文章编号:1674 - 2869(2009)03 - 0074 - 03

支持协同工作的加解密系统设计与实现

何成万,焦素廷,李　健
(武汉工程大学计算机科学与工程学院,湖北 武汉 430074)

摘　要:介绍了一种支持协同工作的加解密系统.与一般加解密系统不同,此系统在逻辑上将员工划分为不同的组.各组的加密文件可以在组内成员间解密,达到了加密文件共享的目的.

关键词:加密;解密;协同工作
中图分类号:TP 309.7　　文献标识码:A

0　引　言

一般的加解密工具加密的文件,解密时需要解密密钥[1]和加密文件一起使用才能解密.不同的客户使用的加密密钥是不同的,因此就产生了许多不同的解密密钥.有时候,需要在一个部门或单位共享加密文件,当共享的加密文件逐渐增多时,解密文件的时候必须将加密文件与对应的解密密钥一起使用来解密文件,这样显然很不方便,而且容易造成混乱和信息的泄漏.为了方便文件在本部门共享,又不会将信息泄漏给别的部门,需要设计一个新型的支持协同工作的加解密系统.这也是协同工作的加解密系统与一般加解密系统的主要区别.

1　支持协同工作的加解密系统介绍

在一个企业内部常常划分为几个部门,各个部门之间需要共同合作完成一项任务.但是有的部门的文件是不需要公开的.比如财务部门的财务信息通常只在本部门内部传递;技术部门的核心技术资料也只在一些员工之间交流.为了使各个部门协同工作[2]又不至于使信息在不适当的范围流动,支持协同工作的加解密系统自动的实现加密文件在逻辑空间的隔离,实现了在特定范围使用加密文件.加密与解密文件的过程不需要用户输入密钥,由系统完成协同工作的过程.协作过程如图 1 所示,假设用户 A 与用户 B 属于同一组,用户 A 使用系统加密文件,并将加密文件传给用户 B.用户 B 使用系统解密,得到解密文件.

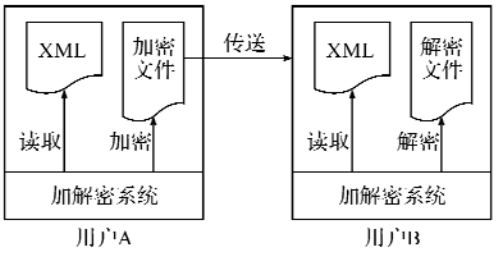

图 1　加密文件的共享

Fig. 1　Encryption file sharing

此系统可以在网络环境下运行,地域上分散的用户可以方便使用此系统协同工作.系统将用户划分成一些组,组与组之间的关联体现也被系统描述,并将此信息存于文件中.

2　支持协同工作的加解密系统实现

为了使加解密工作清晰有序的进行,系统的各个部件之间需要协同实现有效的控制逻辑.

2.1　加解密系统中,员工、组与密钥之间的关系表示

利用 XML(The Extensible Markup Language)[3]文件可以非常方便表示出这三者之间的关系.利用三个 XML 文件,将这三者关联起来.在用户验证 XML 文件中,存储了员工的用户名、密码、所属组与个人的密钥信息.其格式如下:

```
<Employee>
<Name> aaa </Name>
<Psw> bbb </Psw>
< Key ID = " {E6AC7573-6396-113c-9EC3-7B55B87ACCDD}" > ccc </Key>
<Part> A </Part>
</ Employee>
```

收稿日期:2008 - 05 - 21
基金项目:国家自然科学基金项目(60873024);湖北省教育厅重点研究项目(D20081505).
作者简介:何成万(1967 -),男,湖北荆门人,副教授,博士.研究方向:软件工程.

它描述了一个用户名为 aaa，密码为 bbb 的员工、他的 private 密钥与该用户属于的组 Λ.

把密钥分为 public、protected、private 三个等级，后面将会讲述. 此处存放的是 private 密钥. 在存储密钥的 XML 文件当中，存放了各个组的密钥信息. 存储在文件里面的每一个密钥都有 ID、Access、Current 属性. ID 属性是密钥的唯一标识，使用微软提供的全局唯一标识(GUID). Access 属性取值空间为 public、protected，说明此密钥的访问等级. Current 属性取值为 0 或 1. 1 表示此密钥是最新的正在使用的密钥，否则为 0. 其格式如下：

<Part Name－"Λ">

< Key ID －" { B0B95ED7-CD3C-45ad-82CD-70D884279B22}" Acccss="public" Current="1" >vvvvv</Key>

< Key ID －" { 2D1D6E2D-E830-4cc6-9C66-0E4775C1C6DE}" Acccss－"protected" Current－"1" > zzzzz</Key>

</Part>

它描述了组 A 中存在两个密钥，Access 属性分别为 public、protected. 在存储组关系的 XML 文件中表示各组之间的关系. 其格式如下：

<Part Name－"Λ">

<Relation rc－"Super"> B</Relation>

<Relation rc="Sub"> C</Relation>

</Part>

它描述了组 A 有一个上级组 B 和一个下级组 C.

