文章编号:1673-0291(2016)02-0058-005 DOI:10.11860/j.issn.1673-0291.2016.02.010

跳表和认证跳表的性能分析及对比

彭双和,陈志阁,陈得恩
(北京交通大学 计算机与信息技术学院,北京 100044)

摘 要:为了解决现有研究对跳表性能分析的不清晰,对认证跳表认证性能分析存在偏差的问题,深入研究了跳表的性能,认证跳表数据完整性的认证性能.本文采用公式证明的方式对两种数据结构就性能分析上存在的问题和两种结构之间的性能差异进行了分析.分析结果表明:本文对跳表遍历性能和认证跳表认证性能的分析比以往的性能分析更加清晰准确,同时对两个结构的性能差异也给出了正确的分析结果,以便为相关研究提供重要的参考.

关键词:跳表;认证跳表;性能分析
中图分类号:TP302.7 **文献标志码**:A

Performance analysis and comparison of skip lists and authenticated skip lists

PENG Shuanghe, CHEN Zhige, CHEN De'en
(School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China)

Abstract: In order to solve the existing problems that the performance analysis of skip lists is not clear and the performance analysis of authenticated skip lists is not rigorous in current studies, this paper further studies the performance of the skips lists and data integrity certification performance of authenticated skip lists. The problems on performance analysis of the two data structures and the performance differences between skip lists and authenticated skip lists are analyzed based on formula method. The results show that: this paper gives a more clear and accurate analysis of skip lists' traversal performance and authenticated skip lists' certification performance than previous researches, and it also gives a accurate analysis result about the differences of performance between skip lists and authenticated skip lists, which will provide references for subsequent studies.

Key words: skip lists; authenticated skip lists; performance analysis

云计算是一个多租户环境,云上资源均需共享.用户将数据存储到云服务器上具有一定风险,无论在传输过程或者服务器上都有可能遭到未授权第三方篡改.怎样让用户确信他们的数据在云端被正确存储和处理,即数据的完整性验证,是云存储发展必须要解决的一个问题.可证明数据持有(Provable Data Possession,PDP)[1]提供了一个解决该问题的模型.在该模型下,客户端和服务方之间通过一种挑战应答协议来验证数据的完整性.但验证过程中最为重要的是完整性证明信息的生成,即利用数据结构和算法实现完整性证明信息的生成.2001年Goodrich提出的认证跳表数据结构[2]是基于1989年William Pugh提出的跳表[3]数据结构建立的,利用跳表节点存储数据元素的哈希值,并通过相应的哈希

收稿日期:2015-06-05
基金项目:中央高校基本科研业务费专项资金资助(2015JBM034);国家留学基金委提供部分基金支持(201407095023)
作者简介:彭双和(1974—),女,湖南衡阳人,讲师,博士.研究方向为信息安全.**email**: shhpeng@bjtu.edu.cn.

计算来实现对数据可信认证.由于认证跳表具有动态操作优点,它在数字时间戳[4-5]、无线射频系统[6]、数据库外包[7-8]及云存储数据完整性检测中都得到广泛的应用.

证明信息生成的效率直接影响用户数据完整性,认证即数据的认证代价.为取得高效的认证,需要一个性能分析方法来分析认证跳表的认证算法.已有的研究曾对跳表及认证跳表这两种数据结构相应性能进行过分析.跳表[9-10]是一种能有效实现对有序序列进行查找的数据结构,其平均查找和更新时间是 $O(\log n)$. 文献[3]对跳表的节点查找进行了性能分析,但部分分析过程简略,不利于后续学者进一步分析跳表.文献[11]在文献[3]的基础上对认证跳表的认证性能进行了分析,其中部分性能分析存在偏差和不清晰的问题,这都为后续认证跳表相关研究带来困难.为了能更准确说明这两种数据结构的性能,本文作者采用公式证明方式对两种数据结构在性能分析上存在的问题进行详细证明,将跳表遍历性能和认证跳表认证性能进行对比,证明其差异性.可以最大程度减少差异带来的额外代价,提高整个系统的认证效率.

1　相关符号定义

定义符号标记,便于以下性能分析及证明.

