# 大容量存储设备身份认证与数据保护方案研究

韩刚，李辉

（西北工业大学 电子信息学院，陕西 西安　710072）

**摘　要**：通用串行总线大容量存储器是一种被广大用户广泛使用的存储设备。为了保护 USB 存储设备中用户数据安全，提出了一种适用于 USB 大容量存储器有效的身份认证与数据保护方案。利用属性权限控制技术，实现了对 USB 设备用户的安全认证、实现了对 USB 设备中隐私数据细粒度控制。安全性分析表明该方案具有较强的安全性，能够抵抗重放攻击、拒绝服务攻击、在线口令猜测攻击、用户伪装攻击等；实验性能分析表明该方案在用户注册、认证、文件解密等方面具有较高的效率。

**关　键　词**：大容量存储器；属性基权限控制；细粒度控制；实验设计；可靠性分析

**中图分类号**：TP309G　　**文献标志码**：A　　**文章编号**：1000-2758(2018)03-0550-08

通用串行总线（universal serial bus，USB）是连接计算机系统与外部设备的一个串口总线标准，也是一种输入输出接口的技术规范，被广泛应用于外围设备与个人电脑连接过程之中，例如键盘、鼠标、打印机以及大容量存储设备（mass storage devices，MSD）等。USB 重要的优势包括高可用性、高数据传输率以及连接便利性等。但从应用层面讲，大容量存储设备存在着重大的安全隐患，例如非授权用户能够随意非法读取并复制存储器中的明文数据；其次，大容量设备在通过 USB 与个人电脑连接过程中，攻击者能够轻易截获此过程中的数据。因此，对于 MSD-USB 用户身份的认证以及 MSD 中明文数据的保护至关重要。

如今很多学者提出了相应的保护方案。2004 年 Ku 和 Chen 等人[1]仅仅利用口令提出了一种认证方案，但随后 Yoon 和 Ryu 等人[2]发现 Ku 等人的方案对于预防平行会话攻击非常脆弱，因此 Yoon 等人提出了一个增强版的认证方案。2010 年 Yang 等人[3]提出了一种利用人的生物特征（比如指纹等）生成口令的方案，用于提升安全性。2013 年 Lee 和 Chen 等人[4]基于生物特征以及第三方认证中心实现了一种认证协议。但 2014 年，He 等人[5]发现 Lee 等人的方案不能有效抵抗口令猜测攻击、拒绝服务攻击以及重放攻击等，于是 He 等人提出了一种基于三方认证的安全协议。2015 年 Giri 等人[6]提出了一种基于口令、生物特征以及第三方认证的安全协议，该协议能够有效实现用户身份认证以及保护 MSD 中的数据安全。

上述方案将研究重点放在了用户身份认证以及保护 MSD 中数据安全上。但可以从以下几个方面拓展 MSD 的应用场景：①能否实现对 MSD-USB 用户身份的规范化管理，例如某个 MSD 设备属于某一个用户组，只有属于该用户组的用户才能读取并利用该 MSD 设备中的文件，因此需要实现从该用户组中删除或添加一位用户；②能否实现对某个 MSD 设备中文件的细粒度权限控制。这类似于 Linux 系统中 root 用户与普通用户的权限不一样，但与 Linux 系统中的权限类型又有点不同，只有拥有特定权限的用户解密该 MSD 设备中对应隐私密文数据。

因此，本文所做的工作包括以下几点：

1）结合相关密码技术，将 MSD 设备的应用场景进一步拓展，提高了 MSD 设备的实际应用价值；

2）采用新的认证协议，有效地实现了用户身份认证，能够有效防止重放攻击、中间人攻击等等；

3）采用基于属性加密技术实现了对 MSD 设备用户组的有效管理，实现了向该用户组中删除或添

**收稿日期**：2017-04-12　　**基金项目**：国家自然科学基金（61571364）、陕西省自然科学基金（2017JM6037）与西北工业大学研究生创意创新种子基金（Z2017022）资助

**作者简介**：韩刚（1990—），西北工业大学博士研究生，主要从事信息与通信安全研究。

加一位用户;

4) 实现了对某个 MSD 设备中隐私文件的细粒度权限控制。用户组中不同的用户拥有不同的权限,拥有哪种权限才能解密具有对应权限的隐私密文文件。

# 1　基础知识

## 1.1　模糊提取器

模糊提取器是由安全概略和强提取器来定义的,先给出安全概略的定义,安全概略是由一对随机过程$\langle SS,REC\rangle$定义的,$SS$是概略过程,$REC$是恢复过程[7]。

