软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2013,24(2):266−278 [doi: 10.3724/SP.J.1001.2013.04324]

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

标准模型下可证明安全的入侵容忍公钥加密方案*

于 佳[1,2], 程相国[1], 李发根[3,4], 潘振宽[1], 孔凡玉[5,6], 郝 蓉[1]

[1](青岛大学 信息工程学院,山东 青岛 266071)

[2](信息安全国家重点实验室(中国科学院 信息工程研究所),北京 100093)

[3](电子科技大学 计算机科学与工程学院,四川 成都 610054)

[4](Faculty of Mathematics, Kyushu University, Fukuoka 819-0395, Japan)

[5](山东大学 网络信息安全研究所,山东 济南 250100)

[6](密码技术与信息安全教育部重点实验室(山东大学),山东 济南 250100)

通讯作者: 于佳, E-mail: qduyujia@gmail.com, http://iec.qdu.edu.cn/

摘 要: 在传统的公钥加密方案中,一旦解密密钥泄漏,系统的安全性将完全丧失.特别是随着越来越多的加密系统被应用到移动的、安全性低的设备中,密钥泄漏显得难以避免.入侵容忍公钥加密的提出就是为了减小密钥泄漏对加密系统的危害,具有比前向安全加密、密钥隔离加密更强的安全性.在这种体制下,整个生命周期被分割成离散的时间阶段,公钥固定不变,密钥信息分享在解密者和基地中,前者独立完成解密操作,而后者则在每个时间周期中提供一个更新信息来帮助演化解密密钥.此外,每个时间段内有多次密钥刷新的操作,可以刷新解密者的密钥和基密钥.当解密者和基地被入侵时,只要不是同时被入侵,安全性就可以得到保证.即使入侵者同时入侵解密者和基地,也不会影响以前时间段密文的安全性.提出了一个入侵容忍公钥加密方案,所有费用参数关于总共时间段数的复杂性均不超过对数的平方.证明了该方案是标准模型下安全的.这是一个不需要随机预言的可证明安全的入侵容忍公钥加密方案.

关键词: 密钥泄漏;前向安全加密;密钥隔离加密;入侵容忍加密;标准模型

中图法分类号: TP309 **文献标识码**: A

中文引用格式: 于佳,程相国,李发根,潘振宽,孔凡玉,郝蓉.标准模型下可证明安全的入侵容忍公钥加密方案.软件学报,2013,24(2):266−278. http://www.jos.org.cn/1000-9825/4324.htm

英文引用格式: Yu J, Cheng XG, Li FG, Pan ZK, Kong FY, Hao R. Provably secure intrusion-resilient public-key encryption scheme in the standard model. Ruanjian Xuebao/Journal of Software, 2013,24(2):266−278 (in Chinese). http://www.jos.org.cn/1000-9825/4324.htm

Provably Secure Intrusion-Resilient Public-Key Encryption Scheme in the Standard Model

YU Jia[1,2], CHENG Xiang-Guo[1], LI Fa-Gen[3,4], PAN Zhen-Kuan[1], KONG Fan-Yu[5,6], HAO Rong[1]

[1](College of Information Engineering, Qingdao University, Qingdao 266071, China)

[2](State Key Laboratory of Information Security (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

[3](School of Computer Science and Engineering, University of Electronic Science and Technology, Chengdu 610054, China)

[4](Faculty of Mathematics, Kyushu University, Fukuoka 819-0395, Japan)

[5](Institute of Network Security, Shandong University, Ji'nan 250100, China)

[6](Key Laboratory of Cryptographic Technology and Information Security (Shandong University), Ministry of Education, Ji'nan 250100, China)

* 基金项目: 国家自然科学基金(61272425, 60703089, 61073176, 61202475); 山东省自然科学基金(ZR2010FQ019, ZR2009GQ008, ZR2010FQ015); 青岛市科技计划(12-1-4-2-(16)-jch); 华为科技基金; 信息安全国家重点实验室开放课题

收稿时间: 2012-03-25; 定稿时间: 2012-09-13

Corresponding author: YU Jia, E-mail: qduyujia@gmail.com, http://iec.qdu.edu.cn/

Abstract: In traditional public-key encryption schemes, security guarantees will be fully lost once decryption secret keys are exposed. With the ever-increase in encryption systems used in mobile and low secuirity devices, key exposure seems unavoidable. An intrusion-resilient public-key encryption is proposed to mitigate the damage for the encryption systems brought by key exposure, which provides more security than the forward-secure encryption and key-insulated encryption. In its primitive, the whole lifetime is divided into discrete periods where the public key is fixed. Secret keys are shared in a decrypter and a base. The former performs the decrypting operations on his own while the latter provides an updated message to help evolve secret keys in each period. Furthermore, multiple operations of refresh secret keys are performed to refresh decrypter and base secrets periodically. The security can be preserved when both the user and base are compromised, as long as they are not compromised simultaneously. In addition, the simultaneous compromise doesn't affect the security of the ciphertext generated in previous periods. This paper proposes an intrusion-resilient public-key encryption scheme. All the parameters in this scheme have at most a log-squared complexity in terms of the total number of time periods. The proposed scheme is proven to be secure in the standard model and is a provably secure intrusion-resilient public-key encryption scheme without random oracles.

Key words: key exposure; forward secure encryption; key-insulated encryption; intrusion-resilient encryption; standard model

标准的公钥加密方案都需要假定解密密钥绝对安全,一旦解密密钥泄漏,所有的密文都不安全了,无论这些密文是在密钥泄漏之前生成,还是在密钥泄漏之后生成的.然而,随着越来越多的加密算法被应用在便携的、安全性低的移动设备中,这使得密钥泄漏变得很难避免.在实际应用中,攻击者入侵安全性低的移动设备要比攻破实际的密码假设更加容易.密钥泄漏可以导致加密系统安全性的完全丧失.因此,如何减小密钥泄漏对加密体制的危害是一项十分重要的研究工作.

密钥演化技术是一种减小密钥泄漏危害的有效方法.最先出现的密钥演化技术是前向安全的方法.1997 年,Anderson 最先提出了非交互环境中的前向安全性的概念[1],随后出现了大量关于前向安全签名的研究工作,如文献[2,3].前向安全公钥加密方案最先由 Canetti,Halevi 和 Katz 在 2003 年欧密会上提出[4].该方案基于 Gentry 和 Silverberg 提出的分级的基于身份加密方案[5],在双线性 Diffie-Hellman 判定假设下是可证明安全的.在前向安全公钥加密体制中,解密密钥可以周期性地更新,当前的解密密钥泄漏了不会影响以前时间段产生的密文的安全性.

然而,前向安全加密/签名无法保护密钥泄漏之后密文/签名的安全性.针对这个不足,密钥隔离机制被提出来.它既可以保护密钥泄漏之前,又可以保证密钥泄露之后时间段密文/签名的安全性.密钥隔离加密/签名[6,7]和密钥隔离基于身份加密/签名[8,9]也已经得到了广泛的研究.在密钥隔离加密/签名方案中,需要一个物理安全的设备(又称基地或协助器),在每个时间段帮助用户完成密钥更新的操作.

入侵容忍安全是另一种具有更高安全性的解决方法,结合了先动安全(proactive security)、前向安全、密钥隔离安全的思想.像密钥隔离加密/签名一样,在入侵容忍加密/签名体制中,解密者/签名者可以独立完成解密/签名操作,密钥更新需要基地提供一个更新信息才能完成.不同于密钥隔离加密/签名,入侵容忍加密/签名不需要再假定基地是安全的,这是因为用户和基地在每个时间阶段中都频繁地更新其私钥.因此,无论用户和基地被入侵多少次,只要入侵不是同时发生的,其他时间段的密文/签名就都是安全的.另外,即使入侵者同时入侵用户和基地,也不能解密以前时间段的密文(伪造以前时间段的签名).在 2002 年美密会上,Itkis 和 Reyzin 提出了第一个入侵容忍签名方案[10],其构造采用了乘法共享的技术,将更新签名密钥的秘密分享在签名用户和外围设备中.随后,文献[11,12]对入侵容忍签名做了进一步的研究.文献[13]给出了入侵容忍基于身份签名的安全性模型,并提出了一个有效的方案.Dodis 等人[14]提出了目前为止唯一一个具体的入侵容忍加密方案,其方法实质上是对文献[4]中前向安全加密方案的一般化构造[15],其方案的安全性是基于随机预言模型的,其基于的困难假设是双线性 Diffie-Hellman(BDH)假设.然而,随机预言模型下的安全性证明仅仅是启发式的[16],并不能保证实际应用中的安全[17].因此,构造标准模型下可证明安全的入侵容忍加密方案显得更有吸引力.

我们的贡献是:提出了一个标准模型下可证明安全的入侵容忍公钥加密方案,基于 *l*-wBDHI 判定假设,给出了该方案在标准模型下的安全性证明.该方案也具有很好的平均性能,所有的费用参数包括:密钥产生、密钥更

新、密钥刷新、签名、验证时间的复杂性和公钥、私钥和签名长度的复杂性都不超过 $O(\log^2 T)$.同时,也给出了该方案与文献[14]方案的性能对比.