在如图 2 所示的上下级关系当中，假如 Λ 是 A1 和 A2 的直属上级，A 组的加密文件可在 A 组内解密，也可以选择在 A、A1、A2 中被解密使用. A1 与 A2 中的加密文件一定能被 A 解密使用. 也就是，Λ1 与 Λ2 的加密文件一定可以被 Λ 解密，反之却不一定. 系统通过这三个 XML 文件表示了它们之间的协作关系，并自动提供加解密服务. 为了系统的安全性，每一个员工只属于一个组. 如果有一个员工属于组 Λ1，同时又属于组 Λ2，他就有可能在 A1 与 A2 之间泄漏信息. 像这样角色比较特殊的员工，可以考虑将他加入组 A 当中.

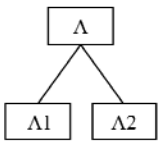

图 2　上下级关系

Fig. 2　Relation between superior and subordinate

2.2　密钥的更新与管理

系统采用 DES[4] 加解密算法. 此算法属于对称密钥[1] 的加解密算法，加密密钥与解密密钥相等. 在密钥管理当中，把密钥分为三个访问等级：public、protected、private. 每个通过系统验证的用户可选择其中之一，进行文件加密. 不同的级别的加密文件使用的范围也不同. 使用 public 访问等级加密的文件将可在本组和直属上属组中解密. 以 protected 访问等级加密的文件则只能在本组范围被解密. 而使用 private 访问等级加密的文件，只能由加密者本人解密. 系统中的每一个密钥被唯一的 ID 标识，系统可随时更新正在使用的密钥，新增密钥的 Current 属性被置 1，并将旧密钥的 Current 属性设为 0.

2.3　加密过程

系统中用到的三个 XML 文件被保存在另一台机器上，第一次用户验证过程和查找密钥的过程序要通过网络，在另一台机器上完成验证[5] 与查找，返回结果，并将结果存放在系统注册表中，之后使用就可以先通过注册表得到结果. 只有在注册表中查找不到所需信息时，才使用网络. 用户使用系统加密文件的操作十分简单. 加密文件时，只需输入用户名与密码即可使用系统的加密功能. 首先，系统通过在用户验证 XML 文件中查找匹配的用户名与密码，并从中获取用户的 private 访问属性的密钥与所属组的信息. 如果匹配过程成功，则用户选择具有 public、protected、private 之一的访问属性的密钥进行加密. 默认访问属性为 protected，即在本组中的成员可以解密该加密文件. 如果用户选择 public 访问权限进行加密，系统根据用户所属组在存储密钥的 XML 文件中找到 Access 属性为 public，Current 属性为 1 的密钥加密. 如果用户选择 protected 访问权限加密，系统根据用户所属组在存储密钥的 XML 文件中找到 Access 属性为 protected，Current 属性为 1 的密钥加密. 如果用户选择 private 访问权限加密，系统直接使用在用户验证 XML 文件中匹配节点中的密钥加密. 之后，系统会生成后缀名为 ept 的文件. 加密过程程序流程如图 3 所示. 在生成的文件中，保存着加密信息及描述该文件的元信息. 描述信息被保存在文件头部，其结构如图 4 所示，整个文件头长度为 33 字节.

