Breaking and repairing asymmetric public-key traitor tracing[C]// DRM 2002. Heidelberg: Springer Press, 2003: 32-50

[11] Jin H X, Jeffery L, Mrod M. Efficient coalition detection in traitor tracing scheme[C] // International Federation for Information Processing. Boston: Springer Press, 2008: 365-380


（下转第 130 页）

这些检查单可以用于指导软件不同阶段不同性质的测试工作，提高了发现缺陷的效率。表5给出了根据需求缺陷模式制定的测试检查单示例。

根据缺陷模式所制定的测试检查单已成功地指导了多个测评项目的文档审查、代码审查及系统测试的用例设计，受到了测试人员的青睐和肯定。

### 3.2 软件开发过程的缺陷模式应用

缺陷模式可以指导开发人员在软件开发过程中避免引入类似缺陷，即开发人员可以在开发过程中考虑采用什么样的开发技术预防这些缺陷模式的再次出现，将软件缺陷模式转换为软件设计准则。如何将软件缺陷模式转化为软件设计准则呢？通过对软件缺陷的因果分析及软件缺陷预防措施的研究可知：软件设计准则在某种程度上就是软件缺陷的预防措施。为此，需对缺陷模式作进一步的分析，本文引入软件缺陷模式库的概念，将缺陷模式库定义为四元组，由软件缺陷模式属性的集合构成，属性包括软件缺陷模式的名称、软件缺陷引入的原因及后果、软件缺陷预防措施，缺陷模式库定义如下：

$$\psi=(N,I,O,P)$$

式中，$N$ 是软件缺陷模式的名称；$I$ 是该缺陷模式引入原因的集合；$O$ 是该缺陷模式引发后果的集合；$P$ 是该缺陷模式预防措施集合，均不能为空值。

建立软件缺陷模式库之后，分析缺陷模式预防措施就可以制定相应的软件设计准则。

表6和表7给出了缺陷模式库及相应设计准则的示例。

表6 需求缺陷模式库示例

| 缺陷模式 | 引入原因 | 引发后果 | 预防措施 |
|---|---|---|---|
| 缺少对异常情况的判断 | 1. 需求分析人员与用户沟通不充分；2. 需求分析人员对“应该做什么、不能做什么”考虑不全；3. 粗心导致漏掉某些需求 | 软件功能、无法满足用户要求，软件需求存在缺陷 | 与用户充分沟通，既要考虑软件要做什么，也应考虑软件不能做什么，应该明确输入、输出存在哪些异常情况，对其处理措施应详细说明 |

表7 需求设计准则示例

| 缺陷模式 | 需求阶段软件设计准则 |
|---|---|
| 缺少对异常情况的判断 | 准则：必须仔细分析软件运行过程中各种可能的异常情况，处理过程应考虑相应的保护措施。特别当采用现成软件时，必须仔细分析原有的异常保护措施对于现有的软件需求是否足够且完全使用 |

根据软件缺陷模式制定的设计准则已应用于某光学系统控制软件和某测试过程管理软件的开发中。软件开发人员认为，软件设计准则正是他们开展软件开发实践工作所迫切需要的，便于开发人员间吸取经验、避免已经发生缺陷的再发生，同时能更快地提高开发人员的熟练程度。

**结束语** 本文收集了大量软件缺陷数据，对整个软件生命周期中引入缺陷数据进行了深入研究，发现软件缺陷本身及其产生遵循一定的规律，同时结合模式的概念，提出软件缺陷模式的概念。软件缺陷模式是对软件缺陷的抽象描述，可以清晰描述某类具有共同特征的软件缺陷。之后分别对软件需求阶段、设计及编码阶段的缺陷模式进行了所属分类划分，同时整理归纳了相应的缺陷模式。最后，从软件开发和测试两个方面阐述了如何应用缺陷模式，将缺陷模式的预防措施转换为设计准则指导软件开发，避免相似缺陷的再发生；同时也可以根据缺陷模式制定测试检查单，更快更准确地识别相似缺陷并改正，为如何利用缺陷数据提高软件可靠性提供了思路。

后续工作一方面应随着软件缺陷数据的不断积累从不同类型软件、不同编程语言等角度扩充并完善缺陷模式。另一方面需加强基于缺陷模式自动测试和验证系统的研究，切实有效地指导软件开发和测试工作。

## 参考文献


[1] 阮廉，陆民燕，韩峰岩. 装备软件质量和可靠性管理[M]. 北京：国防工业出版社，2006

[2] 韩卫岗，周红建，赵禄丰. 软件缺陷信息分析研究[J]. 计算机工程与设计，2008，7：3381-3383，3447

[3] 聂林波，刘孟仁. 软件缺陷分类的研究[J]. 计算机应用研究，2004，7：84-86，98

[4] IEEE Std 729-1983. Standard Glossary of Software Engineering Terminology[S]. IEEE，1990

[5] Paulk M C. Capability Maturity-model SM for software [R]. Pittsburgh，Pennsylvania：Carnegie Mellon University，1993

[6] IEEE 982. 1-2005 Standard Dictionary of Measures of the Software Aspects of Dependability[S]. IEEE，2005

[7] 石柱，何新贵，武庄. 软件可靠性及其评估[J]. 计算机应用，2000，20(11)：1-5

[8] Frederieks M，Basili V. Using Defect Tracking and Analysis to Improve Software Quality[J]. IBM Journal of Research and Development，1998，19(10)：23-26

[9] 唐为明. 嵌入式软件缺陷模式知识库研究[D]. 北京：北京航空航天大学，2007

[10] Allen，Eric. Bug Patterns in Java [M]. Springer-Verlag：New York Inc，2005

[11] 宫云战，赵瑞莲，等. 软件测试教程[M]. 北京：机械工业出版社，2008

[12] 熊节. 模式的乐趣[M]. 北京：清华大学出版社，2003

[13] 刘海，郝克刚. 软件缺陷原因分析方法[J]. 计算机科学，2009，36(1)：242-243，251


(上接第109页)


[12] Hervé C，Duong H P，Vid P. Public traceability in traitor tracing scheme[C]//EUROCRYPT 2005. Berlin：Springer Press，2005：542-558

[13] Tzeng W G，Tzeng Z J. A public-key traitor tracing scheme with revocation using dynamic shares[J]. Designs，Codes and Cryptography，2005，35(1)：47-61

[14] Yevgeniy D，Nelly F. Public key trace and revoke scheme secure against adaptive chosen ciphertext attack[C]//PKC2002. Berlin：Springer Press，2003：100-115

[15] Kim C H，Hwang Y H，Lee P J. An efficient public key trace and revoke scheme secure against adaptive chosen ciphertext attack [C]//ASIACRYPT2003. Berlin：Springer Press，2003：359-373

[16] Kim C H，Hwang Y H，Lee P J. TTS without revocation capability secure against CCA2[C]//ACISP2004. Berlin：Springer Press，2004：36-49

[17] Naor M，Pinkas B. Oblivious transfer and polynomial evaluation [C]//Proc. of STOC'99. Atlanta：ACM Press，1999：245-254