(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 106776904 B
(45)授权公告日 2019.05.28

(21)申请号 201611081331.8
(22)申请日 2016.11.30
(65)同一申请的已公布的文献号
申请公布号 CN 106776904 A
(43)申请公布日 2017.05.31
(73)专利权人 中南大学
地址 410083 湖南省长沙市岳麓区麓山南路932号
(72)发明人 罗跃逸 朱小玉 袁修贵
(74)专利代理机构 长沙市融智专利事务所(普通合伙) 43114
代理人 龚燕妮
(51)Int.Cl.
G06F 16/14(2019.01)
G06F 21/60(2013.01)
H04L 29/06(2006.01)
(56)对比文件
CN 103607405 A,2014.02.26,
CN 104102714 A,2014.10.15,
CN 102938767 A,2013.02.20,
US 2013144879 A1,2013.06.06,
审查员 赖女女

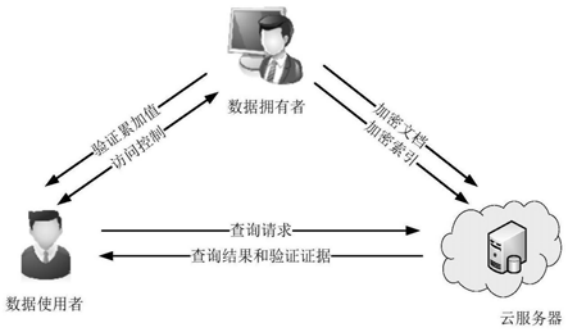
权利要求书3页 说明书8页 附图3页

(54)发明名称

一种不可信云计算环境中支持动态验证的模糊查询加密方法

(57)摘要

本发明公开了一种不可信云计算环境中支持动态验证的模糊查询加密方法,实现了云计算环境中用户查询隐私的保护,提高了云计算环境中加密数据的查询体验。该方法通过编辑距离来定义关键词之间的相似度,利用通配符构造模糊关键词集,基于倒排索引构造安全索引,使得用户能够进行模糊关键词的查询。利用可验证技术,构造可验证集合验证服务器是否篡改查询结果,验证云服务器返回的查询结果是否正确和完整。针对云计算环境中用户需要大量更新数据的问题,实现了数据的高效更新。该方法支持用户同时进行模糊查询、动态更新加密数据、验证查询结果的正确性和完整性,在保护数据隐私的前提下,提升了用户的查询体验。



CN 106776904 B

1.一种不可信云计算环境中支持动态验证的模糊查询加密方法,其特征在于,包括以下几个步骤:

步骤1:数据拥有者利用密钥生成算法,获得私钥集合k和公钥集合pk=(N,g),然后使用对称加密算法和私钥集合k将明文文档集合D加密,生成加密文档集合$\tilde{D}$;

步骤2:数据拥有者依据明文文档中的每个关键词和编辑距离构建关键词模糊集合S_{w_i}和对应的查询陷门T_i,利用查询陷门构建模糊关键词的安全查询索引Index,并对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值;同时,将加密文档集合$\tilde{D}$、安全查询索引Index和公钥集合pk上传至云服务器;

步骤3:数据使用者发出查询请求关键词w_a,并依据查询请求关键词生成查询请求关键词模糊集合S_{w_a},数据拥有者接收到查询请求关键词模糊集合后,计算该查询请求关键词的查询陷门,并将查询陷门返回给数据使用者;

步骤4:数据使用者将从数据拥有者发送来的查询陷门T_a发送至云服务器,从云服务器中存储的安全查询索引集合中寻找与查询陷门T_a匹配的安全查询索引,并从匹配的安全查询索引中提取对应的加密索引$\tilde{I}_a$,再利用加密索引获取对应的加密文档;

步骤5:对步骤4获得的加密文档和对应的安全查询索引计算验证累加值,得到文档验证证据$pf(\tilde{D})$和索引验证证据$pf(\tilde{I})$,并将获得的加密文档查询结果$\tilde{D}(w_a)$和验证证据发送至数据使用者;

步骤6:对步骤5获得的查询结果和验证证据进行验证,若验证通过,则允许数据使用者下载步骤4获得的加密文档,并从云服务器中获取私钥集合k对加密文档进行解密。

2.根据权利要求1所述的方法,其特征在于,所述安全查询索引的构建步骤如下:

步骤1):采用Trapdoor算法对明文文档中每个关键词分别构造一个查询陷门T_i,$T_i=\{\tilde{F}_i,[f_{k_1}(w_i)]_{1...n}\}$;

其中,$\tilde{F}_i$表示关键词模糊集合S_{w_i}的加密集合,$\tilde{F}_i=\{[f_{k_0}(w_i')]_{1...128}\}_{w_i'\in S_{w_i}}$,$f_{k_0}(w_i')$表示利用伪随机函数$f_k$和密钥$k_0$加密关键词模糊集合$S_{w_i}$中的关键词$w_i'$,$w_i'\in S_{w_i}$,$[f_{k_0}(w_i')]_{1...128}$表示取$f_{k_0}(w_i')$前128位;$S_{w_i}$是由属于文档中的关键词$w_i$采用FuzzySet算法生成的集合;

$[f_{k_1}(w_i)]_{1...n}$表示查询辅助信息,$f_{k_1}(w_i)$表示利用伪随机函数f_k和密钥k_1加密关键词w_i,$[f_{k_1}(w_i)]_{1...n}$表示取$f_{k_1}(w_i)$前n位;

步骤2):将查询辅助信息$[f_{k_1}(w_i)]_{1...n}$和第i行索引I_i进行异或运算,获得加密后的第i行索引$\tilde{I}_i=I_i\oplus[f_{k_1}(w_i)]_{1...n}$,所有的关键词$w_i\in W$,获得加密索引为$\tilde{I}=\{\tilde{I}_i\}_{w_i\in W}$;

I_i代表I的第i行,I为m×n的二元矩阵,I=$\{I_{i,j}\}$,关键词w_i包含在文档d_j中,则$I_{i,j}=1$;否则$I_{i,j}=0$;

步骤3):使用随机排列函数γ作用于{1,...,m},m为关键词数量,获得安全查询索引集合$Index=\{(\tilde{F}_{\gamma(i)},\tilde{I}_{\gamma(i)})_{1\le i\le m}\}$。

3.根据权利要求2所述的方法，其特征在于，所述对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值的具体过程如下：