$L(n)$表示层数的期望 $L(n)=\log_{1/p}n$.其中 n 为跳表层数,p 为跳表第 i 层元素出现在第 $i+1$ 层的概率.二项分布 $B(n,p)$表示在 n 次独立重复试验中成功发生的次数,期望为 np.负二项分布 $NB(n,p)$表示在进行独立重复试验中发生第 n 次成功之前失败的次数,期望为 $n(1-p)/p$.其中 n 是一个非负数,p 表示一次试验成功的概率.

跳表中由存储相同元素(如元素 e)副本的节点组成 1 个塔.塔中元素有不同的层,把只出现在 e_1,e_2,…,e_i 而不出现在 e_{i+1}(i 表示元素的层数)及之上的元素叫作层 i 元素.一个元素为层 i 元素的概率是 $(1-p)/p^{i+1}$.如果一个跳表节点是它所在塔的最顶层节点,则称其是平节点.

2　跳表及认证跳表的性能分析

2.1　基于跳表的遍历性能分析

跳表由 5 个链表$\{S_1,S_2,S_3,S_4,S_5\}$构成,如图 1 所示,并且每个链表 $S_i(1\leqslant i\leqslant 5)$存储两个特殊元素 $-\infty$和$+\infty$,即每个链表中最小元素和最大元素.

在跳表中查找节点如查找图 1 的 h_{15}节点,从顶层节点(图 1 中 S_5 的 S_t 节点)开始,对跳表由上至下逐层查找.查找节点索引值小于当前节点 S_t 右指针指向的节点 E_t 的索引值,则跳转到当前节点 S_t 向下指针指向的节点 F 上,链表 S_5 后续节点不再遍历.从 F 节点开始对链表 S_4 后续节点遍历,查找节点索引值大于当前节点 F 右指针指向的节点 K 的索引值,则跳转到当前节点 F 右指针指向的节点 K,从 K 开始继续对链表 S_4 后续节点遍历.重复上述两种情况直至找到节点 h_{15}.其中图 1 中节点集$\{S_t,F,K,J,H,M,L,N\}$形成节点 h_{15}遍历路径.

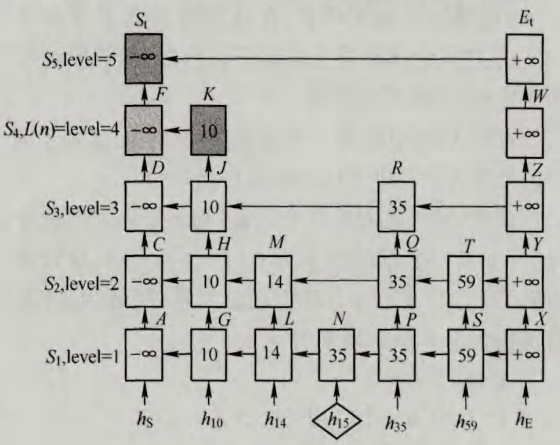

图 1　跳表及遍历路径

Fig.1　Skip lists and traversal path

文献[3]中采用查找跳表节点的逆向方向,以跳表期望层数 $L(n)$为界(图 1 中链表 S_4 所在层)将节点遍历路径分为低于 $L(n)$层路径上遍历访问的节点和高于或等于 $L(n)$层路径上遍历访问的节点,以此分析跳表的遍历代价.文献[3]对跳表节点查找性能进行了分析,低于 $L(n)$层路径上的遍历代价分析的很详细,具体可参见文献[3].高于或等于 $L(n)$层这段路径上遍历代价,文献[3]分成两部分推导分析:1)这段路径向左遍历访问的节点,遍历代价用 C_1 表示即如图 1 浅灰斜线填充的节点 F.2)这段路径上向上遍历访问的节点,遍历代价用 C_2 表示,即如图 1 深灰色斜线填充的节点集$\{K,S_t\}$.对于 C_1 遍历代价,文献[3]没有通过具体公式来推导出现在 $L(n)$层的节点数为何服从二项分布 $B(n,1/np)$.而在分析 C_2 遍历代价中出现的一系列公式,文献[3]没有给出具体的推导和说明.文献[3]对于高于或等于 $L(n)$层这段路径上的遍历代价分析不够清晰,立据不足,不利于后续学者对该部分性能分析的理解.

本文作者通过具体的公式推导来分析跳表在高于或等于 $L(n)$层这段遍历路径上的遍历代价.

