娱乐中的人工智能

主　编　潘云鹤　庄越挺
副主编　耿卫东　于金辉　吴江琴

浙江大学出版社

出版发行　浙江大学出版社
（杭州浙大路 38 号　邮政编码 310027）
（E-mail：zupress@mail.hz.zj.cn）
（网址：http://www.zjupress.com）
责任编辑　余健波　杜希武
排　　版　浙江大学出版社电脑排版中心
印　　刷　浙江大学印刷厂
开　　本　850mm×1168mm　1/16
印　　张　24
字　　数　629 千字
版 印 次　2004 年 9 月第 1 版　2004 年 9 月第 1 次印刷
书　　号　ISBN 7-900674-98-5/TP·34
定　　价　80.00 元

首届智能 CAD 与数字娱乐学术会议 (CIDE2004)

会议时间：

2004 年 10 月 19 日至 10 月 21 日

地点：

浙江 杭州

主办：

中国人工智能学会智能 CAD 与数字艺术专业委员会

中国图像图形学学会计算机动画与数字娱乐专业委员会 (筹)

承办：

浙江大学

协办：

浙江省计算机学会

会议主席：

潘云鹤 (浙江大学)

会议副主席：

戴国忠 (中科院软件所)　　陈　纯 (浙江大学)

程序委员会主席：

Harry Shum (微软亚洲研究院)　　庄越挺 (浙江大学)

程序委员会委员：

鲍虎军 (浙江大学)　　陈宏刚 (重庆宏信软件有限责任公司)

邓建明 (东南大学)　　郭百宁 (微软亚洲研究院)

黄铁军 (中国科学院研究生院)　　李　青 (香港城市大学)

廖祥忠 (北京广播学院)　　梅　宏 (北京大学)

潘志庚 (浙江大学)　　庞云阶 (吉林大学)

齐东旭 (澳门科技大学)　　史元春 (清华大学)

孙守迁 (浙江大学)　　谭群钊 (上海盛大网络发展有限公司)

王小松 (浙江大学)　　王阳生 (中科院自动化所)

王涌天 (北京理工大学)　　王兆其 (中科院计算所)

杨四亦 (北京广播学院)　　殷国富 (四川大学)

Gino Yu (香港理工大学)　　赵　阳 (中国美术学院)

周激流 (四川大学)　　朱淼良 (浙江大学)

组织委员会主席：

耿卫东 (浙江大学)　　于金辉 (浙江大学)

门素琴 (浙江大学)

组织委员会委员：

罗仕鉴 (浙江大学)　　陆系群 (浙江大学)

会议秘书：

吴江琴 (浙江大学)　　吴　飞 (浙江大学)

目　　录

序　言

近年来，计算机图形学和网络多媒体等技术的迅猛发展推动了智能 CAD、计算机动画、数字娱乐及数字艺术的深入研究和广泛应用。从《侏罗纪公园》、《泰坦尼克》、《特洛伊》等影片中，人们看到了用计算机图形技术生成的精彩无比的画面和令人震撼的特技效果；人们在闲暇之余，体验轻松愉快或刺激紧张的网络在线游戏。而现在的游戏软件无论在画面的效果还是在运行速度上，都是过去的单机游戏所无法比拟的。显然，数字娱乐作为新世纪一个引人瞩目的产业而在国内外掀起，并已经引起了各国政府的重视。

中外的计算机科学家、艺术家、影视创作者报以空前的热情，进行着更高起点和更高目标下的计算机艺术创造活动，软件、数字娱乐业界人士在为计算机创造的艺术成果而欣喜的同时，也更积极地研究开发相应的新的技术和新的产品。

数字娱乐是一个巨大的应用领域，但目前还只是处在初级阶段，有一系列重要的问题需要研究。从技术上来讲，它涉及到计算机图形学、人工智能、多媒体、人机交互、网络协同、智能动画等。由数字娱乐软件的应用中产生的需求，也促进了对上述领域研究的深入。这是一块肥沃的学术与产业可耕之地。