2.4　解密过程

当用户使用系统解密，系统要求用户输入用户名与密码通过验证. 之后，系统读取待解密文件头部，查看加密标识. 如果加密标识不正确，则提示用户退出. 如果加密标识正确，则读取存储在文件中的密钥 ID，并在系统中查找比较各相关密钥属性中的 ID 值. 查找过程如下：首先，系统找到该

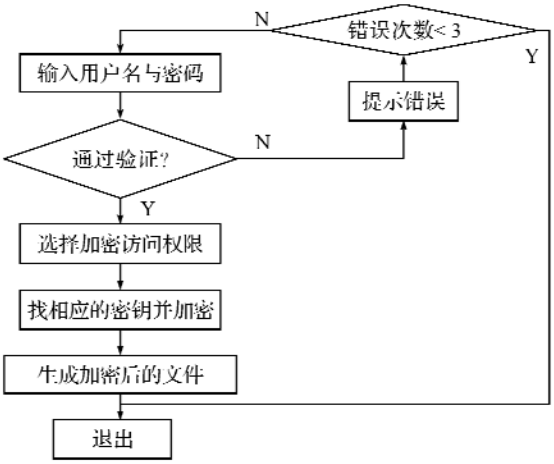

图3 加密过程程序流程图

Fig. 3 Flow chart of encryption process

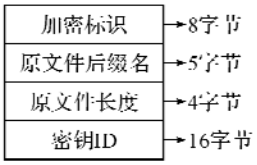

图4 文件头结构

Fig. 4 Header structure

用户的private密钥,并比较其ID是否匹配.如果匹配,则使用此密钥解密.如果不匹配,则继续在用户所在组查找并比较ID是否匹配.如果找到了匹配的密钥,则使用此密钥解密,否则继续在其下级组中查找Access属性为public的密钥的ID属性值是否匹配.如果找到匹配项,则使用此密钥解密,否则提示用户无法解密此文件后退出.密钥查找过程的程序流程图如图5所示.在此过程中找到与待解密文件中ID匹配项后,即可使用相应得密钥解密.解密完毕后,还原该解密后的文件后缀名.

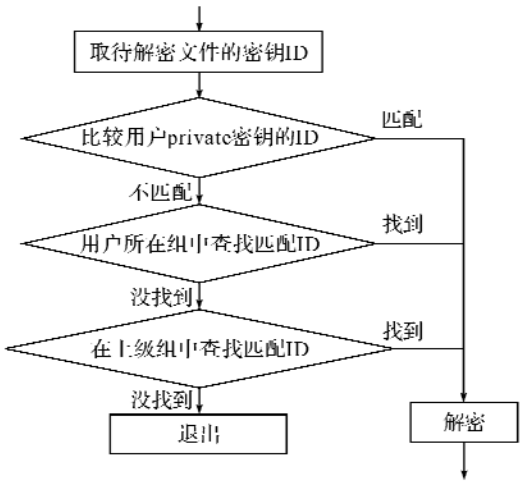

图5 密钥查找流程图

Fig. 5 Flow chart of finding secret key

2.5 加解密系统的安全性及保密性分析

此系统将加解密文件的密钥存放在一台安全的机器上的一个单独的XML文件中,该文件不会被非法修改.所有用户都没有直接使用加解密文件的密钥.任何用户使用系统都必须通过系统的验证.所以对系统的攻击,要求攻击者能获取相应的权限.这就要看用户使用的登录口令的安全性.也就是说,只要用户的身份没有被利用,被此系统加密过的文件不会被非法使用,系统就是安全的.

由于解密使用DES算法,所以保密性与该算法的强度相关.

3 Windows Shell扩展与文件关联

使用Windows Shell扩展技术实现鼠标右键菜单的功能.当用户选中某文件时,单击鼠标右键,会弹出一个菜单,其中有"加密文件"、"解密文件"两个菜单项,分别实现加密与解密.另外,使用文件关联,使的双击后缀名为ept的文件,即可自动运行系统进行解密,方便用户操作.

4 结 语

在支持协同工作的加解密系统中,使用了XML文件描述员工与组,组与组之间的关系,实现了信息在逻辑单位上的隔离[6].用户使用时只需要输入用户名与密码通过系统验证,控制逻辑存于系统内部,用户操作过程非常简单.

参考文献:

[1] STALLINGS. 密码编码学与网络安全[M]. 北京:电子工业出版社,2005.

[2] 倪强,朱光喜. 计算机支持下的协同工作的研究现状综述[J]. 计算机工程与应用,2000,36(4):5-7.

[3] 栗松涛. XML程序设计[M]. 北京:清华大学出社,2001.

[4] 马银华,刘明生,王书海,等. 提高DES加密强度的密钥选位方法研究[J]. 计算机工程,2000,26(3):68-69.

[5] 刘黎志,刘军. 基于Internet的ASP.NET应用程度安全模型研究[J]. 武汉工程大学学报,2008;30(3):90-93.

[6] 何成万,李健,焦素廷. 基于MVC模式的科研成果管理系统的开发[J]. 武汉工程大学学报,2009,31(1):79-81.

(下转第88页)