1)分析 C_1 遍历代价.对于跳表的结构,下层节点以一定概率出现在上层,因此从 $L(n)$层到跳表

根节点(S_t元素)向左遍历访问的节点个数小于出现在$L(n)$层节点的个数.下层的节点出现到上层发生的概率为p,可推导在$L(n)$层的节点个数为$1/p$.假设level表示跳表层数且最底层的level为1.最底层level=1有n个节点;level=2有np个节点;level=$L(n)$有$np^{[L(n)-1]}$个节点.将$L(n)=\log_{1/p}n$代入$np^{[L(n)-1]}$,得第$L(n)$层节点的个数为$1/p$.因此跳表在高于或等于$L(n)$层这段遍历路径上向左遍历访问的节点数即遍历代价$C_1\leqslant 1/p$.

2)分析C_2遍历代价.在计算跳表高于或等于$L(n)$层这段遍历路径上的遍历代价之前,需要证明两个定理,为证明做准备.

定理1:对于任意一个元素a,其出现在大于k层的概率为p^k,即$P(\text{Level}(a)>k)=p^k$.

证明:对于跳表中某个元素,则1层元素的概率为$1-p$,2层元素的概率为$p(1-p)$,…,k层元素的概率为$p^{k-1}(1-p)$.对于任意元素,层数为L,其出现在大于k层的概率为

$$\begin{aligned}&P(L>k)=\\&P(L>k+1)+P(L>k+2)+\cdots=\\&p^k(1-p)+p^{k+1}(1-p)+\cdots=\\&p^k(1-p)(1+p+p^2+\cdots)=\\&p^k(1-p)\frac{1}{1-p}=p^k\end{aligned}\tag{1}$$

定理2:若M代表跳表层数,则有

$$P(M>k)=1-(1-p^k)^n\leqslant np^k.$$

证明:由定理1可知,不能出现在大于k层事件的概率为$1-p^k$.对于底层有n个节点的跳表,均不能出现在大于k层的概率为$(1-p^k)^n$.则存在1个元素,出现在大于k层的概率为$1-(1-p^k)^n$.依据伯努利不等式,任意整数$n\geqslant 0$及任意实数$x\geqslant -1$,$(1+x)^n\geqslant 1+nx$.由于$1-\leqslant -p^k\leqslant 0$,因此有

$$P(M>k)=1-(1-p^k)^n\leqslant np^k\tag{2}$$

$NB(1,1-p)$是1个负二项分布,表示在进行独立重复试验中,在发生1次成功(以概率$1-p$向左走)之前失败(以概率p向上走)的次数.如果在第1层就成功,$NB(1,1-p)=0$,该元素层数是$NB(1,1-p)+1=1$;如果在第1层失败,第2层成功,则$NB(1,1-p)=1$,$NB(1,1-p)+1=2$,可知$NB(1,1-p)+1$即失败次数加1表示该元素的层数.由定理1知$P(\text{Level}(a)>k)=p^k$则$NB(1,1-p)+1+L(n)>k$

$$\begin{aligned}&P(NB(1,1-p)+1+L(n)>k)=\\&P(NB(1,1-p)+1>k-L(n))=\\&p^{k-L(n)}\end{aligned}\tag{3}$$

将$L(n)=\log_{1/p}n$代入得

$$np^k=\frac{p^k}{1/n}=\frac{p^k}{p^{\log_{1/p}n}}=\frac{p^k}{p^{L(n)}}=p^{k-L(n)}\tag{4}$$

由定理2及式(4)可知

$$\begin{aligned}P(M>k)\leqslant P(&NB(1,1-p)+\\&1+L(n)>k)\end{aligned}\tag{5}$$

由此可得M的上限值即

$$M\leqslant_{\text{prob}}L(n)+NB(1,1-p)+1\tag{6}$$

M的期望值为

$$E[M]\leqslant L(n)+1/(1-p)\tag{7}$$

C_2为高于或等于$L(n)$层路径向上遍历访问节点个数的期望值,所以$E[C_2]\leqslant E[M]-L(n)$.将式(7)代入得

$$E[C_2]\leqslant L(n)+\frac{1}{1-p}-L(n)=\frac{1}{1-p}\tag{8}$$

跳表在高于或等于$L(n)$层这段遍历路径向上遍历访问的节点数即遍历代价$C_2\leqslant 1/(1-p)$.

