云南大学学报(自然科学版),2014,36(3):321~328 DOI:10.7540/j.ynu.20130402
Journal of Yunnan University

具有长时安全性的高性能异或秘密共享协议的研究*

陆正福, 晁 巍
(云南大学 数学与统计学院,云南 昆明 650091)

摘要:用于长数据分散存储的秘密共享协议面临着2方面问题,其一为长时安全性问题——份额在长期存储过程中可能渐次泄露,其二为份额分解与重构时的性能问题.已有秘密共享协议无法同时解决上述2个问题,鉴于此,选取Kurihara等的异或秘密共享(XORSS)协议和拜占庭协商协议为基础协议,设计了2个具有长时安全性的增强型异或秘密共享协议——用于份额更新的主动异或秘密共享(PXORSS)协议和用于门限提升的动态门限异或秘密共享(DTXORSS)协议.PXORSS协议和DTXORSS协议基于异或运算进行实现,延续了XORSS协议的高效性,此外给出了数学证明和过程分析,表明2协议满足长时安全性;设计了基于MapReduce模式的云存储原型系统,实验结果表明2协议性能较高、在长数据分散存储方面实用性较强.

关键词:秘密共享;异或;主动安全;动态门限;长时安全;MapReduce
中图分类号:TP 393.08 **文献标志码**:A **文章编号**:0258-7971(2014)03-0321-08

从数学观点看,秘密共享[1-2]是一种在此时(此地)对某个数学对象实施分解、而在彼时(彼地)重构某个数学对象的方法.从算法观点看,秘密共享是一种分布式算法,由几个子算法构成:针对原始秘密的分解算法、份额(即分解后的数据)分发协议、份额收集协议、根据一定门限值的份额重构原始秘密的算法.秘密共享首次出现于1979年,近30年来从多方面得到了发展.

秘密共享早期主要用于现代密码学中的密钥管理[3-4],其中密钥是秘密共享协议中的原始秘密.密钥通常都属于短数据范畴(一般只有几十、几百、几千比特),如果密钥更新频度不高,则性能一般不会受太大影响.后期(尤其是21世纪以来)的情形出现了一些需求变化:①在一些信息系统中,秘密共享中的原始秘密往往是比较长的数据(一般有几GB、几十GB甚至几百GB,在程序实现时往往被分块并当作大整数或者长位串处理),分解、重构等操作的计算量很大,这些应用领域包括网络存储(如云计算系统的云端存储)、容侵系统、灾备系统、图像安全等;②在另一些信息系统中,虽然秘密共享中的原始秘密依然是密钥,但是其密钥更新频度高,例如大规模动态群组通信系统的密钥更新[5]等;③作为安全多方计算[6]的基础协议之一,秘密共享的原始秘密不再局限于密钥;一个熟知的事实是安全多方计算的性能通常比较低下,所以秘密共享的性能会直接影响其上层的安全多方计算协议.上述3方面的要求变化引出了共同的研究问题:高性能秘密共享协议的设计.

此外,上述需求变化中除了性能需求还涉及一些安全性需求,典型的情况是涉及长时安全性威胁:可能会因长时间的系统运行给敌手造成机会,各计算主体所拥有的份额渐次泄露,达到一定门限值后,最终危及受保护的原始秘密.上述长时安全性威胁的本质是被动攻击的成功因素(用于重构的份额集合)会因为时间的推移而单调递增,进而可能达到门限值,因此长时安全性的应对策略是份额定期更新、动态门限提升.

综上所述,新一代的需求变化导致秘密共享必须考虑至少2点新约束:①高性能的基础构造方法;

* **收稿日期**:2013-07-07
基金项目:国家自然科学基金(10861012);云南省教育厅科学研究基金(09Y0347);云南大学理(工)科校级科研资助(2011YB27);云南大学中青年骨干教师培养计划专项(XT412003).
作者简介:陆正福(1965-),男,安徽人,教授,主要从事信息安全、协议工程、网络计算研究.E-mail:zhflu@ynu.edu.cn.

②与基础构造方法相匹配的高性能安全性增强协议. 本文设计的秘密共享协议兼顾了上述高性能和长时安全性 2 个方面的需求.首先,从高性能需求出发,本文选择了 Kurihara 等在 2008 年提出的(k,n)门限的异或秘密共享协议[7](下文简称 XORSS)作为本文协议的基础构造方法;其次,从长时安全性需求出发,在 XORSS 原理分析的基础上结合拜占庭协商协议,设计了份额定期更新协议和动态门限提升协议,从而形成了 2 个能够应对长时安全性威胁的安全增强协议,分别称为主动异或秘密共享协议(Proactive XOR Secret Sharing,PXORSS)和动态门限异或秘密共享协议(Dynamic Threshold XOR Secret Sharing,DTXORSS). PXORSS 和 DTXORSS 都是在异或运算的基础上对 XORSS 进行的扩展,2 协议既能保证长时安全性,又能保证协议整体的高效性.

1 相关工作

1.1 秘密共享相关进展 Blakley 和 Shamir 分别独立地提出了秘密共享概念,其中 Shamir 的(k,n)门限秘密共享协议(Shamir's Secret Sharing,SSS),因其完美性(perfect,从少于 k 个份额中不仅无法获取原始秘密,而且也无法推测出原始秘密的任何信息)和理想性(ideal,各个份额的长度都等于原始秘密的长度)而备受推崇.此后,秘密共享研究主要集中于以 SSS 为基础的安全提升研究、性能提升研究以及实际应用研究等领域.安全提升研究主要包括防止分发者或参与者欺骗的可验证秘密共享(Verifiable Secret Sharing,VSS)研究[8-9],保证长时安全性的主动秘密共享(Proactive Secret Sharing,PSS,又译为动态秘密共享)协议研究[10-11]和动态门限秘密共享(Dynamic Threshold Secret Sharing,DTSS)协议研究[12-13]等,后期衍生出了适于长数据运算的性能提升秘密共享协议研究[14-16].上述扩展研究都是以 SSS 为基础的上层拓展,而 SSS 受制于大整数运算效率低下的限制,无法提供高性能支持,所以基于 SSS 的性能优化的秘密共享协议整体效率仍然偏低.