**定义 1**(随机提取器)　一个$(k,\varepsilon)$的提取器是一个函数$EXT:\{0,1\}^n\times\{0,1\}^d\rightarrow\{0,1\}^m$,对于任意定义在$\{0,1\}n$上具有最小熵$k$的分布$X$,分布$EXT(X;U_d)$,也就是函数的输出$\varepsilon$接近$\{0,1\}m$的均匀分布$U_m$。直观上来说,随机提取器借助一个正随机的种子把最小熵从随机变量$X$中提取出来[7]。

**定义 2**(安全概略)　一个$(M,m,m',n)$的安全概略是一对随机过程$\langle SS,REC\rangle$,这对随机过程具有如下的几个性质[7]。

1) 概略过程:$SS$输入$M$上的一个元素$w$,输出$s\in\{0,1\}^*$

2) 恢复过程:$REC$输入的元素有2个,$w'\in M,s\in\{0,1\}^*$。如果$w$和$w'$的统计距离小于$t$,那么就有$REC(w',SS(w))=w$。

安全概略提供以下几个安全保证,对于任意$M$上的分布$W$,只要$W$上的最小熵大于$m$,安全概略可以保证其输出位串$s$后,$W$的条件最小熵大于等于$m'$,也就是$H_\infty(W\mid SS(W))\geqslant m'$,有了安全概略的概念之后,接下来就可以定义模糊提取器的概念了。

**定义 3**(模糊提取器[7])　一个$(M,m,l,t,\varepsilon)$模糊提取器是满足以下特性的一对随机过程$\langle GEN,REP\rangle$。

1) 产生过程:$GEN$输入$w\in M$,输出$s\in\{0,1\}l$上的一个随机串$R$和$\{0,1\}^*$一个帮助串$P$。

2) 重生过程:$REP$输入$w'\in M$和$P\in\{0,1\}^*$。如果$dis(w,w')\leqslant t,GEN(w)=<R,P>$,那么$REP(w',P)=w$。

模糊提取器提供以下保证:如果定义在$M$上的随机变量$W$的最小熵大于$m$,那么,即便敌手可以观察到$P,R$依旧非常的接近于均匀分布。或者说,如果$GEN(w)=\langle R,P\rangle$,那么$dis((R,P),(U_1,P))\leqslant\varepsilon$。

**引理 1**(从安全概略中得到模糊提取器)　保证$(SS,Rec)$是一个$(M,m,m',n)$的安全概略,并且$Ext$是一个$(M,m,l,t,\varepsilon)$的强提取器,那么下面的$(GEN,REP)$是一个$(M,m,l,t,\varepsilon)$的模糊提取器[8];

1) $GEN(w;r,x)$:设定$P=(SS(w,r),x),R=EXT(w,x)$,输出为$(R,P)$。

2) $REP(w',s,c)$:恢复出$w=REC(w',s)$,输出$R=EXT(w,c)$。

模糊提取器的原理框图结构如图 1 所示

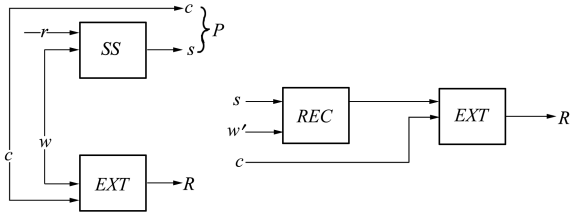


图 1　模糊提取器框图

## 1.2　权限控制树 $T_{pr}$

本文通过权限控制树$T_{pr}$描述访问控制策略,权限树中的每个叶子节点代表某个属性,每个非叶子节点代表一个阈值门。文中所采用的权限控制树类似于操作系统中的权限控制,例如用户$a$可以访问文件$f$,而用户$b$则不能访问文件$f$。

文件拥有者将文件存储于 MSD 之前,会根据某种属性集合在文件中定义某种权限控制树$T_{pr}$,如果某个用户的属性满足该特权树$T_{pr}$,那么该用户便被授予该特权$P$。给定一个权限控制树,如果$num_x$是结点$x$的孩子结点的数量,$k_x$是$x$的阈值且$0<k_x<num_x$,如果至少有$k_x$个孩子结点被赋值为真,那么该结点将被赋值为真。在某种情况下,$k_x=1$时,该结点为 OR 门;$k_x=num_x$时,该结点为 AND 门。

如图 2 所示为某种权限控制树,每个叶子结点均用某个用户属性表示,非叶子结点为设定的阈值门。该权限控制树的满足运算规则如下,$S$代表着用户属性集合,$T_{pr}$代表某个权限控制树,$x$代表权限树中的某个结点。如果用$S$满足$T_{pr}$或结点$x$,则有$T_{pr}(S)=1$或$x(S)=1$。通过递归运算便可计算出$T_{pr}(S)$。如果$x$为叶子结点,当且仅当$att(x)\in S$时,则$x(S)=1$;如果$x$为非叶子结点,当结点$x$至少