2.2 基于认证跳表的数据完整性认证性能分析

认证跳表是在跳表基础上,利用跳表节点存储数据元素哈希值即标签值,通过相应的哈希计算实现数据认证.具体节点标签值计算见文献[4].

完整性证明信息的获取是在认证跳表节点遍历过程中,根据一定的算法获取的.根据完整性证明信息对节点进行完整性认证.对图2节点h_{15}的遍历过程中,根据一定算法获取节点h_{15}的完整性证明信息集$\{D,G,R,W,h_{14}\}$,如图浅灰色填充节点.这些节点的标签值是需要根据一定规则重新计算的.因此认证跳表的认证代价实际是节点遍历过程中有多少个跳表节点需要重新计算标签值.

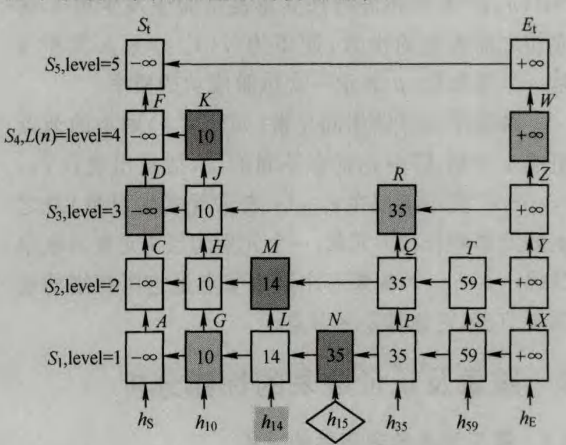

图2 节点h_{15}的完整性证明信息和两种路径差异

Fig.2 Integrity information of node h_{15} and differences between the two paths

文献[11]对认证跳表的认证代价进行了分析,它是以节点遍历的逆向方向将节点的认证代价分3

部分进行分析:1)计算低于 $L(n)$层;2)高于或等于 $L(n)$层向左走;3)高于或等于 $L(n)$层向上走.

相应路径上与节点认证相关的节点数,需重新计算哈希的节点数,以此衡量彼此路径上的认证代价.

文献[11]对低于 $L(n)$层和高于或等于 $L(n)$层向左走这两段路径上的认证代价分析的很详细,也有具体的推导,这部分的性能具体可参见文献[11].但是对于高于或等于 $L(n)$层向上走的认证代价,文献[11]没有给出相关证明,分析存在偏差.下面给出具体推导来证明这段路径的认证代价.

在 2.1 节求解 C_2 中,推导出 $L(n)$到最高层的层数期望是 $1/(1-p)$,定义 $N=1/(1-p)$.当前节点的右节点是平节点,这个事件发生的概率$(1-p)$.$NB(1,1-p)$是一个二项分布,表示在进行 N 次独立重复试验中(在这 N 层中),成功发生概率(根据文献[4]计算标签值的规则,当前节点的右节点是平节点,需要哈希计算一次)的次数,也就是从 $L(n)$层到根节点向上走这 N 层中进行哈希计算的次数,用变量 Num 表示.由于从 $L(n)$层到最高层向上走哈希计算的次数一定小于或等于 $L(n)$层到最高层的层数即 $\mathrm{Num}\leqslant_{\mathrm{prob}} B(N,1-p)$.将 $N=1/(1-p)$和 $E[B(n,p)]=np$,代入 $\mathrm{Num}\leqslant_{\mathrm{prob}} B(N,1-p)$得 $\mathrm{Num}\leqslant 1$.

认证跳表在高于或等于 $L(n)$层这段遍历路径向上走的认证代价小于等于 1.

3　跳表与认证跳表的性能对比

3.1　性能对比

本文根据文献[3]和文献[11]将跳表遍历性能和认证跳表认证性能进行性能对比.分成 3 部分进行比较,分别用 N_1、N_2 和 N_3 表示低于 $L(n)$层、高于 $L(n)$向左遍历和高于 $L(n)$向上遍历的性能代价.具体对比结果如表 1 所示,对数的底数可取任意值,为简化操作取 2.