为了进一步促进我国在智能 CAD、计算机动画、数字娱乐及数字艺术等领域的研究与应用，召开一次以“智能 CAD 与数字娱乐”为主题的学术会议具有重大的现实意义。首届智能 CAD 与数字娱乐学术会议（CIDE2004）正是在这样的背景下召开的。

CIDE2004 由中国人工智能学会智能 CAD 与数字艺术专业委员会（筹）和中国图像图形学学会计算机动画与数字娱乐专业委员会（筹）联合主办，浙江大学承办，浙江省计算机学会协办。会议将为智能 CAD、数字娱乐及相关领域的学者提供一个交流最新研究成果、进行广泛学术讨论的平台。希望这次会议抛砖引玉，引起更多学者和业界人士的兴趣，最终促进中国数字娱乐产业的跨越式的大发展。

本次会议共选择收录代表性论文 63 篇，这些论文在很大程度上反映了我国在智能 CAD、计算机动画、数字娱乐及数字艺术等领域的最新研究成果和发展趋势。论文内容包括娱乐中的网络支撑环境和信息安全、娱乐中的多媒体技术、娱乐中的数字艺术、娱乐中的计算机动画、娱乐中的智能 CAD 、娱乐中的虚拟现实、娱乐中的图像处理、计算机游戏、游戏中的人工智能等理论研究和实际应用成果。

10 月的杭州，桂花飘香，气候宜人。愿各位代表在浙江大学度过愉快的学术时光。

祝贺这次会议取得圆满成功！

大会主席 

2004－9－12

一个基于 RDF/XML 的异构信息集成模型*

陈慧芳,邓建明

(东南大学计算机科学与工程系 0902 信箱,江苏,南京 210096)

摘要:元数据概念的提出和使用使得 web 上海量的信息有可能为机器所理解,而 RDF 为各种元数据格式之间的互操作提供了统一的资源描述框架。本文提出的基于 RDF/XML 的层次型异构信息集成模型,通过在信息中加入元数据描述的方式,利用 RDF 在元数据之间互操作方面的特性,能够有效地屏蔽各种信息源之间的异构性,提供了一种对异构信息的统一访问机制,为数据资源的共享和利用提供了新的可能解决途径。

关键词:资源描述框架,异构,元数据

A Heterogeneous Information Integration Model Based On RDF/XML

Chen Huifang

(*Computer Science & Engineering Department*, *Southeast University*, *Nanjing JiangSu*, 210096)

Abstract: The advancing and usage of the concept of metadata make it possible to understand information on web, and RDF provides a uniform resource description framework for interoperation between all kinds of metadata models. The heterogeneous information integration model based on RDF/XML introduced in this article can effectively shield the heterogeneity between different information resources and provide a uniform access method for them by adding metadata into information and characteristic of RDF in the way of interoperation of metadata. It provides a new possible method to information share.

Keywords: Resource Description Framework, heterogeneous, metadata

1 前 言

自上世纪 80 年代末 web 技术出现以来,它正以前所未有的影响力改变着世界。然而随着其应用领域研究的不断深入,它的一些缺陷也日渐显露[1]。主要来说,存在着两方面的问题:

一方面,web 内容无论是文本还是图片绝大多数只能由人浏览而不可以被机器所理解,从而导致人们很少能利用计算机对这些信息进行自动辨析和提取来提高对它们的访问效率。

另一方面,虽然 web 上浩如烟海的数据形成了一个巨大的信息库,给人们提供了丰富的资源,但是要在这个庞大的信息库中查找自己所需要的信息却变得越来越困难。这主要是由于 web 上各个系统的自治性导致了信息的异构性(这里的异构指的是数据模型、数据模式和

* 本研究受国家自然科学基金 60373067 资助

数据实例之间的异构),使得人们不能基于统一的机制来访问它们,从而影响了人们对这些信息的有效利用。

由于上述两方面问题的存在,人们无法对 web 上的海量信息进行充分地利用,从而造成了 web 资源的极大浪费。如果能将这些信息集成起来,通过某种机制屏蔽各种信息之间的异构性,那么将有助于人们对 web 信息访问效率的提高和对信息的充分利用。本文在分析现有的异构信息集成方法和其他相关技术的基础上,提出了一个基于 RDF/XML 的异构信息集成模型,并由此提供一种对异构信息的统一访问机制,从而屏蔽各种信息之间的异构性,使得集成后的异构数据对用户来说是无差异的。本文第 2 节就有关技术和方法作了简单的分析;第 3 节详细介绍模型的体系结构并对其进行了分析;第 4 节对全文进行了总结。