1 预备知识

1.1 密码定义和假设

令 G_1 和 G_2 是阶为素数 p 的两个乘法循环群.映射 $\hat{e}: G_1 \times G_1 \to G_2$ 被称为双线性映射,当且仅当下面的条件满足:

1. 双线性:对于任意 $g_1,g_2 \in G_1$ 和 $a,b \in Z_p^*$, 满足: $\hat{e}(g_1^a, g_2^b) = \hat{e}(g_1, g_2)^{ab}$.
2. 非退化性:存在 $g_1,g_2 \in G_1$,满足: $\hat{e}(g_1, g_2) \neq 1$.
3. 可计算性:存在一种有效算法,对任意的 $g_1,g_2 \in G_1$,可以计算 $\hat{e}(g_1, g_2)$.

G_1 中的 l-wBDHI 判定问题描述如下[18]:称输出 $b \in \{0,1\}$的算法 B 具有ε的概率来解决 G_1 中的 l-wBDHI 判定问题,当且仅当

$$|\Pr[B(g,h,g^{\alpha},...,g^{\alpha^l},\hat{e}(g,h)^{\alpha^{l+1}}) = 0] - \Pr[B(g,h,g^{\alpha},...,g^{\alpha^l},T) = 0]| \geqslant \varepsilon.$$

这里 g,h 是 G_1 中的随机生成元,α是 Z_p^* 中随机选择的元素, $T \in G_1^*$ 是随机元素.表示左边的分布为 P_{wBDHI},右边的分布为 R_{wBDHI}.

定义 1(l-wBDHI 判定假设). 称 G_1 中的(t,ε) l-wBDHI 判定假设成立,当且仅当不存在运行时间不超过 t 的算法能够以不少于ε的概率解决 G_1 中的 l-wBDHI 判定问题.

1.2 入侵容忍公钥加密方案

一个入侵容忍公钥加密方案由解密者和基地两个实体构成,系统的整个生命周期被分割成 T 个时间段,公钥始终保持不变,而私钥则分享在这两个实体中.解密者每个时间段持有的私钥可以独立解密这个时间段产生的密文.基地的私钥并不直接用于解密操作,在每个时间段开始时,基地利用它的私钥生成一个更新消息来帮助解密者更新解密密钥.系统的秘密信息通过两种算法进行修改:一种是密钥更新算法,另一种是密钥刷新算法.密钥更新算法在每个时间段都进行一次更新密钥的操作,而密钥刷新算法则是在一个时间段内进行多次刷新密钥的操作,这些刷新操作对解密者完全透明.令 $SKS_{i,r}$ 和 $SKB_{i,r}$ 分别表示第 i 时间段第 r 次更新后的解密者密钥和基地密钥(简称基密钥).当 $i=i'$ 且 $r=r'$ 时,称 $i,r=i',r'$;当 $i<i'$ 或者 $i=i'$ 且 $r<r'$ 时,称 $i,r<i',r'$.下面的定义来源于文献[14],并进行了一些修改.

定义 2(入侵容忍公钥加密方案). 一个入侵容忍公钥加密方案 IRPKE 由以下 7 种算法构成:*IRPKE*=(*IRPKE.setup*,*IRPKE.updbase*,*IRPKE.upduser*,*IRPKE.refbase*,*IRPKE.refuser*,*IRPKE.encrypt*,*IRPKE.decrypt*),其中,

1. *IRPKE.setup*:密钥产生算法,是一种概率算法.输入安全参数 k、总共时间段数 T,输出初始化的解密密钥 $SKS_{0,0}$,初始化的基密钥 $SKB_{0,0}$、公钥 PK.
2. *IRPKE.updbase*:基密钥更新算法,是一种概率算法.输入当前的基密钥 $SKB_{i,r}$,输出下一个时间段的新基密钥 $SKB_{i+1,0}$ 和更新消息 SKU_i.
3. *IRPKE.upduser*:解密密钥更新算法,是一种概率算法.输入当前的解密密钥 $SKS_{i,r}$,更新消息 SKU_i,输出下一时间段新的解密密钥 $SKS_{i+1,0}$.
4. *IRPKE.refbase*:基密钥刷新算法,是一种概率算法.输入当前的基密钥 $SKB_{i,r}$,输出新的基密钥 $SKB_{i,r+1}$ 和刷新消息 $SKR_{i,r}$.
5. *IRPKE.refuser*:解密密钥刷新算法,是一种概率算法.输入当前的解密密钥 $SKS_{i,r}$,刷新消息 $SKR_{i,r}$,输出一个新的解密密钥 $SKS_{i,r+1}$.
6. *IRPKE.encrypt*:加密算法,是一种概率算法.输入当前的时间段 i、公钥 PK 和消息 M,产生消息 M 在第 i 时间段的密文 C.
7. *IRPKE.decrypt*:解密算法,是一种确定性算法.输入当前的时间段 i、解密密钥 $SKS_{i,r}$ 和密文 C,输出明

文 M.

上述加密方案应该满足标准的一致性要求,即,如果 C 是 *IRPKE.encrypt* 算法生成的第 i 时间段消息 M 的加密密文,则 $IRPKE.decrypt(i,C,SKS_{i,r})=M$.

1.3 安全性定义

令 $RN(i)$表示第 i 时间段密钥的刷新次数,遵循文献[2]的定义,密钥产生之后,紧跟着一个密钥更新算法来得到 i=1 的密钥;密钥更新之后,紧跟着一个密钥刷新算法来得到 r=1 的密钥.所以,r=0 和 i=0 不会真正使用,RN 的采用是为了表达方便,不需要事先知道.因此,敌手可以访问的解密者密钥、基密钥、密钥更新消息、密钥刷新消息的集合定义如下:

$$SKS^*_{ID}=\{SKS^{ID}_{i,r}\mid 1\leqslant i\leqslant T,1\leqslant r\leqslant RN(i,ID)\},$$
$$SKB^*_{ID}=\{SKB^{ID}_{i,r}\mid 1\leqslant i\leqslant T,1\leqslant r\leqslant RN(i,ID)\},$$
$$SKU^*_{ID}=\{SKU^{ID}_{i}\mid 1\leqslant i\leqslant T-1\},$$
$$SKR^*_{ID}=\{SKR^{ID}_{i,r}\mid 1\leqslant i\leqslant T-1,0\leqslant r<RN(i,ID)\}\setminus\{SKR^{ID}_{1,0}\}.$$

下面的实验包括了在方案的整个生命周期中所有密钥信息的情况.

Experiment *Gen-keys*(k,T,RN)

$i\leftarrow 0$; $r\leftarrow 0$

$(SKS_{i,r},SKB_{i,r},PK)\leftarrow setup(1^k,T)$

for i=1 to T

$(SKB_{i,0},SKU_{i-1})\leftarrow updbase(SKB_{i-1,r})$

$SKB_{i,0}\leftarrow upduser(SKB_{i-1,r},SKU_{i-1})$

for r=1 to $RN(i)$

$(SKB_{i,r},SKR_{i,r-1})\leftarrow refbase(SKB_{i,r-1})$

$SKS_{i,r}\leftarrow refuser(SKS_{i,r-1},SKR_{i,r-1})$

令 SKS^*,SKB^*,SKU^*,SKR^*分别表示上述实验中所产生的解密者密钥、基密钥、更新信息、刷新信息构成的集合.敌手可以窃取上述集合中的秘密,定义敌手 F 是一个概率多项式时间的预言图灵机,能够访问以下预言:

- *LR*:"左-或-右"加密预言:输入$(i,m_0,m_1)(1\leqslant i\leqslant T)$,选择随机比特 b,输出 $Encrypt(PK,i,m_b)$.这个预言可以只被敌手查询 1 次,通过此预言,我们可以方便地定义方案的安全性.
- Q_{SKS}:解密密钥预言:输入("d",i,r),$1\leqslant i\leqslant T,1\leqslant r\leqslant RN(i)$,输出 $SKS_{i,r}$.
- Q_{SKB}:基密钥预言:输入("b",i,r),$1\leqslant i\leqslant T,1\leqslant r\leqslant RN(i)$,输出 $SKB_{i,r}$.
- Q_{SKU}:密钥更新预言:输入("u",i),$1\leqslant i\leqslant T-1$,输出 SKU_i.
- Q_{SKR}:密钥刷新预言:输入("r",i,r),$1\leqslant i\leqslant T-1,0\leqslant r<RN(i)$,并且 $i\neq 1$ 和 $r\neq 0$ 中至少 1 个成立,输出 $SKR_{i,r}$.

我们称 O_{SKS},O_{SKB},O_{SKU} 和 O_{SKR} 预言为密钥泄漏预言.敌手查询这些预言意味着它入侵了解密者,或入侵了基地,或获取了密钥更新消息,或获取了密钥刷新消息.

