ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.21, No.9, September 2010, pp.2352−2366
doi: 10.3724/SP.J.1001.2010.03772

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

可证安全的入侵容忍签名方案*

于 佳[1+], 孔凡玉[2,3], 程相国[1], 郝 蓉[1], GUO Xiangfa[4]

[1](青岛大学 信息工程学院,山东 青岛 266071)

[2](山东大学 网络信息安全研究所,山东 济南 250100)

[3](密码技术与信息安全教育部重点实验室,山东 济南 250100)

[4](Department of Computer Science, National University of Singapore, 117590, Singapore)

Intrusion-Resilient Signature Scheme with Provable Security

YU Jia[1+], KONG Fan-Yu[2,3], CHENG Xiang-Guo[1], HAO Rong[1], GUO Xiangfa[4]

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

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

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

[4](Department of Computer Science, National University of Singapore, 117590, Singapore)

+ Corresponding author: E-mail: qduyujia@gmail.com

Yu J, Kong FY, Cheng XG, Hao R, Guo XF. Intrusion-Resilient signature scheme with provable security. *Journal of Software*, 2010,21(9):2352−2366. http://www.jos.org.cn/1000-9825/3772.htm

Abstract: An intrusion-resilient signature scheme with provable security is presented in this paper. The scheme has a stronger security than a forward secure signature scheme and a key-insulated signature scheme. It satisfies that signatures in other time periods are secure even after arbitrarily many compromises of base and signer, as long as the compromises do not happen simultaneously. Furthermore, the intruder cannot generate signatures pertaining to previous time periods, even if she compromises base and signer simultaneously to get their secret information. The scheme has a nice average performance. There is no cost parameter for key generation time, base (user) key updating time, base (user) key refresh time, signing time, verifying time, signature size, public key size, and base (user) storage size having a complexity more than $O(\log T)$ in this scheme. At last, this scheme is proven secure in the random oracle model, assuming CDH problem is hard.

Key words: forward security; proactive security; intrusion-resilient signature; bilinear maps; provable security

摘 要: 提出了一种可证安全的入侵容忍签名方案,方案具有比前向安全签名和密钥隔离签名更强的安全性,满足无论签名者和基地被入侵多少次,只要入侵不是同时发生的,其他任何时间段的签名都是安全的.另外,即使入侵

* Supported by the National Natural Science Foundation of China under Grant No.60703089 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2004AA001260 (国家高技术研究发展计划(863)); the Science and Technology Project of Provincial Education Department of Shandong of China under Grant No.J08LJ02 (山东省教育厅科技计划项目); the Scientific Research Foundation for the Excellent Middle-Aged and Youth Scientists of Shandong Province of China under Grant No.2008BS01011 (山东省优秀中青年科学家科研奖励基金); the Shandong Provincial Natural Science Foundation of China under Grant No.ZR2009GQ008 (山东省自然科学基金)

Received 2008-11-28; Revised 2009-07-06; Accepted 2009-11-09

者同时入侵签名者和基地,获得所有秘密信息,仍然不能伪造以前时间段的签名.方案具有良好的平均性能,所有费用参数包括密钥产生、基密钥(用户密钥)演化、基密钥(用户密钥)更新、签名、验证的时间和签名长度、公钥长度和基地(用户)存储空间大小的复杂性都不超过 $O(\log T)$.最后证明,假定 CDH 问题是困难的,方案在随机预言模型中是安全的.

关键词: 前向安全性;动态安全性;入侵容忍签名;双线性映射;可证安全性

中图法分类号: TP309 **文献标识码**: A

密钥泄漏严重威胁着数字签名的安全,因此,如何减少密钥泄漏对数字签名的危害是一项重要的研究工作.其中一种有效的方法是赋予数字签名前向安全的特性,将系统的整个生命周期划分成若干时间段,每个时间段都通过单向的演化函数演化新密钥,并删除以前的旧密钥,而公钥在整个生命周期中保持不变,这就使得即便当前的密钥泄漏了,以前时间段的签名仍然是有效的.数字签名的前向安全性最早由 Anderson[1]提出,随后文献[2−6]提出了一系列前向安全签名方案.后来,Canetti 等人[7]提出了基于双线性配对的第 1 个前向安全的公钥加密方案,方案使用的是二进制树加密(BTE)的方法,这种方法是分级的基于身份加密(HIBE)[8]的一种变形,受其启发,文献[9−11]提出了基于双线性映射的前向安全签名方案,这些方案具有更好的平均性能,全部费用参数的复杂性均小于 $O(T)$.

但前向安全签名仍然存在不足,它不能保证密钥泄漏以后时间段签名的安全性,由于从入侵成功到密钥泄漏被检测到可能需要一段时间,这就意味着密钥撤销之前,可能有多个时间段的签名是不安全的,入侵者完全可以伪造这些时间段内的签名,所以对于密钥泄漏以后时间段签名的安全性保证显得尤其重要.密钥隔离签名(key-insulated signature)可以解决这个问题,在这种模型中,需要两个模块,一是签名者,另一个是基地(或外围设备).签名者使用它持有的当前时间段的签名密钥进行签名,但在每个时间段结束进行密钥演化时,需要基地的演化信息来计算新的签名密钥.因此假定基地是安全的,即使入侵者得到当前的签名密钥,没有基地的帮助也无法计算密钥泄漏后时间段的签名密钥,文献[12−14]对密钥隔离的签名进行了研究.

近些年来,入侵容忍安全[15−19]作为一种新的方法被提出来,它结合了动态安全、前向安全、密钥隔离安全的方法,具有更高的安全性.与密钥隔离签名一样,在入侵容忍签名中,签名仅需签名用户单独完成,而密钥演化时,用户则需要基地的交互信息来完成.与密钥隔离签名不同,它不需假定基地是安全的,这是因为用户和基地每个时间段都有多个更新密钥的操作,更新可以任意频繁,这就使得即使敌手入侵用户和基地任意多次,只要不是同时入侵,就无法计算其他时间段的签名密钥.另一方面,即使敌手能够同时入侵签名用户和基地,得到所有的秘密信息,也无法伪造之前时间段的签名.Itkis 对入侵容忍签名的一般化工作进行了探讨[15],提出了通用的方法构建入侵容忍签名,但效率不高.文献[16]提出了具体的入侵容忍签名方案,它是基于文献[5]的前向安全方案,安全性依赖于强 RSA 假设,但其密钥产生和基密钥(用户密钥)演化是低效的,复杂性都为 $O(T)$,方案没有明确给出安全性证明.文献[19]提出了另一个新的标准模型下的入侵容忍签名,方案具有更好的性能,所有的性能参数的复杂性都不超过 $O(\log^2 T)$,但方案仍然也没有任何入侵容忍的安全性证明.因此,构建可证安全的、高效的入侵容忍签名方案是一项非常有意义的研究工作[20].

本文基于文献[9]方案,提出了一个可证安全的入侵容忍签名方案.方案不但满足入侵容忍签名的所有属性,而且由于采用二进制树结构,所有的费用参数的复杂性都不超过 $O(\log T)$,这比以前存在的入侵容忍签名方案都要好.基于计算 Diffie-Hellman(CDH)假设,证明了提出的入侵容忍签名方案在随机预言模型下的安全性.由于方案中的双线性映射是建立在特定椭圆曲线上的,因此继承了短签名的优点,签名和密钥的长度都较短.

1 预备知识

1.1 功能定义

入侵容忍签名是一种密钥演化签名,它将签名的整个生命周期划分成 T 个时间段.在这种签名系统中,共有

两个实体:签名者和基地.密钥产生算法(Gen)为系统产生初始化的签名用户密钥、基密钥和公钥.所有签名都是由签名者通过签名算法(Sign)生成的,而基地并不直接参与签名,它只用来帮助签名者进行密钥演化和更新.任何人都可以利用公钥 PK 通过验证算法(Ver)来验证签名的正确性.整个系统的密钥通过两种算法(密钥演化算法和密钥更新算法)进行更改.密钥演化算法包括基密钥演化算法(UB)和签名用户密钥演化算法(US),它们在每个时间段结束时执行.密钥更新算法包括基密钥更新算法(RB)和签名用户密钥更新算法(RS),它们可以在每个时间段内频繁使用.通过密钥演化算法和密钥更新算法不断地更新签名者和基地的密钥,使得即使敌手入侵用户和基地任意多次,只要不是同时入侵,就无法计算其他时间段的签名密钥,从而保证其他时间段签名的安全性.

令 $SKS_{i,r}$ 和 $SKB_{i,r}$ 分别表示第 i 时间段第 r 次更新后的签名用户密钥和基地密钥(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 $i,r<i',r'$.仍然遵循文献[2]的定义,密钥产生之后,紧跟着一个密钥演化算法来得到 $i=1$ 的密钥,密钥演化之后,再紧跟着一个密钥更新算法来得到 $r=1$ 的密钥.下面的定义来源于文献[16],并进行了一些修改.

定义 1. 密钥演化签名方案是一个由 PPT 算法构成的七元组(Gen,Sign,Ver,UB,US,RB,RS).

1. Gen, 密钥产生算法:

输入:安全参数 k 和总共时间段数 T.

输出:初始化的签名用户密钥 $SKS_{0,0}$,初始化的基密钥 $SKB_{0,0}$,公钥 PK.

2. Sign, 签名算法:

输入:当前的签名用户密钥 $SKS_{i,r}$,消息 M.

