# 基于 XML 数据安全交换的方法

周杭霞[1]，夏荣钊[2]，何利力[2]

(1. 中国计量学院 信息工程学院，浙江 杭州 310018；2. 浙江大学 计算机科学与技术学院，浙江 杭州 310027)

**摘 要**：Web Services 使用基于 XML 的消息机制作为服务的创建和访问机制。客户通过 Web 协议就可以方便地访问 Web Services 所封装的特定的功能和商业逻辑。通过对 Web Services 中可能实现的安全防范措施进行研究，分析比较了现有解决方案的优劣。最后，提供了一种实现基于 XML 数据安全交换的方法，依据此加密方法可以解决大多常见的安全漏洞问题。

**关键词**：Web Services；XML 加密；DOM；Cryptography

**中图法分类号**：TP309.2 **文献标识码**：A **文章编号**：1001-3695(2006)04-0126-03

# Mechanism Security Exchange for XML Data

ZHOU Hang-xia[1], XIA Rong-zhao[2], HE Li-li[2]

*(1. College of Information & Engineering, China Jiliang University, Hangzhou Zhejiang 310018, China; 2. College of Computer Science & Technology, Zhejiang University, Hangzhou Zhejiang 310027, China)*

**Abstract:** XML Message as the services creation and accessing ways of Web Service. By Through the web protocols, customer can access the functions and business logical which be encapsulated by Web Service. In this paper, we summarize the possible security ways of Web Service and analysis the advantage and disadvantage of current encryption solutions. At the end, we provide a kind of XML-based encryption method, which is one way to handle complex requirements for security in data interchange applications. By the way, we can solve most of the security problems.

**Key words:** Web Services; XML Encryption; DOM; Cryptography

## 1 引言

随着企业级互联网应用中对事物处理的要求进一步提高，原有集中的信息存储模式已经不能满足企业对信息及时性、交互性的要求，为了更好地满足企业的这些需求，真正实现企业在互联网上的自主性、独立性，Web Services 应运而生，但在 Web Services 的应用中，出现了一些新的问题，对应用的实现也提出了一些新的要求。Web Services 与传统的组件化思想非常相似，它们都封装了特定的功能，包括商业逻辑和函数，但是 Web Services 与传统组件不同，传统组件要求客户使用特定的对象模型进行访问，而 Web Services 并无此要求，客户通过 Web 协议就可以轻松地访问它们。Web Services 使用基于 XML 的消息机制作为服务的创建和访问机制，允许软件开发者无缝地集成一个功能多样的客户应用和服务应用。

然而，XML 语言是一种面向数据的标记规范，XML 标记通常总是力求准确清晰地说明数据本身的涵义，即使对于一些非常陌生的 XML 文件，人们也很容易理解其所要表达的内容，从这个意义上讲，XML 数据是完全开放的，所以，一旦含有特殊机密的商业信息的 XML 文档被别有用心的人直接得到，泄密几乎是必然的。因此要设计一个基于 Web Services 传递数据的企业级互联网应用系统，首要问题将是数据安全和交换，这是新一代 Web 服务普遍应引起关注的话题，这也是本文要讨论的问题。

收稿日期：2005-04-04；修返日期：2005-05-13

## 2 现有解决方案及存在问题

### 2.1 当前 Web Services 中可能实现的安全防范措施

将会出台的 Web 服务基本的安全规范包括：

(1) 用于整合的 Web 服务描述语言(WSDL)，WSDL 是采用 XML 语言来描述 Web 服务的属性，例如它做什么，位于哪里和怎样呼叫它。

(2) 用于认证和授权的安全性声明标记语言。

(3) 用于渠道保密的安全槽层(SSL)。

(4) 用于高度机密的 XML 加密标准和用于高级授权的 XML 数字签名。

此外，其他几项规范也会陆续出台，包括：Web 服务安全性规范(包括 XML—加密和 XML—数字签名)、XML 密钥管理规范和用于授权的可扩展访问控制标记语言规范等。

为 Web 服务提供安全功能和组件的模型需要把现有的流程和技术与将来的应用程序的安全性需求集成起来。统一的安全技术就必须把应用程序对安全的需求从特定的机制中抽象出来，目的是让开发者能够容易地使用异类系统建立可互操作的安全解决方案。成功的 Web 服务安全方法需要一组灵活的、可互操作的基本元素，通过策略和配置，这些安全性基本元素可以使多种安全解决方案成为可行的方案。可行的 Web 服务安全性机制需要满足和包括下列组件的要求：