文档验证累加值$acc(\tilde{D})$：$acc(\tilde{D})=g^{\prod_{j=1}^{n}P(H(j,H(\tilde{d}_j)))} \bmod N$，$\tilde{D}=\{(j,\tilde{d}_j)\,|\,1\le j\le n\}$；

索引验证累加值$acc(\tilde{I})$：$acc(\tilde{I})=g^{\prod_{i=1}^{m}\prod_{j=1}^{n}P(H(j,[\tilde{I}_i]_j))} \bmod N$，$\tilde{I}=\{\tilde{I}_i\}_{w_i\in W}$，$[\tilde{I}_i]_j$代表加密后的第i行索引$\tilde{I}_i$的第j位；

其中，P(•)是一个质数生成函数，H:{0,1}*→{0,1}σ是一个无碰撞哈希函数，m为关键词数量，n为明文文档数量，(N,g)为利用密钥生成算法生成的公钥集合。

4.根据权利要求3所述的方法，其特征在于，所述对步骤5获得的查询结果和验证证据进行验证的具体过程如下：

步骤A：对于查询结果中包含的所有文档$(j,\tilde{d}_j)\in\tilde{D}(w_a)$，利用无碰撞哈希函数H生成文档哈希值，再通过质数生成函数P(•)生成一个质数x_j，$x_j=P(H(j,H(\tilde{d}_j)))$；

步骤B：从数据拥有者处获得验证累加值$acc=(acc(\tilde{D}),acc(\tilde{I}))$，从云服务器处获得验证证据$pf=(pf(\tilde{D}),pf(\tilde{I}))$，判断$acc(\tilde{D})$与$pf(\tilde{D})^{\prod_{e_j=1}x_j} \bmod N$是否相等，若相等，则查询结果正确且完整，验证通过，若不相等，则验证失败，退出整个查询加密过程；

步骤C：根据查询结果$\tilde{D}(w_a)$重建查询关键词w_a对应的索引行I_a，通过I_a和查询辅助信息$[f_{k_1}(w_a)]_{1\ldots n}$重建出加密后的索引行$\tilde{I}_a$，$\tilde{I}_a=I_a\oplus[f_{k_1}(w_a)]_{1\ldots n}$；

步骤D：对于所有的1≤j≤n，利用H和P(•)生成一个质数z_j，$z_j=P(H(j,[\tilde{I}_a]_j))$；

步骤E：判断$acc(\tilde{I})$与$pf(\tilde{I})^{\prod_{j=1}^{n}z_j} \bmod N$是否相等，若相等，则索引未被篡改，验证通过，若不相等，则退出整个查询加密过程。

5.根据权利要求4所述的方法，其特征在于，在动态云存储环境中，数据拥有者按照以下步骤进行任意的增加、删除或修改文档，实现动态数据更新：

1)增加一个文档d_{n+1}

首先对矩阵索引新增一列，如果文档d_{n+1}中包含文档关键词w_i，令$I_{i,n+1}=1$，否则令$I_{i,n+1}=0$；

其次，数据拥有者首先使用加密算法将文档d_{n+1}加密成$\tilde{d}_{n+1}$；

对于1≤i≤m，计算$b_i=[f_{k_1}(w_i)]_{n+1}\oplus I_{i,n+1}$，再计算出$b_{n+1}=(b_{\gamma(1)},\ldots,b_{\gamma(m)})$，其中，γ(1)...γ(m)为随机排列函数γ作用于{1,...,m}得到，数据拥有者将$(\tilde{d}_{n+1},b_{n+1})$发送到云服务器；

对于1≤i≤m，云服务器将安全加密索引$\tilde{I}_{\gamma(i)}$更新为$\tilde{I}'_{\gamma(i)}=\tilde{I}_{\gamma(i)}\,\|\,b_{\gamma(i)}$，其中“||”代表连接词；

最后，计算出$acc(\tilde{D})'=acc(\tilde{D})^{P(H(n+1,H(\tilde{d}_{n+1})))} \bmod N$，$acc(\tilde{I})'=acc(\tilde{I})^{\prod_{i=1}^{m}P(H(n+1,b_i)))} \bmod N$，将$acc(\tilde{D})$更新为$acc(\tilde{D})'$，$acc(\tilde{I})$更新为$acc(\tilde{I})'$；云服务器更新加密文档集合、安全查询索引和

验证累加值；

2)删除文档d_j

云服务器收到数据拥有者发出的文档d_j删除请求后，计算$x_j = P(H(j,H(\tilde{d}_j)))$，计算累加值$acc(\tilde{D})' = acc(\tilde{D})^{(x_j)^{-1}} \bmod N$；云服务器删除密文$\tilde{d}_j$，将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$；

3)修改：数据拥有者将文档d_j修改为文档d'_j，且d_j和d'_j拥有相同的关键词；

云服务器收到数据拥有者发出的修改请求后，计算$x_j = P(H(j,H(\tilde{d}_j)))$和$x'_j = P(H(j,H(\tilde{d}'_j)))$，其中$\tilde{d}'_j$是$d'_j$的密文；计算$acc(\tilde{D})' = acc(\tilde{D})^{(x_j)^{-1}x'_j}$；最后将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$。

一种不可信云计算环境中支持动态验证的模糊查询加密方法

技术领域

[0001] 本发明涉及计算机科学与技术领域，特别涉及一种不可信云计算环境中支持动态验证的模糊查询加密方法。

背景技术

[0002] 随着云计算的快速发展，可查询加密方案逐渐获得人们的关注和认可。大量的用户通过云盘上传个人文件，然而人们在享用云计算服务带来便利的同时，也面临着敏感信息泄露的风险。在云计算环境中，用户失去了对数据的直接控制权。为了保护用户的隐私信息不被云破解，很多用户会选择将个人的数据加密之后上传。用户需要在加密的数据集上进行查询，而明文的信息查询方法无法适用于加密数据，因此可查询加密方法成为了研究的热点问题。

[0003] 可查询加密方法可按照不同的功能和设定条件进行划分。模糊查询指的是在用户输入的查询请求存在拼写错误时云服务器仍可以返回正确的查询结果。另外，不可信云服务器可能会由于病毒或意外而出现故障，甚至会为了节省存储空间和计算资源而恶意的删除或修改用户的加密数据，或者直接篡改用户的查询结果。而支持验证的可查询加密方法可以验证文档和查询结果的完整性，保护用户的查询结果不被篡改。在实际生活中，用户将大量的数据外包到云存储后，用户可能需要动态频繁地更新数据，如插入、删除、修改数据，此时数据更新成为了一个重要的问题。现有的可查询加密方法都只是单独针对模糊查询、可验证查询或动态更新等问题，而没有方法能够同时支持加密数据的模糊查询、验证结果和动态更新。因此需要针对不可信云计算环境，提供一种支持动态验证的模糊查询加密方法。

