



















RSA 密钥对  $(n, e, d)$ , 大小为  $2\log_2 N$  比特; 日志项的签名密钥对, 大小为  $2\log_2 N$  比特; 持久状态  $\alpha=(B)$ , 大小为  $m \cdot \log_2 T_{\max}$  比特(其中,  $m$  为此 DPA 管理的文件组个数).

为了比较 IDPA-MF-PDP 的性能, 我们引入了两个基本的 MF-PDP 方案作为比较基准, 其中, PDP 方案用文献[2]中的方案分别为文件组中每个文件做数据持有性检查; DPDP 方案以文献[4]中第 3.2 节的方案为基础, 用一个多层次跳跃表(skip list)维护文件组中所有文件. DPDP 以文件为单位, 将文件块的 tag 值作为叶子节点计算一个跳跃表, 并将文件组中每个文件跳跃表的起始节点(start node)作为叶子节点构造父级跳跃表. 假设文件组中有  $n$  个文件, 且这些文件中单个文件块数最大值为  $T_{\max}$ , 则文件组跳跃表高度为  $\log n + \log T_{\max}$ . 在 Prove 阶段, DPDP 返回  $c$  个抽样块的 tag 值和文件块聚合值  $M$  及这些块在跳跃表中的验证路径(verification path); Verify 阶段, 首先根据 DPA 本地保存的文件组起始节点验证  $c$  个 tag 值及其返回路径, 再做  $c$  个 tag 的乘法聚合值, 将其与  $M$  做比较. 3 个方案均基于本文提出的由 DPA 担任隐式第三方的审计架构.

表 1 总结了 IDPA-MF-PDP 的复杂度和开销构成, 并且将其与 DPDP 和 PDP 方案做出比较. 从表中可以得出: IDPA-MF-PDP 将所有开销均减小到  $O(1)$ , 其中, DPA 计算开销和存储开销的大幅降低使基于可信硬件的 DPA 得以实现. 需要特别指出: DPDP 方案的服务器计算开销虽然不包含指数运算, 在表中表示为  $O(1)$ , 但其需为每个挑战块利用哈希计算验证路径, 耗时为  $O(\log n)$ .

Table 1 Complexity of PDP, DPDP and IDPA-MF-PDP

表 1 PDP, DPDP 和 IDPA-MF-PDP 审计交互的开销复杂度与构成

方案	服务器 I/O (抽样块个数)	服务器计算开销 (指数运算次数)	DPA 计算开销 (指数运算次数)	服务器与 DPA 间 通信开销(比特)	DPA 存储 开销(比特)
PDP 复杂度	$O(n)$	$O(n)$	$O(n)$	$O(n)$	$O(n)$
DPDP 复杂度	$O(1)$	$O(1)$	$O(1)$	$O(\log n)$	$O(1)$
IDPA-MF-PDP 复杂度	$O(1)$	$O(1)$	$O(1)$	$O(1)$	$O(1)$
IDPA-MF-PDP 开销构成	$c$	$c$	$c+5$	$\log_2 T_{\max} + 2\kappa + L' + \log_2 N + 2 ER  +  LE $	$4\log_2 N + \log_2 T_{\max}$

说明:  $n$  为文件组中文件的个数(在审计交互协议中, 服务器端向 DPA 返回响应时, 一并将文件组的项引用 ER 返回给 DPA, 其大小为  $|ER|$ . DPA 生成日志, 并将其返回给服务器存储时, 也会产生常数级的通信开销, 包括新的项引用 ER 新生成日志项 LE, 其大小为  $|ER| + |LE|$ ).

## 4.2 安全性分析

我们通过 4 个定理将 IDPA-MF-PDP 的数据持有性审计过程分解为结果的生成(定理 1、定理 2)和结果的保存(定理 3、定理 4)两个环节论证其安全性. 我们假设在 IDPA-MF-PDP 架构中 DPA 是可信的, 即, 外界不能修改 DPA 用于生成日志的时钟、密钥、程序等内部状态和数据(见第 2.2 节).