为了提升秘密共享在长数据处理时的运算效率,Ishizu 等[17]中提出了 XORSS,该方案建立在快速高效的异或运算基础上(将长数据分块当作长位串,然后执行异或运算),完全摆脱了大整数运算性能低下的限制,但该文献只是给出一个(2,3)门限方案,局限性较大.后期在一系列相关研究[18,19]的基础上,Kurihara 等给出了具有一般性的、完美的、理想的、性能优势明显的(k,n)门限方案[7].XORSS 所具备的性能优势,使其可以更好地应用于涉及到长数据分散存储的应用系统中,但是完全基于 XORSS 的应用系统,数据的长时安全性并不能得到保证,主要是由于传统的 PSS 和 DTSS 是对 SSS 的扩展,如果摒弃 SSS 则 2 种安全性增强协议将不再有效,那么也就无法简单地将传统的 PSS 和 DTSS 引入到 XORSS 中.因此需要重新设计基于 XORSS 的 PXORSS 和 DTXORSS,才能在延续 XORSS 高效性的基础上提升其数据长时安全性.

1.2 Kurihara 异或秘密共享介绍

1.2.1 分解算法

$$
\begin{cases}
r_{0\cdot 0+0}^{0} \oplus r_{1\cdot 0+0}^{1} \oplus \cdots \oplus r_{(k-2)\cdot 0+0}^{k-2} \oplus s_{0-0} = w_{(0,0)} \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{第 0 个子组} \\
r_{0\cdot 0+n_p-2}^{0} \oplus r_{1\cdot 0+n_p-2}^{1} \oplus \cdots \oplus r_{(k-2)\cdot 0+n_p-2}^{k-2} \oplus s_{n_p-2-0} = w\ (0,n_p-2) \\
\text{----------} \\
r_{0\cdot 1+0}^{0} \oplus r_{1\cdot 1+0}^{1} \oplus \cdots \oplus r_{(k-2)\cdot 1+0}^{k-2} \oplus s_{0-1} = w_{(1,0)} \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{第 1 个子组} \\
r_{0\cdot 1+n_p-2}^{0} \oplus r_{1\cdot 1+n_p-2}^{1} \oplus \cdots \oplus r_{(k-2)\cdot 1+n_p-2}^{k-2} \oplus s_{n_p-2-1} = w\ (1,n_p-2) \\
\text{----------} \\
\cdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \vdots \\
\text{----------} \\
r_{0\cdot (n-1)+0}^{0} \oplus r_{1\cdot (n-1)+0}^{1} \oplus \cdots \oplus r_{(k-2)\cdot (n-1)+0}^{k-2} \oplus s_{0-(n-1)} = w_{(n-1,0)} \\
\vdots \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{第 } n-1 \text{ 个子组} \\
r_{0\cdot (n-1)+n_p-2}^{0} \oplus r_{1\cdot (n-1)+n_p-2}^{1} \oplus \cdots \oplus r_{(k-2)\cdot (n-1)+n_p-2}^{k-2} \oplus s_{n_p-2-(n-1)} = w_{(n-1,n_p-2)}
\end{cases}
\tag{1}
$$

对于(k,n)门限的异或秘密共享方案,其分解算法首先将一个原始秘密S分解成n_p-1(n_p为一个不小于n的最小素数)份长度为d的秘密子块$s_1,s_2,\cdots,s_{n_p-1}$(不足则补0),写成向量的形式为$\boldsymbol{s}=(s_1,s_2,\cdots,s_{n_p-1})$,为了便于算法介绍引入辅助变量$s_0$(一个长度为$d$的全0子串);然后生成$n_p\cdot(k-1)$个长度为$d$的随机数,分别为$r_0^0,\cdots,r_{n_p-1}^0,r_0^1,\cdots,r_{n_p-1}^1,r_0^{k-2},\cdots,r_{n_p-1}^{k-2}$,由于$r_{n_p-1}^0$不被使用,所以舍弃该随机数,将剩余随机数写成向量的形式为$\boldsymbol{r}=(r_0^0,\cdots,r_{n_p-2}^0,r_0^1,\cdots,r_{n_p-1}^1,\cdots,r_0^{k-2},\cdots,r_{n_p-1}^{k-2})$;接着将$\boldsymbol{r}$和$\boldsymbol{s}$中的分量,按照$w_{(i,j)}\leftarrow(\bigoplus_{h=0}^{k-2}r_{h-i+j}^h)\oplus s_{j-i}$进行异或运算(本文所有的下标运算都是在有限域$GF(n_p)$上进行),从而得到份额子块$w_{(i,j)}$;最后将相应的份额子块按照$w_i\leftarrow w_{(i,0)}\|\cdots\|w_{(i,n_p-2)}$进行合并连接,得到最终的份额$w_i$.