(1) 网络安全性。支持如 SSL 等提供机密性和完整性的安全传输机制。

(2) XML 消息安全性。①XML 数字签名,以便接收方可以证明消息发送方的身份。②XML 加密,提供数据元素的机密性使能够验证交换。W3C 发布 XML 密钥管理服务(XML Key Management Services, XKMS)的备忘录,帮助分发及管理在端点之间进行安全通信所需的密钥。

(3) 端点验证及授权。①支持在企业之间交换信息的合同中定义哪些雇员可以使用哪些服务。中介体负责审计和服务原始性证明。②支持网络内部的、可信任的第三方验证服务,如 Kerberos。

(4) 安全性服务描述。①描述是否支持数字签名、加密、验证和授权以及如何支持它们。Web 服务请求者使用服务描述的安全性元素,来查找符合政策要求及其安全性方法的服务端点。②OASIS 成立了一个技术委员会来定义授权和验证断言安全声明标记语言(Security Assertion Markup Language, SAML),帮助端点接收和决断访问控制权,是基于 XML(可扩展标记语言)面向 Web 服务的架构。SAML 通过因特网对不同的安全系统的信息交换进行处理。③OASIS 同时成立了另一个技术委员会来标准化访问控制权的表达(eXtensible Access Control Markup Language, XACML),帮助端点能够以一致的方式解析 SAML 断言。

### 2.2　现有解决方案存在问题

因特网上常用的传统安全技术对 Web Services 来说是远远不够的,主要问题在它们的传输存在依赖性。如最广泛使用的安全技术 SSL(Secure Socket Layer)受限于网络通信端点。SSL 只能对全部信息进行加密,而不能有选择地对部分信息加密,传送大量数据时会有性能方面的问题,而且 SSL 只能确保点对点的安全,无法保障端到端的安全。采用 VPN 解决方案,但存在费用昂贵,维护配置烦琐的缺点。(原文:更准确地说,身份特性只能指定给常被多个 Web Services 共享的通信端点。)

其他安全技术,如 GSS-API(Generic Security Service Application Programmers Interface)和基于 GSS-API 的安全机制(如 SPKM(Simple Public Key Mechanism)和 Kerberos)主要是为松散连接的体系结构而设计的。GSS-API 独立于传输和安全机制(安全机制独立性意味着密码技术、身份表示和数据签名等潜在技术被完全封装了)。当今 Web Services 中最常用的安全机制仍然是 SSL,但是 SPKM 和 Kerberos 的使用也在不断增长。

基于以上原因,2002 年 12 月 10 日,领先的业界 Web 标准组织“WWW 联盟”(W3C)批准了两个 XML 加密标准,W3C 表示,XML 加密语法和处理以及 XML 签名的加密转换这两个标准将使用 XML 的 Web 网页对在两个网站之间进行交换的部分网页进行加密处理。

## 3　Web Services 中 XML 数据加密方法的研究

### 3.1　Web Services 中 XML 的加密语法

目前,在与 XML 相关的安全性领域方面开发规范和标准中最重要的部分是 XML 加密(Xenc)、XML 签名(XML-SIG)、XML 加密管理规范(XKMS)、XML 访问控制标记语言(XACML)、XACL(XKMS)、SAML(XACML)和安全声明标记语言(SAML)。

本文加密/解密算法主要针对 Web Services 中的 XML 数据。Web Services 中的 XML 加密语法同其他场合的加密语法一样,其核心元素是 EncryptedData 元素,它是从 EncryptedType 抽象类型派生的。要加密的数据可以是任意数据、XML 文档、XML 元素或 XML 元素内容;加密数据的结果是一个包含或引用密码数据的 XML 加密元素。当加密元素或元素内容,EncryptedData 元素替换 XML 文档加密版本中的该元素或内容。当加密的是任意数据时,EncryptedData 元素可能成为新 XML 文档的根,或者可能成为一个子代元素。当加密整个 XML 文档时,EncryptedData 元素可能成为新文档的根。此外,EncryptedData不能是另一个 EncryptedData 元素的父代或子代元素。