发明内容

[0004] 本发明提供了一种不可信云计算环境中支持动态验证的模糊查询加密方法，该方法支持用户对加密数据进行模糊查询、验证结果以及动态更新，可以保护用户的数据隐私，并方便用户进行查询、验证和更新。

[0005] 一种不可信云计算环境中支持动态验证的模糊查询加密方法，包括以下几个步骤：

[0006] 步骤1：数据拥有者利用密钥生成算法，获得私钥集合k和公钥集合pk＝(N,g)，然后使用对称加密算法和私钥集合k将明文文档集合D加密，生成加密文档集合$\tilde{D}$；

[0007] 步骤2：数据拥有者依据明文文档中的每个关键词和编辑距离构建关键词模糊集合S_{w_i}和对应的查询陷门T_i，利用查询陷门构建模糊关键词的安全查询索引Index，并对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值；同时，将加密文档集合$\tilde{D}$、安全查询索引Index和公钥集合pk上传至云服务器；

[0008] 通过模糊关键词集生成算法，输入关键词w和编辑距离ed，输出模糊关键词集S_w。

[0009] 步骤3：数据使用者发出查询请求关键词w_a，并依据查询请求关键词生成查询请求

关键词模糊集合S_{w_a}，数据拥有者接收到查询请求关键词模糊集合后，计算该查询请求关键词的查询陷门，并将查询陷门返回给数据使用者；

[0010] 对于查询请求w_a，首先数据使用者通过FuzzySet算法计算出w_a对应的模糊关键词集S_{w_a}，并将S_{w_a}发送到数据拥有者。接收S_{w_a}后，数据拥有者通过Trapdoor算法计算查询陷门$T_a=\{\tilde{F}_a,[f_{k_1}(w_a)]_{1...n}\}$，并将$T_a$返回给数据使用者。

[0011] 云服务器从数据使用者处接收到查询陷门T_a后，服务器将$T_a=\{\tilde{F}_a,[f_{k_1}(w_a)]_{1...n}\}$与查询索引$Index=\{(\tilde{F}_{\gamma(i)},\tilde{I}_{\gamma(i)})_{1\le i\le m}\}$进行匹配，查找到$(\tilde{F}_a,\tilde{I}_a)\in Index$，获得索引$\tilde{I}_a$。然后服务器再利用查询辅助信息$[f_{k_1}(w_a)]_{1...n}$计算$I_a=\tilde{I}_a\oplus[f_{k_1}(w_a)]_{1...n}$，解密得到$I_a$。令$I_a=(e_1,...,e_n)$，最终服务器计算出查询结果$\tilde{D}(w_a)=\{\tilde{d}_j\mid e_j=1\}$。

[0012] 步骤4：数据使用者将从数据拥有者发送来的查询陷门T_a发送至云服务器，从云服务器中存储的安全查询索引集合中寻找与查询陷门T_a匹配的安全查询索引，并从匹配的安全查询索引中提取对应的加密索引$\tilde{I}_a$，再利用加密索引获取对应的加密文档；

[0013] 步骤5：对步骤4获得的加密文档和对应的安全查询索引计算验证累加值，得到文档验证证据$pf(\tilde{D})$和索引验证证据$pf(\tilde{I})$，并将获得的加密文档的查询结果和验证证据发送至数据使用者；

[0014] 对于所有不在查询结果中的文档即$e_j=0$，先利用无碰撞哈希函数$H:\{0,1\}^*\rightarrow\{0,1\}^o$生成文档哈希值，再通过质数生成函数$P(\bullet)$生成一个质数。再利用密钥生成算法生成的公钥集合(N,g)计算文档的验证证据：$pf(\tilde{D})=g^{\prod_{e_j=0}P(H(j,H(\tilde{d}_j)))}\bmod N$；

[0015] 对于不包含查询关键词w_a的索引即$i\neq a$。先利用无碰撞哈希函数$H:\{0,1\}^*\rightarrow\{0,1\}^o$生成索引哈希值，再通过质数生成函数$P(\bullet)$生成一个质数。再利用密钥生成算法生成的公钥集合(N,g)计算索引的验证证据：$pf(\tilde{I})=g^{\prod_{i\neq a}\{\prod_{j=1}^{n}P(H(j,[\tilde{I}_i]_j))\}}\bmod N$；

[0016] 步骤6：对步骤5获得的查询结果和验证证据进行验证，若验证通过，则允许数据使用者下载步骤4获得的加密文档，并从云服务器中获取私钥集合k对加密文档进行解密。

[0017] 进一步地，所述安全查询索引的构建步骤如下：

[0018] 步骤1)：采用Trapdoor算法对明文文档中每个关键词分别构造一个查询陷门T_i，$T_i=\{\tilde{F}_i,[f_{k_1}(w_i)]_{1...n}\}$；

[0019] 其中，$\tilde{F}_i$表示关键词模糊集合S_{w_i}的加密集合，$\tilde{F}_i=\{[f_{k_0}(w_i')]_{1...128}\}_{w_i'\in S_{w_i}}$，$f_{k_0}(w_i')$表示利用伪随机函数$f_k$和密钥$k_0$加密关键词模糊集合$S_{w_i}$中的关键词$w_i'$，$w_i'\in S_{w_i}$，$[f_{k_0}(w_i')]_{1...128}$表示取$f_{k_0}(w_i')$前128位；$S_{w_i}$是由属于文档中的关键词$w_i$采用FuzzySet算法生成的集合；

[0020] $[f_{k_1}(w_i)]_{1...n}$表示查询辅助信息，$f_{k_1}(w_i)$表示利用伪随机函数f_k和密钥k_1加密关键词w_i，$[f_{k_1}(w_i)]_{1...n}$表示取$f_{k_1}(w_i)$前n位；

[0021] 步骤2)：将查询辅助信息$[f_{k_1}(w_i)]_{1...n}$和第i行索引I_i进行异或运算，获得加密后的第i行索引$\tilde{I}_i = I_i \oplus [f_{k_1}(w_i)]_{1...n}$，所有的关键词$w_i \in W$，获得加密索引为$\tilde{I} = \{\tilde{I}_i\}_{w_i \in W}$；