对于任意的密钥泄漏查询集合 Q,当下列条件之一成立时,我们称 $SKS_{i,r}$ 是 Q-泄漏:

- ("s",i,r)$\in Q$;
- $r>1$,("r",i,$r-1$)$\in Q$,$SKS_{i,r-1}$ 是 Q-泄漏;
- $r=1$,("u",$i-1$)$\in Q$,$SKS_{i-1,RN(i-1)}$是 Q-泄漏.

显然,$SKS_{i,r}$ 的泄漏可以使敌手有效地解密时间段 i 的密文.以上述同样方法可以定义 $SKB_{i,r}$ 的 Q-泄漏.敌手如果同时获得解密者和基地的密钥$(SKS_{i,r},SKB_{i,r})$,则它可以解密所有第 $i'(i'\geqslant i)$时间段的密文.

因此,如果下面两个条件中的任意一个成立,则称方案是(i,Q)-被入侵(compromised):

- $SKS_{i,r}$ 是 Q-泄漏$(1\leqslant r\leqslant RN(i))$;
- $SKS_{i',r}$ 和 $SKB_{i',r}$ 是 Q-泄漏的,这里,$i'\leqslant i$.

下面的实验给出了敌手的模型,如果一个敌手能够有效地解密一个时间段 i 的密文:不是通过查询 LR 预言获得的,仅进行合法查询,且给定时间段的安全性没有受到破坏,就认为它成功了.

Experiment *Run-Adversary*(F,k,T,RN)

Gen-keys(k,T,RN)

$b' \leftarrow A^{LR,O_{SKS},O_{SKB},O_{SKS},O_{SKR}}(1^k,T,PK,RN)$

令 Q 表示 F 对 O_{sec} 预言的查询集合

令(i,m_0,m_1)为执行的 LR 查询

令 b 是 LR 预言在回答查询时使用的比特

if $b\neq b'$ or 方案是(i,Q)-泄漏

then return (0)

else return (1)

定义 3. 令 IRPKE 表示一个入侵容忍公钥加密方案,k 为安全参数,T 为总时间段数,RN 表示 T 个时间段中密钥刷新次数构成的数组.令 F 表示上述描述的敌手.令 $Succ^{IR}(IRPKE[k,T,RN],F)$表示上述实验返回 1 的概率,则方案 IRPKE 的不安全性定义为函数

$$Insec^{IR}(IRPKE[k,T,RN],t,q_{SKS},q_{SKB},q_{SKU},q_{SKR})=\max_{F}\{|Succ^{IR}(IRPKE[k,T,RN],F)-1/2|\}.$$

这里的最大值针对的是满足下列条件的所有敌手:执行上述实验的时间最多为 t,进行的 O_{SKS}预言查询最多为 q_{SKS}次,O_{SKB}预言查询最多为 q_{SKB}次,O_{SKU}预言查询最多为 q_{SKU}次,O_{SKR}预言查询最多为 q_{SKR}次.

如果 $Insec^{IR}(IRPKE[k,T,RN],t,q_{SKS},q_{SKB},q_{SKU},q_{SKR})<\varepsilon$,则称 $IRPKE[k,T,RN]$公钥加密方案是$(t,\varepsilon,q_{SKS},q_{SKB},q_{SKU},q_{SKR})$入侵容忍的.

2 方案描述

构造思想:为了获取标准模型下的安全性,加密算法结合了 Waters 的加密技术[19].

我们使用一个深度为 l 的满二叉树[4]来生成 $T=2^l$ 个时间段的密钥.将二叉树的每个叶节点$\langle i\rangle$按照从左到右的顺序与每个时间段 i 关联,令每个时间段 $i(0\leqslant i\leqslant T-1)$的 l 位二进制表示为$\langle i\rangle=i_1i_2\ldots i_l$.最左边的叶节点表示第 0 时间段,最右边的叶节点表示第 $T-1$ 时间段.每个二叉树的节点标示为一个二进制串 w.令ε表示为一个空串,标示二叉树的根节点为ε.当一个中间节点标示为二进制串 w 时,它的左右儿子节点分别标示为 $w0$ 和 $w1$.令 $w|_k$ 表示为 w 的 k 位前缀, $w|_{\bar{k}}$ 表示为节点 $w|_k$ 的兄弟节点. 第 i 时间段的完全私钥 $SK_i=\{sk_{\langle i\rangle}\}\cup\{sk_{w1}|w0=i_0i_1\ldots i_j,0\leqslant j<l\}$,$sk_{\langle i\rangle}$用来解密,而$\{sk_{w1}|w0=i_0i_1\ldots i_j,0\leqslant j<l\}$用来生成下一时间段的解密密钥.

- 解密者持有的密钥为 $SKS_{i,r}=\{sk_{\langle i\rangle}\}\cup\{sk'_{w1}\mid w0=i_0i_1..i_j,0\leqslant j<l\}$;
- 基地持有的密钥为 $SKB_{i,r}=\{sk''_{w1}\mid w0=i_0i_1..i_j,0\leqslant j<l\}$.

这里,对于所有的 $w0=i_0i_1\ldots i_j,0\leqslant j<l$,满足 $sk_{w1}=sk'_{w1}\cdot sk''_{w1}$.因此,解密者可利用 $sk_{\langle i\rangle}$独立完成解密操作,而密钥更新则需要基地和解密者合作完成.

(1) *IRPKE.setup*:输入安全参数 k、总时间阶段数 $T=2^l$,做以下操作:

① 运行 $IG(1^k)$,产生素数 p 阶循环群 G_1,G_2 和一个双线性配对 $\hat{e}:G_1\times G_1\rightarrow G_2$.

② 选择生成元 $g\in_R G_1$ 和随机数 $\alpha\in_R Z_q^*$,令 $g_1=g^\alpha$.选择 $g_2,w',w_1,\ldots,w_l\in_R G_1$,计算 $Z=\hat{e}(g_1,g_2)$. 令 k 元向量 $W=(w_j),j=1,\ldots,k$ 且 $k\leqslant l$.

③ 令 x 是一个 k 比特的二进制串表示的二叉树节点,x_j 表示 x 的第 j 比特,$X\subseteq\{1,\ldots,k\}$表示满足 $x_j=1$ 的所有 j 构成的集合.这意味着 X 是比特串 x 的所有比特位为 1 的位数构成的集合.

定义函数 $F_1:\{0,1\}^{\leqslant l}\rightarrow G_1$ 为

$$F_1(x_1\ldots x_k)=w'\prod_{j\in X}w_j.$$

④ 首先,为了构造初始化的解密密钥,选择 $r_0,r_1\in_R Z_p$,计算:

$$sk_0=(g_2^{\alpha}\cdot(w')^{r_0},g^{r_0},w_2^{r_0},...,w_l^{r_0}),$$
$$sk_1=(g_2^{\alpha}\cdot(w'w_1)^{r_1},g^{r_1},w_2^{r_1},...,w_l^{r_1}).$$

选择 $\alpha',r_0',r_1'\in_R Z_p$,计算 $\alpha''=\alpha-\alpha',r_0''=r_0-r_0',r_1''=r_1-r_1'$,

$$sk_0'=(g_2^{\alpha'}\cdot(w')^{r_0'},g^{r_0'},w_2^{r_0'},...,w_l^{r_0'}),$$
$$sk_0''=(g_2^{\alpha''}\cdot(w')^{r_0''},g^{r_0''},w_2^{r_0''},...,w_l^{r_0''}),$$
$$sk_1'=(g_2^{\alpha'}\cdot(w'w_1)^{r_1'},g^{r_1'},w_2^{r_1'},...,w_l^{r_1'}),$$
$$sk_1''=(g_2^{\alpha''}\cdot(w'w_1)^{r_1''},g^{r_1''},w_2^{r_1''},...,w_l^{r_1''}).$$

⑤ for k=2 to l do

begin

解析:

$$sk_{0^{k-1}}=(a_0,a_1,b_k,...,b_l)=(g_2^{\alpha}\cdot F_1(0^{k-1})^{r},g^{r},w_k^{r},...,w_l^{r}),$$
$$sk_{0^{k-1}}'=(a_0',a_1',b_k',...,b_l')=(g_2^{\alpha'}\cdot F_1(0^{k-1})^{r'},g^{r'},w_k^{r'},...,w_l^{r'}),$$
$$sk_{0^{k-1}}''=(a_0'',a_1'',b_k'',...,b_l'')=(g_2^{\alpha''}\cdot F_1(0^{k-1})^{r''},g^{r''},w_k^{r''},...,w_l^{r''}).$$

选择 $t_0,t_1,t_0',t_1'\in_R Z_p^*$,计算 $t_0''=t_0-t_0',t_1''=t_1-t_1'$,

$$sk_{0^k}=(a_0\cdot F_1(0^k)^{t_0},a_1g^{t_0},b_{k+1}w_{k+1}^{t_0},...,b_lw_l^{t_0})=(g_2^{\alpha}\cdot F_1(0^k)^{r_0},g^{r_0},w_{k+1}^{r_0},...,w_l^{r_0}),$$
$$sk_{0^k}'=(a_0'\cdot F_1(0^k)^{t_0'},a_1'g^{t_0'},b_{k+1}'w_{k+1}^{t_0'},...,b_l'w_l^{t_0'})=(g_2^{\alpha'}\cdot F_1(0^k)^{r_0'},g^{r_0'},w_{k+1}^{r_0'},...,w_l^{r_0'}),$$
$$sk_{0^k}''=(a_0''\cdot F_1(0^k)^{t_0''},a_1''g^{t_0''},b_{k+1}''w_{k+1}^{t_0''},...,b_l''w_l^{t_0''})=(g_2^{\alpha''}\cdot F_1(0^k)^{r_0''},g^{r_0''},w_{k+1}^{r_0''},...,w_l^{r_0''}).$$

其中,$r_0=r+t_0,r_0'=r'+t_0',r_0''=r''+t_0''$.