**定理 1.** 在 DPA 持有正确的文件组持久状态的前提下, MF-PDP 具有和基本 PDP 方案<sup>[2]</sup>等同的安全性.

证明: 在 MF-PDP 中, 我们把文件组作为数据持有性审计的单元. 文件组中包含不定数目的文件, 每个文件都被分成特定长度的数据块, 每个数据块都被赋予一个在文件组中的虚拟下标. 在 Prove 阶段, SSP 先根据文件组持久状态为挑战生成一组在文件组中的虚拟下标, 然后, 通过这组虚拟下标生成同态认证元, 计算需要返回的证据  $P$ . 同样, 在 Verify 阶段, DPA 也需先计算出挑战块的虚拟下标, 再将自己计算的验证值与 SSP 返回的证据  $P$  相比较(见第 1.2 节). 这样, 一个文件组就可以被看作一个虚拟的大文件, 对文件组中任意文件数据块的损坏等价于对虚拟文件数据块的损坏, 对文件组内所有文件的审计就归约为对这个虚拟文件的审计, MF-PDP 的安全性就可以规约为单文件数据持有性的安全性. 而在文献[2]中, Ateniese 等人已给出严格的单文件 PDP 的安全性证明, 本文将不再赘述(关于 MF-PDP 安全性更详尽的分析, 参见文献[16]). □

**定理 2.** 在一个文件组的由正常操作组成的生命周期内, DPA 始终持有正确的该文件组持久状态.

证明: 在 IDPA-MF-PDP 中, DPA 为每个文件组保持一个持久状态, 作为生成挑战的依据. 由于 DPA 是可信的, 外界只能通过 SSP 和 DPA 间的交互协议. 在创建文件组时, DPA 将文件组的  $\alpha$  初始化为 0, 保证其初始状态的正确性(第 3.1 节). 在向文件组添加文件时, 用户与 DPA 都将自身存储的  $\alpha$  值进行修改, 其中, 用户修改后的  $\alpha$  值一定

是正确的.DPA 将修改后的 $\alpha$ 值用自己的私钥进行签名,传递给用户进行比对.用户用 DPA 的公钥对签名消息进行验证并提取出签名消息中的 $\alpha$ 与自身的 $\alpha$ 进行比对,如果结果一致,则 DPA 中现有的 $\alpha$ 也是正确的(第 3.2 节).因此,只要文件组的创建过程以及向文件组添加文件的过程正确执行,则 DPA 中保存的 $\alpha$ 一直是正确的.  $\square$

定理 1 和定理 2 保证了每次数据持有性审计生成结果的可信性.为了给出关于记录审计结果的审计历史的可信性的定理 3 和定理 4,下面先给出 3 个相关定义.

**定义 1(项引用 ER 的可验证性).** 称项引用 ER 是可验证的,当且仅当:

- (1) 用 DPA 的公钥  $dpk$  可验证  $ER.sig$ ;
- (2)  $ER.time$  落在距离当前时刻最近的  $epoch$  内.

**定义 2(日志项 LE 的可验证性).** 称日志项 LE 是可验证的,当且仅当用 DPA 公钥  $dpk$  可验证  $LE.sig$ .

**定义 3(审计历史的一致性).** 称文件组  $G$  的审计历史  $h$  处于一致状态,当且仅当:

- (1) 项引用 ER 是可验证的;
- (2)  $ER.eid$  指向的日志项 LE 存在且可验证;
- (3) 对于日志链表中每一日志项 LE,若  $LE.prev\_id$  不为空,其指向的 LE 存在且可验证.

**定理 3.** 当 SSP 正确按照第 3.4 节的交互协议与 DPA 进行交互时,所生成的关于文件组  $G$  的审计历史  $h$  一定处于一致状态.

**证明:**根据第 3.4 节中的审计过程交互协议,当 DPA 收到来自 SSP 的响应  $P$  及当前文件组的项引用 ER 时,会先验证  $ER.sig$  的正确性和  $ER.time$  的新鲜性,验证通过后方可继续执行.DPA 周期性地发起挑战,在每个  $epoch$  内至少发生一次 ER 的更新,只要  $ER.time$  落在最近的  $epoch$  内,则此 ER 确实为在最新一次审计过程中修改过的 ER.DPA 生成用其私钥签名的保存本次审计结果的 LE,并生成新的 ER 使其  $eid$  指向 LE.根据定义 1~定义 3 易知:若 SSP 正确按照交互协议与 DPA 进行交互,得到的审计历史  $h$  一定处于一致状态.