[0022] I_i代表I的第i行，I为m×n的二元矩阵，$I = \{I_{i,j}\}$，关键词w_i包含在文档d_j中，则$I_{i,j}=1$；否则$I_{i,j}=0$；

[0023] 步骤3)：使用随机排列函数γ作用于{1,...,m}，m为关键词数量，获得安全查询索引集合$Index = \{(\tilde{F}_{\gamma(i)}, \tilde{I}_{\gamma(i)})_{1 \le i \le m}\}$。

[0024] 进一步地，所述对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值的具体过程如下：

[0025] 文档验证累加值$acc(\tilde{D})$：$acc(\tilde{D}) = g^{\prod_{j=1}^{n} P(H(j,H(\tilde{d}_j)))} \bmod N$，$\tilde{D} = \{(j,\tilde{d}_j) \mid 1 \le j \le n\}$；

[0026] 索引验证累加值$acc(\tilde{I})$：$acc(\tilde{I}) = g^{\prod_{i=1}^{m}\prod_{j=1}^{n} P(H(j,i,[\tilde{I}_i]_j))} \bmod N$，$\tilde{I} = \{\tilde{I}_i\}_{w_i \in W}$，$[\tilde{I}_i]_j$代表加密后的第i行索引$\tilde{I}_i$的第j位；

[0027] 其中，P(•)是一个质数生成函数，$H:\{0,1\}^* \rightarrow \{0,1\}^q$是一个无碰撞哈希函数，m为关键词数量，n为明文文档数量，(N,g)为利用密钥生成算法生成的公钥集合。

[0028] 所述对步骤5获得的查询结果和验证证据进行验证的具体过程如下：

[0029] 步骤A：对于查询结果中包含的所有文档$(j,\tilde{d}_j) \in \tilde{D}(w_a)$，利用无碰撞哈希函数H生成文档哈希值，再通过质数生成函数P(•)生成一个质数x_j，$x_j = P(H(j,H(\tilde{d}_j)))$；

[0030] 步骤B：从数据拥有者处获得验证累加值$acc = (acc(\tilde{D}), acc(\tilde{I}))$，从云服务器处获得验证证据$pf = (pf(\tilde{D}), pf(\tilde{I}))$，判断$acc(\tilde{D})$与$pf(\tilde{D})^{\prod_{e_j=1} x_j} \bmod N$是否相等，若相等，则查询结果正确且完整，验证通过，若不相等，则验证失败，退出整个查询加密过程；

[0031] 步骤C：根据查询结果$\tilde{D}(w_a)$重建查询关键词w_a对应的索引行I_a，通过I_a和查询辅助信息$[f_{k_1}(w_a)]_{1...n}$重建出加密后的索引行$\tilde{I}_a$，$\tilde{I}_a = I_a \oplus [f_{k_1}(w_a)]_{1...n}$；

[0032] 步骤D：对于所有的1≤j≤n，利用H和P(•)生成一个质数z_j，$z_j = P(H(j,[\tilde{I}_a]_j))$；

[0033] 步骤E：判断$acc(\tilde{I})$与$pf(\tilde{I})^{\prod_{j=1}^{n} z_j} \bmod N$是否相等，若相等，则索引未被篡改，验证通过，若不相等，则退出整个查询加密过程。

[0034] 进一步地，在动态云存储环境中，数据拥有者按照以下步骤进行任意的增加、删除或修改文档，实现动态数据更新：

[0035] 1)增加一个文档d_{n+1}

[0036] 首先对矩阵索引新增一列，如果文档d_{n+1}中包含文档关键词w_i，令$I_{i,n+1}=1$，否则令$I_{i,n+1}=0$；

[0037] 其次，数据拥有者首先使用加密算法将文档d_{n+1}加密成$\tilde{d}_{n+1}$；

[0038] 对于1≤i≤m，计算$b_i = [f_{k_1}(w_i)]_{n+1} \oplus I_{i,n+1}$，再计算出$b_{n+1} = (b_{\gamma(1)},...,b_{\gamma(m)})$，其中，γ(1)...γ(m)为随机排列函数γ作用于{1,...,m}得到，数据拥有者将$(\tilde{d}_{n+1}, b_{n+1})$发送到云服

务器；

[0039] 对于1≤i≤m，云服务器将安全加密索引$\tilde{I}_{\gamma(i)}$更新为$\tilde{I}'_{\gamma(i)}=\tilde{I}_{\gamma(i)}\|b_{\gamma(i)}$，其中“||”代表连接词；

[0040] 最后，计算出$acc(\tilde{D})'=acc(\tilde{D})^{P(H(n+1,H(\tilde{d}_{n+1})))} \bmod N$，$acc(\tilde{I})'=acc(\tilde{I})^{\prod_{i=1}^{m}P(H(n+1,b_i)))} \bmod N$，将$acc(\tilde{D})$更新为$acc(\tilde{D})'$，$acc(\tilde{I})$更新为$acc(\tilde{I})'$；云服务器更新加密文档集合、安全查询索引和验证累加值；

[0041] 2)删除文档d_j

[0042] 云服务器收到数据拥有者发出的文档d_j删除请求后，计算$x_j=P(H(j,H(\tilde{d}_j)))$，计算累加值$acc(\tilde{D})'=acc(\tilde{D})^{(x_j)^{-1}} \bmod N$；云服务器删除密文$\tilde{d}_j$，将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$；

[0043] 3)修改：数据拥有者将文档d_j修改为文档d'_j，且d_j和d'_j拥有相同的关键词；

[0044] 云服务器收到数据拥有者发出的修改请求后，计算$x_j=P(H(j,H(\tilde{d}_j)))$和$x'_j=P(H(j,H(\tilde{d}'_j)))$，其中$\tilde{d}'_j$是$d'_j$的密文；计算$acc(\tilde{D})'=acc(\tilde{D})^{(x_j)^{-1}x'_j}$；最后将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$。

[0045] 有益效果

[0046] 本发明提供了一种不可信云计算环境中支持动态验证的模糊查询加密方法，在对称密码学的研究基础之上，提出了模糊查询的加密方法，实现了云计算环境中用户查询隐私的保护，提高了云计算环境中加密数据的查询体验。该方法通过编辑距离来定义关键词之间的相似度，利用通配符构造模糊关键词集，基于倒排索引构造安全索引，使得用户能够进行模糊关键字的查询。利用可验证技术，构造可验证集合验证服务器是否篡改查询结果，验证云服务器返回的查询结果是否正确和完整。针对云计算环境中用户需要大量更新数据的问题，实现了数据的高效更新。该方法支持用户不可信云环境中同时进行模糊查询、动态更新加密数据、验证查询结果的正确性，在保护数据隐私的前提下，提升了用户的查询体验。