2 相关方法和技术

随着计算机网络的发展,数据资源共享成为许多应用的一个重要需求。因此许多研究者考虑了将不同数据资源集成起来的各种方法。大体上,这些传统的数据集成的方法可以分为下列两类[2]:

1. 数据仓库方法:事先将所有要处理的数据都集中到一个中心仓储 - 数据仓库中,然后提供对这个数据仓库的查询机制。用户提出的查询实际上是作用在数据仓库中的数据之上,由于集中式的存贮故有查询速度快的优点。缺点是当数据源的数据发生变化时数据仓库中的数据也要作相应修改,会带来一致性和可扩展性的问题。

2. 需求驱动的方法:数据仍存储在数据源,集成系统提供一个虚拟的集成视图以及对这个视图的查询机制。用户对集成视图提出查询时系统自动地将其转换为对各信息源的查询,然后动态地搜集数据。此方法的优点是减少了数据冗余并且能保证查到最新的数据,但由于是动态搜集,当重复查询某一资源时就会增加许多不必要的操作。

要注意的是,单纯地采用上述的方法还不能充分利用 web 信息,这主要是由于信息的机器不可理解性。元数据概念的提出和使用朝着机器可理解方向迈出了很重要的一步。因为它能够被用来描述信息资源的基本特征及其相互关系,利用对元数据的解析,计算机可以对该元数据标记的信息内容进行识别和理解。

虽然元数据的应用有助于实现信息资源的机器可理解性,从而给信息查找提供便利,但不同领域(甚至同一领域)存在多个不同的元数据格式,在不同的元数据格式描述的信息资源之间进行检索、资源描述和资源利用时就存在元数据的互操作性问题。因此 W3C 组织提出了 RDF (Resource Description Framework 资源描述框架)以解决这个问题。

RDF 采用 XML 语法,基本 RDF 模型由资源、属性和声明三种对象类型组成。资源是由 RDF 表达式描述的所有事物。属性用于描述资源的特征、属性和关系。而由资源、属性及属性的值构成一个 RDF 声明。RDF 本身并不直接定义具体元数据,而是定义元数据的基本描述模式。它通过 XML Namespace 机制调用已有的元数据集定义,独立于具体元数据格式,又可引用和集成多个元数据格式。同时又由于 RDF 本身采用 XML 语言标记,可在任何基于 XML 的系统平台上方便地进行解析,从而提供了统一的和机器可读的元数据标记和交换机制[3,4]。

文献[2]提出了一个基于 RDF 的异构集成模型,它采用的是将语义元数据与需求驱动的方法相结合的方式,当收到来自应用层的查询后由中介层解析此查询并分解为多个子查询,由于它采用动态方法搜集数据,因此它没有能够避免需求驱动方法的缺点。

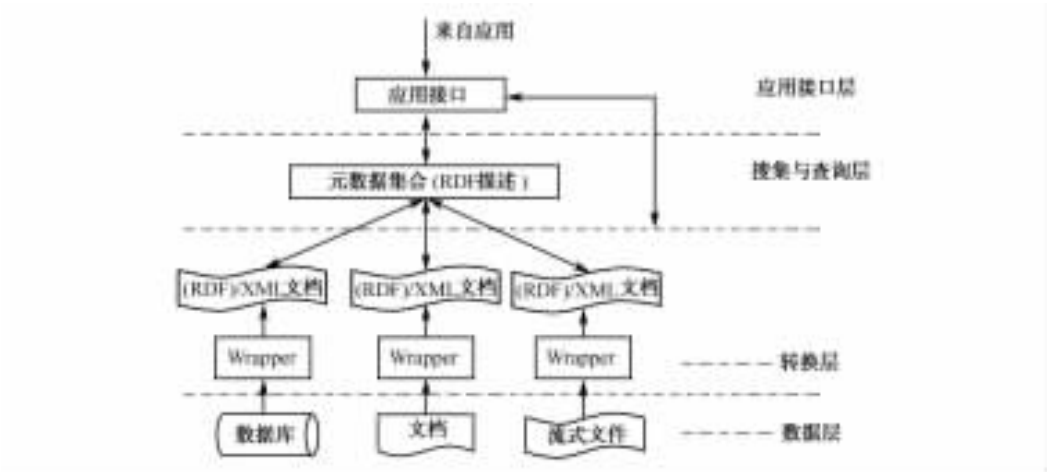
在异构集成方面也有研究者采用单纯基于 XML 的方法,它需要在数据模式不同的各个异构数据源之间建立统一的 XML 元模型,这在同一领域容易做到,但不同领域却比较困难。