输出:第 i 时间段对 M 的签名$\langle i,Sig\rangle$.

3. Ver,验证算法:

输入:消息 M,签名$\langle i,Sig\rangle$和公钥 PK.

输出:“有效”或“无效”.

4. UB,基密钥演化算法:

输入:当前的基密钥 $SKB_{i,r}$.

输出:新的基密钥 $SKB_{i+1,0}$ 和密钥演化信息 SKU_i.

5. US,签名用户密钥演化算法:

输入:当前的签名用户密钥 $SKS_{i,r}$ 和密钥演化信息 SKU_i.

输出:新的签名用户密钥 $SKS_{i+1,0}$.

6. RB,基密钥更新算法:

输入:当前的基密钥 $SKB_{i,r}$.

输出:新的基密钥 $SKB_{i,r+1}$ 和对应的密钥更新消息 $SKR_{i,r}$.

7. RS,签名用户密钥更新算法:

输入:当前的签名用户密钥 $SKS_{i,r}$ 和密钥更新信息 $SKR_{i,r}$.

输出:新的签名用户密钥 $SKS_{i,r+1}$.

上述 7 个算法的作用和关系如下:

(1) 算法 Gen 在整个系统建立时执行,用来产生初始化的签名用户密钥、基密钥和公钥.生成的签名用户密钥可以作为算法 Sign 的一个输入用来产生有效签名,也可以作为算法 US 和 UR 的一个输入用来产生新的签名用户密钥.生成的基密钥可以作为算法 UB 的输入用来产生新的基密钥和演化信息,也可以作为算法 RB 的输入用来产生新的基密钥和对应的更新消息.

(2) 算法 Sign 在生成签名时执行,使用签名用户的密钥生成某一时间段对消息的签名,与标准数字签名不同,生成的签名还应包含其所在的时间阶段 i.

(3) 算法 Ver 在签名验证时执行,不仅验证签名的有效性,还要验证生成签名所在时间阶段的有效性.

(4) 算法 UB 在某个时间段结束时执行,它可以产生新的基密钥和密钥演化信息.产生的密钥演化信息作为算法 US 的一个输入可以用来演化下一时间段的签名用户密钥.

(5) 算法 US 也在某个时间段结束时执行(执行在算法 UB 之后),使用当前签名用户密钥和算法 UB 生成的密钥演化信息来演化下一时间段的签名用户密钥.

(6) 算法 RB 在每个时间段内执行,它可以产生新的基密钥和对应的密钥更新消息.密钥更新消息作为算法 RS 的一个输入用来更新签名用户密钥.

(7) 算法 RS 也在某个时间内执行(执行在算法 RB 之后),通过当前签名用户密钥和算法 RB 生成的密钥更新消息来更新签名用户密钥.

1.2 安全性定义

令 $RN(i)$表示第 i 时间段的密钥更新次数,因为假定每次密钥演化后都紧跟着一个密钥更新,密钥产生后再紧跟着一个密钥演化,所以 r=0 和 i=0 不会真正使用,RN 的使用是为了表达方便,无须事先知道.下列实验表示了在方案的整个生命周期中所有密钥信息的产生.

Experiment $Gen\text{-}keys(k,T,RN)$

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

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

for $i = 1$ to T

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

$SKS_{i,0} \leftarrow US(SKS_{i-1,r}, SKU_{i-1})$

for $r = 1$ to $RN(i)$

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

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

令 $SKS^*, SKB^*, SKU^*, SKR^*$ 分别表示上述实验中所产生的签名用户密钥、基密钥、演化信息、更新信息构成的集合.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,能够访问下面预言:

$-Osig$,签名预言:输入(m,i,r) $(1 \leqslant i \leqslant T, 1 \leqslant r \leqslant RN(i))$,输出$Sign(SKS_{i,r}, m)$

$-O\text{sec}$,密钥泄漏预言:1. 输入$("s",i,r), 1 \leqslant i \leqslant T, 1 \leqslant r \leqslant RN(i)$,输出$SKS_{i,r}$;

2. 输入$("b",i,r), 1 \leqslant i \leqslant T, 1 \leqslant r \leqslant RN(i)$,输出$SKB_{i,r}$;

3. 输入$("u",i), 1 \leqslant i \leqslant T$,输出$SKU_i$和$SKR_{i+1,0}$;

4. 输入$("r",i,r), 1 \leqslant i \leqslant T, 1 \leqslant r \leqslant RN(i)$,输出$SKR_{i,r}$.

查询这些预言表示敌手攻破签名用户或基地,或者获取演化或更新的信息.

对于任意的密钥泄漏查询集合 Q,当下列条件成立之一时,我们称 $SKS_{i,r}$ 是 Q-泄漏:

$-("s",i,r) \in Q$;

$-r > 1, ("r",i,r-1) \in Q$,且$SKS_{i,r-1}$是$Q$-泄漏;

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

显然,$SKS_{i,r}$ 的泄漏可以使敌手产生阶段 i 的有效签名.以上述同样方法可以定义 $SKB_{i,r}$ 的 Q-泄漏.敌手如果同时获得签名用户和基地的密钥$(SKS_{i,r}, SKB_{i,r})$,则它可以伪造所有第 $i'(i' \geqslant i)$ 时间段的有效签名.

因此,如果下面两个条件任意一个成立,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$SKB_{i',r}$都是Q-泄漏,且$i' \leqslant i$.

下面实验给出了敌手的模型,如果一个敌手能伪造一个有效的签名:不是通过查询 Osig 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)

$(m,j,Sig)\leftarrow F^{H,Osig,O\text{sec}}(1^k,T,PK,RN)$

令Q是F进行Osec密钥泄漏查询的集合

if *Ver*(m,j,Sig)="无效"或者(m,j)已经被F进行Osig查询了

或者为非法查询或者方案是(j,Q)-被入侵的

then return (0)

else return (1)

下面给出入侵容忍安全性定义:

定义 2. 令 IRSIG=(Gen,Sign,Ver;UB,US;RB;RS)是一个密钥演化签名方案,k 为安全参数,T 为总共的时间段数,RN 表示 T 个时间段中密钥更新次数构成的数组,H 表示一个随机预言,F 表示上述描述的敌手.令 $Succ^{IR}(IRSIG[k,T,RN],F)$表示上述实验返回 1 的概率,则方案 IRSIG 的不安全性定义为函数

$$Insec^{IR}(IRSIG[k,T,RN]),t,q_{sig},q_{hash})=\max_{F}\{Succ^{IR}(IRSIG[k,T,RN],F)\}.$$

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,进行的签名预言查询最多为 q_{sig} 次,进行的随机 H-预言查询最多为 q_{hash} 次.如果 $Insec^{IR}(IRSIG[k,T,RN]),t,q_{sig},q_{hash})<\varepsilon$,则称 IRSIG[$k$,$T$,RN])签名方案是$(t,\varepsilon,q_{sig},q_{hash})$入侵容忍的.

以上定义主要来源于文献[16],并经过简单的修改.

1.3 密码知识和假设

令 G_1 是阶为素数 q 的加法循环群,G_2 是阶为素数 q 的乘法循环群,$P\in G_1$ 是 G_1 的一个生成元.双线性映射 $e:G_1\times G_1\to G_2$,满足:

1. 双线性:对于任意 $P,Q\in G_1,a,b\in Z_q$,满足: $e(aP,bQ)=e(P,Q)^{ab}$.
2. 非退化性:存在 $P,Q\in G_1$,满足: $e(P,Q)\neq 1$.
3. 可计算性:存在一个有效算法,对任意的 $P,Q\in G_1$,可以计算 $e(P,Q)$.

计算 Diffie-Hellman(CDH)问题:给定 (P,aP,bP), $a,b\in Z_q$,计算 abP.算法 A 解决群 G_1 上的 CDH 问题的优势概率为 $AdvCDH_A=\Pr\left[A(P,aP,bP)=abP\mid a,b\overset{R}{\leftarrow}Z_q\right]$.

定义 3(CDH 假设). 概率算法 A 攻击群 G_1 上的 CDH 问题,如果使用的时间最多为 t,获得的优势概率 $AdvCDH_A\geqslant\varepsilon$,就称算法 $A(t,\varepsilon)$攻击群 G_1 上的 CDH 问题.如果不存在概率算法能够(t,ε)攻击群 G_1 上的 CDH 问题,就称群 G_1 是(t,ε)-CDH 群.

IΓ表示 CDH 参数产生器,输入安全参数 k,产生素数 q 阶群 G_1,G_2,所需要的双线性映射 $e:G_1\times G_1\to G_2$.说 IΓ满足 CDH 假设(方案中使用的),当且仅当对于 PPT 算法 A 下列概率是可忽略的:

$$\Pr[A(G_1,G_2,e,P,aP,bP)=abP\mid(G_1,G_2,e)\leftarrow I\Gamma(1^k),P\leftarrow G_1^*,a,b\leftarrow Z_q].$$

2 入侵容忍签名方案

2.1 符号表示和相关说明

方案使用的是二进制树的结构,同样的结构已广泛应用在其他密码方案[7,9]中.为方便起见,定义总共的时间段个数 T 为 2 的幂的形式($T=2^l$),使用一个深度为 l 的满二进制树,每个节点表示一个二进制串ζ,根节点表示空二进制串$\zeta=\varepsilon$,若节点ζ所在的深度小于 l,它的左右儿子节点表示为$\zeta 0$ 和$\zeta 1$,令每个时间段 $i(0\leqslant i\leqslant T-1)$ 的 l 位二进制表示为 $\langle i\rangle=i_1i_2...i_l$,显然二进制树从左到右的每个叶节点$\langle i\rangle$可以表示每个时间段 i.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.中间节点ζ二进制表示为 $\zeta=\zeta_1\zeta_2...\zeta_j(1\leqslant j<l)$,$j$ 为节点ζ所在的层数.