附图说明

[0047] 图1为本发明所述方法的整体架构示意图；

[0048] 图2为本发明中关键词模糊集的生成时间示意图；

[0049] 图3为本发明中安全查询索引生成时间示意图；

[0050] 图4位本发明中所述方法中进行查询的时间示意图；

[0051] 图5为本发明中所述方法中进行验证的时间示意图。

具体实施方式

[0052] 下面将结合附图和实施例对本发明做进一步的说明。

[0053] 实验硬件环境为Windows 7操作系统，CPU为Intel Core i5-4590(3.30GHz)，内存为4GB，采用Java编程语言实现。数据集为近10年的IEEE INFOCOM论文集，包含超过3500篇文章，通过提取文档中包含的关键词，形成关键词集合。实验采用256位AES对称加密算法来

加密和解密文档，采用密钥长度1024位的RSA累加器生成验证证据，采用SHA-256作为哈希函数。

[0054] 一种不可信云计算环境中支持动态验证的模糊查询加密方法，整体架构如图1所示，包括以下几个步骤：

[0055] 步骤1：数据拥有者利用密钥生成算法，获得私钥集合k和公钥集合pk＝(N,g)，然后使用对称加密算法和私钥集合k将明文文档集合D加密，生成加密文档集合$\tilde{D}$；

[0056] 步骤2：数据拥有者依据明文文档中的每个关键词和编辑距离构建关键词模糊集合S_{w_i}和对应的查询陷门T_i，利用查询陷门构建模糊关键词的安全查询索引Index，并对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值；同时，将加密文档集合$\tilde{D}$、安全查询索引Index和公钥集合pk上传至云服务器；

[0057] 步骤3：数据使用者发出查询请求关键词w_a，并依据查询请求关键词生成查询请求关键词模糊集合S_{w_a}，数据拥有者接收到查询请求关键词模糊集合后，计算该请求关键词的查询陷门，并将查询陷门返回给数据使用者；

[0058] 步骤4：数据使用者将从数据拥有者发送来的查询陷门T_a发送至云服务器，从云服务器中存储的安全查询索引集合中寻找与查询陷门T_a匹配的安全查询索引，并从匹配的安全查询索引中提取对应的加密索引$\tilde{I}_a$，再利用加密索引获取对应的加密文档；

[0059] 步骤5：对步骤4获得的加密文档和对应的安全查询索引计算验证累加值，得到文档验证证据$pf(\tilde{D})$和索引验证证据$pf(\tilde{I})$，并将获得的加密文档的查询结果和验证证据发送至数据使用者；

[0060] 步骤6：对步骤5获得的查询结果和验证证据进行验证，若验证通过，则允许数据使用者下载步骤4获得的加密文档，并从云服务器中获取私钥集合k对加密文档进行解密。

[0061] 所述安全查询索引的构建步骤如下：

[0062] 步骤1)：采用Trapdoor算法对明文文档中每个关键词分别构造一个查询陷门T_i，$T_i=\{\tilde{F}_i,[f_{k_1}(w_i)]_{1...n}\}$；

[0063] 其中，$\tilde{F}_i$表示关键词模糊集合S_{w_i}的加密集合，$\tilde{F}_i=\{[f_{k_0}(w_i')]_{1...128}\}_{w_i'\in S_{w_i}}$，$f_{k_0}(w_i')$表示利用伪随机函数$f_k$和密钥$k_0$加密关键词模糊集合$S_{w_i}$中的关键词$w_i'$，$w_i'\in S_{w_i}$，$[f_{k_0}(w_i')]_{1...128}$表示取$f_{k_0}(w_i')$前128位；$S_{w_i}$是由属于文档中的关键词$w_i$采用FuzzySet算法生成的集合；

[0064] $[f_{k_1}(w_i)]_{1...n}$表示查询辅助信息，$f_{k_1}(w_i)$表示利用伪随机函数f_k和密钥k_1加密关键词w_i，$[f_{k_1}(w_i)]_{1...n}$表示取$f_{k_1}(w_i)$前n位；

[0065] 步骤2)：将查询辅助信息$[f_{k_1}(w_i)]_{1...n}$和第i行索引I_i进行异或运算，获得加密后的第i行索引$\tilde{I}_i=I_i\oplus[f_{k_1}(w_i)]_{1...n}$，所有的关键词$w_i\in W$，获得加密索引为$\tilde{I}=\{\tilde{I}_i\}_{w_i\in W}$；

[0066] I_i代表I的第i行，I为m×n的二元矩阵，I＝$\{I_{i,j}\}$，关键词w_i包含在文档d_j中，则$I_{i,j}$＝1；否则$I_{i,j}$＝0；

[0067] 步骤3)：使用随机排列函数γ作用于{1,...,m}，m为关键词数量，获得安全查询索

引集合$Index=\{(\tilde{F}_{\gamma(i)},\tilde{I}_{\gamma(i)})_{1\le i\le m}\}$。

[0068] 所述对加密文档和对应的安全查询索引采用RSA累加器计算验证累加值的具体过程如下：

[0069] 文档验证累加值$acc(\tilde{D})$：$acc(\tilde{D})=g^{\prod_{j=1}^{n}P(H(j,H(\tilde{d}_j)))} \bmod N$，$\tilde{D}=\{(j,\tilde{d}_j)\,|\,1\le j\le n\}$；

[0070] 索引验证累加值$acc(\tilde{I})$：$acc(\tilde{I})=g^{\prod_{i=1}^{m}\prod_{j=1}^{n}P(H(j,[\tilde{I}_i]_j))} \bmod N$，$\tilde{I}=\{\tilde{I}_i\}_{w_i\in W}$，$[\tilde{I}_i]_j$代表加密后的第i行索引$\tilde{I}_i$的第j位；

[0071] 其中，P(•)是一个质数生成函数，H:{0,1}*→{0,1}^o是一个无碰撞哈希函数，m为关键词数量，n为明文文档数量，(N,g)为利用密钥生成算法生成的公钥集合。

[0072] 所述对步骤5获得的查询结果和验证证据进行验证的具体过程如下：

[0073] 步骤A：对于查询结果中包含的所有文档$(j,\tilde{d}_j)\in\tilde{D}(w_a)$，利用无碰撞哈希函数H生成文档哈希值，再通过质数生成函数P(•)生成一个质数x_j，$x_j=P(H(j,H(\tilde{d}_j)))$；