另外,

$$sk_{0^{k-1}1}=(a_0\cdot F_1(0^{k-1}1)^{t_1},a_1g^{t_1},b_{k+1}w_{k+1}^{t_1},...,b_lw_l^{t_1})=(g_2^{\alpha}\cdot F_1(0^{k-1}1)^{r_1},g^{r_1},w_{k+1}^{r_1},...,w_l^{r_1}),$$
$$sk_{0^{k-1}1}'=(a_0'\cdot F_1(0^{k-1}1)^{t_1'},a_1'g^{t_1'},b_{k+1}'w_{k+1}^{t_1'},...,b_l'w_l^{t_1'})=(g_2^{\alpha'}\cdot F_1(0^{k-1}1)^{r_1'},g^{r_1'},w_{k+1}^{r_1'},...,w_l^{r_1'}),$$
$$sk_{0^{k-1}1}''=(a_0''\cdot F_1(0^{k-1}1)^{t_1''},a_1''g^{t_1''},b_{k+1}''w_{k+1}^{t_1''},...,b_l''w_l^{t_1''})=(g_2^{\alpha''}\cdot F_1(0^{k-1}1)^{r_1''},g^{r_1''},w_{k+1}^{r_1''},...,w_l^{r_1''}).$$

其中,$r_1=r+t_1,r_1'=r'+t_1',r_1''=r''+t_1''$.

end

⑥ 令加密公钥为 $PK=(G_1,G_2,\hat{e},g,g_1,g_2,w',w_1,...,w_l,Z)$.

令 $SKB_{0,0}=\{(sk_1'',sk_{01}'',...,sk_{0^{l-1}1}'')\},SKS_{0,0}=\{sk_{0^l},(sk_1',sk_{01}',...,sk_{0^{l-1}1}')\}$.

删除中间数据,返回 $PK,SKS_{0,0},SKB_{0,0}$.

(2) *IRPKE.updbase*:输入当前时间段 i,当前解密密钥 $SKB_{i,r}$,做以下操作:

令$\langle i\rangle=i_0i_1\ldots i_l(i_0=\varepsilon)$,解析 $SKB_{i,r}=\{(\{sk_{i_0\ldots i_{k-1}1}''\}_{i_k=0,1\leqslant k\leqslant l})\}$.

① 如果 i_l=0,则新的基密钥为 $SKB_{i+1,0}=\{(\{sk_{i_0\ldots i_{k-1}1}''\}_{i_k=0,1\leqslant k<l})\}$.

更新消息为 $SKU_i=\{sk_{i_0\ldots i_{l-1}1}''\}$ (注意到$\langle i+1\rangle=i_0\ldots i_{l-1}1$).

② If i_l=1,找到满足 i_j=0 的最长的 j,令 $x_1\ldots x_i=i_0\ldots i_{j-1}1$.

for $k=j+1$ to l do

begin

解析 $sk_{x_1\ldots x_{k-1}}''=(a_0'',a_1'',b_k'',...,b_l'')=(R''\cdot F_1(x_1...x_{k-1})^{r_i''},g^{r_i''},w_k^{r_i''},...,w_l^{r_i''})$.

选择 $t_0'',t_1''\in_R Z_p^*$,计算

$sk_{x_1\ldots x_{k-1}0}''=(a_0''\cdot F_1(x_1...x_{k-1}0)^{t_0''},a_1''g^{t_0''},b_{k+1}''w_{k+1}^{t_0''},...,b_l''w_l^{t_0''})=(R''\cdot F_1(x_1...x_{k-1}0)^{r_0''},g^{r_0''},w_{k+1}^{r_0''},...,w_l^{r_0''})$,

其中,$r_0''=r_i''+t_0''$.

另外,计算

$$sk''_{x_1\ldots x_{k-1}1}=(a''_0\cdot F_1(x_1\ldots x_{k-1}1)^{t''_1},a''_1g^{t''_1},b''_{k+1}w_{k+1}^{t''_1},\ldots,b''_lw_l^{t''_1})=(R''\cdot F_1(x_1\ldots x_{k-1}1)^{r''_1},g^{r''_1},w_{k+1}^{r''_1},\ldots,w_l^{r''_1}),$$

其中, $r''_1=r''_i+t''_1$.

end

③ 删除 $SKB_{i,r}$,新的基密钥为 $SKB_{i+1,0}=\{(\{sk''_{i_1\ldots i_{j-1}10^{k}1}\}_{0\leqslant k\leqslant l-j-1})\}$.

密钥更新消息为 $SKU_i=\{sk''_{i_1\ldots i_{j-1}10^{l-j}}\}$ (注意到$\langle i+1\rangle=i_1\ldots i_{j-1}10^{l-j}$).

(3) *IRPKE.upduser*:输入当前时间段 i、刷新次数 r、解密密钥 $SKS_{i,r}$、更新消息 SKU_i,做以下操作:

令$\langle i\rangle=i_0i_1\ldots i_l(i_0=\varepsilon)$.解析 $SKS_{i,r}=\{sk_{\langle i\rangle},(\{sk'_{i_0\ldots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l})\},SKU_i=\{sk''_{\langle i+1\rangle}\}$.

① If i_l=0,这意味着$\langle i+1\rangle=i_0\ldots i_{l-1}1$.新的解密密钥为

$$SKS_{i+1,0}=\{sk'_{i_0\ldots i_{l-1}1}\cdot sk''_{i_0\ldots i_{l-1}1},(\{sk'_{i_0\ldots i_{k-1}1}\}_{i_k=0,1\leqslant k<l})\}.$$

② If i_l=1,找到满足 i_j=0 的最长的 j.

令 $x_1\ldots x_i=i_0\ldots i_{j-1}1$(注意到$\langle i+1\rangle=i_1\ldots i_{j-1}10^{l-j}$).

for k=j+1 to l do

begin

解析 $sk'_{x_1\ldots x_{k-1}}=(a'_0,a'_1,b'_k,\ldots,b'_l)=(R'\cdot F_1(x_1\ldots x_{k-1})^{r'_1},g^{r'_1},w_k^{r'_1},\ldots,w_l^{r'_1})$.

选择 $t'_0,t'_1\in_R Z_p^*$,计算

$$sk'_{x_1\ldots x_{k-1}0}=(a'_0\cdot F_1(x_1\ldots x_{k-1}0)^{t'_0},a'_1g^{t'_0},b'_{k+1}w_{k+1}^{t'_0},\ldots,b'_lw_l^{t'_0})=(R'\cdot F_1(x_1\ldots x_{k-1}0)^{r'_0},g^{r'_0},w_{k+1}^{r'_0},\ldots,w_l^{r'_0}),$$

其中, $r'_0=r'_i+t'_0$.

另外,计算

$$sk'_{x_1\ldots x_{k-1}1}=(a'_0\cdot F_1(x_1\ldots x_{k-1}1)^{t'_1},a'_1g^{t'_1},b'_{k+1}w_{k+1}^{t'_1},\ldots,b'_lw_l^{t'_1})=(R'\cdot F_1(x_1\ldots x_{k-1}1)^{r'_1},g^{r'_1},w_{k+1}^{r'_1},\ldots,w_l^{r'_1}),$$

其中, $r'_1=r'_i+t'_1$.

end

③ 删除 $SKS_{i,r}$,新的基密钥为 $SKS_{i+1,0}=\{sk'_{i_1\ldots i_{j-1}10^{l-j}}\cdot sk''_{i_1\ldots i_{j-1}10^{l-j}},(\{sk'_{i_1\ldots i_{j-1}10^{k}1}\}_{0\leqslant k\leqslant l-j-1})\}$.

(4) *IRPKE.refbase*:输入当前时间段 i、刷新次数 r、基密钥 $SKB_{i,r}$,做以下操作:

对于每个 $sk''_{i_0\ldots i_{k-1}1}\in SKB_{i,r}$ (这里,i_k=0,1≤k<l),选择 $R''_{i_0\ldots i_{k-1}1}\in_R G_1$,

令 $SKB_{i,r+1}=\{sk''_{i_0\ldots i_{k-1}1}\cdot R''_{i_0\ldots i_{k-1}1}\mid i_k=0,1\leqslant k<l\},SKR_{i,r}=\{R''_{i_0\ldots i_{k-1}1}\mid i_k=0,1\leqslant k<l\}$,

删除所有中间数据,并返回 $SKB_{i,r+1},SKR_{i,r}$.