使用 XML 加密,所有与密钥相关的问题划分成两个部分:

(1) 密钥的交换(非对称加密);

(2) 使用预先交换的密钥(对称加密)。

这样,用户可以交换密钥并在以后使用它们。一方将它的公钥发送给另一方;另一方使用这个公钥加密其密钥。

### 3.2　加密算法的设计目标

(1) XML 加密(XML Encryption)无意替换或取代 SSL/TLS。相反,它提供了用于 SSL 未涵盖的安全性需求的机制。以下是两个 SSL 未涉及的重要领域:加密交换数据的一部分;多方(不止两方)之间的安全会话。

(2) 使用 XML 加密,每一方都可以保持与任何通信方的安全或非安全状态。可以在同一文档中交换安全的和非安全的数据。

(3) XML 加密能够处理 XML 和非 XML(如二进制)数据。

### 3.3　XML 加密算法(类)的结构

XML 加密实现由一组类组成。XmlEncryption 类是其余类的封装器,用户只需要与这个类交互。在内部它使用其他类的功能,包括 XmlEncryption 封装器类,EncryptedData 类,EncryptionMethod 类,KeyInfo 类,CipherData 类。XmlEncryption 类包含各种公用 Get/Set 方法。用户将调用 Set 方法来指定加密参数,包括下列:

(1) 要加密的文件名称。

(2) 所产生的 XML 加密文件的名称。

(3) 加密算法的名称。

(4) 将用于加密的密钥名称。

(5) 用于 < EncryptedData > 结构标志的 ID。

### 3.4　XML 加密类的原理与实现

XML 加密类 XmlEncrypt 和解密类 XmlDecrypt 利用 DOM 接口完成 XML 数据分析、读取、存储等功能。原理图如图 1 所示。

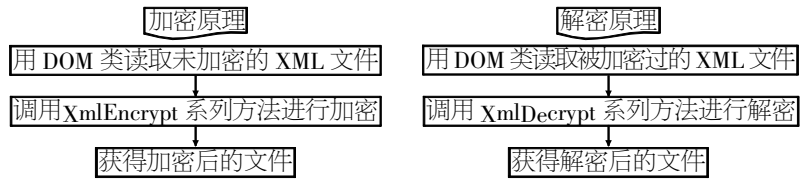



图 1　原理图

同时结合以下方法具体实现 XML 加密、解密:

(1) 加密类提供的主要接口方法和流程。以下是加密类对以上 XmlEncrypt 和 XmlDecrypt 内部方法作封装后暴露的主要接口和方法:

① void setAlgoName( String name) //设置要采用的加密算法
② void setEncKey( Key encKey) //设置密钥
③ void setEncID( String EncID)
//设置用于 < EncryptedData > 结构标志的 ID
④ void encryptCompleteXmlFile( ) //整个文件的加密
⑤ void encryptElementOfXmlFile( String strElement)
//用于加密 XML 文件中的特定元素
⑥ void encryptElementContentOfXmlFile( String strElement, String strContent) //用于加密 XML 文件中特定元素的内容
⑦ string getDecryptedData( ) //加密 XML 文件的解密

通过使用 XML 加密类的第⑤ ~⑦三种 XML 加密方式中的任何一种,就可以保护企业 XML 数据中的敏感信息:加密整个 XML 文件;加密 XML 文件中的元素;加密 XML 文件中元素的内容。其流程图如图 2 所示。