[0074] 步骤B：从数据拥有者处获得验证累加值$acc=(acc(\tilde{D}),acc(\tilde{I}))$，从云服务器处获得验证证据$pf=(pf(\tilde{D}),pf(\tilde{I}))$，判断$acc(\tilde{D})$与$pf(\tilde{D})^{\prod_{e_j=1}x_j} \bmod N$是否相等，若相等，则查询结果正确且完整，验证通过，若不相等，则验证失败，退出整个查询加密过程；

[0075] 步骤C：根据查询结果$\tilde{D}(w_a)$重建查询关键词w_a对应的索引行I_a，通过I_a和查询辅助信息$[f_{k_1}(w_a)]_{1...n}$重建出加密后的索引行$\tilde{I}_a$，$\tilde{I}_a=I_a\oplus[f_{k_1}(w_a)]_{1...n}$；

[0076] 步骤D：对于所有的1≤j≤n，利用H和P(•)生成一个质数z_j，$z_j=P(H(j,[\tilde{I}_a]_j))$；

[0077] 步骤E：判断$acc(\tilde{I})$与$pf(\tilde{I})^{\prod_{j=1}^{n}z_j} \bmod N$是否相等，若相等，则索引未被篡改，验证通过，若不相等，则退出整个查询加密过程。在动态云存储环境中，数据拥有者按照以下步骤进行任意的增加、删除或修改文档，实现动态数据更新：

[0078] 1)增加一个文档d_{n+1}

[0079] 首先对矩阵索引新增一列，如果文档d_{n+1}中包含文档关键词w_i，令$I_{i,n+1}=1$，否则令$I_{i,n+1}=0$；

[0080] 其次，数据拥有者首先使用加密算法将文档d_{n+1}加密成$\tilde{d}_{n+1}$；

[0081] 对于1≤i≤m，计算$b_i=[f_{k_1}(w_i)]_{n+1}\oplus I_{i,n+1}$，再计算出$b_{n+1}=(b_{\gamma(1)},...,b_{\gamma(m)})$，其中，γ(1)...γ(m)为随机排列函数γ作用于{1,...,m}得到，数据拥有者将$(\tilde{d}_{n+1},b_{n+1})$发送到云服务器；

[0082] 对于1≤i≤m，云服务器将安全加密索引$\tilde{I}_{\gamma(i)}$更新为$\tilde{I}'_{\gamma(i)}=\tilde{I}_{\gamma(i)}\,\|\,b_{\gamma(i)}$，其中“||”代表连接词；

[0083] 最后，计算出$acc(\tilde{D})'=acc(\tilde{D})^{P(H(n+1,H(\tilde{d}_{n+1})))} \bmod N$，$acc(\tilde{I})'=acc(\tilde{I})^{\prod_{i=1}^{m}P(H(n+1,b_i)))} \bmod N$，将$acc(\tilde{D})$更新为$acc(\tilde{D})'$，$acc(\tilde{I})$更新为$acc(\tilde{I})'$；云服务器更新加密文档集合、安全查询索引

和验证累加值；

[0084] 2)删除文档d_j

[0085] 云服务器收到数据拥有者发出的文档d_j删除请求后，计算$x_j = P(H(j, H(\tilde{d}_j)))$，计算累加值$acc(\tilde{D})' = acc(\tilde{D})^{(x_j)^{-1}} \bmod N$；云服务器删除密文$\tilde{d}_j$，将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$；

[0086] 3)修改：数据拥有者将文档d_j修改为文档d'_j，且d_j和d'_j拥有相同的关键词；

[0087] 云服务器收到数据拥有者发出的修改请求后，计算$x_j = P(H(j, H(\tilde{d}_j)))$和$x'_j = P(H(j, H(\tilde{d}'_j)))$，其中$\tilde{d}'_j$是$d'_j$的密文；计算$acc(\tilde{D})' = acc(\tilde{D})^{(x_j)^{-1}x'_j}$；最后将将累加值$acc(\tilde{D})$更新为$acc(\tilde{D})'$。

[0088] 生成模糊关键词集合的时间开销如图2所示。在编辑距离变化时，时间开销与关键词数都几乎呈线性增长，而编辑距离为2比编辑距离为1的时间开销大很多，因为编辑距离越大，生成的模糊关键词集的数目将呈指数级增长。编辑距离是影响模糊查询效率的一个非常重要的因子。

[0089] 生成安全查询索引的时间开销如图3所示。设定编辑距离为1，安全查询索引的生成时间与文件数呈正相关性。随着文件数增加，关键词数量也不断增加，构造安全查询索引的时间开销随之增加。安全查询索引只需要构造一次，在文档增加、更新、删除时，只需给服务器发送请求，服务器在原有的安全查询索引上进行更新，而无需再次重新构造索引，节省了数据拥有者的时间开销。

[0090] 查询的时间开销如图4所示。查询时间随着文件数的增加呈线性增长，查询陷门由数据拥有者生成并发送给云服务器，云服务器将查询陷门与安全查询索引匹配得到查询结果。模糊关键词集由数据拥有者完成，因而在云服务器端的查询时间开销与文件数呈正相关性，与生成模糊关键词集的开销无关。

[0091] 验证的时间开销如图5所示。验证时间随着文件数的增加而增加，数据使用者首先验证文档的完整性，然后根据查询结果重建索引并验证查询结果的完整性。验证时间与文件数呈正相关性。

[0092] 隐私安全：在整个可查询加密过程中，云服务器仅获取上传的加密文档、安全查询索引、验证累加值、查询陷门、查询结果和验证证据。除此之外，云服务器无法获取文档对应的明文、查询陷门对应的查询请求等其他任何信息，从而做到隐私保护。

[0093] 定理1：本发明提出的支持可验证模糊查询的加密方案可以隐私安全。

[0094] 证明：假定$\mathcal{S}$是一个模拟器，$\mathcal{S}$首先从敌手$\mathcal{A}$处接收到$|d_1|, \ldots, |d_n|$和m。对于$1 \leqslant j \leqslant n$，$\mathcal{S}$可以模拟出密文文档$\tilde{d}_j = \mathrm{Enc}_{sk}(0^{|d_j|})$，其中sk在Enc算法中随机选取，然后生成$\tilde{D}' = \{\tilde{d}_1, \ldots, \tilde{d}_n\}$。对于$1 \leqslant i \leqslant m$，$\mathcal{S}$随机选择随机数为$\tilde{F}_i$，随机选择$\tilde{I}_i \in \{0,1\}^n$。用一个随机排列函数$\gamma$作用于$\{1, \ldots, m\}$，生成$Index' = \{(\tilde{F}_{\gamma(i)}, \tilde{I}_{\gamma(i)})_{1 \leq i \leq m}\}$，最后将$\{\tilde{D}', Index'\}$发送给$\mathcal{A}$。