(5) *IRPKE.refuser*:输入当前时间段 i、刷新次数 r、解密密钥 $SKS_{i,r}$、刷新消息 $SKR_{i,r}$,做以下操作:

解析 $SKS_{i,r}=\{sk_{\langle i\rangle},(\{sk'_{i_0\ldots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l})\},SKR_{i,r}=\{R''_{i_0\ldots i_{k-1}1}\mid i_k=0,1\leqslant k<l\}$.

令 $SKS_{i,r+1}=\{sk_{\langle i\rangle},(\{sk'_{i_0\ldots i_{k-1}1}/R''_{i_0\ldots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l})\}$,

删除所有中间数据,并返回 $SKS_{i,r+1}$.

(6) *IRPKE.encrypt*:输入当前时间段 i、公钥 PK、消息 M,做以下操作:

① 解析$\langle i\rangle=x_1\ldots x_l$;

② 选择 $s\in Z_p$,计算$(C_0,C_1,C_2)=(Z^s\cdot M,F_1(x_1\ldots x_l)^s,g^s)$;

③ 输出密文$\langle i,(C_0,C_1,C_2)\rangle$.

(7) *IRPKE.decrypt*:输入当前解密密钥 $SKS_{i,r}$、密文$\langle i,(C_0,C_1,C_2)\rangle$,做以下操作:

解析$\langle i\rangle=x_0x_1\ldots x_l(x_0=\varepsilon)$, $SKS_{i,r}=(sk_{\langle i\rangle},\{sk'_{x_0\ldots x_{k-1}1}\}_{x_k=0,1\leqslant k\leqslant l})$,

其中, $sk_{\langle i\rangle}=(a_0,a_1)=(g_2^{\alpha}\cdot F_1(x_1\ldots x_l)^r,g^r)$;

解密获取明文: $M=C_0\dfrac{\hat{e}(a_1,C_1)}{\hat{e}(a_0,C_2)}$.

3 安全性分析

定理 1. 令$\langle i,(C_0,C_1,C_2)\rangle$为 *IRPKE.encrypt* 算法产生的第 i 时间段消息 M 的加密密文,则

$$IRPKE.decrypt(\langle i,(C_0,C_1,C_2)\rangle,SKS_{i,r})=M.$$

证明:

$$C_0\frac{\hat{e}(a_1,C_1)}{\hat{e}(a_0,C_2)}=Z^s\cdot M\cdot\frac{\hat{e}(g^r,F_2(x_1...x_l)^s)}{\hat{e}(g_2{}^{\alpha}\cdot F_1(x_1...x_l)^r,g^s)}=Z^s\cdot M\cdot\frac{\hat{e}(g,F_2(x_1...x_l))^{rs}}{\hat{e}(g_2{}^{\alpha},g^s)\cdot\hat{e}(F_1(x_1...x_l)^r,g^s)}=Z^s\cdot M\cdot\frac{1}{Z^s}=M.\quad\square$$

定理 2. 如果 G_1 中的(t',ε') l-wBDHI 判定假设成立,则我们提出的方案是$(t,\varepsilon,q_{SKS},q_{SKB},q_{SKU},q_{SKR})$入侵容忍的,这里,

- $t'=t+O(\log T(T\log T+q_{SKB}+q_{SKS}+\log T\cdot q_{SKU}+q_{SKU}+1)\cdot t_{G_1})$,
- $\varepsilon'=\frac{1}{T}\cdot\varepsilon$,

其中,t_{G_1} 表示 G_1 中的一个运算最大的执行时间.

证明:我们按以下方法来构造一种算法 $I(t',\varepsilon')$来解决 G_1 中的 l-wBDHI 判定问题.

假定输入一个随机元组 $(g,h,z_1=g^{\alpha},z_2=g^{\alpha^2},...,z_l=g^{\alpha^l},T')\in G_1^{l+3}$ 给算法 I,这个随机元组或者是从 P_{wBDHI} 中选取的(这时,$T'=\hat{e}(g,h)^{\alpha^{l+1}}$),或者是从 R_{wBDHI} 中选取的(这时,T'在 G_1^* 中均匀分布).算法 I 的目的是:若输入元组是从 P_{wBDHI} 中选取的,则输出 1;否则,输出 0.I 随机猜测一个 $i^*(0\leqslant i^*\leqslant T-1)$作为 F 进行 LR 加密预言查询的时间段,表示$\langle i^*\rangle=i_1^*...i_l^*$,令 $z_l=g^{\alpha^{l+1}}$.

首先,算法 I 产生初始化的参数,选取 $\gamma\in_R Z_p^*$, 令 $g_1=g^{\alpha}=z_1$, $g_2=g^{\gamma+\alpha^l}=g^{\gamma}\cdot z_l$. 因此, $g_2{}^{\alpha}=(g^{\gamma+\alpha^l})=z_1{}^{\gamma}\cdot z_{l+1}$, I 并不知道它的值.I 选取 $\gamma_1,\gamma_2,...,\gamma_l,\delta\in Z_p^*$, 令 $w'=g^{\delta}\prod_{j=1}^{l}z_{l-j+1}^{i_j^*}$ 和 $w_j=g^{\gamma_j}/z_{l-j+1}$($j$=1,…,$l$).然后,$I$ 令 $\tau_u=q_e$,并选取 $\lambda_u\in_R(0,1,\ldots,n_u)$.假定 $\tau_u(n_u+1)<p$,算法 I 选取 $x'\in_R Z_{\tau_u}$ 和向量 $(x_1,...,x_{n_u})$, 这里,$x_j\in_R Z_{\tau_u}$ for all $1\leqslant j\leqslant n_u$.算法 I 也选择 $y'\in_R Z_p$ 和向量 $(y_1,...,y_{n_u})$, 这里,$y_j\in_R Z_p$(for all $1\leqslant j\leqslant n_u$).

I 计算 $Z=\hat{e}(g_1,g_2)$,提供给 F 公钥 $PK=(G_1,G_2,\hat{e},g,g_1,g_2,w',w_1,...,w_l,Z)$.

I 可以产生除了阶段 i^*以外的所有时间段的完全密钥.

首先,I 先计算从根节点到叶节点$\langle i^*\rangle$路径上所有节点的左兄弟节点(如果存在)的节点密钥,即 I 产生满足 $i_k^*=1$ ($k\in\{1,\ldots,l\}$)的所有节点 $i^*|_{\bar{k}}=i_1^*...i_{k-1}^*0$ 的节点密钥,I 选择 $\bar{r}\in_R Z_p$, 如果定义 $r=\bar{r}-a^k\in Z_p$, 则 I 可以计算

$$sk_{i^*|_{\bar{k}}}=(g_2^{\alpha}\cdot F_1(i_1^*...i_{k-1}^*0)^r,g^r,w_{k+1}^r,...,w_l^r).$$

观察下面的关系成立:

$$g_2^{\alpha}\cdot F_1(i_1^*...i_{k-1}^*0)^r=z_1^{\gamma}\cdot z_{l+1}\cdot F_1(i_1^*...i_{k-1}^*0)^r,$$

$$\begin{aligned}F_1(i_1^*...i_{k-1}^*0)^r&=\left(w'\cdot\prod_{j=1}^{k-1}w_j^{*i_j}\right)^r\\&=\left(w'/w_k\cdot\prod_{j=1}^{k}w_j^{*i_j}\right)^r\\&=\left(g^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*-\gamma_k}\cdot z_{l-k+1}\cdot\prod_{j=k+1}^{l}z_{l-j+1}^{i_j^*}\right)^r\\&=\left(g^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*-\gamma_k}\cdot z_{l-k+1}\cdot\prod_{j=k+1}^{l}z_{l-j+1}^{i_j^*}\right)^{\bar{r}-a^k}\\&=(g^{\bar{r}}/z_k)^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*-\gamma_k}\cdot z_{l-k+1}^{\bar{r}}/z_{l+1}\cdot\prod_{j=k+1}^{l}(z_{l-j+1}^{\bar{r}}/z_{l+k+1-j})^{i_j^*}.\end{aligned}$$

因此,

$$g_2^{\alpha}\cdot F_1(i_1^*...i_{k-1}^*0)^r = z_1^{\gamma}\cdot(g^{\bar r}/z_k)^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*-\gamma_k}\cdot z_{l-k+1}^{\bar r}\cdot\prod_{j=k+1}^{l}(z_{l-j+1}^{\bar r}/z_{l+k+1-j})^{i_j^*}.$$

此外, $g^r = g^{\bar r}/z_k, w_j^r = (g^{\gamma_j}/z_{l-j+1})^{\bar r-a^k} = (g^{\bar r}/z_k)^{\gamma_j}\cdot z_{l+k+1-j}/z_{l-j+1}^{\bar r}$ (这里,$j=k+1,\ldots,l,k<l$).