如式(1)所示,上述分解过程可看成已知$\boldsymbol{r}$和$\boldsymbol{s}$求解份额子块$w_{(i,j)}$,第i个子组则表示第i个份额的各个子块的求解过程.将式(1)用矩阵表示,可写为$\boldsymbol{A}\cdot\boldsymbol{e}=\boldsymbol{w}$,$\boldsymbol{A}$为二元域上$n\cdot(n_p-1)\times(k\cdot n_p-2)$的系数矩阵,$\boldsymbol{A}=(\boldsymbol{v}_{(0,0)},\cdots,\boldsymbol{v}_{(0,n_p-2)},\boldsymbol{v}_{(1,0)},\cdots,\boldsymbol{v}_{(1,n_p-2)},\cdots,\boldsymbol{v}_{(n-1,0)},\cdots,\boldsymbol{v}_{(n-1,n_p-2)})^{\mathrm{T}}$,其中$\boldsymbol{v}_{(i,j)}=[\boldsymbol{i}_j^{n_p-1}\boldsymbol{i}_{i+j}^{n_p}\boldsymbol{i}_{2i+j}^{n_p}\cdots\boldsymbol{i}_{(k-2)}^{n_p}\boldsymbol{i}_{j-i-1}^{n_p-1}]$,$0\leqslant i\leqslant n-1$,$0\leqslant j\leqslant n_p-2$,$\boldsymbol{i}_y^x$表示一个$x$维的行向量,其中只有第$y$个分量为1,其余分量全为0;$\boldsymbol{e}$为由随机数和秘密子块组成的$k\cdot n_p-2$维列向量,记为$\boldsymbol{e}=(\boldsymbol{r},\boldsymbol{s})^{\mathrm{T}}$;$\boldsymbol{w}$为份额子块组成的$n\cdot(n_p-1)$维列向量,记为$\boldsymbol{w}=(w_{(0,0)},\cdots,w_{(0,n_p-2)},w_{(1,0)},\cdots,w_{(1,n_p-2)},\cdots,w_{(n-1,0)},\cdots,w_{(n-1,n_p-2)})^{\mathrm{T}}$.易知,$\boldsymbol{A}\cdot\boldsymbol{e}=\boldsymbol{w}$的执行结果与$w_{(i,j}\leftarrow(\bigoplus_{h=0}^{k-2}r_{h-i+j}^h)\oplus s_{j-i}$的执行结果是一致的.

1.2.2　*重构算法*　XORSS分解算法是在已知$\boldsymbol{A}$和$\boldsymbol{e}$的情况下求解$\boldsymbol{w}$,其重构算法则是在已知$\boldsymbol{w}$部分信息(只有k个子组)和$\boldsymbol{A}$的情况下求解$\boldsymbol{e}$中的分量$\boldsymbol{s}$,即通过部分份额求解原始秘密S.首先,从式(1)中任意取出k个子组(假设取出的子组序号为$t_0,t_1,\cdots,t_{k-1}$),形成一个方程组,该方程组可以用矩阵表示为$\boldsymbol{G}\cdot\boldsymbol{e}=\boldsymbol{w}'$,$\boldsymbol{G}$为二元域上的系数矩阵,为$k$个份额的子块组成的$k(n_p-1)$维列向量,记为$\boldsymbol{w}'(w_{t_0,0)},\cdots,w_{(t_0,n_p-2)},w_{(t_1,0}},\cdots,w_{(t_1,n_p-2)},\cdots,w_{(t_{k-1},0)},\cdots,w_{(t_{k-1},n_p-2)})^{\mathrm{T}}$.然后,对方程组做高斯消元法求解,即可解出$s=(s_1,s_2,\cdots,s_{n_p-1},s_0)$,文献[7]已经对其正确性和可行性做了分析和证明.

2　基于 XORSS 的主动秘密共享协议(PXORSS)

2.1　问题描述及模型假设　PSS主要用于解决外部敌手的攻击问题,即存储份额的服务器存在永久性漏洞,外部敌手能够不断地从中获取份额,当敌手获取的份额足够多(大于等于门限值k)时,将能够从中重构原始秘密.PSS通过周期性更新份额,以缩短份额的有效周期,从而达到抑制敌手获取足够多有效份额的目的.本节根据XORSS的特点提出了一个基于XORSS的PSS,命名为PXORSS.

PXORSS协议的设计对于份额存储服务器做了以下假设:①敌手能获取存储在外存中的数据,无法获取加载到内存中的数据.②敌手能够监听窃取份额服务器间的通信信息,但无法控制份额存储服务器.③敌手攻破k个服务器的时间T'大于份额的更新时间周期T,如果T'小于T,则需执行3.2节的动态门限提升协议来提升门限值k,或者缩短更新时间周期T.④份额存储服务器可能出现故障,但出现故障的服务器数量所占比例小于1/3.

2.2　重构算法分析

$$
\begin{aligned}
&w_{(0,2)}\oplus w_{(0,3)}\oplus w_{(1,1)}\oplus w_{(1,2)}\oplus w_{(2,3)}\oplus w_{(4,1)}=\\
&r_2^0\oplus r_2^1\oplus r_2^2\oplus s_2\oplus\\
&r_3^0\oplus r_3^1\oplus r_3^2\oplus s_3\oplus\\
&r_1^0\oplus r_2^1\oplus r_3^2\oplus s_0\oplus\\
&r_2^0\oplus r_3^1\oplus r_4^2\oplus s_1\oplus\\
&r_3^0\oplus r_0^1\oplus r_2^2\oplus s_1\oplus\\
&r_1^0\oplus r_0^1\oplus r_4^2\oplus s_2.
\end{aligned}\tag{2}
$$