3 基于 RDF 的异构信息集成模型

3.1 体系结构

本文提出的基于 RDF/XML 异构平台信息集成模型,通过在信息中加入元数据描述信息的手段,给信息查找提供可用的线索,它利用 RDF 提供的在各种元数据之间提供互操作方面的特性,可以屏蔽各个资源之间的异构性,为用户提供了对异构数据源的统一的访问机制。

如图 1 所示,模型采用层次型结构。之所以选择分层的结构,是因为这样会使每一层完成的功能都比较单一,有利于实现和维护;而且数据是随着层次的升高而逐渐抽象的,它们依次是数据源、XML 数据 + RDF 元数据、RDF 全局文档。模型由下到上共分为 4 个层次:数据层、转换层、搜集与查询层及应用接口层。数据层和转换层位于资源提供者这一方,而搜集与查询层和应用接口层位于集成服务器上。在下列各段中我们按从下到上的顺序分别讨论各层的功能。由于转换层和搜集与查询层是本模型的核心,我们将重点讨论这两部分。



3.2 数据层

数据层为整个模型提供各种类型的数据资源,其中包括关系型数据库、普通文本、流式文件等等,它们是模型要处理的数据信息的来源,分布于 web 上不同的站点。它们之间的异构包括了数据模型、数据模式、数据实例三方面的异构。例如,数据库数据与普通文本之间的异构属于数据模型的异构;在关系型数据库中描述同样的内容可以采用不同的模式,这是数据模式之间的异构;在 XML 数据中,同样的词有可能表示不同的意义,或者一个概念可能由不同的词来表示,这属于数据实例方面的异构。在这一层还没有对数据进行操作,它只是为整个模型的处理机制提供了多样性的数据源。

3.3 转换层

转换层转换由数据层提供的数据,向其上层提供 XML 数据和 RDF 描述的元数据,这一层要完成的工作交由资源提供者来做。这样做就能够提高资源提供者的资源被检索的几率和准确率,对资源提供者是有利的。

这一层首先要完成的是原有信息格式与 XML 格式之间的转换,为达到这个目标需要一些包装(wrapper)过程[5]。本模型并不规定采用何种具体的 XML Schema,资源提供者可以根据各自特定的信息采用其认为合适的格式。如果数据源本身已经是 XML 格式,则转换过程可以省略。

在转换的同时,要在 XML 文档中以附加元素的形式来加入元数据,此元数据采用 RDF

来描述，在编辑期间用 XML 编辑器来插入。为简单起见，本模型直接将 RDF 描述作为 XML 文档的根元素的第一个子元素。比如为了表示数据源的作者的信息，加入下面的一段文档。在这个例子中，我们采用了 DC 元数据（以类似图书馆卡片目录的方式来定义资源的元数据集）和 VCARD（定义个人信息的元数据集）元数据[6]。

```
<?xml version="1.0"?>
<xmlrootelement>
<rdf:RDF xmlns:rdf="http://www.
        w3.org/1999/02/22-rdf-
        syntax-ns#"
        xmlns:dc="http://purl.
        org/dc/elements/1.0/"
        xmlns:vcard="http://
        www.imc.org/vcard/3.0/"
        >
<rdf:Description rdf:about="http://
www.sjtu.edu.cn/mydoc.htm">
  <dc:creator>
    <rdf:Description>
      <vcard:fn>John</vcard:fn>
      <vcard:email>John@163.com
    </vcard:email>
    </rdf:Description>
  </dc:creator>
</rdf:Description>
</rdf:RDF>
<!—other content—>
</xmlrootelement>
```