因此,I 可以计算所有的 $sk_{i^*|_{\bar k}}(i_k^*=1,k<l)$.然后,I 使用这些 $sk_{i^*|_{\bar k}}$ 计算所有 $t<i^*$时间段的完全密钥.

此外,I 也可以使用相近的方法计算从根节点到叶节点$\langle i^*\rangle$路径上所有节点的右兄弟节点(如果存在)的节点密钥,即 I 产生满足 $i_k^*=0\,(k\in\{1,\ldots,l\})$的所有节点 $i^*|_{\bar k}=i_1^*...i_{k-1}^*1$ 的节点密钥,I 选择 $\bar r\in_R Z_p$, 如果定义 $r=\bar r+a^k\in Z_p$, 则 I 可以计算

$$sk_{i^*|_{\bar k}} = (g_2^{\alpha}\cdot F_1(i_1^*...i_{k-1}^*1)^r, g^r, w_{k+1}^r,...,w_l^r).$$

类似地,观察下面的关系成立:

$$g_2^{\alpha}\cdot F_2(i_1^*...i_{k-1}^*1)^r = z_1^{\gamma}\cdot z_{l+1}\cdot F_1(i_1^*...i_{k-1}^*1)^r,$$

$$F_1(i_1^*...i_{k-1}^*1)^r = \left(w'\cdot w_k\cdot\prod_{j=1}^{k}w_j^{i_j^*}\right)^r$$

$$= \left(g^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*+\gamma_k}\cdot z_{l-k+1}^{-1}\cdot\prod_{j=k+1}^{l}z_{l-j+1}^{i_j^*}\right)^r$$

$$= \left(g^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*+\gamma_k}\cdot z_{l-k+1}^{-1}\cdot\prod_{j=k+1}^{l}z_{l-j+1}^{i_j^*}\right)^{\bar r+a^k}$$

$$= (g^{\bar r}\cdot z_k)^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*+\gamma_k}\cdot z_{l-k+1}^{-\bar r}/z_{l+1}\cdot\prod_{j=k+1}^{l}(z_{l-j+1}^{\bar r}\cdot z_{l+k+1-j})^{i_j^*}.$$

因此,

$$g_2^{\alpha}\cdot F_1(i_1^*...i_{k-1}^*1)^r = z_1^{\gamma}\cdot(g^{\bar r}\cdot z_k)^{\delta+\sum_{j=1}^{k-1}\gamma_j i_j^*+\gamma_k}\cdot z_{l-k+1}^{-\bar r}\cdot\prod_{j=k+1}^{l}(z_{l-j+1}^{\bar r}\cdot z_{l+k+1-j})^{i^*}.$$

此外, $g^r = g^{\bar r}\cdot z_k, w_j^r = (g^{\gamma_j}/z_{l-j+1})^{\bar r+a^k} = (g^{\bar r}\cdot z_k)^{\gamma_j}/(z_{l-j+1}^{\bar r}\cdot z_{l+k+1-j})$ (这里,$j=k+1,\ldots,l,k<l$).

因此,I 可以计算所有的 $sk_{i^*|_{\bar k}}(i_k^*=0,k<l)$.然后,I 使用这些 $sk_{i^*|_{\bar k}}$ 计算所有 $t>i^*$时间段的完全密钥.

为了方便回答 F 对 O_{SKS} 预言、O_{SKB} 预言、O_{SKU} 预言和 O_{SKR} 预言的查询,我们保持一张记录表 $List_k$ 来记录必要的秘密信息,其每个元组表示为$(i,r,SKB_{i,r},SKS_{i,r},SKR_{i,r})$.

- 基密钥 O_{SKB} 预言查询

当 F 从 O_{SKB} 预言中查询(“b”,i,r)的值时,I 做以下操作:

1. 如果$(i,r,SKB_{i,r},*,*)$已在 $List_k$ 中存在,那么 I 从 $List_k$ 中提取 $SKB_{i,r}$.
2. 否则,如果$(i,r,*,SKS_{i,r},*)$已在 $List_k$ 中存在,那么,
   I 解析 $SKS_{i,r}=\{sk_{\langle i\rangle},\{sk'_{i_0\ldots i_{k-1}}\}_{i_k=0,1\leqslant k\leqslant l}\}$;
   I 计算 $SKB_{i,r}=\{sk_{i_0\ldots i_{k-1}}/sk'_{i_0\ldots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}$;
   I 用$(i,r,SKB_{i,r},SKS_{i,r},*)$代替$(i,r,*,SKS_{i,r},*)$以更新 $List_k$.
3. 否则,如果$(i,r-1,SKB_{i,r-1},*,SKR_{i,r-1})$已在 $List_k$ 中存在:
   I 解析 $SKB_{i,r-1}=\{sk''_{i_0\ldots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}, SKR_{i,r-1}=\{R''_{i_0\ldots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}$;
   I 计算 $SKB_{i,r}^{ID}=\{sk''_{i_0\ldots i_{k-1}}\cdot R''_{i_0\ldots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}$, 将元组$(i,r,SKB_{i,r},*,*)$加入 $List_k$.
4. 否则,I 选择 $sk''_{i_0\ldots i_{k-1}}\in_R G_1$ (for i_k=0,1≤k≤l),令 $SKB_{i,r}=\{sk''_{i_0\ldots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}$, 将元组$(i,r,SKB_{i,r},*,*)$加入 $List_k$.
5. 将 $SKB_{i,r}$ 返回给 F.

- 解密密钥 O_{SKS} 预言查询

当 F 从 O_{SKS} 预言中查询(“s”,i,r)的值时,I 做以下操作:

1. 如果$(i,r,*,SKS_{i,r},*)$已在 $List_k$ 中存在,那么 I 从 $List_k$ 中提取 $SKS_{i,r}$.
2. 否则,如果$(i,r,SKB_{i,r},*,*)$已在 $List_k$ 中存在:
   I 解析 $SKB_{i,r}=\{sk''_{i_0\cdots i_{k-1}}\mid i_k=0,1\leqslant k\leqslant l\}$, 令 $SKS_{i,r}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}/sk''_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$;
   I 用$(i,r,SKB_{i,r},SKS_{i,r},*)$代替$(i,r,SKB_{i,r},*,*)$以更新 $List_k$.
3. 否则,如果$(i,r-1,*,SKS_{i,r-1},SKR_{i,r-1})$已在 $List_k$ 中存在,那么,
   I 解析 $SKS_{i,r-1}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$, $SKR_{i,r-1}=\{R''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$;
   I 计算 $SKS_{i,r}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}/R''_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$, 将元组$(i,r,*,SKS_{i,r},*)$加入 $List_k$.
4. 否则,I 选择 $sk'_{i_0\cdots i_{k-1}1}\in_R G_1$ (for i_k=0,1≤k≤l),令 $SKS_{i,r-1}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$,
   将元组$(i,r,*,SKS_{i,r},*)$加入 $List_k$.
5. 将 $SKS_{i,r}$ 返回给 F.

- 密钥更新 O_{SKU} 预言查询

当 F 从 O_{SKU} 预言中查询("u",i)的值时,I 做以下操作:

1. 如果$(i,RN(i),SKB_{i,RN(i)},*,*)$已在 $List_k$ 中存在,那么 I 执行 $UB(SKB_{i,RN(i)})$算法生成 SKU_i.
2. 否则,如果$(i,RN(i),*,SKS_{i,RN(i)},*)$已在 $List_k$ 中存在,解析 $SKS_{i,RN(i)}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$;
   如果 i_l=0,解析 $SK_{i+1,0}=\{sk_{i_0\cdots i_{l-1}},(\{sk_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k<l})\}$, I 计算 $SKR_{i,r}=\{sk_{i_0\cdots i_{l-1}1}/sk'_{i_0\cdots i_{l-1}1}\}$;
   如果 i_l=1,解析 $SK_{i+1,0}=\{sk_{i_1\cdots i_{j-1}10^{l-j}},(\{sk_{i_1\cdots i_{j-1}10^k1}\}_{0\leqslant k\leqslant l-j-1})\}$, I 设置 $SKU_i=e\in G_1$(e 为 G_1 中的单位元);
   I 执行 $IRPKE.upduser(SKS_{i,RN(i)},e)$算法生成 $SKS'_{i+1,0}=\{sk'_{i_1\cdots i_{j-1}10^{l-j}},(\{sk'_{i_1\cdots i_{j-1}10^k1}\}_{0\leqslant k\leqslant l-j-1})\}$;
   I 计算 $SKU_i=\{sk_{i_1\cdots i_{j-1}10^{l-j}}/sk'_{i_1\cdots i_{j-1}10^{l-j}}\}$, 用$(i,r,*,SKS_{i,r},SKR_{i,r})$代替$(i,r,*,SKS_{i,r},*)$以更新 $List_k$.
3. 否则,I 选择 $sk''_{i_0\cdots i_{k-1}1}\in_R G_1$ (for i_k=0,1≤k≤l),令 $SKB_{i,RN(i)}=\{sk''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$,
   将$(i,RN(i),SKB_{i,RN(i)},*,*)$加入 $List_k$,然后执行 $IRPKE.updbase(SKB_{i,RN(i)})$算法生成 SKU_i.
4. 将 SKU_i 返回给 F.