以$(k,n)=(4,5)$方案中s_3的重构为例,利用重构算法得到的求解表达式,如式(2)所示.对于等号后面的式子,每一行对应一个秘密子块(第1行是$w_{(0,2)}$的表达式,最后1行是$w_{(4,1)}$的表达式),每一列都存

在相等的随机数(如第1列的 r_2^0 与 r_2^0,r_3^0 与 r_3^0 等),在异或运算的作用下相等的数被消去,最后只剩余 $s_3 \oplus s_0$,而 s_0 则是一个全为0的字符串,所以最终 s_3 被成功重构,类似地 s_1, s_2, s_4 也以相同的方式被成功重构.整个重构过程利用了异或运算的性质——相同元素异或值为0,任何元素与0异或为其本身,最终可重构将被随机数隐藏的秘密子块.从上述分析可知,如果为每个份额子块 $w_{(i,j)}$ 都异或共同的随机数 r_{add},在原始秘密的重构过程中该随机数 r_{add} 会被消去,那么该操作可达到更新份额的目的,且不影响原始秘密的重构.

2.3 主动异或秘密共享协议—PXORSS 综合2.2节的分析,现给出PXORSS协议,该协议在不改变原始秘密的情况下,对份额进行定期更新,从而抵制敌手获取足够多的有效份额.由于协议中需要用到一致性参数,如果系统中存在故障节点,则会影响一致性参数的获取,为此在PXORSS协议中引入了BAP(Byzantine Agreement Protocol,拜占庭协商协议)原语,该原语在故障节点数量所占比例小于1/3情况下,保证无故障节点协商出一致性参数.具体协议如下.

协议1:PXORSS协议

输入:w_i(旧份额),$p_{\text{agreement}}$的份额

输出:w_1'(新份额)

步骤1:各参与者执行BAP协议原语,得到共同的随机数 $r_{\text{agreement}} \in \{0,1\}^d$

步骤2:利用 $p_{\text{agreement}}$ 的份额重构 $p_{\text{agreement}}$,计算 $r_{\text{add}} \leftarrow r_{\text{agreement}} \oplus p_{\text{agreement}}$

步骤3:$w_{(i,0)} \| \cdots \| w_{(i,n_p-2)} \leftarrow w_i$

步骤4:for $j \leftarrow 0$ to n_p-2 do $w'_{i,j)} \leftarrow w_{(i,j)} \oplus r_{\text{add}}$ end for

步骤5:$w_i' \leftarrow w'_{(i,0)} \| \cdots \| w'_{(i,n_p-2)}$

步骤6:删除 $w_i, r_{\text{agreement}}, p_{\text{agreement}}, r_{\text{add}}$

步骤7:返回 w_1'.

为了防止协商的随机数 $r_{\text{agreement}}$ 被外部敌手窃取,不直接将 $r_{\text{agreement}}$ 作为 r_{add},而是在协议中引入长度为 d 的约定参数 $p_{\text{agreement}}$(该参数使用秘密共享分解算法进行分解,以份额的形式独立存放,只有在执行PXORSS协议时才重构使用).将 $r_{\text{agreement}}$ 与 $p_{\text{agreement}}$ 做本地处理得到 r_{add}(步骤2),然后将自身拥有的份额 w_i 分解为长度为 d 的子块 $w_{(i,0)}, w_{(i,1)}, \cdots, w_{(i,n_p-2)}$,接着让每个子块与 r_{add} 做异或操作,得到更新后的份额子块 $w'_{(i,0)}, w'_{(i,1)}, \cdots, w'_{(i,n_p-2)}$,最后将各个份额子块合并成更新后的秘密份额 w_i'.

2.4 PXORSS正确性证明

定理1 对于一个 (k,n) 门限的XORSS,设 $\boldsymbol{P}$ 是一个由任意 k 个参与者组成的集合,若对所有参与者 $P_i \in \boldsymbol{P}, (i=0,1,\cdots,k-1)$,都将其拥有的份额子块 $w_{(i,j)}$ 与一个相同的增量 r_{add} 做异或操作形成新的份额子块 $w'_{(i,j)}$,则按照XORSS的重构算法,仍然可以从 $w'_{(i,j)}$ 中重构原始秘密.

证明 设 $\boldsymbol{P} = \{P_{t_0}, P_{t_1}, \cdots, P_{t_{k-1}}\}$ 表示一个拥有 k 个参与者的集合,其中 $t_0, \cdots, t_{k-1} \in DF(n_p)$ 为任意数值,但需满足 $0 \leqslant t_i, t_j \leqslant n-1$ 且如果 $i \neq j$ 则 $t_j \neq t_j$.

现在定义2个矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$,使其满足:$\boldsymbol{w} = \boldsymbol{U} \cdot \boldsymbol{r} \oplus \boldsymbol{V} \cdot \boldsymbol{s} = (w_{(t_0,0)}, \cdots, w_{(t_0,n_p-2)}, \cdots, w_{(t_{k-1},0)}, \cdots, w_{(t_{k-1},n_p-2)})^{\mathrm{T}}$,

其中 $\boldsymbol{r}$ 和 $\boldsymbol{s}$ 分别为:$\boldsymbol{r} = (r_0^0, \cdots, r_{n_p-2}^0, r_0^1, \cdots, r_{n_p-2}^1, \cdots, r_0^{k-2}, \cdots, r_{n_p-2}^{k-2})^{\mathrm{T}}$,$\boldsymbol{s} = (s_1, \cdots, s_{n_p-1})^{\mathrm{T}}$,

由XORSS的重构算法可知 $[\boldsymbol{UV}] = \boldsymbol{G}$.对于 $k(n_p-1)$ 维向量 $\boldsymbol{R} = (r_{\text{add}}, r_{\text{add}}, \cdots, r_{\text{add}})$,令:

$\boldsymbol{w'} = \boldsymbol{w} \oplus \boldsymbol{R} = (w_{(t_0,0)} \oplus r_{\text{add}}, \cdots, w_{(t_0,n_p-2)} \oplus r_{\text{add}}, \cdots, w_{(t_{k-1},0)} \oplus r_{\text{add}}, \cdots, w_{(t_{k-1},n_p-2)} \oplus r_{\text{add}})^{\mathrm{T}}$

$\boldsymbol{w'}$ 是 $\boldsymbol{P}$ 中 k 个参与者的份额子块经过与 r_{add} 异或操作后得到的份额向量.下证通过 w' 可重构原始秘密 S.由XORSS的分解算法可知 $w_{(i,j)} = (\bigoplus_{h=0}^{k-2} r_{h-i+j}^h) \oplus s_{j-i}$,其中 $0 \leqslant i \leqslant n-1, 0 \leqslant j \leqslant n_p-2$,

则 $w'_{(i,j)} = w_{(i,j)} \oplus r_{\text{add}} = (\bigoplus_{h=0}^{k-2} r_{h-i+j}^h) \oplus s_{j-i} \oplus r_{\text{add}} = (\bigoplus_{h=0}^{k-2} r_{h-i+j}^h) \oplus s'_{j-i}, (s'_{j-i} = s_{j-i} \oplus r_{\text{add}})$.

所以可得:$\boldsymbol{w'} = \boldsymbol{w} \oplus \boldsymbol{R} = \boldsymbol{U} \cdot \boldsymbol{r} \oplus \boldsymbol{V} \cdot \boldsymbol{s'}$,其中:$\boldsymbol{s'} = (s'_1, s'_2, \cdots, s'_{n_p-1})^{\mathrm{T}}, s'_m = s'_m \oplus r_{\text{add}}, m = 1,2,\cdots,n_p-1.$

由文献[7]中的定理2可知,可以通过对 $[\boldsymbol{UV}]$ 作行变换得到 $[\overline{\boldsymbol{UV}}]$,使得:

$$[\overline{U}\overline{V}]\cdot\begin{bmatrix} r \\ \cdots \\ s' \end{bmatrix}=\begin{pmatrix} * \\ \cdots \\ * \\ \hline (s'_{\alpha}\oplus s'_{\beta}) \\ (s'_{\alpha+1}\oplus s'_{\beta+1}) \\ \cdots \\ (s'_{\alpha-2}\oplus s'_{\beta-2}) \end{pmatrix}=\begin{pmatrix} * \\ \cdots \\ * \\ \hline ((s_{\alpha}\oplus r_{\text{add}})\oplus(s_{\beta}\oplus r_{\text{add}})) \\ ((s_{\alpha+1}\oplus r_{\text{add}})\oplus(s_{\beta+1}\oplus r_{\text{add}})) \\ \cdots \\ ((s_{\alpha-2}\oplus r_{\text{add}})\oplus(s_{\beta-2}\oplus r_{\text{add}})) \end{pmatrix}=$$

$$\begin{pmatrix} * \\ \cdots \\ * \\ \hline (s_{\alpha}\oplus s_{\beta})\oplus(r_{\text{add}}\oplus r_{\text{add}}) \\ (s_{\alpha+1}\oplus s_{\beta-1})\oplus(r_{\text{add}}\oplus r_{\text{add}}) \\ \cdots \\ (s_{\alpha-2}\oplus s_{\beta-2})\oplus(r_{\text{add}}\oplus r_{\text{add}}) \end{pmatrix}=\begin{pmatrix} * \\ \cdots \\ * \\ \hline (s_{\alpha}\oplus s_{\beta}) \\ (s_{\alpha+1}\oplus s_{\beta-1}) \\ \cdots \\ (s_{\alpha-2}\oplus s_{\beta-2}) \end{pmatrix}. \tag{3}$$

其中 $\alpha=-\sum_{i=0}^{k-3}t_i-t_{k-2}+t_{k-1}$,$\beta=-\sum_{i=0}^{k-3}t_i+t_{k-2}-t_{k-1}$,而文献[7]中定理2的证明过程表明可以从集合:$\{(s_{\alpha}\oplus s_{\beta}),(s_{\alpha-1}\oplus s_{\beta-1}),\cdots,(s_{\alpha-2}\oplus s_{\beta-2})\}$ 中通过递推的方式得到集合$\{s_1,s_2,\cdots,s_{n_p-1}\}$,又原始秘密可以表示为 $s=s_1\|s_2\|\cdots\|s_{n_p-1}$,综上可知,定理 1 得证.

2.5　协议安全性分析　假设敌手需要时间 T' 才能获取足够多的份额,份额存储服务器按照周期 T 进行份额更新,且 $T<T'$,则敌手无法及时获取足够多的有效份额,因为旧份额在 PXORSS 执行后就会失效,所以敌手无法从中重构原始秘密.但是如果 r_{add} 泄露,则敌手可以将失效的旧份额按照步骤 3~5 作更新就能得到有效的新份额,所以需要重点保证 r_{add} 的安全性.

BAP 协议的引入可以在存在故障节点的情况下,保证无故障节点正确执行协议,但协商出的 $r_{\text{agreement}}$ 可能在协商过程中泄露,所以协议中没有直接让 $r_{\text{agreement}}$ 作为 r_{add},而是将 $r_{\text{agreement}}$ 与重构的 $p_{\text{agreement}}$ 做本地处理得到 r_{add}.由于 $p_{\text{agreement}}$ 份额较短,可以利用外部设备存储,有效地保证了 $p_{\text{agreement}}$ 份额的安全.$p_{\text{agreement}}$ 的引入降低了 $r_{\text{agreement}}$ 泄露所带来的损失,因为即使外部敌手获取 $r_{\text{agreement}}$,也无法推测出最终的 r_{add}.