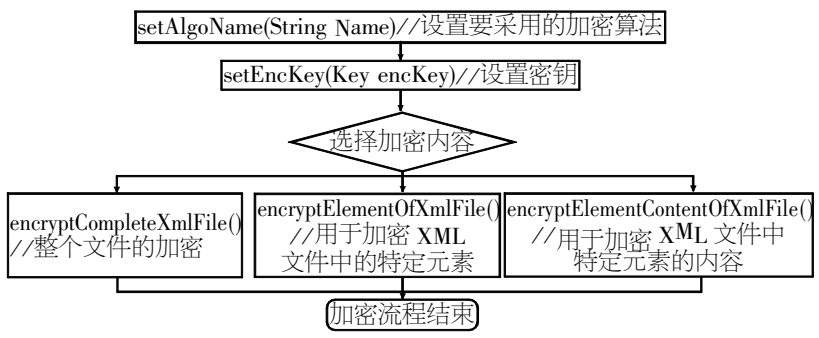


图 2 流程图

(2) 加密 XmlEncrypt。这个方法接收文本字符串,并使用 TripleDES 算法对它加密。对于对称加密,除了加密的明文数据外,还需要加密密钥,这个密钥可以用作为类的私有属性 encKey。输入文本字符串转换成了字节数组,并用它填充。对需要将明文数据字节放在八字节块的加密使用块密码算法,所以采用一些字符填充最后一个不完整的块。样本实现使用空格(" ")作为填充字符。接着创建 Cipher 类对象,并以加密方式对它初始化。随后将初始向量(IV)抽取到字节数组,对明文数据进行块加密,并将 IV 字节置于加密的数据字节之前。使用类的方法 getBase64Encoded 时,结果字节数组转换成了用 Base64 编码的字符串,随后被返回。实现的 Java 核心代码如下:

```
Public String xmlEncrypt ( String data) {
  String encData = new String( data) ;
  String cipherDataBase64 = null; //使用空格" "作为填充字符
  int pad = encData . length( ) % 8;
  for( int i = 0; i < ( 8 - pad) ; i + + )
    encData + = " "; //创建 Cipher 类对象
  Cipher cipherObj = Cipher. getInstance ( this. algoName + " /CBC/ NoPadding") ; //以加密方式对它初始化 Cipher 类对象
  cipherObj. init( Cipher. ENCRYPT_MODE, this. encKey) ;
  //将初始向量( IV) 抽取到字节数组
  byte [ ] iv = cipherObj. getIV( ) ; //对明文数据进行块加密
  byte[ ] cipherTemp = cipherObj. doFinal( encData . getBytes( ) ) ;
  //将 IV 字节置于加密的数据字节之前
  int ivlen = iv. length;
  int cTlen = cipherTemp. length;
  int cDlen = ivlen + cTlen;
  byte [ ] cipherData = new byte[ cDlen] ;
  for ( int i = 0; i < ivlen; i + + )
    cipherData[ i] = ( byte) iv[ i] ;
  for ( int i = ivlen; i < cDlen; i + + )
    cipherData[ i] = ( byte) cipherTemp[ i - 8] ;
  //字节数组转换成了用 Base64 编码的字符串
  cipherDataBase64 = getBase64 Encoded( cipherData) ;
  //结果返回
  return cipherDataBase64;
}
```

(3) 解密 XmlDecrypt。这个方法与 XmlEncrypt 方法相反。它接收用 Base64 编码的密码字节数组,并对它译码和解密以返回明文数据字符串。接着将 IV 与真正的密码字节数组分离。这样就用解密方式创建并初始化了 Cipher 类对象。接着对该密码字节数组解密,解密的字节数组作为文本字符串返回。实现的核心代码如下:

```
//接受用 Base64 编码的密码字节数组,并译码
Public String xmlDecrypt ( String encString) {
  byte[ ] g = getBase64Decoded( encString) ;
  int glen = g. length;
  //将 IV 与真正的密码字节数组分离
  byte [ ] iv = new byte[ 8] ;
  for( int t = 0; t < 8; t + + )
    iv[ t] = g[ t] ;
  byte [ ] Enc = new byte[ glen - 8] ;
  for( int p = 8; p < glen; p + + )
    Enc[ p - 8] = g[ p] ;
  String decString = null; //明文数据字符串变量
  //设置 IV
  IvParameterSpec ivps = new IvParameterSpec( iv) ;
  AlgorithmParameters aparam = AlgorithmParameters. getInstance( algo) ;
  aparam. init( ivps) ;
  //创建 Cipher 类对象
  Cipher cipherObj = Cipher. getInstance( algo +" /CBC/ NoPadding") ;
  //以解密方式初始化 Cipher 类对象
  cipherObj. init( Cipher. DECRYPT_MODE, decKey, aparam) ;
  //对该密码字节数组解密
  decString = new String( cipherObj. update( Enc) ) ;
  //结果返回
  return decString;
}
```

加密类目前已经对 TripleDES 块密码算法提供了支持。进行少量修改,还可以使用诸如 AES 等其他算法。

### 3.5 XML 加密类应用实例

#### 3.5.1 加密 XML 数据

首先实例化 XmlEncrypt 对象,然后调用 XmlEncrypt 的各种设置方法来设置如下属性:

(1) FileSource 和 FileResult——要加密的 XML 文件和加密后文件路径。

(2) encKey——用于加密的密钥。

(3) AlgoName——加密算法名称。

(4) keyName——加密密钥的名称。它是 KeyName 元素的子文本节点值。

(5) encId——文档中赋予 EncryptedData 标记唯一的名称。

根据调用应用程序时所传递的命令行参数,调用 XML 加密引擎(XmlEncrypt 类)的不同方法。依据 XML 加密规范,还可以对任意数据(如. jpg 图像或 Web 上几乎任何内容)加密,这几乎等同于整个 XML 文件的加密;唯一区别是上述 XML 加密语法的核心元素 EncryptedData 元素类型属性的值不同。另外,XML 加密规范列出了许多必需的和可选的密码术算法,包括块加密、流加密、密钥传送、密钥协议、对称密钥包、消息摘要、消息认证。另外,Base64 编码算法也是必需的,在可以将每个密码插入到 XML 文档之前,它们都必须用 Base64 编码。本加密类使用了 Cryptography 密码术供应商的 TripleDES 块加密算法,加密方式是块大小为八字节的 CBC(Cipher Block Chaining,密码块链接)。

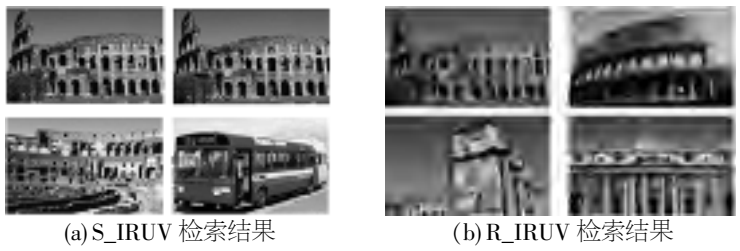
#### 3.5.2 解密加密过的 XML 数据

对于前面所述的所有三种加密方式, (下转第 148 页)

相对于图像库中与样本同类的图像数量而言的,正确率是相对于返回结果个数而言的。在我们的系统中,参与检索的图像特征包括: RGB 颜色空间的 8 ×8 ×8 级直方图, HSV 颜色空间的 8 ×3 ×3 级量化主颜色特征,灰度共生矩阵纹理特征和基于奇异值的图像全局特征。

表 1 检索检中率及正确率

| 图像 | 海滩图像 | 建筑图像 | 恐龙图像 |
|---|---|---|---|
| 检中率 | 64% | 68% | 100% |
| 正确率 | 57% | 62% | 70% |



(a) S_IRUV 检索结果 (b) R_IRUV 检索结果

图 2 古典建筑检索实例

在实验中,也发现了系统的一些缺点,最主要体现在对图像进行检索时,第一次检索出来的结果往往不尽如人意,虽然经过反馈后重新检索能够达到良好的效果,但是如果能够在第一次检索时就能有较高的命中率,这对于减少反馈次数是非常有利的。如何将具有较高首次检索命中率的算法和 Vague 集检索算法结合起来,这将是我们需要进一步研究的内容。

参考文献:


[1] N Gudivada, V V Raghavan. Content Based Image Retrieval Systems[J]. IEEE Computer, 1995, 28(9): 18-22.

[2] 苏中,马少平,张宏江. 基于内容图像检索的特征子空间抽取[J]. 软件学报, 2003, 14(2): 190-193.

[3] Zadeh L A. The Concept of a Linguistic Variable and Its Application to Approximate Reasoning[J]. Information Sciences, 1975, 8(3): 199-249.

[4] Zadeh L A. Fuzzy Sets[J]. Information and Control, 1965, 8(3): 338-353.

[5] 刘增良. 模糊技术与应用选编(1)[M]. 北京: 北京航空航天大学出版社, 1997.

[6] Gau W L, Buehrer D J. Vague Sets[J]. IEEE Transactions on Systems, Man, and Cybernetics, 1993, 23(2): 610-614.

[7] Bustince H, Burillo P. Vague Sets are Intuitionistic Fuzzy Sets[J]. Fuzzy Sets and Systems, 1996, 79: 403-405.

[8] Chen S M. Measures of Similarity Between Vague Sets and Between Elements[J]. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, 1997, 27(1): 153-158.