根节点ε包含根密钥s_ε和根验证点Q,每个中间节点ζ都包含一个节点秘密s_ζ,一个局部秘密$S_\zeta \in G_1$和一个验证点Q_ζ.每个叶节点$\langle i\rangle$的局部密钥$sk_{\langle i\rangle}=(S_{\langle i\rangle},\mathcal{Q}_{\langle i\rangle})$,其中$\mathcal{Q}_{\langle i\rangle}=(Q_{i_1},...,Q_{i_1 i_2 ... i_{l-1}},Q_{\langle i\rangle})$.所有的这些验证点并非真正的秘密,而将作为签名的一部分公开.

令叶节点$\langle i\rangle = i_0 i_1 i_2 ... i_l$(这里,$i_0=\varepsilon$),阶段$i$的完全密钥$SK_i$包括(1) $s_{\langle i\rangle}$ (2) $sk_{\langle i\rangle}$ (3) $\{sk_{i_0 i_1 ... i_{j-1}1}\}$(每个$i_j=0,1 \leqslant j \leqslant l$).前两部分分别是叶结点秘密和局部密钥.对于每个$i_j=0 (1 \leqslant j \leqslant l)$,内节点$\zeta 0 = i_0 i_1 ... i_j$都有一个右兄弟节点$\zeta 1 = i_0 i_1 ... i_{j-1} 1$,我们用$\psi(i)$表示对应节点$i$的所有这些右兄弟节点构成的集合.表示$SK_i$为$\{s_{\langle i\rangle}, S_{\langle i\rangle}, Set_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}\}$,其中$Set_{\langle i\rangle}=\{S_\zeta \mid \zeta \in \psi(i)\}$.由$Set_{\langle i\rangle}$可以演化所有$i'>i$时间段的完全密钥$SK_{i'}$,而无法演化之前时间段的完全密钥,而由密钥$s_{\langle i\rangle}, S_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}$可以计算所需签名.我们令基密钥为$Set'_{\langle i\rangle}=\{S'_\zeta \mid \zeta \in \psi(i)\}$,签名用户密钥为$\{s_{\langle i\rangle}, S_{\langle i\rangle}, Set''_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}\}$,其中$Set''_{\langle i\rangle}=\{S''_\zeta \mid \zeta \in \psi(i)\}$,并且满足$S_\zeta = S'_\zeta + S''_\zeta (\zeta \in \psi(i))$.签名用户密钥中$s_{\langle i\rangle}, S_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}$用来产生签名,而签名用户密钥的$Set''_{\langle i\rangle}$和基密钥$Set'_{\langle i\rangle}$用来共同演化下一阶段的密钥.在密钥更新时:对于每个$\zeta \in \psi(i)$,基地只需要选择$R_\zeta \in_R G_1$,计算$S'_\zeta = S'_\zeta - R_\zeta$,将$R_\zeta$传送给用户;对于每个$\zeta \in \psi(i)$,用户计算$S''_\zeta = S''_\zeta + R_\zeta$,这样仍然满足$S_\zeta = S'_\zeta + S''_\zeta$.密钥演化时,基地利用$SKB_{i,r}=Set'_{\langle i\rangle}$,计算$SKB_{i+1,0}=Set'_{\langle i+1\rangle}$,计算$s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}$等演化信息并将其传送给用户;用户利用$SKS_{i,r}$和$s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}$等演化信息,计算下一阶段的密钥.

2.2 方案描述

(1) 算法$Gen(k,l,T)$

运行IG(1^k)产生阶为素数q的加法群G_1和乘法群G_2,及双线性映射$e: G_1 \times G_1 \to G_2$.

选择生成元$P \in_R G_1$和秘密$s_\varepsilon \in_R Z_q$,令$Q = s_\varepsilon P$.

选择密码hash函数$H: \{0,1\}^* \to G_1$.

令公钥$PK=\{G_1, G_2, e, H, P, Q\}$.计算$S_0 = s_\varepsilon H(0), S_1 = s_\varepsilon H(1)$.

For j=1 to l−1

选择$s_{0^j} \in_R Z_q$,计算$Q_{0^j} = s_{0^j} P$.

计算$S_{0^j 0} = S_{0^j} + s_{0^j} H(0^j 0)$, $S_{0^j 1} = S_{0^j} + s_{0^j} H(0^j 1)$.

选择$s_{0^l} \in_R Z_q$,计算$Q_{0^l} = s_{0^l} P$.

令$SK_0 = \{s_{0^l}, S_{0^l}, Set_{0^l}, \mathcal{Q}_{0^l}\}$,这里,$\mathcal{Q}_{0^l}=(Q_0,...,Q_{0^{l-1}},Q_{0^l})$, $Set_{0^l}=(S_1, S_{01},..., S_{0^{l-1}1})$.

选择$S'_{0^{j-1}1} \in_R G_1 (1 \leqslant j \leqslant l)$,计算$S''_{0^{j-1}1} = S_{0^{j-1}1} - S'_{0^{j-1}1}$.

令$Set'_{0^l}=(S'_1, S'_{01},..., S'_{0^{l-1}1})$, $Set''_{0^l}=(S''_1, S''_{01},..., S''_{0^{l-1}1})$.

令$SKB_{0,0}=Set'_{0^l}$, $SKS_{0,0}=\{s_{0^l}, S_{0^l}, Set''_{0^l}, \mathcal{Q}_{0^l}\}$.

删除所有中间数据,返回$PK, SKS_{0,0}, SKB_{0,0}$.

(2) 算法$UB(SKB_{i,r})$

If i=T−1 then

令$SKB_{i+1,0}=\varnothing$, $SKU_i=\varnothing$.

Else

解析$\langle i\rangle = i_0 i_1 i_2 ... i_l$, $(i_0=\varepsilon)$, $SKB_{i,r}=Set'_{\langle i\rangle}=\{S'_\zeta \mid \zeta \in \psi(i)\}$.

If i_l=0 then

从$Set'_{\langle i\rangle}$中找到$S'_{\langle i+1\rangle}$,令$Set'_{\langle i+1\rangle}=Set'_{\langle i\rangle}-\{S'_{\langle i+1\rangle}\}$.

选择$s'_{\langle i+1\rangle} \in_R Z_q$,计算$Q'_{\langle i+1\rangle}=s'_{\langle i+1\rangle}P$.

令$SKB_{i+1,0}=Set'_{\langle i+1\rangle}$, $SKU_i=(s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}, Q'_{\langle i+1\rangle})$.

Else

找到满足$i_j=0$的最大值$j (1 \leqslant j < l)$,令$\eta = i_0 i_1 ... i_{j-1} 1$, $(i_0=\varepsilon)$,显然$\langle i+1\rangle = \eta 0^{l-j}$.

从$Set'_{\langle i\rangle}$中找到S'_η,令$Set'_{\langle i+1\rangle}=Set'_{\langle i\rangle}-\{S'_\eta\}$.

For m=1 to l–j

选择 $s'_{\eta 0^{m-1}} \in_R Z_q$，计算 $Q'_{\eta 0^{m-1}} = s'_{\eta 0^{m-1}} P$，$S'_{\eta 0^m} = S'_{\eta 0^{m-1}} + s'_{\eta 0^{m-1}} H(\eta 0^m)$ 和 $S'_{\eta 0^{m-1}1} = S'_{\eta 0^{m-1}} + s'_{\eta 0^{m-1}} H(\eta 0^{m-1}1)$.

令 $Set'_{\langle i+1\rangle} = Set'_{\langle i+1\rangle} \cup \{S'_{\eta 0^{m-1}1}\}$.

选择 $s'_{\langle i+1\rangle} \in_R Z_q$，计算 $Q'_{\langle i+1\rangle} = s'_{\langle i+1\rangle} P$.

令 $SKB_{i+1,0} = Set'_{\langle i+1\rangle}$，$SKU_i = (s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}, Q'_\eta, Q'_{\eta 0}, ..., Q'_{\eta 0^{l-j-1}}, Q'_{\langle i+1\rangle})$.

删除所有中间数据，返回 $SKB_{i+1,0}$，SKU_i.

(3) 算法 $US(SKS_{i,r}, SKU_i)$

If i=T–1 then

令 $SKS_{i+1,0} = \varnothing$.

Else

解析 $\langle i\rangle = i_0 i_1 i_2 ... i_l$, $(i_0 = \varepsilon)$，$SKS_{i,r} = \{s_{\langle i\rangle}, S_{\langle i\rangle}, Set''_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}\}$，$\mathcal{Q}_{\langle i\rangle} = (Q_{i_1}, ..., Q_{i_1 i_2 ... i_{l-1}}, Q_{\langle i\rangle})$.

If i_l=0 then

解析 $SKU_i = (s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}, Q'_{\langle i+1\rangle})$.

从 $Set''_{\langle i\rangle}$ 中找到 $S''_{\langle i+1\rangle}$，令 $Set''_{\langle i+1\rangle} = Set''_{\langle i\rangle} - \{S''_{\langle i+1\rangle}\}$.