3　基于 XORSS 的动态门限提升协议(DTXORSS)

虽然通过份额的定期更新可以很好地提升 XORSS 的长时安全性,但是如果份额更新的周期较长,则敌手仍然可能在某个周期内获取足够多的份额,通常可以采取缩短份额更新周期来防止上述情况的出现,但是如果频繁执行更新操作将会严重影响存储服务器的运行效率,所以需要新的手段来削弱敌手获取足够多份额的能力,从而达到既能保证份额长时安全性又能保证系统运行效率的目的.DTSS 可以有效地满足上述需求,利用该思想可在不重新分配份额的情况下动态提升秘密共享的门限值 k,从而有效延长敌手获取足够多份额的时间.鉴于传统的 DTSS 无法适用于 XORSS,本文在 XORSS 的基础上设计了 DTSS,命名为 DTXORSS.

3.1　分解算法分析　从 XORSS 的分解算法可知,门限值 k 对份额的主要影响体现在份额子块形成表达式:$w_{(i,j)}\leftarrow(\bigoplus_{h=0}^{k-2}r_{h-i+j}^{h})\oplus s_{j-i}$.给定2个 XORSS 协议,其参数分别为$(k_1,n)$ 和(k_2,n),$k_2>k_1$,产生份额的主要区别在于,(k_2,n) 方案的每个份额子块较之(k_1,n) 方案多异或了 k_2-k_1 个随机数,例如$(k,n)=(4,5)$ 方案的份额子块 $w_{(i,j)}=r_j^0\oplus r_{2i+j}^2\oplus s_{j-i}$,相比于$(k,n)=(3,5)$ 方案的份额子块 $w_{(i,j)}=r_j^0\oplus r_{i+j}^1\oplus s_{j-i}$ 多异或了一个随机数 r_{2i+j}^2.

将(k_1,n) 方案提升为(k_2,n) 方案,只需将(k_1,n) 方案得到的各个秘密份额中的份额子块,按照(k_2,n) 方案的规则补充异或上 k_2-k_1 个随机数即可实现.鉴于上述思路,下文给出了 DTXORSS 协议.

3.2　基于 XORSS 的动态门限协议—DTXORSS　给出 DTXORSS 协议,该协议执行后完成份额的更新,门限值 k 提升为 k+1,其中 $\boldsymbol{h}_x^y$ 表示 x 维的二元行向量,只有第 y 维为 1,其余维全为 0.$\boldsymbol{R}$ 是由协商出的 n_p 个随机数组成的 n_p 维向量,$\boldsymbol{H}$ 是 $n_p\times(n_p-2)$ 的矩阵,$\boldsymbol{R}$ 与 $\boldsymbol{H}$ 相乘以后得到 n_p-1 维向量 $\boldsymbol{R}^{\text{add}}$,$\boldsymbol{R}^{\text{add}}$相当于 $\boldsymbol{R}$

中各元素进行了重排后的结果.步骤3将 $\boldsymbol{R}^{\text{add}}$ 中的分量与份额的各个子块进行异或运算,该步相当于补齐份额在(k,n)方案中相比于$(k+1,n)$方案中缺少的异或运算.

协议2:DTXORSS协议

输入:w_i(旧份额), $p_{\text{agreement}}$的份额

输出:w_i'(新份额)

步骤1:参与者服务器 P_i 与其他参与者服务器执行 BAP 协议原语,共同协商出 n_p 个随机数 $r_1,r_2,\cdots,r_{n_p\in|1,0|^d}$

步骤2:利用 $p_{\text{agreement}}$ 的份额重构 $p_{\text{agreement}}$

步骤3:对于向量 $\boldsymbol{R}=(r_1,r_2,\cdots,r_{n_p})$ 和 $\boldsymbol{H}=\boldsymbol{h}_{n_p}^{i(k-1)},\boldsymbol{h}_{n_p}^{i(k-1)+1},\cdots,\boldsymbol{h}_{n_p}^{i(k-1)+n_p-2})$ 计算 $\boldsymbol{R}^{\text{add}}=\boldsymbol{R}\cdot\boldsymbol{H}=(r_{t_0},r_{t_1},\cdots,r_{t_{n_p-2}})$

步骤4:for $j\leftarrow 0$ to n_p-2 do $w'_{(i,j)}\leftarrow w_{(i,j)}\oplus p_{\text{agreement}}$ end for

步骤5:$w_i'\leftarrow w'_{(i,0)}\|\cdots\|w'_{(i,n_p-2)}$

步骤6:删除 $r_1,r_2,\cdots,r_{n_p},p_{\text{agreement}}$ and w_i

步骤7:返回 w_i'.

3.3 协议正确性和安全性分析 上述协议通过份额与随机数的异或运算补齐操作,达到门限提升的目的,该过程中的异或运算补齐操作与XORSS的分解算法保持一致,所以可以按照XORSS的重构算法进行原始秘密的重构.该协议可以保证XORSS的门限值由 k 提升为 $k+1$,对任意 $k'>k$,将该协议执行 $k'-k$ 次,就可以完成(k,n)方案向(k',n)方案的提升,从而可以有效地延长攻击者获取足够份额的时间.