[9] Hong D H, Kim C. A Note on Similarity Measures Between Vague Sets and Between Elements[J]. Information Sciences, 1999, 135: 83-96.

[10] 李凡,徐章艳. Vague 集之间的相似度量[J]. 软件学报, 2001, 12(6): 922-926.

[11] Li Deng-feng, Cheng Chun-tian. New Similarity Measures of Intuitionistic Fuzzy Sets and Application to Pattern Recognition[J]. Pattern Recognition Letters, 2002, 23: 221-225.


作者简介:


易璨(1981-),男,湖南株洲人,硕士,主要研究方向为基于内容的图像和视频检索;石跃祥(1964-),男,湖南南县人,教授,博士,主要研究领域为模式识别与智能系统;肖晋,女,湖南长沙人,硕士,研究方向为数字图像处理。


(上接第 128 页)解密的方法都是 getDecryptedData。加密类的内部方法 simulateXml 将 XML 加密的文件序列化成文本字符串,并将它传递给 getDecryptedData 方法进行解密。getDecryptedData 方法标志 EncryptedData 标记,并抽取出用 Base64 编码的密码值。XML 加密标记中存在着解密需要的所有信息,即加密算法的名称、加密的数据类型以及加密密钥的名称。getDecryptedData 方法现在根据密钥和算法名称生成解密密钥,接着它将抽取出的用 Base64 编码的密码值、解密密钥以及算法名称传递给名为 Decrypt 的内部方法。Decrypt 内部方法将已译码和解密的数据作为文本字符串返回。依照 EncryptionData 元素的类型属性对该文本字符串操作,该类型属性可能是以下属性中的任意一个:

(1) 整个 XML 文件,作为新的 XML 文件保存到磁盘。

(2) 元素,替代 EncryptedData 标记以产生原始 XML 文件。

(3) 元素内容,替代 EncryptedData 标记以产生原始 XML 文件。

## 4 结论

本文针对 Web Service 中 XML 数据安全交换问题以及相应的应用程序设计,提出了一种保护 XML 数据的方法,经过实践证明(本方法已经使用于绍兴县政府综合资源信息库项目中,对导入、导出的机密文件进行加密,取得了良好的效果),采用 XML 加密具有简单性、开放性、灵活性、可扩充性等优点。除了上述的加密方法外,合理选择一些其他辅助的安全策略也是非常重要的,如进程访问令牌, XML 数据的有效期,数据交互点对点跟踪,适当采用防火墙等。Web Services 中 XML 数据安全的本质仍是计算机数据安全,由于数据采用了 XML 格式,因此我们可以结合此特点衍生出各种符合应用实际需求的安全加密技术, XML 数据安全不仅仅是一个技术问题,还是整个企业系统安全的重要组成部分。在一般情况下,灵活使用上述的加密方法与补充策略可以解决绝大多数安全方面的常见漏洞。

参考文献:


[1] 李力鸿. XML 文档信息的几种转换方法分析与应用实例[J]. 计算机科学, 2003, 1(30): 40-44.

[2] 陈博,许满武. 基于 Web 的计算模型——Web Service[J]. 计算机应用研究, 2003, 20(1): 41-42.

[3] 吴晨,王忠民. Web Service 中身份验证体系的研究与应用[J]. 计算机应用研究, 2003, 20(7): 80-82.

[4] Mark Bartel. XML-Signature Syntax and Processing[EB/OL]. http://www.w3.org/TR/xmldsig-core/, 2003-08.

[5] Joseph Reagle. XML Encryption Requirements[EB/OL]. http://www.w3.org/TR/xml-encryption-req, 2002-05.

[6] Merlin Hughes. Decryption Transform for XML Signature[EB/OL]. http://www.w3.org/TR/xmlenc-decrypt, 2003-12.

[7] Bilal Sodium. Exploring XML Encryption[EB/OL]. http://www.106.ibm.com/developerworks/xml/, 2003-04.


作者简介:


周杭霞(1963-),女,浙江杭州人,高级实验师,硕士,主要研究方向为网络应用软件;夏荣钊(1975-),男,湖北宜春人,工程师,硕士,研究方向为 XML 以及大型数据库;何利力(1952-),男,浙江杭州人,教授,硕士,研究方向为人工智能。