用户可按照如下过程验证某个文件组  $G$  的审计历史  $h$  的一致状态:向 SSP 发送文件组  $GID$ ,SSP 根据用户请求返回该文件组的日志链表  $l$ ;用户接收到  $l$  后首先验证 ER,接着逐项验证链表中日志项 LE,直至最后一个可验证的 LE 的  $prev\_id$  域为空为止.  $\square$

**定理 4.** 文件组  $G$  的审计历史  $h$  在某个  $epoch$  内处于一致状态  $s_1$ ,则在  $h$  达到下一个  $epoch$  内的一致状态  $s_2$  之前,任何针对  $h$  一致性的攻击(修改或丢弃)都无法在 DPA 与 SSP 审计交互过程中通过交互协议,或者无法在用户审查  $h$  时正确通过用户审计,而使  $h$  仍处于一致状态.

**证明:**假设  $h$  在一致状态  $s_1$  时的索引项为  $ER_1$ ,敌手在  $h$  到达下一个一致状态  $s_2$  前试图对其篡改.下面分 3 种情况讨论:(1) 如果敌手试图修改  $ER_1$ ,由于敌手无法掌握 DPA 的私钥,则其不能伪造  $ER.sig$ ,所以其对 ER 的任何字段的修改将违背定义 3 的条件(1),使审计历史  $h$  偏离一致状态;(2) 如果审计日志链表不空且敌手试图修改或丢弃第一个 LE,将违背定义 3 的条件(2), $h$  偏离一致状态;(3) 假设审计日志链表日志项个数大于 1 且敌手试图修改或丢弃除第一个 LE 外的其他 LE,将违背定义 3 的条件(3), $h$  偏离一致状态.由以上分析可知:对审计历史  $h$  的任何部分的修改或丢弃,都会使其偏离在某个  $epoch$  内的一致状态.  $\square$

定理 3 和定理 4 保证了 SSP 和 DPA 审计交互过程中生成处于一致状态的审计历史,且该历史一旦生成就无法篡改.综上,IDPA-MF-PDP 的数据持有性审计是安全的.

## 5 系统实现和实验评估

### 5.1 系统实现

我们在 Linux 平台上用 C 语言实现了 IDPA-MF-PDP 原型系统,同时实现了第 4.1 节中提到的适用于文件组的 DPDP 方案以及基于单文件 PDP 方案<sup>[2]</sup>的原始 MF\_PDP 方案作为对照,在后文中,分别用 DPDP 和 PDP 表示.所有密码操作实现均来自于 OpenSSL(版本号 0.9.8o)<sup>[18]</sup>.我们采用两台配置相同的服务器来模拟云服务器和 DPA:Intel Xeon 四核处理器,8G 内存(主频 2.27GHz),10000r/m 146G SAS 硬盘.由于可信硬件运算速度一般比云服务器慢,我们根据对测试平台测试获得的数据加入特定的减速因子,以模拟可信硬件的性能.IBM4764 每秒钟

能产生 2.16 个长度为 1 024 比特的 RSA 密钥对<sup>[17]</sup>.对比在测试平台获得的数据(14.92 个/s),我们设定可信硬件对大数运算的减速因子为 6.91.

测试文件大小在区间[0.5GB,1GB]内均匀分布,分块大小为 4KB.为保证 DPDP 方案中文件组跳跃表层数最少以达到最高性能,我们将 DPDP 方案中各文件大小设为一致.选择每次挑战块个数  $c=460$ ,使得以 99%的置信度检测 1%的块损坏比例.

## 5.2 实验结果

我们测试了 IDPA-MF-PDP,DPDP 和 PDP 方案的审计性能.一次审计交互时间由 4 部分组成:云服务器读取挑战数据块的 I/O 时间、云服务器计算持有性证据的计算时间、DPA 验证证据的时间和 DPA 与服务器的通信时间.由于在我们的隐式第三方审计架构中 DPA 整合在云端,DPA 和云服务器通信属于本地通信,且传输数据量为常数量级,其时间开销可忽略不计.