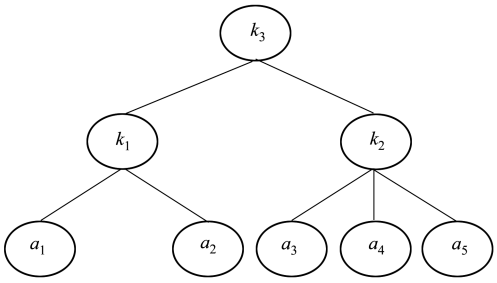


图 2 权限控制树

有 $k_x$ 个孩子结点返回 1 时，则 $x(S)=1$。对于特权树 $T_{pr}$ 的根结点 $R_{pr}$ 来说，只有当 $R_{pr}(S)=1$ 时，$T_{pr}(S)=1$。

### 1.3 双线性对[9]

**定义 4** $G_0$ 是素数阶为 $p$ 的循环群，$g$ 是 $G_0$ 的生成元，运算 $e:G_0\times G_0\rightarrow G_T$ 映射拥有如下 3 个性质：

1）双线性：$e(g^a,g^b)=e(g,g)^{ab}$，$a,b\in Z_p$；

2）不可退化性：存在 $x,y\in G_0$ 使得 $e(x,y)\neq 1$；

3）对称性：对于所有的 $x,y\in G_0$，存在 $e(x,y)=e(y,x)$。

### 1.4 双线性对累积器

双线性对累加器是一种高效的数据认证机制，该机制对于任意大小的集合输入提供一个固定长度的消息摘要，并且为集合中的每一个元素提供一个固定长度的证据，该证据可以用来验证集合中的成员身份。

为了构造一个双线性对累加器，对于 $Z_p^*$ 中的 $n$ 个元素 $\{a_1,a_2,\cdots,a_n\}$ 组成的集合 $L$，先在 $G$ 中计算一个累加值 $acc(L)=g_{\prod_{a_i\in L}(a_i+s)}$，这里的 $s\in Z_p^*$ 是一个随机选取的值，$\prod_{a_i\in L}(a_i+s)$ 是集合 $L$ 的特征多项式。对于任意子集 $L'\subseteq L$，可以计算出一个证据 $Wit_{L',L}=g_{\prod_{a_i\in L-L'}(a_i+s)}$，随后通过判断(1)式是否相等，进行子集验证。

$$e\left(g_{\prod_{a_i\in L'}(a_i+s)},Wit_{L',L}\right)\overset{?}{=\!=}e(acc(L),g)\quad(1)$$

注意，仅给定对应元素 $a$ 和 $\{g^{s^i}:0\leqslant i\leqslant q\}$，$g^{\prod(a+s)}$ 能够通过多项式内插法构造出来。例如，在集合 $L$ 中插入一个新的元素 $a_{n+1}$，从而能够获得一个新的累加值 $acc'(L)=acc(L)^{(a_{n+1}+s)}$，同时从集合 $L$ 中删除一个元素 $a_i$ 后，新的累加值 $acc'(L)=acc(L)^{(a_i+s)^{-1}}$。