基于与PXORSS协议相同的安全性考虑,在DTXORSS协议中引入了BAP协议原语和约定参数 $p_{\text{agreement}}$,其中BAP协议原语用于保证随机数协商的一致性,约定参数 $p_{\text{agreement}}$ 则是为了防止实际用于DTXORSS协议的随机数被外部敌手窃取.

4 PXORSS和DTXORSS的原型实现

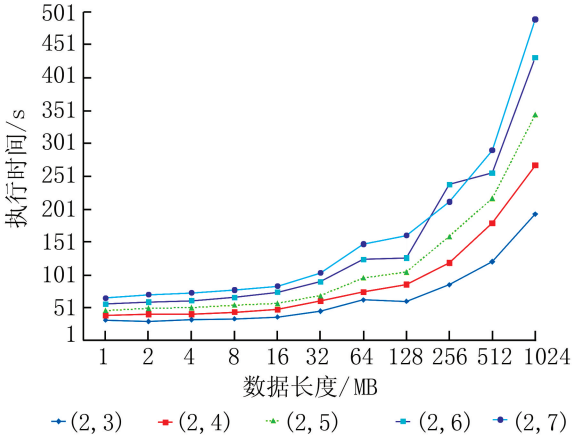
4.1 实验过程 为了验证PXORSS和DTXORSS在长数据分散存储系统中的高效性,我们利用Hadoop平台搭建了小型云存储系统,该系统由1台客户机节点(Intel Pentium处理器,双核双线程,主频为1.6 GHz,2 GB内存,120 GB硬盘)、3台DataNode节点(Intel Pentium处理器,单核双线程,主频为3.0 GHz,2 GB内存,120 GB硬盘)和1台NameNode节点(Intel E8501处理器,四核四线程,主频为2.4 GHz,4 GB内存,120 GB硬盘)组成,各节点操作系统为Ubuntu12.04.1(32位).以该系统为基础,利用MapReduce模式编程实现了对大小为 2^i MB($i=0,1,2,\cdots,10$)的文件数据的相关处理.

由于相同长度的数据的处理效率主要取决于分解的份额数 n,而最小重构数 k 对处理效率影响非常小,所以选取门限值$(k,n)=(2,3),(2,4),(2,5),(2,6),(2,7)$的XORSS作为测试方案,并在MapReduce模式下对分散存储在DataNode中的份额执行份额更新和门限提升的相关操作.在MapReduce的处理过程中,一个map任务所处理的分片总长度为64 MB(不足64 MB则选取数据实际长度),一次处理的记录长度为16 MB(不足16 MB则选取数据实际长度),每次的处理结果被写入单独的子文件中.具体的执行结果如图1和图2所示.

4.2 实验结果分析 上述实验结果表明对于相同长度的数据,门限值中的 n 值越大,则份额更新以及门限提升的时间就越长,而给定门限值 k 和 n,则份额更新和门限提升的时间都呈现出指数级增长,其原因在于数据的长度在以指数级增长.

以$(k,n)=(2,3)$方案为例,各长度数据平均处理速度分别为0.03、0.06、0.12、0.23、0.43、0.69、1.02、2.1、2.9 MB/s和4.2 MB/s,可以看出长数据处理的性能优势明显,主要由于MapReduce模式在任务处理过程中需要启动大量的map任务和reduce任务,任务的启动会消耗掉一部分时间,这些时间相对于长数据的处理时间非常微小,但相对于短数据的处理时间却非常可观,所以在长数据处理上MapReduce模式更具优势.如果扩大集群规模(增加DataNode的数量),那么并行处理的map任务和reduce任务将会更多,长

数据的处理时间将会进一步缩短.



1 PXORSS 协议执行结果

Fig.1 Result of PXORSS Protocol's Implementation

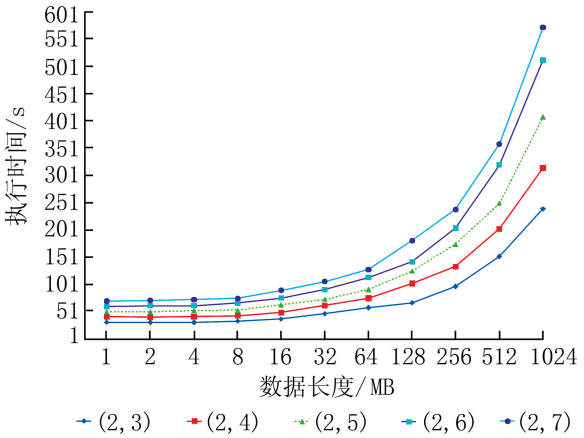


图 2 DTXORSS 协议执行结果

Fig.2 Result of DTXORSS Protocol's Implementation

实验的整体结果表明,本文所设计的 2 个增强型异或秘密共享协议在 MapReduce 模式下可以顺利执行,数据的处理速度能够满足性能要求,可以很好地应用于网络存储(包括云计算系统的云端存储)、容侵系统、灾备系统、图像安全等长数据分散存储的相关系统.

5 结束语

传统的 SSS 协议涉及到有限域上的相关运算,每次处理的数据长度受制于有限域的选取,而且在处理过程中需要进行大量的大整数运算,在待分解秘密较长的情况下,特别是在涉及到长数据操作的应用系统中,其执行效率将会非常低下.Kurihara 等提出的 XORSS 则以位异或运算为基础、从方案构造的数学方法上突破了大整数运算的限制,每次处理的数据长度可以非常大,从而提升了秘密共享的执行效率,为秘密共享用于长数据应用系统奠定了基础.PXORSS 和 DTXORSS 是对 XORSS 的扩展设计,不仅能延续原有 XORSS 协议的高效性,同时又增强了份额在存储过程中的长时安全性,更加有利于秘密共享协议的应用和发展.但是本文的 2 个安全协议只能抵抗被动攻击,如果份额发生错误或被修改,将会影响到系统的可靠性或数据完整性,所以还应考虑加入验证机制,构造可验证的 XORSS 协议,从而达到能够抵抗主动攻击的目的.限于篇幅,另文撰写.