图 8 给出了 IDPA-MF-PDP,DPDP 和 PDP 的 I/O 时间开销.由于 IDPA-MF-PDP 与 DPDP 的 I/O 内容几乎一致,且两个方案在挑战过程中选取的文件块数量均不随文件组大小变化,因此在图 8 中将两方案合并.由图 8 可知:当文件组中的文件数目增加时,IDPA-MF-PDP/DPDP 的 I/O 时间基本维持不变.

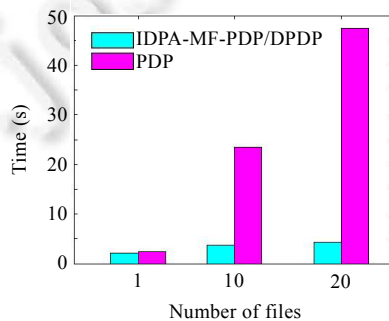


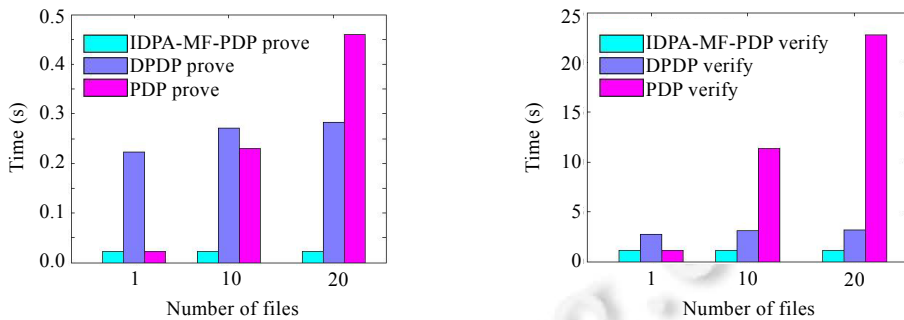
Fig.8 I/O overhead of IDPA-MF-PDP/DPDP and PDP

图 8 IDPA-MF-PDP 和 PDP 的 I/O 开销

图 9(a)给出了 IDPA-MF-PDP,DPDP 和 PDP 方案在 Prove 阶段的计算时间.IDPA-MF-PDP 中,挑战块的数目不随文件组大小而更改,因此计算时间也不会增长;DPDP 挑战块数目虽然恒定,但 Prove 阶段需根据跳跃表为每个挑战块计算验证路径,其耗时随文件块数目增长呈对数增长;而在 PDP 中,计算时间会随着挑战块数的线性增加而显著增长.

图 9(b)给出了 3 个方案 Verify 过程的计算时间.由于此过程由 DPA 验证,因此我们将所有数据乘以减速因子.由图可知:即使 Verify 阶段整体上比 Prove 阶段耗时久,但是随着文件数目的增加,IDPA-MF-PDP 的计算速度基本保持恒定,DPDP 仍因维护跳跃表使耗时呈对数增加,PDP 计算速度随文件数目增长最快(在 Verify 过程中,虽然 DPA 做了生成日志的工作,但测试结果表明,生成日志的时间基本恒定在 3ms 左右,对 Verify 时间影响甚微,因此忽略不计).结合图 8 和图 9 可以看出,I/O 开销占 IDPA-MF-PDP 审计开销的主要部分.

表 2 给出当文件组中文件数目为 100 时,IDPA-MF-PDP,DPDP 和 PDP 总审计时间对比及其组成.为了计算通信开销,我们将 SSP 与 DPA 之间的带宽设为 100Mbps.IDPA-MF-PDP 的通信内容为一个文件块的聚合值及一个认证元的聚合值,易知,PDP 的通信内容是其 100 倍.根据文献[4],DPDP 的 Prove 阶段,SSP 返回给 DPA 的持有性证据包括  $c$  个文件块 tag 值、 $c$  个验证路径以及一个文件块聚合值,其中,每个验证路径包括  $\log n + \log T_{\max}$  个哈希值.除此之外,3 个方案的通信内容均包含日志相关内容,其中,PDP 日志开销为其余两个方案的 100 倍.从表中可以看出:当文件数目增加时,IDPA-MF-PDP 的审计时间和计算时间均保持在常量水平,而 DPDP 和 PDP 的审计时间和计算时间均有明显增加.其中,DPDP 传输验证路径哈希值的通信开销多至 23.4s,导致其总开销显著高于 IDPA-MF-PDP.