表 1　跳表与认证跳表性能差异

Tab.1　Performance differences between skip lists and authentication skip lists

性能	跳表	认证跳表	结构差异
N_1	$2\mathrm{lb}n-2$	$3/2\ \mathrm{lb}n-3/2$	$1/2\mathrm{lb}n-1/2$
N_2	2	2	0
N_3	2	1	1
N	$2\mathrm{lb}n+2$	$3/2\mathrm{lb}n+3/2$	$1/2\mathrm{lb}n+1/2$

从表 1 中可以看出跳表和认证跳表的遍历代价差异在 N_1 和 N_3.跳表在 N_1 这段路径上访问的是所有节点,而认证跳表访问的是需要重新计算哈希的节点,因此它们之间的差异在于这段路径上不需要计算哈希的节点.N_3 同理.

3.2　跳表与认证跳表性能差异证明

跳表遍历性能和认证跳表认证性能的差异代价是 $1/2(\mathrm{lb}n)$.如图 2 所示,节点集$\{S_t,F,K,J,H,M,L,N\}$形成偏压节点 h_{15}需要访问的节点,节点集$\{D,G,R,W,h_{14}\}$形成节点 h_{15}认证路径上需重新计算哈希的节点.从图 2 上可以看出,两者的差异之处是不需要哈希计算的节点,如图 2 节点集$\{K,M,N\}$.

本文作者通过推导证明它们两者之间的差异.采用分析认证跳表遍历路径法,分成 3 部分进行分析:1)从底层到 $L(n)$层中不需要计算哈希的节点代价,用 D_1 表示;2)从 $L(n)$到最高层这段路径中向左方向不需要计算哈希的节点代价,用 D_2 表示;3)从 $L(n)$到最高层向上走不需要计算哈希的节点代价,用 D_3 表示.下面开始逐个进行分析证明.

1)计算 D_1 的代价.定义 C_k 表示从 level $i\rightarrow$ level $(i+k)$ 这 k 层不需计算哈希的节点数,其中 k 为向上走的层数,初始 $C_0=0$.处于某层的节点有两种情况,可能以概率 p 向上走,也可能以概率 $1-p$ 向左走.①向上走,已知当前节点(向上走到的节点).如果当前节点没有右节点或者是它的右节点不是平节点,它的哈希值直接继承它的下层节点哈希值,不需计算哈希值,用变量 X_i 表示,即 $X_i=1$,如图 2 中节点 N.如果当前节点有右节点且是平节点,需哈希计算 1 次,$X_i=0$,如图 2 中节点 J.这种情况减少了需要向上走的层数,因此 $C_k=p(X_i+C_{k-1})$,其中 $X_i\sim\mathrm{Bern}(p)$是一个 0−1 伯努利分布.②当向左走,当前节点如图节点 H(从节点 M 向左走到节点 H)的右节点一定存在且是平节点,根据哈希机制,需要哈希计算一次,那么不需要计算哈希的节点个数为 0.向左走并没有减少需要向上走层数,因此 $C_k=(1-p)\ C_k$.

由此从 level $i\rightarrow$ level $(i+k)$ 这 k 层不需要哈希计算的节点数,即 C_k 的期望值是

$$\begin{cases}E[C_k]=E[p([E(X_i]+C_{k-1})+(1-p)C_k]\\ E[C_k]=E[p(p+C_{k-1})+(1-p)C_k]\\ E[C_k]=p^2+pE[C_{k-1}]+E[C_k]-pE[C_k]\\ E[C_k]=kp\end{cases}\tag{9}$$

D_1 为从 level 1→ level $(1+L(n))$这 $L(n)-1$ 层不需要重新计算哈希的节点数,将 $L(n)-1$ 代入式(9),得 $E[D_1]\leqslant(L(n)-1)p$.又因为 $L(n)=\log_{1/p}n$,由此可得

$$E[D_1] \leqslant p\log_{1/p} n - p \tag{10}$$

2)计算 D_2 的代价.当前节点(如图2节点 F)的右节点是(节点 K)平节点,才可以从 K 向左走到 F.根据哈希机制,需要重新计算节点 F 哈希值.因此在遍历路径上向左走到的节点都需要重新计算哈希值,而 D_2 为在 $L(n)$ 或者高于 $L(n)$ 向左走不需要重新计算哈希的节点数.因此