[0095] $\mathcal{A}$发出查询请求w_a，$\mathcal{S}$得知查询结果$\tilde{D}(w_a) = \{\tilde{d}_j \mid e_j = 1\}$。首先计算$[f'_{k_1}(w_a)]_{1\ldots n} = \tilde{I}_a \oplus (e_1, \ldots, e_n)$，关键词$w_i$对应的陷门为$T'_a = (\tilde{F}'_a, [f'_{k_1}(w_a)]_{1\ldots n})$，$\mathcal{S}$将$T'_a$发送给$\mathcal{A}$。

[0096] $\mathcal{A}$发出增加文档的请求，$\mathcal{S}$模拟出$\tilde{d}_{n+1}=\mathrm{Enc}_{sk}(0^{|d_{n+1}|})$，对于1≤i≤m，随机选取b′$_i$∈{0,1}，使用一个随机排列函数γ作用于{1,...,m}，并计算出b′$_{n+1}$=(b′$_{\gamma(1)}$,...,b′$_{\gamma(m)}$)，S将$(ins,n+1,\tilde{d}_{n+1},b'_{n+1})$发送给$\mathcal{A}$。

[0097] $\mathcal{A}$发出删除文档的请求，$\mathcal{S}$将(del,j)发送给$\mathcal{A}$。

[0098] $\mathcal{A}$发出修改文档的请求，$\mathcal{S}$模拟出$\tilde{d}'_j=\mathrm{Enc}_{sk}(0^{|d'_j|})$，然后将$(\mathrm{mod},j,\tilde{d}'_j)$发送给$\mathcal{A}$。

[0099] 由于加密算法Enc是CPA安全的，所以$\mathcal{A}$无法区分密文$\tilde{D}$和$\tilde{D}'$。由于伪随机函数f和随机排列函数γ，导致$(T'_a,\tilde{d}'_j,\tilde{d}'_{n+1},b'_{n+1})$和$(T_a,\tilde{d}_j,\tilde{d}_{n+1},b_{n+1})$也是不可区分的。所以$\mathcal{A}$无法获知更多的信息，所以保护了隐私安全。

[0100] 可验证安全：在整个可查询加密过程中，假定恶意攻击者存在篡改用户查询结果等恶意行为，那么用户能够快速识别。

[0101] 本发明提出的支持可验证模糊查询的加密方案可以满足定义3中的可验证安全。

[0102] 证明：为了证明本文提出的方案是可验证安全的，需要证明攻击者无法伪造正确的查询结果和验证证据。

[0103] 假设$(\tilde{D}(w_a),pf(\tilde{D}),pf(\tilde{I}))$是正确的查询结果和验证证据，需要证明攻击者伪造的查询结果和验证证据$(\tilde{D}'(w_a),pf'(\tilde{D}),pf'(\tilde{I}))$无法通过数据使用者的验证算法，需要证明伪造的查询结果和证据与原有的证据不符，即$(\tilde{D}(w_a),pf(\tilde{D}),pf(\tilde{I}))\neq(\tilde{D}'(w_a),pf'(\tilde{D}),pf'(\tilde{I}))$。分为三种可能的情况：1) $\tilde{D}(w_a)=\tilde{D}'(w_a)$且$(pf(\tilde{D}),pf(\tilde{I}))\neq(pf'(\tilde{D}),pf'(\tilde{I}))$；2) $\tilde{D}(w_a)=\tilde{D}'(w_a)$且{z$_j$}≠{z′$_j$}；3) $\tilde{D}(w_a)\neq\tilde{D}'(w_a)$且{z$_j$}={z′$_j$}。

[0104] 接下来证明这三种情况下，验证过程失败的概率可以忽略不计。1)因为$(pf(\tilde{D}),pf(\tilde{I}))\neq(pf'(\tilde{D}),pf'(\tilde{I}))$，因此验证失败的概率可以忽略不计；2)因为{z$_j$}≠{z′$_j$}，在强RSA假设下，$pf(\tilde{I})$验证失败的概率可以忽略不计；3)因为$\tilde{D}(w_a)\neq\tilde{D}'(w_a)$，这说明存在两种情况$(j,\ \tilde{d}_j)\in\tilde{D}(w_a)$和$(j,\ \tilde{d}_j)\in\tilde{D}'(w_a)$可以使得$\tilde{d}_j\neq\tilde{d}'_j$。对于这种情况，由于哈希函数H的无冲撞特性，导致$H(j,H(\tilde{d}_j))\neq H(j,H(\tilde{d}'_j))$。因此，在强RSA假设下，由于$P(H(j,H(\tilde{d}_j)))\neq P(H(j,H(\tilde{d}'_j)))$，$pf(\tilde{D})$验证失败的概率可以忽略不计。

[0105] 基于以上分析，攻击者不能伪造出真实可信的查询结果和验证证据，因此本发明提出的方案是可以满足可验证安全的。

[0106] 综上所述，生成模糊关键词集需要较大的时间开销，因而扩展后的安全查询索引耗时较多，但是构造索引只需一次，而查询和文档更新操作较为频繁，本方案在搜索、更新操作上有较高的效率，可以满足实际环境的需求。

[0107] 以上所述仅是本发明技术的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本发明技术原理的前提下，还可以做出若干改进和替换，这些改进和替换也应视为本发明的保护范围。