(a) IDPA-MF-PDP, DPDP 和 PDP 在 Prove 阶段耗时比较 (b) IDPA-MF-PDP, DPDP 和 PDP 在 Verify 阶段耗时比较

Fig.9 Computational overhead in phase Prove, Verify of IDPA-MF-PDP, DPDP and PDP

图 9 IDPA-MF-PDP, DPDP 和 PDP 在 Prove 和 Verify 阶段耗时比较

**Table 2** Overheads of IDPA-MF-PDP, DPDP and PDP

表 2 IDPA-MF-PDP, DPDP 和 PDP 的时间开销

时间开销	IDPA-MF-PDP (s)	DPDP (s)	PDP (s)
I/O	4.708 (80.15%)	4.6	235.5
Prove	0.023 (0.39%)	0.307	2.3
Verify	1.14 (19.41%)	3.356	114.015
Communication	0.000 3 (0%)	23.388	0.032
Log Generation	0.003 (0.05%)	0.003	0.3
Total	5.874	31.654	352.147

## 6 相关工作

数据持有性和可取回性证明,近年来作为云存储安全中一个重要问题<sup>[19,20]</sup>已得到广泛关注。Ateniese 等人<sup>[2]</sup>和 Juels, Kaliski<sup>[21]</sup>首先提出了数据持有性证明和可取回性证明的定义和方案。Ateniese 等人<sup>[2]</sup>定义了 PDP 模型,并且给出了更高效的实现方案——E-PDP。在他们的方案中,通过随机抽样来减少检查的开销。基于 RSA 的同态认证标签用来提供抽样块的集合值用于验证,因此,此方案支持公开验证。同态认证元同样也是我们方案中的一个重要组成部分。

Ateniese 等人的原始 PDP 方案只支持静态文件。在他们的后续工作中,Ateniese 等人提出了一个动态的 PDP 方案<sup>[3]</sup>。其基本思想是:在系统建立阶段,提出所有后续可能的挑战,并且把预计算的持有性证据当作元数据存储。这样,验证端发起的挑战次数是限定的。此外,每一次对于数据的更新都需要重新计算所有的挑战,这在大文件的存储中会出现问题。Wang 等人<sup>[5]</sup>研究了分布系统中的动态数据存储问题,他们的方案不仅可以判定数据的正确性,还可以定位数据出错的位置。在他的另一个研究中<sup>[6]</sup>,构造了一个整合了 TTP 公开审计和动态数据的方案。全动态数据更新,特别是数据块的插入,通过用默克尔哈希树存储认证标签来实现。Erway 等人<sup>[4]</sup>对 Ateniese 提出的 PDP 模型<sup>[2]</sup>进行扩展,分别利用跳跃表(skip list)和 RSA 树构建了基于秩次的认证字典,以此提出了支持全动态更新的 PDP 方案。但是这个方案的更新操作的效率同样较低。Li 等人<sup>[7]</sup>提出了 SN-BN 表结构来支持数据块的插入操作,他们采用这个表将数据块的逻辑下标和实际位置对应起来。在 PDP 过程中,通过序列号(SM)标识挑战的数据块,通过块号(BN)取得这些数据块。当插入一个新数据块时,按照当前数据块的数目,顺序设定这个数据块的 SN,以此来完成数据的更新。

以上支持动态数据的 PDP 方案<sup>[3-7]</sup>都针对一般的单文件更新,且方案复杂更新和检查开销较大,尤其是在简单扩展到处理多文件时( $O(n)$ )。其中,只有文献<sup>[4]</sup>考虑了多文件检查问题,他们将多个可更新文件的组成和目录结构相同的树结构。由于针对最一般的文件系统更新模式,使更新和检查复杂度为  $O(d \cdot \log_2 n)$ ,其中, $d$  为树的深度, $n$  为跳跃表中最大叶节点个数。本文针对云存储中数据更新的特殊模式提出的 MF-PDP 方案,在保持方案

简单性的同时,显著降低了多文件检查的开销( $O(1)$ ).