选择 $s''_{\langle i+1\rangle} \in_R Z_q$，计算 $Q''_{\langle i+1\rangle} = s''_{\langle i+1\rangle} P$.

计算 $s_{\langle i+1\rangle} = s'_{\langle i+1\rangle} + s''_{\langle i+1\rangle}$，$S_{\langle i+1\rangle} = S'_{\langle i+1\rangle} + S''_{\langle i+1\rangle}$，$\mathcal{Q}_{\langle i+1\rangle} = (\mathcal{Q}_{\langle i\rangle} - \{Q_{\langle i\rangle}\}) \cup \{Q'_{\langle i+1\rangle} + Q''_{\langle i+1\rangle}\}$.

令 $SKS_{i+1,0} = (s_{\langle i+1\rangle}, S_{\langle i+1\rangle}, Set''_{\langle i+1\rangle}, \mathcal{Q}_{\langle i+1\rangle})$.

Else

找到满足 i_j=0 的最大值 $j(1 \leqslant j < l)$，令 $\eta = i_0 i_1 ... i_{j-1} 1$, $(i_0 = \varepsilon)$，显然 $\langle i+1\rangle = \eta 0^{l-j}$.

解析 $SKU_i = (s'_{\langle i+1\rangle}, S'_{\langle i+1\rangle}, Q'_\eta, Q'_{\eta 0}, ..., Q'_{\eta 0^{l-j-1}}, Q'_{\langle i+1\rangle})$.

从 $Set''_{\langle i\rangle}$ 中找到 S''_η，令 $Set''_{\langle i+1\rangle} = Set''_{\langle i\rangle} - \{S''_\eta\}$.

令 $\mathcal{Q}_{\langle i+1\rangle} = \mathcal{Q}_{\langle i\rangle} - \{Q_{i_1 i_2 ... i_j}, Q_{i_1 i_2 ... i_j 1}, ..., Q_{i_1 i_2 ... i_j 1^{l-j-1}}, Q_{\langle i\rangle}\}$.

For m=1 to l–j

选择 $s''_{\eta 0^{m-1}} \in_R Z_q$，计算 $Q''_{\eta 0^{m-1}} = s''_{\eta 0^{m-1}} P$，$S''_{\eta 0^m} = S''_{\eta 0^{m-1}} + s''_{\eta 0^{m-1}} H(\eta 0^m)$ 和 $S''_{\eta 0^{m-1}1} = S''_{\eta 0^{m-1}} + s''_{\eta 0^{m-1}} H(\eta 0^{m-1}1)$.

令 $Set''_{\langle i+1\rangle} = Set''_{\langle i+1\rangle} \cup \{S''_{\eta 0^{m-1}1}\}$，$\mathcal{Q}_{\langle i\rangle} = \mathcal{Q}_{\langle i\rangle} \cup \{Q'_{\eta 0^{m-1}} + Q''_{\eta 0^{m-1}}\}$.

计算 $S_{\langle i+1\rangle} = S'_{\langle i+1\rangle} + S''_{\langle i+1\rangle}$.

选择 $s''_{\langle i+1\rangle} \in_R Z_q$，计算 $Q''_{\langle i+1\rangle} = s''_{\langle i+1\rangle} P$.

令 $s_{\langle i+1\rangle} = s'_{\langle i+1\rangle} + s''_{\langle i+1\rangle}$，$\mathcal{Q}_{\langle i+1\rangle} = \mathcal{Q}_{\langle i+1\rangle} \cup \{Q'_{\langle i+1\rangle} + Q''_{\langle i+1\rangle}\}$.

令 $SKS_{i+1,0} = (s_{\langle i+1\rangle}, S_{\langle i+1\rangle}, Set''_{\langle i+1\rangle}, \mathcal{Q}_{\langle i+1\rangle})$.

删除所有中间数据，返回 $SKS_{i+1,0}$.

(4) 算法 $RB(SKB_{i,r})$

解析 $SKB_{i,r} = Set'_{\langle i\rangle} = \{S'_\zeta \mid \zeta \in \psi(i)\}$.

对于每个 $\zeta \in \psi(i)$，选择 $R_\zeta \in_R G_1$，令 $SKB_{i,r+1} = \{S'_\zeta - R_\zeta \mid \zeta \in \psi(i)\}$，$SKR_{i,r} = \{R_\zeta \mid \zeta \in \psi(i)\}$.

删除所有中间数据，返回 $SKB_{i,r+1}$，$SKR_{i,r}$.

(5) 算法 $RS(SKS_{i,r}, SKR_{i,r})$

解析 $SKS_{i,r} = \{s_{\langle i\rangle}, S_{\langle i\rangle}, Set''_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}\}$，$Set''_{\langle i\rangle} = \{S''_\zeta \mid \zeta \in \psi(i)\}$，$SKR_{i,r} = \{R_\zeta \mid \zeta \in \psi(i)\}$.

令 $Set''_{\langle i\rangle} = \{S''_\zeta + R_\zeta \mid \zeta \in \psi(i)\}$，$SKS_{i,r+1} = \{s_{\langle i\rangle}, S_{\langle i\rangle}, Set''_{\langle i\rangle}, \mathcal{Q}_{\langle i\rangle}\}$.

删除所有中间数据,返回 $SKS_{i,r+1}$.

(6) 算法 $Sign(SKS_{i,r},M)$

解析 $\langle i\rangle=i_1i_2...i_l$, $SKS_{i,r}=\{s_{\langle i\rangle},S_{\langle i\rangle},Set''_{\langle i\rangle},\mathcal{Q}_{\langle i\rangle}\}$.计算 $V=S_{\langle i\rangle}+s_{\langle i\rangle}H(i_1i_2...i_lM)$.

返回签名 $\langle i,sign=(V,\mathcal{Q}_{\langle i\rangle})\rangle$.

(7) 算法 $Ver(M,PK,\langle i,sig\rangle)$

解析 $\langle i\rangle=i_1i_2...i_l$, $sign=(V,\mathcal{Q}_{\langle i\rangle}),\mathcal{Q}_{\langle i\rangle}=(Q_{i_1},...,Q_{i_1i_2...i_{l-1}},Q_{\langle i\rangle})$.

验证: $e(P,V)\overset{?}{=}e(Q,H(i_1))\cdot e(Q_{\langle i\rangle},H(i_1i_2...i_lM))\cdot\prod_{j=2}^{l}e(Q_{i_1i_2...i_{j-1}},H(i_1i_2...i_j))$

若成立,返回“有效”;否则,返回“无效”.

3 性能分析

因为提出的方案是采用二进制树结构,所以具有这种结构独特的优势.具体来说,方案具有良好的平均性能,所有的性能参数包括密钥产生、基密钥(用户密钥)演化、基密钥(用户密钥)更新、签名、验证的时间和公钥、私钥和签名的长度的 T 项复杂性都不超过 $O(\log T)$.而入侵容忍签名方案[16,19]不具有这种效率,文献[16]的方案的 T 项复杂性为 $O(T)$,文献[19]的方案的 T 项复杂性则为 $O(\log^2 T)$.另外,由于方案中的双线性映射是建立在特定椭圆曲线上的,因此继承了基于双线性配对短签名的特点,当 T 不是特别大时,签名和各种密钥的长度都很短.验证算法很大程度上依赖于双线性配对的运算效率,使用一个好的配对运算算法,可以获得较好的验证效率.

下面分析构成提出方案的 7 个算法所需要的运算次数.分析时,考虑的是 G_1 中的乘法、加法和减法运算,G_2 中的乘法运算,以及配对运算.而对于像集合中增加元素和删除元素之类简单运算则忽略不计(时间远远小于其他运算).

(1) 密钥产生算法

计算 $Q=s_\varepsilon P$ 需要 1 次 G_1 中的乘法运算,计算 $S_0=s_\varepsilon H(0),S_1=s_\varepsilon H(1)$ 需要 2 次 G_1 中的乘法运算.For j=1 to l–1,计算 $Q_{0^j}=s_{0^j}P$, $S_{0^j0}=S_{0^j}+s_{0^j}H(0^j0)$, $S_{0^j1}=S_{0^j}+s_{0^j}H(0^j1)$,需要 3($l$–1)次 G_1 中的乘法运算和 2(l–1)次 G_1 中的加法运算.计算 $Q_{0^l}=s_{0^l}P$ 需要 1 次 G_1 中的乘法运算.计算 $S''_{0^{j-1}1}=S_{0^{j-1}1}-S'_{0^{j-1}1}$ ($1\leqslant j\leqslant l$) 需要 l 次 G_1 中的减法运算.

所以,该算法总共需要 3l+1 次 G_1 中的乘法运算,2(l–1)次 G_1 中的加法运算和 l 次 G_1 中的减法运算.

(2) 基密钥演化算法.

(a) 当 $i=T-1$ 时,不需任何耗时运算.

(b) 当 $i\neq T-1$ 时,

(I) 当 $i_l=0$ 时,

$Set'_{\langle i+1\rangle}=Set'_{\langle i\rangle}-\{S'_{\langle i+1\rangle}\}$ 是集合中删除元素,时间可以忽略不计.

计算 $Q'_{\langle i+1\rangle}=s'_{\langle i+1\rangle}P$ 需要 1 次 G_1 中的乘法运算.

(II) 当 $i_l\neq 0$ 时,

$Set'_{\langle i+1\rangle}=Set'_{\langle i\rangle}-\{S'_\eta\}$ 是集合中删除元素,时间可忽略不计.