- 密钥刷新 O_{SKR} 预言查询

当 F 从 O_{SKR} 预言中查询("r",i,r)的值时,I 做以下操作:

1. 如果$(i,r,*,*,SKR_{i,r})$已在 $List_k$ 中存在,那么 I 从 $List_k$ 中提取 $SKR_{i,r}$.
2. 否则,如果$(i,r,*,SKS_{i,r},*)$和$(i,r,*,SKS_{i,r+1},*)$已在 $List_k$ 中存在,那么,
   I 解析 $SKS_{i,r}=\{sk_{\langle i\rangle},\{sk'_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$, $SKS_{i,r+1}=\{sk_{\langle i\rangle},\{sk''_{i_0\cdots i_{k-1}1}\}_{i_k=0,1\leqslant k\leqslant l}\}$;
   I 计算 $SKR_{i,r}=\{sk'_{i_0\cdots i_{k-1}1}/sk''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$, 用$(i,r,*,SKS_{i,r},SKR_{i,r})$代替$(i,r,*,SKS_{i,r},*)$以更新 $List_k$.
3. 否则,如果$(i,r,SKB_{i,r},*,*)$和$(i,r,SKB_{i,r+1},*,*)$已在 $List_k$ 中存在,那么,
   I 解析 $SKB_{i,r}=\{sk''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$ 和 $SKB_{i,r+1}=\{sk'_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$;
   I 计算 $SKR_{i,r}=\{sk'_{i_0\cdots i_{k-1}1}/sk''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$, 用$(i,r,SKB_{i,r},*,SKR_{i,r})$代替$(i,r,SKB_{i,r},*,*)$以更新 $List_k$.
4. I 选择 $R''_{i_0\cdots i_{k-1}1}\in_R G_1$ (for i_k=0,1≤k≤l),设置 $SKR_{i,r}=\{R''_{i_0\cdots i_{k-1}1}\mid i_k=0,1\leqslant k\leqslant l\}$, 将$(i,r,*,*,SKR_{i,r})$加入 $List_k$.
5. 将 $SKR_{i,r}$ 返回给 F.

- LR 预言查询

当 F 完成上述查询以后,就输出两个消息 $M_0,M_1\in G_1$ 和所对应的时间阶段 i^*.令 $\langle i^*\rangle=i_1^*i_2^*\ldots i_l^*$. 算法 I 选择一个随机比特 $b\in\{0,1\}$,并返回密文

$$CT=\left(M_b\cdot T'\cdot\hat{e}(z_1,h^{\gamma}),h^{\delta+\sum_{i=1}^{l}i_1^*\gamma_l},h\right).$$

这里,h 和 T'来源于输入元组.如果定义 $h=g^c$(对于某个已知的 c),则

$$\hat{e}(g,h)^{\alpha^{l+1}}\cdot\hat{e}(z_1,h^{\gamma})=(\hat{e}(z_1,z_l)\cdot\hat{e}(z_1,g^{\gamma}))^c=\hat{e}(z_1,g^{\gamma}z_l)^c=\hat{e}(g_1,g_2)^c,$$

$$h^{\delta+\sum_{i=1}^{l}i_1^*\gamma_i}=\left(g^{\delta+\sum_{i=1}^{l}i_1^*\gamma_i}\right)^c=\left(\prod_{i=1}^{l}(g^{\gamma_i}/z_{l-i+1})^{i^*}\cdot\left(g^{\delta}\prod_{i=1}^{l}z_{l-i+1}^{i^*}\right)\right)^c=\left(w'\cdot\prod_{i=1}^{l}(w_i)^{i^*}\right)^c=F_1(i_1^*i_2^*...i_l^*)^c.$$

如果 $T'=\hat{e}(g,h)^{\alpha^{l+1}}$, 则返回的是对消息 M_b 有效的密文,这是因为

$$CT=(M_b\cdot\hat{e}(g_1,g_2)^c,F_1(i_1^*i_2^*...i_l^*)^c,g^c);$$

否则,当 T 在 G_1^* 中均匀分布时,CT 在敌手看来是独立于 b 的.

- 猜测

最后,F 输出其猜测 $b'\in\{0,1\}$,算法 I 按照下面的方法输出其猜测:

如果 $b=b'$,则 I 输出 1,意味着 $T'=\hat{e}(g,h)^{\alpha^{l+1}}$; 否则输出 0,意味着 T 在 G_1 中均匀分布.

当输入元组是从 P_{wBDHI} 中选取时(这里, $T'=\hat{e}(g,h)^{\alpha^{l+1}}$),F 的观察等同于实际攻击游戏中的观察,因此,

$$|\Pr[b=b']-1/2|\geqslant\varepsilon.$$

当输入元组是从 R_{wBDHI} 中选取时(其中,T'在 G_1 中均匀分布),$\Pr[b=b']=1/2$.由于 g,h 在 G_1 中均匀分布,α在 Z_p 中均匀分布,T'在 G_1 中均匀分布,因此,

$$|\Pr[B(g,h,g^{\alpha},...,g^{\alpha^l},\hat{e}(g,h)^{\alpha^{l+1}})=0]-\Pr[B(g,h,g^{\alpha},...,g^{\alpha^l},T')=0]|\geqslant|(1/2\pm\varepsilon)-1/2|=\varepsilon.$$

- 概率分析

I 猜测的 i^*正好是 F 进行 LR 加密预言查询的时间段的概率是 $1/T$,因此, $\varepsilon'=\frac{1}{T}\cdot\varepsilon$.

- 时间分析

从上述模拟可知,I 的运行时间主要由下面的时间构成:

(1) I 产生除了阶段 i^*的所有时间段密钥的时间,总共需要的时间不超过 $O(T\log^2T\cdot t_{G_1})$;

(2) 基密钥 O_{SKB} 预言查询的时间,总共需要的时间不超过 $O(\log T\cdot q_{SKB}\cdot t_{G_1})$;

(3) 解密密钥 O_{SKS} 预言查询的时间,总共需要的时间不超过 $O(\log T\cdot q_{SKS}\cdot t_{G_1})$;

(4) 密钥更新 O_{SKU} 预言查询的时间,总共需要的时间不超过 $O(\log^2T\cdot q_{SKU}\cdot t_{G_1})$;

(5) 密钥刷新 O_{SKR} 预言查询的时间,总共需要的时间不超过 $O(\log T\cdot q_{SKR}\cdot t_{G_1})$;

(6) LR 语言查询的时间,总共需要的时间不超过 $O(\log T\cdot t_{G_1})$;

(7) F 完成伪造的时间,该时间为 t.

因此,I 完成伪造总共所需要的时间最大不超过

$$t+O(\log T(T\log T+q_{SKB}+q_{SKS}+\log T\cdot q_{SKU}+q_{SKU}+1)\cdot t_{G_1}).$$

证明完成. □

下面的定理 3 直接来源于文献[14].

定理 3. 任何对标准公钥加密方案的 CCA2 安全转化也是对入侵容忍公钥加密方案的 CCA2 安全转化[14].

由定理 2、定理 3 和文献[20,21]的方法,可以直接获得下面的定理:

定理 4. 我们提出的方案在 l-wBDHI 判定假设下,是标准模型中满足 CCA2 安全的入侵容忍公钥加密方案.

4 效率分析

一般情况下,标准模型下安全的方案效率要低于随机预言模型下安全的方案.然而,我们的方案在标准模型下仍然是十分高效的.由于在第 2 节中已定义 $T=2^l$,因此 $l=\log T$.也就是说,l 次运算总共需要 $O(\log T)$的时间.在密钥产生算法的第①步~第④步中,最多需要 $O(l)$个运算;第⑤步中最多需要 $O(l^2)$个运算.基密钥更新算法平均需要 $O(l)$个运算,复杂度为 $O(\log T)$.同理,解密密钥更新算法复杂度为 $O(\log T)$.基密钥和解密密钥刷新算法的运算不超过 $O(l)$个,复杂性不超过 $O(\log T)$.在加密算法中,计算 $F_1(i_1i_2...i_l)$及计算独立于消息可以预计算,总复杂性为 $O(1)$.解密运算的复杂性为 $O(1)$.公钥、用户密钥(基密钥)、密文长度的复杂性分别为 $O(\log T)$,$O(\log^2T)$和 $O(1)$.另外,我们方案中的加密算法无需配对运算,而文献[14]需要一次配对运算;我们方案中的解密运算中只需 2 次

配对运算,而文献[14]需要 $O(\log T)$次配对运算.我们可以使用文献[4]的方法进一步优化密钥生成算法和密钥更新算法,利用二叉树的前序遍历技术,将二叉树的每个节点(包括中间节点)与各时间段关联,在这种情况下,密钥生成算法和密钥更新算法在最坏情况下均可以在$O(\log T)$时间内完成.公钥的长度也可以进一步减小,其代价是将方案的安全强度降低到随机预言模型中安全.如果令 $g_2,w',w_1,\ldots,w_l$ 从一个随机预言中演化而来,使用另一个随机预言来代替函数 F_1,方案的公钥长度将降低到 $O(1)$bit.表 1 将我们提出的方案、进行上述优化后的方案和文献[14]的随机预言模型下安全的入侵容忍公钥加密方案的性能参数进行了对比.