## 2 系统结构与功能模块描述

### 2.1 系统结构

图 3 为本文提出的系统模型结构。图 3a）表示用户注册模型，图 3b）表示认证与文件加解密模型。

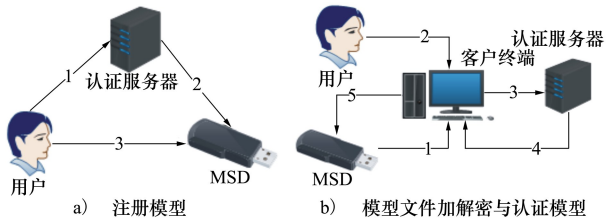


图 3 系统结构

### 2.2 用户添加与删除

1）用户添加：如图 4 所示，属于该 MSD 用户组的用户人数为 2，添加 1 位用户后属于该用户组的用户人数为 3。

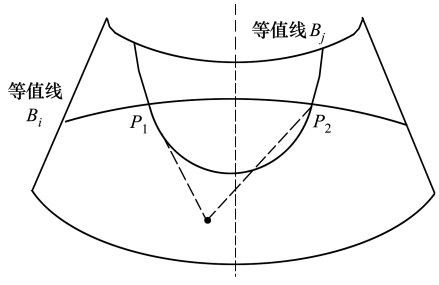


图 4 用户添加

2）用户删除：如图 5 所示，属于该 MSD 用户组的用户人数为 3，添加 1 位用户后属于该用户组的用户人数为 2。

图 5 用户删除

### 2.3 文件细粒度解密权限关系

在这里先定义一种表达式用于表示用户 $u$ 对文件 $f$ 拥有解密权限。例如，$u\prec f$ 表示用户 $u$ 能够解密文件 $f$；$u\succ f$ 表示用户 $u$ 不能够解密文件 $f$。图 6 为 MSD 中文件细粒度解密权限结构关系图，图中显示该 MSD 中共包含 4 份数据文件 $f_1,f_2,f_3,f_4$。用户

对于文件的细粒度解密权限关系可表示为 $u_a < f_1$、$u_b < f_2, f_4$、$u_c < f_1, f_4$、$u_d < f_2, f_4$、$u_e < f_3$。反之,只要不满足此关系,用户便不能解密相应的文件。

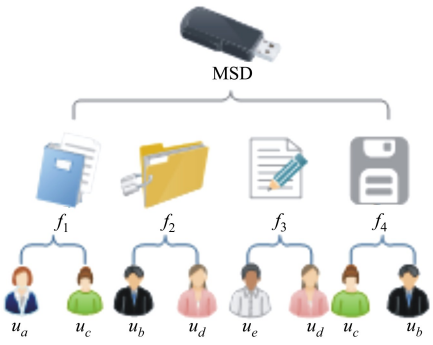


图 6　文件细粒度解密权限关系图

图 6 所示的每一个文件中均包含了某种访问策略,而该策略是文件上传者利用某种用户属性集合生成的权限树 $T_{pr}$,这里的 $pr$ 表示某种特权。例如,文件 $f_1$ 与 $f_2$ 中分别包含着权限 $T_{pr_1}$ 和 $T_{pr_2}$,则说明用户 $u_a$ 拥有权限 $T_{pr_1}$;用户 $u_b$ 则拥有权限 $T_{pr_2}$,那么 $u_a < f_1, u_b < f_2$,但 $u_a > f_2, u_b > f_1$。

## 3　算法阶段描述

### 3.1　用户注册阶段

图 3a)为用户注册模型,在此阶段,用户 U 与认证服务器 AS 按照如下算法过程实现:

1) U、AS、MSD 分别存储自己的私钥 $sk_U, sk_{AS}, sk_{MSD}$ 并发布自己的公钥 $pk_U, pk_{AS}, pk_{MSD}$。

### 3.2　用户身份认证阶段

用户 U 为了向 MSD 证明自己是合法的用户,则该用户 U 需要向认证中心获得授权,当用户将 MSD 插入个人终端后,认证过程执行,具体步骤如下所述:

1) 用户 U 提供 $ID_U$、生物特征 $B_U$ 以及时间戳 $TM$, 并将 $c_{U-AS} = ENC_{pk_{AS}}[ID_U \| B_U \| TM]$ 发送给 AS。

2) AS 执行 $DEC_{sk_{AS}}[c_{U-AS}]$ 后,获得 $ID_U$、生物特征 $B_U$ 以及时间戳 $TM$, 计算得到 $c_{AS-U} = ENC_{pk_{MSD}}[ID_U \| B_U \| TM]$,并将 $c_{AS-U}$ 发送给用户 U;

3) 用户 U 重新获得自身的 $ID_U^*, B_U^*$ 以及 $TM^*$,并计算得到 $c_{U-MSD} = ENC_{pk_{MSD}}[ID_U^* \| B_U^* \| TM^*]$。最后,用户 U 将 $\{c_{AS-U}, c_{U-MSD}\}$ 发送给 MSD。

4) MSD 得到 $\{c_{AS-U}, c_{U-MSD}\}$ 后, 执行 $DEC_{sk_{MSD}}[c_{AS-U}]$ 以及 $DEC_{sk_{MSD}}[c_{U-MSD}]$, 重新获得 $ID_U, B_U, TM, ID_U^*, B_U^*$ 以及 $TM^*$, 并执行如下验证步骤;

(1) 若 $ID_U \neq ID_U^*$, 则认证失败, 否则执行第(2)步;

(2) 若 $B_U \neq B_U^*$, 则认证失败; 否则执行第(3)步;