For m=1 to l–j,计算 $Q'_{\eta 0^{m-1}}=s'_{\eta 0^{m-1}}P$, $S'_{\eta 0^m}=S'_{\eta 0^{m-1}}+s'_{\eta 0^{m-1}}H(\eta 0^m)$ 和 $S'_{\eta 0^{m-1}1}=S'_{\eta 0^{m-1}}+s'_{\eta 0^{m-1}}H(\eta 0^{m-1}1)$ 需要 3(l–j)次 G_1 中的乘法运算和 2(l–j)次 G_1 中的加法运算. $Set'_{\langle i+1\rangle}=Set'_{\langle i+1\rangle}\cup\{S'_{\eta 0^{m-1}1}\}$ 是集合中增加元素,时间可忽略不计.因为 $1\leqslant j<l$,因此,当 j=1 时,这个循环语句的运算最多需要 3(l–1)次 G_1 中的乘法运算和 2(l–1)次 G_1 中的加法运算.

计算 $Q'_{\langle i+1\rangle}=s'_{\langle i+1\rangle}P$ 需要 1 次 G_1 中的乘法运算.

因此,这种情况下最多需要 3l–2 次 G_1 中的乘法运算和 2(l–1)次 G_1 中的加法运算.

所以,此算法最多需要 $3l-2$ 次 G_1 中的乘法运算和 $2(l-1)$次 G_1 中的加法运算.

(3) 签名用户密钥演化算法

(a) 当 $i=T-1$ 时,不需任何耗时运算.

(b) 当 $i\neq T-1$ 时:

(I) 当 $i_l=0$ 时,

$Set''_{\langle i+1\rangle}=Set''_{\langle i\rangle}-\{S''_{\langle i+1\rangle}\}$ 是集合中删除元素,时间可忽略不计.

计算 $Q''_{\langle i+1\rangle}=s''_{\langle i+1\rangle}P$ 需要 1 次 G_1 中的乘法运算.

计算 $s_{\langle i+1\rangle}=s'_{\langle i+1\rangle}+s''_{\langle i+1\rangle}$, $S_{\langle i+1\rangle}=S'_{\langle i+1\rangle}+S''_{\langle i+1\rangle}$,$\mathcal{Q}_{\langle i+1\rangle}=(\mathcal{Q}_{\langle i\rangle}-\{Q_{\langle i\rangle}\})\cup\{Q'_{\langle i+1\rangle}+Q''_{\langle i+1\rangle}\}$ 需要 3 次 G_1 中的加法运算,其中 $\mathcal{Q}_{\langle i\rangle}-\{Q_{\langle i\rangle}\}$ 是集合中删除元素,时间可忽略不计.

(II) 当 $i_l\neq 0$ 时,

$Set''_{\langle i+1\rangle}=Set''_{\langle i\rangle}-\{S''_{\eta}\}$ 和 $\mathcal{Q}_{\langle i+1\rangle}=\mathcal{Q}_{\langle i\rangle}-\{Q_{i_1i_2\ldots i_j},Q_{i_1i_2\ldots i_j1},\ldots,Q_{i_1i_2\ldots i_j1^{l-j-1}},Q_{\langle i\rangle}\}$ 是集合中删除元素,时间可以忽略不计.

For $m=1$ to $l-j$, $Q''_{\eta 0^{m-1}}=s''_{\eta 0^{m-1}}P$, $S''_{\eta 0^m}=S''_{\eta 0^{m-1}}+s''_{\eta 0^{m-1}}H(\eta 0^m)$ 和 $S''_{\eta 0^{m-1}1}=S''_{\eta 0^{m-1}}+s''_{\eta 0^{m-1}}H(\eta 0^{m-1}1)$ 共需要 $3(l-j)$次 G_1 中的乘法运算和 $2(l-j)$次 G_1 中的加法运算. $Set''_{\langle i+1\rangle}=Set''_{\langle i+1\rangle}\cup\{S''_{\eta 0^{m-1}1}\}$ 是集合中增加元素,时间可以忽略不计.计算 $\mathcal{Q}_{\langle i+1\rangle}=\mathcal{Q}_{\langle i+1\rangle}\cup\{Q'_{\eta 0^{m-1}}+Q''_{\eta 0^{m-1}}\}$ 共需要 $l-j$ 次 G_1 中的加法运算.因为 $1\leqslant j<l$,因此,当 $j=1$ 时,这个循环语句需要的运算最多,共需要 $3(l-1)$次 G_1 中的乘法运算和 $3(l-1)$次 G_1 中的加法运算.

计算 $S_{\langle i+1\rangle}=S'_{\langle i+1\rangle}+S''_{\langle i+1\rangle}$ 需要 1 次 G_1 中的加法运算,计算 $Q''_{\langle i+1\rangle}=s''_{\langle i+1\rangle}P$ 需要 1 次 G_1 中的乘法运算.

计算 $s_{\langle i+1\rangle}=s'_{\langle i+1\rangle}+s''_{\langle i+1\rangle}$,$\mathcal{Q}_{\langle i+1\rangle}=\mathcal{Q}_{\langle i+1\rangle}\cup\{Q'_{\langle i+1\rangle}+Q''_{\langle i+1\rangle}\}$ 需要 2 次 G_1 中的加法运算.

所以,此算法最多需要 $3l-2$ 次 G_1 中的乘法运算和 $3l$ 次 G_1 中的加法运算.

(4) 基密钥更新算法

因为 $\psi(i)$中元素不会超过 l 个,所以,计算 $SKB_{i,r+1}=\{S'_\zeta-R_\zeta\mid\zeta\in\psi(i)\}$ 最多需要 l 次 G_1 中的减法运算.

(5) 签名用户密钥更新算法

因为 $\psi(i)$中元素不会超过 l 个,所以,计算 $Set''_{\langle i\rangle}=\{S''_\zeta+R_\zeta\mid\zeta\in\psi(i)\}$ 最多需要 l 次 G_1 中的加法运算.

(6) 签名算法

计算 $V=S_{\langle i\rangle}+s_{\langle i\rangle}H(i_1i_2\ldots i_lM)$ 需要 1 次 G_1 中的加法运算和 1 次 G_1 中的乘法运算.

(7) 验证算法

需要 l 次 G_2 中的乘法运算和 $l+2$ 次配对运算.

正如文献[9−11,19]分析的一样,影响系统性能的最主要因素是关于总共时间段个数 T 项的复杂性.下面,我们分析提出方案各性能参数的 T 项复杂性.注意:常数次 G_1 中的乘法、加法或减法运算,常数次 G_2 中的乘法运算,以及常数次配对运算所需要的运行时间 T 项复杂性为 $O(1)$.$O(\log T)$次上述运算的运行时间 T 项复杂性为 $O(\log T)$.

命题 1. 提出的入侵容忍签名方案整个性能的复杂性为 $O(\log T)$,即所有上述 7 个算法的运行时间,公钥、私钥和签名长度的 T 项复杂性均不超过 $O(\log T)$.

证明:

由第 2.1 节的定义可知总共的时间段个数 $T=2^l$.因此,$l=\log T$.

由前面的分析可以知道:

(1) 密钥产生算法总共需要 $3l+1$ 次 G_1 中的乘法运算,$2(l-1)$次 G_1 中的加法运算和 l 次 G_1 中的减法运算.因此,所需运算时间的 T 项复杂性为 $O(\log T)$.

(2) 基密钥演化算法最多需要 $3l-2$ 次 G_1 中的乘法运算和 $2(l-1)$次 G_1 中的加法运算.因此,所需运算时间的 T 项复杂性为 $O(\log T)$.

(3) 签名用户密钥演化算法最多需要 $3l-2$ 次 G_1 中的乘法运算和 $3l$ 次 G_1 中的加法运算.因此,所需运算时

间的 T 项复杂性为 $O(\log T)$.

(4) 基密钥更新算法最多需要 l 次 G_1 中的减法运算.因此,所需运算时间的 T 项复杂性为 $O(\log T)$.

(5) 签名用户密钥更新算法最多需要 l 次 G_1 中的加法运算.因此,所需运算时间的 T 项复杂性为 $O(\log T)$.

(6) 签名算法需要 1 次 G_1 中的加法运算和 1 次 G_1 中的乘法运算.因此,所需运算时间的 T 项复杂性为 $O(1)$.

(7) 验证算法需要 l 次 G_2 中的乘法运算和 l+2 次配对运算.因此,所需运算时间的 T 项复杂性为 $O(\log T)$.

所以,7 个算法运行时间的 T 项复杂性不超过 $O(\log T)$.

系统公钥 $PK=\{G_1,G_2,e,H,P,Q\}$,包含的元素个数完全独立于 T,其长度的 T 项复杂度为 $O(1)$.