Table 1 Comparison of full parameters

表 1 完全性能参数比较

	方案运行时间							各参数长度(bit)		
	Setup	*updbase*	*upduser*	*Refbase*	*refuser*	*Encrypt*	*Decrypt*	Public key	User (base) key	Ciphertext
我们的方案	$O(\log^2 T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(1)$	$O(1)$	$O(\log T)$	$O(\log^2 T)$	$O(1)$
优化后的方案	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(1)$	$O(1)$	$O(1)$	$O(\log^2 T)$	$O(1)$
文献[14]的方案	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(1)$	$O(\log T)$	$O(\log T)$

5 结 论

密钥泄漏问题严重威胁着公钥加密体制的安全性.本文提出了一个新的入侵容忍的公钥加密方案,当敌手入侵解密者和基地时,只要不是同时入侵,加密方案的安全性就可以保证;即使敌手同时入侵解密者和基地,也不会影响以前时间段密文的安全性.该方案具有良好的性能,证明了它在方案标准模型下是入侵容忍安全的.

References:

[1] Anderson R. Two remarks on public key cryptology. Invited Lecture, ACM-CCS'97. 1997. http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-549.pdf

[2] Bellare M, Miner S. A forward-secure digital signature scheme. In: Wiener M, ed. Proc. of the CRYPTO'99. LNCS 1666, Berlin: Springer-Verlag, 1999. 431−448. [doi: 10.1007/3-540-48405-1_28]

[3] Abdalla M, Reyzin L. A new forward-secure digital signature scheme. In: Okamoto T, ed. Proc. of the Asiacrypt 2000. LNCS 1976, Berlin: Springer-Verlag, 2000. 116−129. [doi: 10.1007/3-540-44448-3_10]

[4] Canetti R, Halevi S, Katz J. A forward-secure public-key encryption scheme. In: Biham E, ed. Proc. of the EUROCRYPT 2003. LNCS 2656, Berlin: Springer-Verlag, 2003. 255−271. [doi: 10.1007/3-540-39200-9_16]

[5] Gentry C, Silverberg A. Hierarchical ID-based cryptography. In: Zheng Y, ed. Proc. of the ASIACRYPT 2002. LNCS 2501, Berlin: Springer-Verlag, 2002. 548−566. [doi: 10.1007/3-540-36178-2_34]

[6] Dodis Y, Katz J, Xu S, Yung M. Key-Insulated public-key cryptosystems. In: Knudsen LR, ed. Proc. of the EUROCRYPT 2002. LNCS 2332, Berlin: Springer-Verlag, 2002. 65−82. [doi: 10.1007/3-540-46035-7_5]

[7] Dodis Y, Katz J, Xu S, Yung M. Strong key-insulated signature schemes. In: Desmedt Y, ed. Proc. of the Public Key Cryptography 2003. LNCS 2567, Berlin: Springer-Verlag, 2003. 130−144. [doi: 10.1007/3-540-36288-6_10]

[8] Weng J, Li XX, Chen KF, Liu SL. Identity-Based parallel key-insulated encryption without random oracles: Security notions and construction. In: Barua R, Lange T, eds. Proc. of the INDOCRYPT 2006. LNCS 4329, Berlin: Springer-Verlag, 2006. 1143−1157. [doi: 10.1007/11941378_29]

[9] Weng J, Chen KF, Liu SL, Li XX. Identity-Based strong key-insulated signature without random oracles. Ruanjian Xuebao/Journal of Software, 2008,19(6):1555−1564 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/1555.htm [doi: 10.3724/SP.J.1001.2008.01555]

[10] Itkis G, Reyzin L. SiBIR: Signer-Base intrusion-resilient signatures. In: Yung M, ed. Proc. of the CRYPTO 2002. LNCS 2442, Berlin: Springer-Verlag, 2002. 499−514. [doi: 10.1007/3-540-45708-9_32]

[11] Gong Z, Li XX, Zheng D, Chen KF. A generic construction for intrusion-resilient signatures from linear feedback shift register. Journal of Information Science and Engineering, 2008,24(5):1347−1360.

[12] Yu J, Kong FY, Cheng XG, Hao R, Guo XF. A provably secure intrusion-resilient signature scheme. Ruanjian Xuebao/Journal of Software, 2010,21(9):2352−2366 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3772.htm [doi: 10.3724/SP.J.1001.2010.03772]

[13] Yu J, Kong FY, Cheng XG, Hao R, Fan JX. Intrusion-Resilient identity-based signature: Security definition and construction. Journal of Systems and Software, 2012,85(2):382−391. [doi: 10.1016/j.jss.2011.08.034]

[14] Dodis Y, Franklin M, Katz J, Miyaji A, Yung M. Intrusion resilient public-key encryption. In: Joye M, ed. Proc. of the CT-RSA 2003. LNCS 2612, Berlin: Springer-Verlag, 2003. 19−32. [doi: 10.1007/3-540-36563-X_2]

[15] Dodis Y, Franklin M, Katz J, Miyaji A, Yung M. A generic construction for intrusion-resilient public-key encryption. In: Okamoto T, ed. Proc. of the CT-RSA 2004. LNCS 2964, Berlin: Springer-Verlag, 2004. 81−98. [doi: 10.1007/978-3-540-24660-2_7]

[16] Gennaro R, Halevi S, Rabin T. Secure hash-and-sign signatures without the random oracle. In: Proc. of the EUROCRYPT'99. LNCS 1592, Berlin: Springer-Verlag, 1999. 123−139. [doi: 10.1007/3-540-48910-X_9]

[17] Canetti R, Goldreich O, Halevi S. The random oracle methodology, revisited. Journal of the ACM, 2004,51(4):557−594. [doi: 10.1145/1008731.1008734]

[18] Boneh D, Boyen X, Goh E. Hierarchical identity based encryption with constant size ciphertext. In: Cramer R, ed. Proc. of the Eurocrypt 2005. LNCS 3493, Berlin: Springer-Verlag, 2005. 440−456. [doi: 10.1007/11426639_26]

[19] Waters B. Efficient identity-based encryption without random oracles. In: Cramer R, ed. Proc. of the Eurocrypt 2005. LNCS 3494, Berlin: Springer-Verlag, 2005. 114−127. [doi: 10.1007/11426639_7]

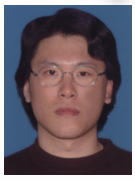
[20] Canetti R, Halevi S, Katz J. Chosen-Ciphertext security from identity-based encryption. In: Camenisch J, ed. Proc. of the Eurocrypt 2004. LNCS 3027, Berlin: Springer-Verlag, 2004. 207−222. [doi: 10.1007/978-3-540-24676-3_13]

[21] Boneh D, Katz J. Improved efficiency for CCA-secure cryptosystems built using identity based encryption. In: Menezes A, ed. Proc. of the RSA-CT 2005. LNCS 3376, Berlin: Springer-Verlag, 2005. 87−103. [doi: 10.1007/978-3-540-30574-3_8]

附中文参考文献:

[9] 翁健,陈克非,刘胜利,李祥学.标准模型下基于身份的强密钥隔离签名.软件学报,2008,19(6):1555−1564. http://www.jos.org.cn/1000-9825/19/1555.htm [doi: 10.3724/SP.J.1001.2008.01555]

[12] 于佳,孔凡玉,程相国,郝蓉,Guo Xiangfa.可证安全的入侵容忍签名方案.软件学报,2010,21(9):2352−2366. http://www.jos.org.cn/1000-9825/3772.htm [doi: 10.3724/SP.J.1001.2010.03772]



于佳(1976－),男,山东青岛人,博士,教授,CCF 高级会员,主要研究领域为密码学,网络信息安全.
E-mail: qduyujia@gmail.com

潘振宽(1966－),男,博士,教授,CCF 高级会员,主要研究领域为图像处理,信息安全.
E-mail: zkpan@qdu.edu.cn

程相国(1969－),男,博士,副教授,主要研究领域为密码学.
E-mail: chengxg@qdu.edu.cn

孔凡玉(1978－),男,博士,副教授,CCF 会员,主要研究领域为密码学.
E-mail: fanyukong@sdu.edu.cn

李发根(1979－),男,博士,副教授,主要研究领域为密码学.
E-mail: fagenli@uestc.edu.cn

郝蓉(1976－),女,讲师,主要研究领域为密码学,网络信息安全.
E-mail: hr@qdu.edu.cn