参考文献:

[1] SHAMIR A.How to share a secret[J].Communications of the ACM,1979:612-613.

[2] BLAKLEY G R.Safeguarding cryptographic keys[C]//Proceedings of the National Computer Conference,New York,1979:313-317.

[3] 刘木兰,张志芳.密钥共享体制和安全多方计算[M].北京:电子工业出版社,2008.
LIU M L,ZHANG Z F.Secret sharing schemes and secure multiparty computation[M].Beijing:Publishing House of Electronics Industry,2008.

[4] 杨春尧,陆正福,李军.UC 安全的动态群组密钥协商协议设计与分析[J].通信技术,2014,47(1):81-85.
YANG C Y,LU Z F,LI J.Design and analysis of dynamic group key agreement protocol with UC security[J].Communications Technology,2014,47(1):81-85.

[5] 陆正福,杜飞.基于安全多播的大规模 VSS 协议优化[J].云南大学学报:自然科学版,2012,34(5):514-520.
LU Z F,DU F.Optimization for large-scale VSS based on secure multicast[J].Journal of Yunnan University:Natural Sciences Edition,2012,34(5):514-520.

[6] YAO A C.Protocols for secure computation[C]//Proceeding of 23rd Annual IEEE Symposium on the Foundation of Computer Science,1982:160-164.

[7] KURIHARA J,KIYOMOTO S,FUKUSHIMA K,et al.A new (k,n)-threshold secret sharing scheme and its extension[C].

Proceedings of the 11th International Conference on Information Security, ISC 2008, Taipei, Taiwan, 2008: 455-470.

[8] FELDMAN P.A practical scheme for non-interactive verifiable secret sharing[C]//28th Annual Symposium on Foundations of Computer Science Los Sngeles, USA, 1987: 427-438.

[9] PEDERSEN T P.Non-interactive and information-theoretic secure verifiable secret sharing[C]//Lecture Notes in Computer Seienee, 1992, 129-140. Doi: 10.100713-540-46766-1_9.

[10] HERZBERG A, JARECKI S, KRAWCZYK H, et al. Proactive secret sharing or: How to cope with perpetual leakage[C]// Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, 1995: 339-352.

[11] SCHULTZ D A, LLSKOV B, LLSKOV M. Mobile proactive secret sharing[C]//Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing, 2008: 458-490.

[12] MARTIN K, PIEPRZYK J, SARAWI-NAINI R, et al. Changing threshold in the absense of secure channel[C]//Information Securrzy and Privaey Springer Berlin Heidelberg, 1999, 1587: 177-191.

[13] TARTARY C, WANG Hua-xiong. Dynamic threshold and cheater resistance for shamir secret sharing scheme[C]//2nd SKLOIS Conference on Information Security and Cryptology, LNCS, 2006, 4318: 103-117.

[14] YANG X Y. The short share secret sharing agreement with matrix factorization[C]//International Conference on Information Engineering and Computer Science, 2009: 1-4.

[15] SHEN F, MEI C. Towards secure and reliable data storage with multi-coefficient secret sharing[C]//IEEE 10th International Conference on Computer and Information Technology (CIT), 2010: 797-802.

[16] PARAKH A, KAK S. Recursive secret sharing for distributed storage and information hiding[C]//IEEE 3rd International Symposium on Advanced Networks and Telecommunication Systems (ANTS), 2009: 1-3.

[17] ISHIZU H, OGIHARA T. A study on long-term storage of electronic data[C]//Proc IEICE General Conf, 2004: 125.

[18] HOSAKA N, TOCHIKUBO K, FUJII Y, et al. $(2, n)$-threshold secret sharing systems based on binary matrices[C]//Proc SCIS, 2007.

[19] KURIHARA J, KIYOMOTO S, FUKUSHIMA K, et al. A fast $(3, n)$-threshold secret sharing scheme using exclusive-or operations[C]//IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2008, E91-A (1): 127-138.

Research on high-performance XOR secret sharing protocols with long-term security

LU Zheng-fu, CHAO Wei

(School of Mathematics and Statistics, Yunnan University, Kunming 650091, China)

Abstract: Secret sharing protocols for decentralized storage of long-data face two problems. The first one is long term security threatening—shares may leak during the period of long-term storage. The second one is performance degrading when constructing shares and reconstructing secret. Traditional secret sharing shcemes cannot solve both of the two problems at the same time. For this reason, we chose XOR Secret Sharing (XORSS) protocol proposed by Kurihara et al. and Byzantine Agreement Protocol as the underlying protocols, then designed two long-term security enhanced XOR Secret Sharing protocols—Proactive XOR Secret Sharing (PXORSS) protocol for share updating and Dynamic Threshold XOR Secret Sharing (DTXORSS) protocol for threshold increasing. PXORSS protocol and DTXORSS protocol are designed based on XOR operations, and both of the two protocols extend the high-performance of XORSS protocol. Mathematical proof and procedure analysis show that the above two protocols can meet the requirement of long-term security; furthermore, prototyping cloud storage system using MapReduce paradigm illustrates that PXORSS protocol and DTXORSS protocol designed by this paper are high-performance and practical for decentralized storage of long-data.

Key words: secret sharing; XOR (Exclusive OR); proactive security; dynamic threshold; long-term security; MapReduce