系统私钥包括:基密钥 $SKB_{i,r}$、签名用户密钥 $SKS_{i,r}$、密钥演化消息 SKU_i 和密钥更新消息 $SKR_{i,r}$.基密钥 $SKB_{i,r}=Set'_{\zeta}$ 它最多由 l 个元素构成,长度的 T 项复杂度为 $O(\log T)$.签名用户密钥 $SKS_{i,r}=\{s_{\langle i\rangle},S_{\langle i\rangle},Set''_{\langle i\rangle},\mathcal{Q}_{\langle i\rangle}\}$,其中 $Set''_{\langle i\rangle}$ 和 $\mathcal{Q}_{\langle i\rangle}$均最多由 l 个元素构成,所以 $SKS_{i,r}$ 最多由 2l+2 个元素构成,其长度的 T 项复杂度为 $O(\log T)$.密钥演化消息 $SKU_i=(s'_{\langle i+1\rangle},S'_{\langle i+1\rangle},Q'_{\eta},Q'_{\eta 0},\ldots,Q'_{\eta 0^{l-j-1}},Q'_{\langle i+1\rangle})$ 最多由 l+2 个元素构成,其长度的 T 项复杂度为 $O(\log T)$.密钥更新消息 $SKR_{i,r}=\{R_{\zeta}\mid\zeta\in\psi(i)\}$ 最多由 l 个元素构成,其长度的 T 项复杂度为 $O(\log T)$.因此,所有私钥长度的 T 项复杂度为 $O(\log T)$.

签名 $\langle i,sign=(V,\mathcal{Q}_{\langle i\rangle})\rangle$ 由一个时间标志和 l+1 个元素构成,其长度的 T 项复杂度为 $O(\log T)$.

因此,提出的入侵容忍签名方案整个性能的复杂性为 $O(\log T)$. □

表 1~表 3 分别给出提出方案的 7 个算法运行时间的 T 项复杂性,提出方案的公私钥和签名长度的 T 项复杂性,以及文献[16,19]中的方案和本文提出的方案整个性能的 T 项复杂性.

Table 1 Complexities of seven algorithms in the presented scheme (in terms of T)

表 1 提出方案的 7 个算法复杂性(T 项)

	Algorithm Gen	Algorithm Sign	Algorithm Ver	Algorithm UB	Algorithm US	Algorithm RB	Algorithm RS
Algorithm complexity	$O(\log T)$	$O(1)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$	$O(\log T)$

Table 2 Complexities of the public key size, the secret key size and the signature size in the presented scheme (in terms of T)

表 2 提出方案的公钥、私钥和签名长度复杂性(T 项)

	Public key size	Secret key size	Signature size
Complexity	$O(1)$	$O(\log T)$	$O(\log T)$

Table 3 Full performance complexities of schemes in Refs.[16,19] and our scheme (in terms of T)

表 3 文献[16,19]中的方案和本文提出的方案整个性能的复杂性(T 项)

	Scheme in Ref.[16]	Scheme in Ref.[19]	Our scheme
Full performance complexities	$O(T)$	$O(\log^2 T)$	$O(\log T)$

4 安全性证明

定理 1. 假定 $\langle i,sign=(V,\mathcal{Q}_{\langle i\rangle})\rangle$ 是 Sign 算法第 i 时间段对消息 M 产生的签名,则 $Ver(M,PK,\langle i,sig=(V,\mathcal{Q}_{\langle i\rangle})\rangle)$=“有效”.

证明:

$$
\begin{aligned}
&e(Q,H(i_1))\cdot e(Q_{\langle i\rangle},H(i_1i_2...i_lM))\cdot\prod_{j=2}^{l}e(Q_{i_1i_2...i_{j-1}},H(i_1i_2...i_j))\\
&=e(s_\varepsilon P,H(i_1))\cdot e(Q'_{\langle i\rangle}+Q''_{\langle i\rangle},H(i_1i_2...i_lM))\cdot\prod_{j=2}^{l}e(Q'_{i_1i_2...i_{j-1}}+Q''_{i_1i_2...i_{j-1}},H(i_1i_2...i_j))\\
&=e(P,s_\varepsilon H(i_1))\cdot e(P,(s'_{\langle i\rangle}+s''_{\langle i\rangle})\cdot H(i_1i_2...i_lM))\cdot\prod_{j=2}^{l}e(P,(s'_{i_1i_2...i_{j-1}}+s''_{i_1i_2...i_{j-1}})H(i_1i_2...i_j))\\
&=e(P,s_{\langle i\rangle}H(i_1i_2...i_lM)+s_\varepsilon H(i_1)+\sum_{j=2}^{l}s_{i_1i_2...i_{j-1}}H(i_1i_2...i_j))\\
&=e(P,s_{\langle i\rangle}H(i_1i_2...i_lM)+S_{\langle i\rangle})\\
&=e(P,V).
\end{aligned}
$$

□

为了简化分析,认为群 G_1 中的运算最多用 $O(k^{n_1})$ 的比特操作时间, Z_q^* 中运算最多用 $O(k^{n_2})$ 的比特操作时间,这里 k 为参数产生器 IG 的输入参数.在时间分析方面,考虑的是关于 k 和 T 的复杂度.

定理 2. 如果存在一个伪造者 F 运行时间最多为 t,进行的签名预言查询最多为 q_{sig} 次,进行的随机 H-预言查询最多为 q_{hash} 次,并且满足 $Succ^{IR}(IRSIG[k,T,RN],F)>\varepsilon$,则存在一个敌手 A 能(t',ε')-攻击群 G_1 上的 CDH 问题,这里,

$$t'=t+O(T\cdot\log T\cdot(k^{n_1}+k^{n_2})),$$

$$\varepsilon'=\frac{\varepsilon}{T\cdot(q_{sig}+q_{hash}+1)}.$$

证明:假定如果敌手 F 输出了一个对消息 M 的伪造签名$\langle i,\text{sign}\rangle$,那么 hash 预言必定在 $(i_1i_2...i_lM)$ 处被查询过,这里 $\langle i\rangle=i_1i_2...i_l$.任何敌手都可以修改进行这种操作,这可以使得对 hash 预言的查询次数增加到 q_{hash}+1 次,这是因为这次的 hash 查询可能不包含在 F 以前的 q_{hash} 次 hash 查询中.我们也假定当 F 在某个时间段 i 对消息 M 进行签名查询时,hash 预言$(i_1i_2\ldots i_lM)$也需要同时查询.同前面假设一样,任何敌手都可以修改进行这种操作,这可以使得对 hash 预言的查询次数增加到 $q_{\text{hash}}+q_{\text{sig}}+1$ 次.假定 F 保留所有必要的记录,不会查询相同的 hash 预言两次.

首先,算法 A 输入 IG 生成的 (G_1,G_2,e) 和 $(P,Q=\alpha P,P'=\beta P)$,目标是输出$\alpha\beta P$,这里 $\alpha=s_\varepsilon,\beta\in_R Z_q$,A 和 F 都不知道 α,β 的值,A 将 F 作为一个子程序来运行.A 选取总共的阶段数 T,并随机猜测 F 产生伪造签名的时间段 i^*,其猜对概率是 $1/T$,令 $\langle i^*\rangle=i_1^*i_2^*...i_l^*$.

A 提供给 F: $PK=(G_1,G_2,e,P,Q)$ 和总共阶段数 T.

A 保存 3 张表:H 预言表、Osig 签名预言表和 Osec 密钥泄漏预言表,用来回答 F 对 H 预言(hash 预言)、Osig 签名预言表和 Osec 密钥泄漏预言的查询.H 预言表中的每个元素为四元组(I,x,y,Δ),其中 I 表示输入的节点下标,x 表示关于 P 的一个随机的幂,y 表示关于 P' 的一个随机的幂(如果存在),Δ表示输出的 hash 值.$\Delta=h(I)$表示输入和输出的关系.

对 *H* 预言查询的模拟

当 F 进行 hash 查询时,A 必须保证回答是随机的,设 k 为 F 查询数据的输入长度,A 执行以下操作:

(1) 若 $1\leqslant k\leqslant l$,令 $w_1w_2...w_k$ 为输入 w 的比特表示,A 首先查询 w 是否出现在 H 表中元素的 I 项,如果出现了,则直接回答相应的Δ值;否则选择 $x_{w_1w_2...w_k}\in_R Z_q$,H 表中增加 $(w_1w_2...w_k,x_{w_1w_2...w_k},\phi,x_{w_1w_2...w_k}P)$,回答为$\Delta=x_{w_1w_2...w_k}P$.

(2) 若 $k>l$,则将输入 w 表示为 $w_1w_2...w_lM_g$,这里 M_g 可能是 F 选择的消息,下标 g 表示第 g 次查询长度大于 l 比特的输入,A 要随机猜测一个序号 g',希望伪造来源于第 g'次查询,因为总共需要 $q_{hash}+q_{sig}+1$ 次 H 查询,所以这就使得 A 猜对 g'的概率至少为 $1/q_{hash}+q_{sig}+1$.当 $g=g'$时,选择 $x_{w_1w_2...w_lM_{g'}}\in_R Z_q$, $y_{i_1^*i_2^*...i_l^*}\in_R Z_q^*$,增加元素 $(w_1w_2...w_lM_{g'},x_{w_1w_2...w_lM_{g'}},y_{i_1^*i_2^*...i_l^*},x_{w_1w_2...w_lM_{g'}}P+y_{i_1^*i_2^*...i_l^*}^{-1}P')$ 到 H 表中,回答其Δ项,这时有 $w_1w_2...w_l=i_1^*i_2^*...i_l^*=\langle i^*\rangle$.当 $g=g'$时,A 首先查询 w 是否出现在 H 表中元素的 I 项,如果出现了,则直接回答相应的Δ值.如果没有出现,A 选择 $x_{w_1w_2...w_lM_g}\in_R Z_q$,增加元素 $(w_1w_2...w_lM_g,x_{w_1w_2...w_lM_g},\phi,x_{w_1w_2...w_lM_g}P)$ 到 H 表中,回答其Δ项.