(3) 若 $TM^* - TM > \Delta T$,则认证失败;否则,认证通过。

只有当第(1)、(2)以及(3)步均成立时,则用户的身份认证通过;否则用户身份认证失败,MSD 将立即与个人终端断开连接。

以上 1) ~ 4) 步有效保证了 U 的合法身份,而 U 是否属于该 MSD 用户组需要通过第 5) 步验证。

5) 对于该 MSD 用户组中的用户集合可以表示为 $User = \{U_1, U_2, \cdots, U_n\}$,这里的 $n$ 表示用户组中用户的个数;第 $i$ 个用户的属性集合可以表示为 $att_i = \{a_1, a_2, \cdots a_m\}$ 且 $1 \leqslant i \leqslant n$, $m$ 表示每个用户的属性个数且 $a_j \in Z_p^*$, $1 \leqslant j \leqslant m$。对每一用户的属性集合 $att_i = \{a_1, a_2, \cdots a_m\}$ 计算 $acc_i = acc(att_i) = g_{\prod_{a_j \in att_i}(a_j+s)}$, $s \in Z_p^*$ 且是随机选取的值。最终得到 $n$ 个用户的属性累加值集合 $acc = \{acc_1, acc_2, \cdots acc_n\}$,最后计算得到所有用户的累积值 $ACC = g_{\prod_{acc_i \in acc}(acc_i+r)}$, $r \in Z_p^*$ 且是随机算选取的值。

U 根据自己的属性集合计算得到属性累积值 $acc_U$ 并发送给 MSD,MSD 计算得到证据 $ACC_{witness} = g_{\prod_{acc_i \in acc-att_U}(acc_i+r)}$。

MSD 判断 $e(acc_U, ACC_{witness}) = e(ACC, g)$ 是否相等,若相等则 MSD 用户组用户成员身份验证通过;否者认证失败,则 MSD 将立即与个人终端断开连接。

进过以上 1) ~ 5) 步,MSD 验证了用户的身份,同时又验证了 MSD 的用户组身份。

### 3.3　MSD 用户组成员添加与删除

MSD 用户组中的用户集合表示为 $User = \{U_1, U_2, \cdots, U_n\}$,生成的属性累加值集合为 $acc = \{acc_1, acc_2, \cdots acc_n\}$,添加一位用户 $U_{n+1}$ 进入用户组中,关于用户 $U_{n+1}$ 生成属性累加值 $acc_{n+1}$;则新的 $ACC^* = ACC^{(acc_{n+1}+s)}$;删除用户组中的一位用户 $U_i$,则新的 $ACC^* = ACC^{(acc_i+1)^{-1}}$。

### 3.4　文件加密阶段

当 U 经过 3.2 小节的身份认证之后,能够按照

如下算法过程以加密形式上传文件至 MSD 中。

1）U 采用对称加密算法以及对称密钥 $k$（例如 $DES$）加密文件 $F$，生成的密文 $C = ENC_k(F)$；

2）为了保护对称密钥 $k$，随机选取 $s \in Z_p^*$，计算得到 $t = s \oplus k$。将计算得到的 $t$ 隐藏在权限控制树之中，这里将采用 Shamir 的密钥分享协议生成访问控制树 $T_{pr}$，达到隐藏 $t$ 的作用。具体过程如下：

这里采用多项式插值法实现 Shamir 的密钥分享协议。对于 $T_{pr}$ 中的每一结点 $x$ 选择一个多项式 $q_x$，设置多项式的次幂 $d_x$ 比设定的阈值 $k_x$ 小 1，即 $k_x = d_x + 1$。从 $T_{pr}$ 的根结点 $R_{pr}$ 开始，随机选取多项式 $q_x$ 的系数，并使得 $q_{R_{pr}}(0) = t$，采用 Shamir 密钥分享协议，将密钥逐级分享下去。对于其他结点 $x$，每个结点的多项式 $q_x$ 系数随机选取，而常数项为 $q_{\mathrm{parent}(x)}(\mathrm{index}(x))$，从而使得 $q_x(0) = q_{\mathrm{parent}(x)}(\mathrm{index}(x))$，这里的 $\mathrm{parent}(x)$ 表示 $x$ 的父结点。最终生成此文件的权限控制树 $T_{pr}$。最终 U 将 $\{C, s, T_{pr}\}$ 存储于 MSD 中。

### 3.5 文件解密阶段

当用户 U 进过 4.2 小节的身份人认证之后，便能通过如下算法过程访问 MSD 中的文件。具体过程如下所示：

1）U 为了解密文件 $F$，首先 U 的属性必须满足权限控制树 $T_{pr}$，根据 2.2 小节权限控制树的定义，权限控制树的每个叶子结点代表用户的每个属性，而非叶子借点代表着一个阈值门，因此只有用户的属性满足该权限控制树时，根据 Shamir 的原理，最终恢复出隐藏在权限控制树 $T_{pr}$ 的值 t；