在 Wang 等人工作<sup>[6]</sup>的基础上,一些学者考虑了基于第三方审计的 PDP 方案的在实际云存储环境中部署所面临的问题和解决方案,主要集中在多 SSP 和可扩展性两个方面.朱岩等人<sup>[8]</sup>考虑了多个 SSP 协同提供云存储服务的情形,基于多证明者零知识证明系统构造了协同 PDP 方案.杨侃等人<sup>[9]</sup>在支持动态审计的同时,通过多用户和多 SSP 的批量审计进一步提高效率.该思想和本文的 MF-PDP 类似,但他们并没有考虑多文件批量审计问题.王化群等人<sup>[10]</sup>构造了多 SSP 环境下的基于身份的分布式 PDP 方案,提高系统的可扩展性.

这些方案都假设一个 TTP 的存在,而没有关注怎样以可操作的方式实现和部署 TTP,也没有探讨在用户离线的环境下如何将 TTP 的审计结果返回给用户的问题.与这些工作不同,本文并未给出某个具体基于第三方审计的 PDP 方案,而是提出一个一般的隐式 TTP 架构,用部署在云端的可信硬件实现隐式 DPA,通过 DPA 和 SSP 的交互和显篡改审计日志支持用户离线审计.需要指出:该架构具有和多种已有 PDP 方案集成的能力,本文的 IDPA-MF-PDP 即显示了该架构和一种特定的 PDP 方案——MF-PDP 的集成.

## 7 总 结

本文分析了现有 PDP 方案在真实云存储环境中应用存在的问题,并给出了一个解决方案:基于隐式第三方审计框架的多文件数据持有性证明系统 IDPA-MF-PDP.理论分析表明:IDPA-MF-PDP 具有和单文件 PDP 方案等同的安全性,审计日志提供了可信的审计结果历史记录,IDPA-MF-PDP 是安全的.复杂度分析和实验结果表明:持有性审计的计算和通信开销由线性减少到接近常数水平,其性能主要受限于硬盘 I/O 能力,而非密码运算,IDPA-MF-PDP 是高效的.本文工作为最终实现可在真实云存储环境中部署的数据持有性证明提供了可能.

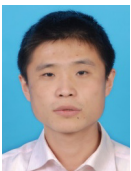
## References:

- [1] Feng DG, Zhang M, Zhang Y, Xu Z. Study on cloud computing security. Ruan Jian Xue Bao/Journal of Software, 2011,22(1): 71–83 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3958.htm> [doi: 10.3724/S.P.J.1001.2011.03958]
- [2] Ateniese G, Burns R, Curtmola R, Herring J. Provable data possession at untrusted stores. In: Ning P, De Capitani di Vimercati S, Syverson PF, eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2007. 598–609. [doi: 10.1145/1315245.1315318]
- [3] Ateniese G, Pietro RD, Mancini LV, Tsudik G. Scalable and efficient provable data possession. In: Proc. of the 4th Int'l Conf. on Security and Privacy in Communication Networks. New York: ACM Press, 2008. [doi: 10.1145/1460877.1460889]
- [4] Erway C, Kupcu A, Papamanthou C, Tamassia R. Dynamic provable data possession. In: Al-Shaer E, Jha S, Keromytis AD, eds. Proc. of the ACM Conf. on Computer and Communications Security. New York: ACM Press, 2009. 213–222. [doi: 10.1145/1653662.1653688]
- [5] Wang C, Wang Q, Ren K, Lou W. Ensuring data storage security in cloud computing. In: Proc. of the 2009 IEEE 17th Int'l Workshop on Quality of Service (IWQoS 2009). Piscataway: IEEE, 2009. 1–9. [doi: 10.1109/IWQoS.2009.5201385]
- [6] Wang Q, Wang C, Li J, Ren K, Lou W. Enabling public verifiability and data dynamics for storage security in cloud computing. In: Backes M, Ning P, eds. Proc. of the 14th European Symp. on Research in Computer Security (ESORICS 2009). Saint-Malo, Berlin: Springer-Verlag, 2009. 355–370. [doi: 10.1007/978-3-642-04444-1\_22]
- [7] Li C, Chen Y, Tan P, Yang G. An efficient provable data possession scheme with data dynamics. In: Proc. of the 2012 Int'l Conf. on Computer Science and Service System. Piscataway: IEEE, 2012. 706–710. [doi: 10.1109/CSSS.2012.182]
- [8] Zhu Y, Hu H, Ahn H, Yu M. Cooperative provable data possession for integrity verification in multi-cloud storage. IEEE Trans. on Parallel and Distributed Systems, 2012,23(12):2231–2244. [doi: 10.1109/TPDS.2012.66]
- [9] Yang K, Jia XH. An efficient and secure dynamic auditing protocol for data storage in cloud computing. IEEE Trans. on Parallel and Distributed Systems, 2013,24(9):1717–1726. [doi: 10.1109/TPDS.2012.278]
- [10] Wang H. Identity-Based distributed provable data possession in multi-cloud storage. Trans. on Services Computing, 2014,PP(99):1. [doi: 10.1109/TSC.2014.1]