为了回答 F 对 Osec 预言的查询,首先,需要对一些必要的基地密钥变化和签名用户密钥变化进行模拟.对基

地密钥变化的模拟.模拟是容易的,这是因为基地第 j 时间段持有的 $SKB_{j,r}$ 仅仅是演化信息 $Set_{\langle j\rangle}$的一部分,且完全是随机的,签名者持有的签名密钥不影响其变化.A 只需选择随机集合 $Set'_{0'}$,按照协议执行的方法依次随机生成 j=1,…,T−1 中每个 r=1,…,$RN(j)$的 $SKB_{j,r}$,其中,每次演化产生的 $SKU_j=(s'_{\langle j+1\rangle},S'_{\langle j+1\rangle},Q'_{\eta},Q'_{\eta 0},...,Q'_{\langle j+1\rangle})$,每次更新选择的 $SKR_{j,r}\in_R G_1$.

对第 $i(0\leqslant i<i^*,i^*<i\leqslant T-1)$ 时间段签名用户密钥的模拟.A 选择 $s_{i_1i_2...i_{j-1}}\in_R Z_q(2\leqslant j\leqslant l)$(如果没有选择过)和 $s_{\langle i\rangle}\in_R Z_q$,令 $\mathcal{Q}_{\langle i\rangle}=\{s_{i_1}P,s_{i_1i_2}P,...,s_{i_1i_2...i_{l-1}}P,s_{\langle i\rangle}P\}$,然后计算 $S_{\langle i\rangle}=x_{i_1}Q+\sum_{m=2}^{l}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P$,这是因为:

$$\begin{aligned}S_{\langle i\rangle}&=s_{\varepsilon}H(i_1)+\sum_{m=2}^{l}s_{i_1i_2...i_{m-1}}H(i_1i_2...i_m)\\&=s_{\varepsilon}x_{i_1}P+\sum_{m=2}^{l}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P\\&=x_{i_1}Q+\sum_{m=2}^{l}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P.\end{aligned}$$

对满足 i_j=0 且 2≤j≤l 所有的 j,计算 $S_{i_1i_2...i_{j-1}1}=x_{i_1}Q+\sum_{m=1}^{j-1}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P+s_{i_1i_2...i_{j-1}}x_{i_1i_2...i_j1}P$,这是因为:

$$\begin{aligned}S_{i_1i_2...i_{j-1}1}&=s_{\varepsilon}H(i_1)+\sum_{m=1}^{j-1}s_{i_1i_2...i_{m-1}}H(i_1i_2...i_m)+s_{i_1i_2...i_{j-1}}H(i_1i_2...i_{j-1}1)\\&=s_{\varepsilon}x_{i_1}P+\sum_{m=1}^{j-1}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P+s_{i_1i_2...i_{j-1}}x_{i_1i_2...i_{j-1}1}P\\&=x_{i_1}Q+\sum_{m=1}^{j-1}s_{i_1i_2...i_{m-1}}x_{i_1i_2...i_m}P+s_{i_1i_2...i_{j-1}}x_{i_1i_2...i_{j-1}1}P.\end{aligned}$$

通过对基地密钥变化的模拟得到 $SKB_{i,0}=Set'_{\langle i\rangle}=\{S'_{\zeta}\mid\zeta\in\psi(i)\}$,计算 $Set''_{\langle i\rangle}=\{S_{\zeta}-S'_{\zeta}\mid\zeta\in\psi(i)\}$.此时 A 模拟产生了阶段 i 开始时的签名用户密钥 $SKS_{i,0}=\{s_{\langle i\rangle},S_{\langle i\rangle},Set''_{\langle i\rangle},\mathcal{Q}_{\langle i\rangle}\}$,通过查询模拟的基地密钥的相应演化和更新信息,它就模拟所有 $i(i\neq i^*),r(r=1...RN(i))$ 的签名用户的所有密钥 $SKS_{i,r}$.

对 Osec 预言查询的模拟

当 F 进行 Osec 预言查询时,而查询不会使方案是(i^*,Q)-被入侵的,A 执行以下操作:

1. 当 F 的查询输入为 ("s",i,r) 时,必定有 $i\neq i^*$ (因为若 $i=i^*$,则方案是 (i^*,Q) -被入侵的),通过上述模拟的签名用户的密钥,回答相应的 $SKS_{i,r}$.

2. 当 F 的查询输入为 ("b",i,r) 时,通过上述模拟的基地密钥,回答相应的 $SKB_{i,r}$.

3. 当 F 的查询输入为 ("u",i) 时,通过模拟的基地密钥的变化,回答相应的 SKU_i 和 $SKR_{i+1,0}$.

4. 当 F 的查询输入为 ("r",i,r) 时,通过模拟的基地密钥的变化,回答相应的 $SKR_{i,r}$.

对 Osig 预言查询的模拟

当 F 对第 i 时间段消息 $M_g(g\neq g')$的 Osig 签名预言进行查询时,A 执行以下操作:

1. 当 i=i*时,选择结点秘密 $s_{i_1^*i_2^*...i_{j-1}^*}\in_R Z_q(2\leqslant j\leqslant l)$(若未选择过),使用定义 $H(i_1^*i_2^*...i_l^*M_{g'})$ 时产生的随机 $y_{\langle i^*\rangle}$,令验证点为 $Q_{\langle i^*\rangle}=y_{\langle i^*\rangle}Q$(暗示 $s_{\langle i^*\rangle}=y_{\langle i^*\rangle}s_{\varepsilon}$,但 A 并不知道 s_{ε}).A 令 $\mathcal{Q}_{\langle i^*\rangle}$ 为由 $s_{i_1^*i_2^*...i_{j-1}^*}P(2\leqslant j\leqslant l)$ 和 $Q_{\langle i^*\rangle}$ 组成的集合,令 $V=x_{i_1}^*Q+\sum_{m=2}^{l}s_{i_1^*i_2^*...i_{m-1}^*}x_{i_1^*i_2^*...i_{m-1}^*}P+y_{\langle i^*\rangle}x_{i_1^*i_2^*...i_l^*M_g}Q$,返回签名 $sign=\langle V,\mathcal{Q}_{\langle i^*\rangle}\rangle$,这是因为:

$$\begin{aligned}V&=s_{\varepsilon}H(i_1^*)+\sum_{m=2}^{l}s_{i_1^*i_2^*...i_{m-1}^*}H(i_1^*i_2^*...i_m^*)+s_{\langle i^*\rangle}H(i_1^*i_2^*...i_l^*M_g)\\&=s_{\varepsilon}x_{i^*_1}P+\sum_{m=2}^{l}s_{i_1^*i_2^*...i_{m-1}^*}x_{i_1^*i_2^*...i_{m-1}^*}P+y_{<i^*>}s_{\varepsilon}x_{i_1^*i_2^*...i_l^*M_g}P\\&=x_{i^*_1}Q+\sum_{m=2}^{l}s_{i_1^*i_2^*...i_{m-1}^*}x_{i_1^*i_2^*...i_{m-1}^*}P+y_{\langle i^*\rangle}x_{i_1^*i_2^*...i_l^*M_g}Q.\end{aligned}$$

2. 当 $i\neq i$*时,因为 A 已经通过模拟知道了 $s\langle i\rangle$,$S\langle i\rangle$和 $\mathcal{Q}_{\langle i\rangle}$,它只需计算 $V=S_{\langle i\rangle}+s_{\langle i\rangle}H(i_1i_2...i_lM_g)=S_{\langle i\rangle}+s_{\langle i\rangle}x_{i_1i_2...i_lM_g}P$,并回答($V$,$\mathcal{Q}_{\langle i\rangle}$).

当 A 猜对了 F 进行伪造签名的 hash 预言查询序号 g'和伪造的阶段 i*时,若 F 可以伪造 $\langle i^*,M_{g'}\rangle$ 的签名 $sign=\langle V,\mathcal{Q}_{\langle i^*\rangle}\rangle$,则有:

$$e(P,V)=e(Q,H(i_1^*))\cdot e(Q_{\langle i^*\rangle},H(i_1^*i_1^*...i_l^*M_{g'}))\cdot\prod_{j=2}^{l}e(Q_{i_1^*i_1^*...i_{j-1}^*},H(i_1^*i_2^*...i_j^*))\Rightarrow$$

$$e(P,V)=e(P,s_\varepsilon H(i_1^*)+s_{\langle i^*\rangle}H(i_1^*i_1^*...i_l^*M_{g'})+\sum_{j=2}^{l}s_{i_1^*i_1^*...i_{j-1}^*}H(i_1^*i_2^*...i_j^*))\Rightarrow$$

$$V=s_\varepsilon H(i_1^*)+s_{\langle i^*\rangle}H(i_1^*i_1^*...i_l^*M_{g'})+\sum_{j=2}^{l}s_{i_1^*i_1^*...i_{j-1}^*}H(i_1^*i_2^*...i_j^*)\Rightarrow$$

$$V=s_\varepsilon x_{i_1^*}P+y_{\langle i^*\rangle}s_\varepsilon(x_{i_1^*i_1^*...i_l^*M_{g'}}P+y_{\langle i^*\rangle}^{-1}P')+\sum_{j=2}^{l}s_{i_1^*i_1^*...i_{j-1}^*}x_{i_1^*i_2^*...i_j^*}P\Rightarrow$$

$$V=x_{i_1^*}Q+y_{\langle i^*\rangle}x_{i_1^*i_1^*...i_l^*M_{g'}}Q+s_\varepsilon P'+\sum_{j=2}^{l}s_{i_1^*i_1^*...i_{j-1}^*}x_{i_1^*i_2^*...i_j^*}P\Rightarrow$$

$$\alpha\beta P=V-x_{i_1^*}Q-y_{\langle i^*\rangle}x_{i_1^*i_1^*...i_l^*M_{g'}}Q-\sum_{j=2}^{l}s_{i_1^*i_1^*...i_{j-1}^*}x_{i_1^*i_2^*...i_j^*}P.$$

所以 A 可以成功计算出$\alpha\beta P$,构造结束.