2）MSD 执行 $k = t \oplus s$，$F = DEC_k(C)$，最终获得文件明文 $F$。

## 4 方案安全性与性能分析

### 4.1 抵抗拒绝服务攻击

3.2 小节用户身份认证过程中，U 需输入个人生物特征（比如指纹等），但根据相关资料[10]显示，同一个人在不同时候获取的个人生物特征会有一定的差别，因此在该认证阶段，U 的身份可能会认证失败，导致 MSD 拒绝提供服务的事件发生。但本方案采用了模糊提取器，U 的个人生物特征采集后经过了模糊提取器算法处理，能够保证认证阶段采集的生物特征值与系统建立阶段采集的生物特征值一致，因此本方案能够抵抗拒绝服务攻击。

### 4.2 能够抵抗用户伪装攻击

在 3.2 小节说明用户 U 的认证过程之中，用户 U、认证中心 AS 以及大容量存储设备 MSD 均分别保存了自己私钥 $sk_{\mathrm{U}}$、$sk_{\mathrm{AS}}$、$sk_{\mathrm{MSD}}$，同时各自发布了公钥证书。

1）用户 U 提供 $ID_{\mathrm{U}}$、生物特征 $B_{\mathrm{U}}$ 以及时间戳 $TM$，并将 $c_{\mathrm{U-AS}} = ENC_{pk_{\mathrm{AS}}}[ID_{\mathrm{U}} \| B_{\mathrm{U}} \| TM]$ 发送给 AS，因此在该阶段中假设攻击者 $A$ 获取了 $c_{\mathrm{U-AS}}$，$A$ 也无法解密该密文，因此无法修改其中的身份信息。

2）AS 执行 $DEC_{sk_{\mathrm{AS}}}[c_{\mathrm{U-AS}}]$ 后，获得 $ID_{\mathrm{U}}$、生物特征 $B_{\mathrm{U}}$ 以及时间戳 $TM$，计算得到 $c_{\mathrm{AS-U}} = ENC_{pk_{\mathrm{MSD}}}[ID_{\mathrm{U}} \| B_{\mathrm{U}} \| TM]$，并将 $c_{\mathrm{AS-U}}$ 发送给用户 U。当 AS 解密 $c_{\mathrm{U-AS}}$ 中的用户身份信息后，重新利用自己的公钥加密用户身份信息，重新生成 $c_{\mathrm{AS-U}}$ 发送给用户 U。因此在此过程中，假设攻击者 $A$ 获取了 $c_{\mathrm{AS-U}}$，$A$ 也无法解密获得用户的身份信息。

3）用户 U 重新获得自身的 $ID_{\mathrm{U}}^*$，$B_{\mathrm{U}}^*$ 以及 $TM^*$，并计算得到 $c_{\mathrm{U-MSD}} = ENC_{pk_{\mathrm{MSD}}}[ID_{\mathrm{U}}^* \| B_{\mathrm{U}}^* \| TM^*]$。最后，用户 U 将 $\{c_{\mathrm{AS-U}}, c_{\mathrm{U-MSD}}\}$ 发送给 MSD。在此过程中，攻击者 $A$ 假设截获了 $\{c_{\mathrm{AS-U}}, c_{\mathrm{U-MSD}}\}$，但 $A$ 依旧不能获取其中的身份信息。

最终进过上述三步用户身份认证过程，攻击者 $A$ 不能获得任何用户信息，也不能伪装成用户 U 执行用户伪装欺骗攻击。

### 4.3 能够抵抗重放攻击

在 3.2 小节用户身份认证过程中，U 向 AS 以及 MSD 均发送了系统当前时间 $TM$，过程如下所示：

1）用户 U 提供 $ID_{\mathrm{U}}$、生物特征 $B_{\mathrm{U}}$ 以及时间戳 $TM$，这里的时间戳 $TM$ 发送给了认证中心，并对该时间戳进行了加密 $c_{\mathrm{U-AS}} = ENC_{pk_{\mathrm{AS}}}[ID_{\mathrm{U}} \| B_{\mathrm{U}} \| TM]$，因此攻击者 $A$ 无法获取该时间戳；