实际上，在数据文件中加入元数据描述信息的思想有点类似于传统搜索引擎中人工分类的思想，基于目前的技术现状这一工作交由资源提供者来做，因为在一般情况下资源提供者对自己的信息更为了解，所加入的元信息就更准确一些。计算机自动加入元数据信息功能的实现还依赖于 web 信息自动抽取技术的进一步完善。

在本模型中，不同资源的元数据描述是不一样的，而如果要定义一种元数据集，包括所有种类的资源，不仅工作量大，而且即使定义出来也并不能保证被广大用户所采纳，故此方法在目前还是不现实的。更何况许多领域（如图书馆）已经有比较成熟的元数据标准，所以本模型采用的是现有的元数据标准。而对于这些种类繁多的元数据集模型采用 RDF 来融合它们。

RDF 能实现各元数据之间的交互主要要归功于它的两大关键支撑技术 - URI 和 XML。由于元数据集也是一种资源，因此它可以用 URI 来标识。这样，在用 RDF 描述资源的时候，就可以使用各种元数据集，只要用 URI 指明它们即可使用。不同元数据集之间的交互通过 RDF Schema[7] 来完成。具体操作时，先使用 XML 语法指定元数据集的 URI（视具体需要可以是一个或多个），再使用指定的元数据集来描述资源，最后使用 RDF Schema 来描述元数据集之间的关系。例如下述描述

```
<rdfs:Property rdf:resource="http://
mymetadata.vocab.org/Author">
<rdfs:subPropertyOf
rdf: resource = http://purlorg/dc/ele-
ments/1.0/Creator/>
</rdf:Property>
```

表示了某个组织自己定义的元数据 Author 是 Dublin Core 的元数据 Creator 的特殊形式。通过诸如此类的 RDF Schema 定义来完成不同元数据集之间的沟通和交互。

3.4 搜集与查询层

该层的主要功能是抽取各个自治的资源站点上的元数据信息，然后调整、归纳这些信息，最终形成一个全局的 RDF 文档，这个 RDF 文档实际上是从各个自治资源上搜集来的元数据信息的汇总。虽然各个资源上可能采用了不同的元数据标准，但由于都采用了 RDF 语法格式来描述，可以运用统一的方法来访问它们，同时也使得不同的元数据标准可以直接交互。此外，这一层还承担查询功能，直接对 RDF 文档中的元数据进行查询。

对于抽取元数据，本模型采用 DOM（文档对象模型）方法[8]。DOM 是以树为基础的

API,能够把 XML 文件转换成一个定制的树状结构,所有后续操作都在这个"树化"后的 XML 文件上执行,处理起来比较直观。具体的处理过程是:首先将要分析的文档解析成 DOM 树;然后定位,即在此 DOM 树中找到用 RDF 描述的元数据内容,方法是遍历 DOM 树,判断节点的名称,如果是"rdf:RDF"则定位过程结束,开始抽取;抽取过程是遍历整个"rdf:RDF"子树的过程,并通过此遍历过程获得各个节点的信息。两个遍历过程采用的都是递归的方法。

元数据的抽取工作由 agent 来完成,模型主控程序定期指派 agent 到各个资源站点上抽取和搜集元数据信息。由于模型采用的是类似于数据仓库的方法,如果全局 RDF 视图中的元数据信息不定期更新的话,就有可能导致数据不一致的情况,所以定期的更新工作是必要的。但是该模型又不同于传统的数据仓库方法,它并不把所有的数据都搜集起来,而只是搜集它的元数据信息,这样在更新的时候也只需要更新元数据信息,减少了工作量。

完成了元数据的抽取工作后,紧接着就是将从各个资源返回的元数据汇总到一起形成一个集合,这个集合中包括了所有抽取和搜集到的资源的元数据信息。汇总要做的是完成一些结点的调整工作,合并相同的结点,删除多余的结点,同时调整文档的结构,形成最终的 RDF 文档。