运行时间分析

A 总共运行的时间包括 F 运行的时间 t 加上以下时间.

(1) 直接回答 H 预言的时间:当 $g=g'$且 $k>l$ 时,主要包括一些 G_1 中的乘、加运算和 Z_q^* 中的逆运算,总共需要的时间不超过 $O(k^{n_1}+k^{n_2})$;当 $k\leqslant l$,或 $k>l$ 且 $g\neq g'$时,仅需要一些 G_1 中的乘运算,总共需要的时间不超过 $O(k^{n_1})$.

(2) 回答 Osec 预言的时间:回答此预言的时间由下面两部分组成:

① 模拟基地密钥变化的时间:生成每个时间段密钥时,最多需要 $O(\log T)$个 G_1 中的乘、加和一些减运算,另外最多间接产生 $O(\log T)$个 hash 计算.所有时间段总共需要的时间不超过 $O(T\cdot\log T\cdot k^{n_1})$(更为精确的分析是不超过 $O(T\cdot k^{n_1})$的时间).

② 模拟所有 $i(0\leqslant i<i^*,i^*<i\leqslant T-1)$ 时间段签名用户密钥的时间:对于每个 i 需要 $O(\log T)$个 G_1 中的乘运算和 $O(\log T)$个 Z_q^* 中的乘运算,以及一些 G_1 中的加、减运算,因此对应所有时间段 i 的总共时间不超过 $O(T\cdot\log T\cdot(k^{n_1}+k^{n_2}))$.

(3) 回答 Osig 预言的时间:最多包括 $O(\log T)$个 G_1 中的乘,$O(\log T)$个 Z_q^* 中的乘和一些 G_1 中的加运算,总共的时间不超过 $O(\log T\cdot(k^{n_1}+k^{n_2}))$.

(4) 解决 CDH 问题的时间:包括 $O(\log T)$个 G_1 中的乘,$O(\log T)$个 Z_q^* 中的乘和一些 G_1 中的减运算,总共的时间不超过 $O(\log T\cdot(k^{n_1}+k^{n_2}))$.

综上所述,A 总共所需要的运行时间为 $t'=t+O(T\cdot\log T\cdot(k^{n_1}+k^{n_2}))$.

成功概率分析

F 无法区分 A 给出的模拟和实际运行中的观察有什么不同,A 猜对伪造阶段的概率为 $1/T$,猜对伪造签名的 H 预言查询序号的概率至少为 $1/(q_{hash}+q_{sig}+1)$,所以如果 F 能以不可忽略的概率ε伪造有效签名,A 就能以不小于 $\dfrac{\varepsilon}{T\cdot(q_{hash}+q_{sig}+1)}$ 的概率计算 G_1 上的 CDH 问题.

因此,定理成立. □

定理 3. 令 IRSIG[k,T,RN]表示上述提出的入侵容忍签名方案,k 为安全参数,T 为总共的时间段数,对于任意 t,q_{sig} 和 q_{hash},下列关系成立:

$$In\sec^{IR}(IRSIG[k,T,RN],t,q_{sig},q_{hash})\leqslant T\cdot(q_{hash}+q_{sig}+1)\cdot In\sec^{CDH}(k,t').$$

这里,$t'=t+O(T\cdot\log T\cdot(k^{n_1}+k^{n_2}))$.

证明:

设 $Insec^{CDH}(k,t')=\varepsilon'$,即不存在敌手 $A(t',\varepsilon')$攻击群 G_1 上的 CDH 问题,由定理 2 得,对于任何敌手 F 都满足下列关系:

$$Succ^{IR}(IRSIG[k,T,RN],F)\leqslant\varepsilon=T\cdot(q_{hash}+q_{sig}+1)\cdot\varepsilon'=T\cdot(q_{hash}+q_{sig}+1)\cdot Insec^{CDH}(k,t').$$

又由定义 2 得

$$Insec^{IR}(IRSIG[k,T,RN],t,q_{sig},q_{hash}) \leqslant T\cdot(q_{hash}+q_{sig}+1)\cdot Insec^{CDH}(k,t').$$

这里 $t'=t+O(T\cdot\log T\cdot(k^{n_1}+k^{n_2}))$. □

5 结 论

入侵容忍签名具有很高的安全性,但目前存在的具体方案较少.本文提出了一个可证安全的入侵容忍签名方案,方案不但满足入侵容忍签名的所有属性,而且具有良好的平均性能,所有费用参数的复杂性均不超过 $O(\log T)$.方案的入侵容忍安全性依赖于 CDH 假设,证明了假定 CDH 问题是难解的,则提出的签名方案在随机预言模型中是入侵容忍安全的.

致谢 在此感谢匿名评审专家对本文工作给予的支持并提出了建设性意见,也感谢家人的细心照顾,使我能够在住院期间完成本文的修改工作.

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, S. Miner. A forward-secure digital signature scheme. In: Wiener M, ed. Proc. of the CRYPTO'99. LNCS 1666, Berlin: Springer-Verlag, 1999. 431–448.

[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.

[4] Kozlov A, Reyzin L. Forward-secure signatures with fast key update. In: Cimato S, Galdi C, Persiano G, eds. Proc. of the Security in Communication Networks 2002. LNCS 2576, Berlin: Springer-Verlag, 2002. 247–262.

[5] Itkis G, Reyzin L. Forward-secure signatures with optimal signing and verifying. Kilian J, ed. Proc. of the Crypto 2001. LNCS 2139, Berlin: Springer-Verlag, 2001. 499–514.

[6] Li RP, Yu J, Li GW, Li DX. Forward secure group signature schemes with efficient revocation. Journal of Computer Research and Development, 2007,44(7):1219–1226 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/19/9.htm

[7] 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.

[8] 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.

[9] Hu F, Wu C, Irwin J. A new forward secure signature scheme using bilinear maps. Cryptology ePrint Archive, 2003. http://eprint.iacr.org/2003/188.pdf

[10] Kang B, Park J, Halm S. A new forward secure signature scheme. Cryptology ePrint Archive, 2004. http://eprint.iacr.org/2004/183

[11] Yu J, Kong F, Cheng X, Hao R, Li G. Construction of yet another forward secure signature scheme using bilinear maps. In: Baek J, Bao F, Chen K, Lai X, eds. Proc. of the ProvSec 2008. LNCS 5324, Berlin: Springer-Verlag, 2008. 83–97.

[12] Dodis Y, Katz J, Xu S, Yung M. Strong key-insulated signature scheme. In: Desmedt Y, ed. Proc. of the PKC 2003. LNCS 2567, Berlin: Springer-Verlag, 2003. 130–144.

[13] Weng J, Li X, Chen K, Liu S. Identity-based parallel key-insulated signature without random oracles: Security notations and construction. In: Barua R, Lange T, eds. Proc. of the INDOCRYPT 2006. LNCS 4329, Berlin: Springer-Verlag, 2008. 1143–1157.

[14] Weng J, Chen KF, Liu SL, Li XX. Identity-Based strong key-insulated signature without random oracles. 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]

[15] Itkis G. Intrusion-resilient signature: Generic constructions, or defeating a strong adversary with minimal assumption. In: Cimato S, Galdi C, Persiano G, eds. Proc. of the Security in Communication Networks 2002. LNCS 2576, Berlin: Springer-Verlag, 2002. 102–118.

[16] 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.

[17] 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.

[18] Dodis Y, Franklin M, Katz J, Miyaji A, Yung M. A generic construction for intrusion-resilient public-key encryption. In: Okamoto T, ed. In: Proc. of the CT-RSA 2004. LNCS 2964, Berlin: Springer-Verlag, 2004. 81−98.

[19] Libert B, Quisquater J, Yung M. Efficient intrusion-resilient signatures without random oracles. In: Lipmaa H, Yung M, Lin D, eds. Proc. of the 2nd SKLOIS Conf. on Information Security and Cryptology. LNCS 4318, Berlin: Springer-Verlag, 2006, 27−41.

[20] Yu J. Research on the cryptosystem related to secret key security [Ph.D. Thesis]. Ji'nan: Shandong University. 2006 (in Chinese with English abstract).

附中文参考文献:

[6] 李如鹏,于佳,李国文,李大兴.高效撤消成员的前向安全群签名方案.计算机研究与发展,2007,44(7):1219−1226.

[14] 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]

[20] 于佳.密钥安全相关密码体系的研究[博士学位论文].济南:山东大学,2006.

于佳(1976－),男,山东青岛人,博士,副教授,CCF 会员,主要研究领域为密码学,网络信息安全.

郝蓉(1976－),女,讲师,主要研究领域为密码学,网络信息安全.

孔凡玉(1978－),男,博士,副教授,主要研究领域为密码学.

GUO Xiangfa(1979－),男,博士生,主要研究领域为网络信息安全.

程相国(1969－),男,博士,副教授,主要研究领域为密码学.