$$E[D_2]=0 \tag{11}$$

3)计算 D_3 的代价.在2.1节求解 C_2 中,推导出 $L(n)$ 到最高层的层数期望是 $1/(1-p)$.定义 $N=1/(1-p)$.向上走到的节点(图2节点 M)的右节点(图2节点 Q)不是平节点,这个事件发生的概率是 p.根据文献[4]计算标签值的规则,当前节点的右节点不是平节点,不需要重新计算该节点的哈希值.$B(N,p)$ 是一个二项分布,表示在进行 N 次独立重复试验中(在这 N 层中),成功发生概率 p 的次数,也就是从 $L(n)$ 层到根节点向上走,这 N 层中不需要哈希计算的次数.从 $L(n)$ 层到最高层向上走不需计算哈希的节点一定小于或等于 N,因此有 $D_3 \leqslant B(N,p)$.将 $N=1/(1-p)$ 及 $E[B(n,p)]=np$,代入 $E[D_3] \leqslant E[B(N,p]$ 得

$$E[D_3] \leqslant p/(1-p) \tag{12}$$

通过上述跳表与认证跳表对同一节点的遍历存在性能差异的分析,可以得出对应节点消耗代价.取 $p=1/2$,对于有 n 个节点的有限跳表,其对应节点的差异代价为 $E[D_H] \leqslant E[D_1] + E[D_2] + E[D_3]$,将式(9~12)代入 $E[D_H] \leqslant E[D_1] + E[D_2] + E[D_3]$,可得

$$E[D_H] \leqslant p\log_{1/p} n - p + 0 + \frac{p}{1-p},$$

$$E[D_H] \leqslant \frac{1}{2}\mathrm{lb}n + \frac{1}{2} \tag{13}$$

4 结语

1)针对现有文献对跳表在高于或等于 $L(n)$ 层这段路径上的性能分析不够清晰,对认证跳表在高于或等于 $L(n)$ 层这段遍历路径向上走的性能分析有偏差的情况,本文作者通过具体公式证明了跳表在这段路径上遍历代价是 $1/p+1/(1-p)$,认证跳表在这段遍历路径上向上走的认证代价小于等于1.

2)本文对跳表的遍历性能和认证跳表的认证性能两者之间存在的性能差异进行了分析,证明其性能差异代价是 $\frac{1}{2}\mathrm{lb}n+\frac{1}{2}$.

本文的研究结论可为后续优化认证跳表认证性能提供理论依据.

参考文献(References):

[1] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//ACM Conference on Computer and Communications Security, 2007: 598－609.

[2] GOODRICH M T, TAMASSIA R, SCHWERIN A. Implementation of an authentication dictionary with skip lists and commutative hashing[C]//DARPA Information Survivability Conference and amp, 2001, 2: 68－82.

[3] WILLIAM Pugh. Skip lists: A probabilistic alternation to balance trees[J]. Lecture Notes in Computer Science, 1990, 33(6): 668－676.

[4] BLIBECH K, GABILLON A. A new timestamping scheme based on skip lists[C]// Computational Science and Its Applications ICCSA, 2006: 395－405.

[5] BLIBECH K, GABILLON A. Chronos: An authenticated dictionary based on skip lists for timestamping systems [C]// *Proceedings of the Workshop on* Secure Web Services, 2005: 84－90.

[6] SAKAI K, SUN M, KU W, et al. Randomized skip lists-based private authentication for large-scale RFID systems[C]//Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2013: 277－280.

[7] ANAGNOSTOPOULOS A, GOODRICH M T, TAMASSIA R. Persistent authenticated dictionaries and their applications[C]//Lecture Notes in Computer Science, 2001: 379－393.

[8] GOODRICH M T, PAPAMANTHOU C, TAMASSIA R, et al. Athos: Efficient authentication of outsourced file systems[C]// Lecture Notes in Computer Science, 2008: 80－96

[9] ANAGNOSTOPOULOS A, GOODRICH M T, TAMASSIA R. Persistent authenticated dictionaries and their applications[C]// Proceedings of the 4th International Conference on Information Security, 2001: 379－393.

[10] ESINER E, KACHKEEV A. FlexDPDP: FlexList-based optimized dynamic provable data possession[R]. IACR Cryptology ePrint Archive, 2013.

[11] TAMASSIA R. On the cost of authenticated data structures [C]// European Symp on Algorithms, 2003: 2－5.