用户 U 给认证中心 AS 发送个人身份信息时，又重新获得自身的 $ID_{\mathrm{U}}^*$，$B_{\mathrm{U}}^*$ 以及 $TM^*$，并计算得到 $c_{\mathrm{U-MSD}} = ENC_{pk_{\mathrm{MSD}}}[ID_{\mathrm{U}}^* \| B_{\mathrm{U}}^* \| TM^*]$。最后，用户 U 将 $\{c_{\mathrm{AS-U}}, c_{\mathrm{U-MSD}}\}$ 发送给 MSD。在此过程中，攻击者 $A$ 假设截获了 $\{c_{\mathrm{AS-U}}, c_{\mathrm{U-MSD}}\}$，但 $A$ 依旧不能获取其中的身份信息。当 MSD 获得 $\{c_{\mathrm{AS-U}}, c_{\mathrm{U-MSD}}\}$ 信息之后，解密获得 2 个时间戳 $TM$ 以及 $TM^*$，只需要判断 $TM$ 与 $TM^*$ 之间的时间差，若 $TM^* - TM > \Delta T$，则说明系统中存在重放攻击，因此本方案可以抵抗重返攻击。

### 4.4　能够抵抗离线口令猜测攻击

用户 U 认证过程中,需要提供用户的个人身份信息,个人生物特征等等,而每个人的生物特征等相关信息是唯一的,因此攻击者无法执行口令猜测攻击。

### 4.5　能够提供有效安全的用户组成员认证

在第 3.2 小节第(5) 步中详细说明了用户组成员的认证过程,对于 MSD 用户组中的用户集合 $User = \{U_1, U_2, \cdots, U_n\}$,最终生成了所有用户的累积值 $ACC = g_{\prod_{acc_i \in acc}^{(acc_i+r)}}$,$r \in Z_p^*$ 且是随机算选取的值。而某个用户 U 根据自己的属性集合计算得到属性累积值 $acc_U$ 并发送给 MSD,MSD 计算得到证据 $ACC_{\text{witness}} = g_{\prod_{acc_i \in acc-att_U}^{(acc_i+r)}}$。MSD 只需要判断 $e(acc_U, ACC_{\text{witness}})$ 与 $e(ACC, g)$ 是否相等,若相等则 MSD 用户组用户成员身份验证通过;否者认证失败。而双线性对累加器的安全性源于 q-SBDH 假设[11]。

### 4.6　性能分析

表 1 将本文所提方案的优势与方案进行了对比,从表 1 中可以发现:

1) 本文所提方案能够抵抗用户伪装攻击、内部攻击、离线口令猜测攻击、拒绝服务攻击以及重放攻击等,因此相对于其他一些方案,在抵抗以上这个攻击上具有一定的优势。

2) 本文所提方案能够进行 MSD 中用户组成员管理,提供了用户组成员添加或删除功能;其次,对于 MSD 中文件实现了细粒度权限控制,严格控制了文件访问权限,提升了用户组间的安全性。

表 2 将本文所提方案与方案在用户注册、用户身份验证、文件加密、文件解密等方面进行了运算效率对比,同时将方案中的用户组成员添加、删除以及权限控制树的运算开销进行了说明。$T_h$ 表示哈希函数运算、$T_m$ 表示标量乘法运算、$T_{\text{div}}$ 表示除法运算、$T_e$ 表示双线性对运算、$T_{En}$ 表示对称加密运算、$T_{De}$ 对称解密运算、$T_{\exp}$ 表示指数运算、$T_{inter}$ 表示插值运算。

**表 1　方案攻击类型或特征对比**

| 抵抗攻击类型或特征 | Ku 等方案[1] | Yoon 等方案[2] | Yang 等方案[3] | Lee 等方案[4] | He 等方案[5] | Giri 等方案[6] | 所提方案 |
|---|---|---|---|---|---|---|---|
| 用户伪装攻击 | no | no | yes | yes | yes | yes | yes |
| 内部攻击 | no | no | no | yes | yes | yes | yes |
| 离线口令猜测攻击 | no | no | yes | no | yes | yes | yes |
| 拒绝服务攻击 | no | no | yes | no | yes | yes | yes |
| 重放攻击 | yes | yes | no | no | no | yes | yes |
| 用户组成员验证 | no | no | no | no | no | no | yes |
| 用户组成员添加/删除 | no | no | no | no | no | no | yes |
| 文件细粒度权限控制 | no | no | no | no | no | no | yes |