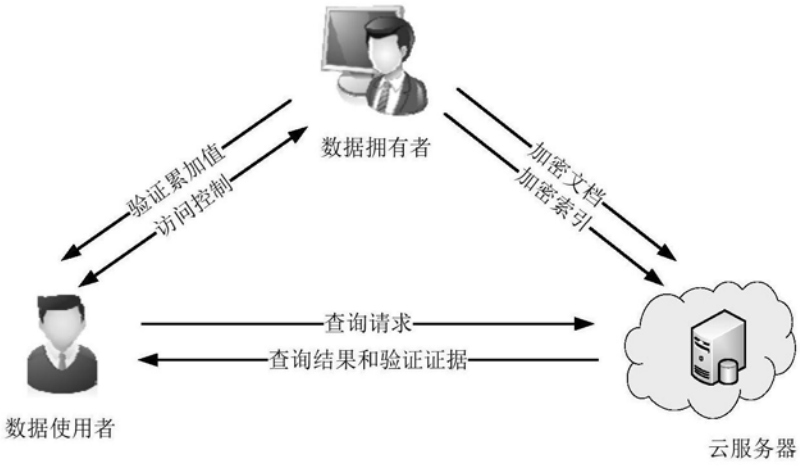

数据拥有者
验证累加值
访问控制
加密文档
加密索引
查询请求
查询结果和验证证据
数据使用者
云服务器

图1

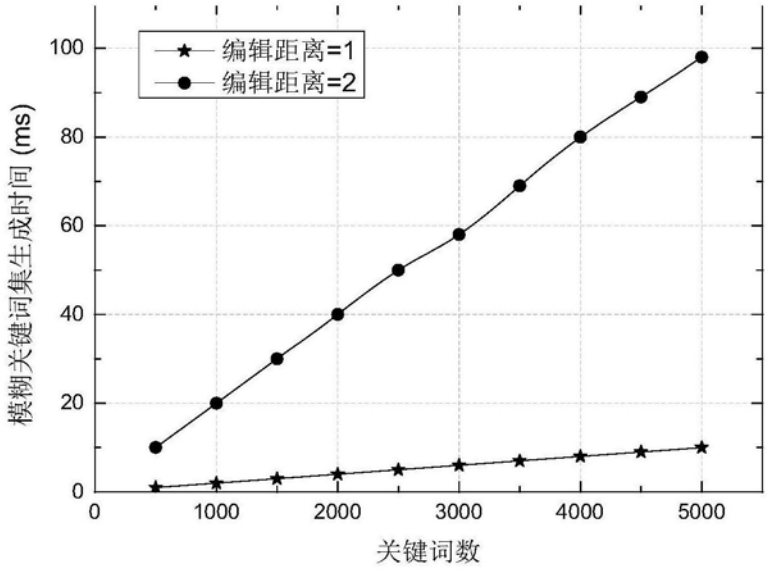

编辑距离=1
编辑距离=2
模糊关键词集生成时间 (ms)
关键词数
0
20
40
60
80
100
0
1000
2000
3000
4000
5000

图2

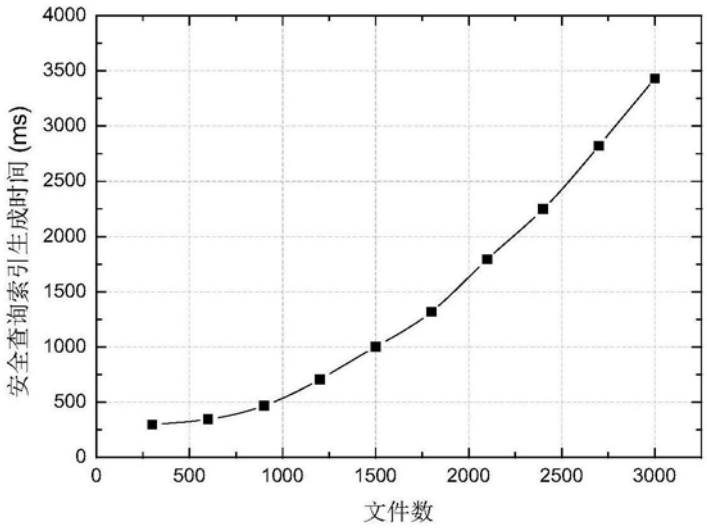

安全查询索引生成时间 (ms)
文件数
0
500
1000
1500
2000
2500
3000
3500
4000

图3

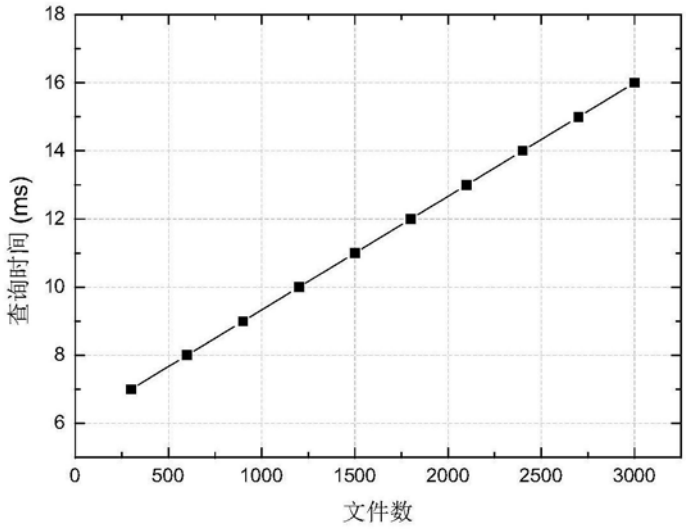

查询时间 (ms)
文件数
6
8
10
12
14
16
18
0
500
1000
1500
2000
2500
3000

图4

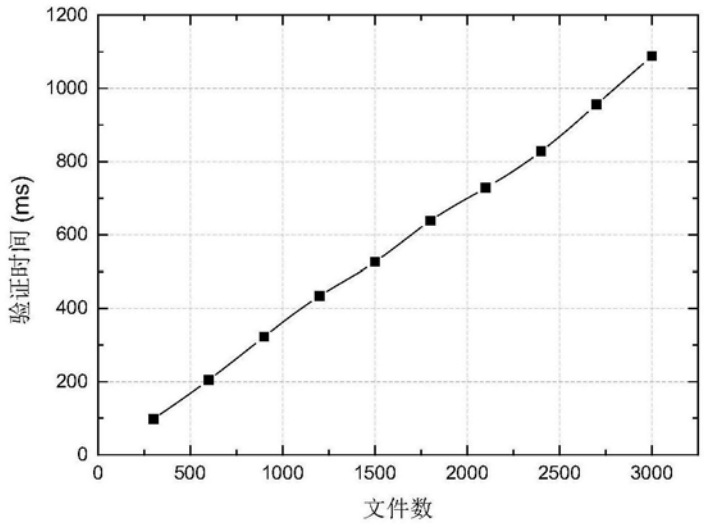

1200
1000
800
600
400
200
0
验证时间 (ms)
0
500
1000
1500
2000
2500
3000
文件数

图5