- [11] Internet and cloud services—Statistics on the use by individuals. [http://ec.europa.eu/eurostat/statistics-explained/index.php/Internet\\_and\\_cloud\\_services\\_-\\_statistics\\_on\\_the\\_use\\_by\\_individuals](http://ec.europa.eu/eurostat/statistics-explained/index.php/Internet_and_cloud_services_-_statistics_on_the_use_by_individuals)
- [12] Librsync. <https://github.com/dropbox/librsync>
- [13] Tarsnap. <http://www.tarsnap.com/>
- [14] Vrable M, Savage S, Voelker GM. Cumulus: Filesystem backup to the cloud. *ACM Trans. on Storage (TOS)*, 2009,5(4):1–28. [doi: 10.1145/1629080.1629084]
- [15] Vrable M, Savage S, Voelker GM. BlueSky: A cloud-backed file system for the enterprise. In: *Proc. of the 10th USENIX Conf. on File and Storage Technologies (FAST 2012)*. 2012.
- [16] Xiao D, Yang Y, Yao W, Wu C, Liu J, Yang Y. Multiple-File remote data checking for cloud storage. *Computers & Security*, 2012, 31(2):192–205. [doi: 10.1016/j.cose.2011.12.005]
- [17] IBM 4764 PCI-X cryptographic coprocessor. <http://www-03.ibm.com/security/cryptocards/pcixcc/overperformance.shtml>
- [18] Openssl Crypto Library. <http://www.openssl.org/>
- [19] Chen LX, Xu L. Research on provable data possession and recovery technology in cloud storage. *Journal of Computer Research and Development*, 2012,49(Suppl.):19–25 (in Chinese with English abstract).
- [20] Fu YX, Luo SM, Shu JW. Survey of secure cloud storage system and key technologies. *Journal of Computer Research and Development*, 2013,50(1):136–145 (in Chinese with English abstract).
- [21] Juels A, Kaliski B. PORs: Proofs of retrievability for large files. In: Ning P, De Capitani di Vimercati S, Syverson PF, eds. *Proc. of the ACM Conf. on Computer and Communications Security*. New York: ACM Press, 2007. 584–597. [doi: 10.1145/1315245.1315317]

#### 附中文参考文献:

- [1] 冯登国,张敏,张妍,徐震.云计算安全研究.软件学报,2011,22(1):71–83. <http://www.jos.org.cn/1000-9825/3958.htm> [doi: 10.3724/S.P.J.1001.2011.03958]
- [19] 陈兰香,许力.云存储服务中可证明数据持有及恢复技术研究.计算机研究与发展,2012,49(Suppl.):19–25.
- [20] 傅颖勋,罗圣美,舒继武.安全云存储系统与关键技术综述.计算机研究与发展,2013,50(1):136–145.



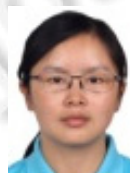
肖达(1981—),男,黑龙江哈尔滨人,博士,讲师,主要研究领域为云存储安全,存储系统.



孙斌(1967—),女,博士,副教授,主要研究领域为计算机网络,信任管理.



杨绿茵(1991—),女,硕士生,主要研究领域为云存储安全.



郑世慧(1979—),女,博士,讲师,主要研究领域为密码学,密码方案的分析与设计.