**表 2　运算效率对比**

| 运算类型 | Ku 等方案[1] | Yoon 等方案[2] | Yang 等方案[3] | Lee 等方案[4] | He 等方案[5] | Giri 等方案[6] | 所提方案 |
|---|---|---|---|---|---|---|---|
| 用户注册 | $2T_h$ | $2T_h$ | $1T_m + 1T_h + 2T_e$ | $4T_h + 1T_{pm}$ | $4T_h + 1T_{pm}$ | $5T_h + 1T_{En}$ | no |
| 用户身份验证 | $3T_h$ | $3T_h$ | $2T_h + 2T_e + 1T_{De} + 1T_{En}$ | $3T_h + 1T_{pm} + 1T_{En} + 1T_{De}$ | $3T_h + 1T_{pm} + 1T_{En} + 1T_{De}$ | $2T_h$ | $3T_{En} + 3T_{De} + 3T_{\exp} + T_e$ |
| 文件加密 | $6T_h$ | $6T_h$ | $3T_h + 1T_m + 4T_e + 1T_{De} + T_{En}$ | $7T_h + 2T_{pm} + 1T_{En} + T_{De}$ | $7T_h + 2T_{pm} + 1T_{En} + 1T_{De}$ | $7T_h + 1T_{En}$ | $1T_{En}$ |
| 文件解密 | $2T_h + 1T_{De}$ | $3T_h + 2T_{De}$ | $2T_{De} + 1T_h$ | $2T_{De} + 1T_h$ | $1T_{De} + 1T_h$ | $1T_{De}$ | $1T_{De}$ |
| 权限控制树生成 | *no* | *no* | *no* | *no* | *no* | *no* | $mnT_{inter}$ |
| 用户组成员添加 | no | no | no | no | no | no | $1T_{\exp} + T_{pm}$ |
| 用户组成员删除 | no | no | no | no | no | no | $1T_{\exp} + 1T_{\text{div}}$ |
| 权限控制树验证 | no | no | no | no | no | no | $mnT_{\text{inter}}$ |

## 5 结 语

本文提出了一种运用于大容量存储设备下的用户认证与数据保护方案。利用属性基加密技术以及权限控制技术,实现了对 USB 设备用户的安全认证、实现了对 USB 设备中隐私数据细粒度控制。安全性分析表明该方案具有较强的安全性,能够抵抗重放攻击、拒绝服务攻击、用户伪装攻击、口令猜测攻击等。

本文提出了适用于大容量存储下的用户认证与数据保护方案,后续需要在文件加解密上进一步提升运算速率;其次,后续会研究如何实现存储器器中密文数据的修改与删除,同时文中方案性能分析部分只是利用了数学理论分析了运算效率,后续将会利用代码具体实现本文方案。

## 参考文献:


[1] Ku W C, Chen S M. Weaknesses and Improvements of an Efficient Password Based Remote User Authentication Scheme Using Smart Cards[J]. IEEE Trans on Consumer Electronics, 2004, 50(1):204-207

[2] Yoon E J, Ryu E K, Yoo K Y. Further Improvement of an Efficient Password Based Remote User Authentication Scheme Using Smart Cards[J]. IEEE Trans on Consumer Electronics, 2004, 50(2): 612-614

[3] Yang F Y, Wu T D, Chiu S H. A Secure Control Protocol for USB Mass Storage Devices[J]. IEEE Trans on Consumer Electronics, 2011, 56(4):2239-2343

[4] Zhang L, Tang S, Chen J, et al. Two-Factor Remote Authentication Protocol with User Anonymity Based on Elliptic Curve Cryptography[J]. Wireless Personal Communications, 2015, 81(1):53-75

[5] He D, Kumar N, Lee J H, et al. Enhanced Three-Factor Security Protocol for Consumer USB Mass Storage Devices[J]. IEEE Trans on Consumer Electronics, 2014, 60(1):30-37

[6] Giri D, Sherratt R S, Maitra T. A Novel and Efficient Session Spanning Biometric and Password Based Three-Factor Authentication Protocol for Consumer Usb Mass Storage Devices[J]. IEEE Trans on Consumer Electronics, 2016, 62(3):283-291

[7] 李西明,张明武. 模糊提取器构造与安全[J]. 重庆理工大学学报:自然科学版,2010,24(8):49-53
Li Ximing, Zhang Mingwu. Conrstruction and Security of Fuzzy Extractors[J]. Journal of Chongqing University of Technology: Natural Science, 2010, 24(8): 49-53 (in Chinese)

[8] Yang B, Pang X Q, Du J Q, et al. Effective Error-Tolerant Keyword Search for Secure Cloud Computing[J]. Journal of Computer Science and Technology, 2014, 29(1): 81-89

[9] Zhang F, Safavi-Naini R, Susilo W. An Efficient Signature Scheme from Bilinear Pairings and Its Applications[J]. Lecture Notes in Computer Science, 2004, 2947(39):277-290

[10] Das A K. Analysis and Improvement on an Efficient Biometric-Based Remote User Authentication Scheme Using Smart Cards[J]. Information Security Iet, 2011, 5(3):145-151

[11] Papamanthou C, Tamassia R, Triandopoulos N. Optimal Verification of Operations on Dynamic Sets[C]//Conference on Advances in Cryptology, 2011:91-110