在上述工作的基础上,此层还要提供对 RDF 文档的查询。这里我们已经把对原始信息的查询转换为对其元数据的查询,因为该 RDF 文档中存放的都是元数据信息。虽然用户关心的是对原始数据查询,但由于元数据提供了数据最一般最本质的特征,因此将对原信息的查询转换为对其元数据的查询的方法是可行的。一个在数据中很少出现的关键词是不能被作为其元数据描述的,这一点又与计算机自动标引的关键词技术的原理有共通之处,而且这里的元数据是资源提供者自己提供的,相比之下准确率更高。

模型采用 Versa 作为查询 RDF 文档的语言。因为它能被集成到 XML 语言中,能提供聚合、子串匹配等的操作[9],这些功能都是其他语言所不具备的。

3.5 应用接口层

应用接口层是模型与用户应用的接口。它为各种应用程序,如 web service、一般的浏览器、决策程序等提供接口。当用户要查询某一局部数据源时,应用接口层将这一请求移交到其下层,由下层来处理查询,接着根据其下层返回的查询结果信息,并使用 HTTP GET 方法来直接访问满足要求的 URI 关联的地址(如图 1 中折线所示),从而得到所需的内容。这样对于用户来说所有的数据都可以通过统一的方式来访问。

3.6 模型分析

如上所述,本文提出的这个异构信息模型把对原始信息的分析、查询等动作转换为对元数据的操作。由于元数据本身具有机器可理解性,计算机可对元数据标记的内容进行自动的识别和理解,自动的对信息进行筛选,并提取其中所需的元素。

另外,由于 RDF 是为了屏蔽元数据的异构性,增强不同元数据间的互操作性而设计的。本文为模型引进的 RDF 机制,可以很好的屏蔽元数据之间的异构性,为用户提供了统一的全局视图。

本文提出的这个模型具有与数据仓库方法同样的查询速度快的优点,但它与传统的数据仓库的方法也有明显的不同,主要表现在以下两个方面:

(1) 该模型并不将要集成的数据源中的所有数据都搜集到中心仓储中,而是搜集它们的元数据信息,减少了所收集的数据的量,也减少了数据的冗余程度。并且由于数据量的大幅减少,也减少了完成一次更新所需的时间;

(2) 模型把对数据源的查询转换为对其元数据的查询,提高了查询的效率。另外由于这些元数据信息是资源提供者自己提供的,比通过计算机自动标引的方法更准确,因而也就提高了查询的准确率。

这个模型有效的利用了 RDF 在元数据交互方面的特性，与单纯的基于 XML 的集成方法比较，并不需要统一的 XML 模式定义就能够屏蔽各种数据源的异构性，提供了对各种不同数据源的统一访问机制。

4 结　论

异构数据源集成不是一个新的问题，但随着相关技术的不断发展，也给解决该问题提供了实施新方法的可能性。本文提出的模型充分利用了 RDF 和 XML 开放性、易于交互的特点，通过在信息中加入元数据描述的方式和 RDF 在元数据之间互操作方面的特性，采用将 RDF 元数据与数据仓库方法相结合的方法，有效地屏蔽了各种信息源的异构性，为用户提供了统一的访问异构信息源的机制。利用这种机制，用户不需考虑信息的具体模式就能够有效地访问 web 信息，从而为数据资源的共享和利用提供了新的可能的解决途径。

参考文献

[1] Frank P. Coyle 著，袁勤勇，莫青等译，《XML、Web 服务和数据革命》，清华大学出版社，2003 年 3 月第一版

[2] Richard Vdovjak，Geert-Jan Houben：RDF Based Architecture for Semantic Integration of Heterogeneous Information Sources，http://www.informatik.uni-trier.de/~ley/db/indices/a-tree/h/Houben:Geert=Jan.html

[3] Resource Description Framework (RDF) Concepts and Abstract Syntax，http://www.w3.org/TR/rdf-concepts/2003-12-10

[4] RDF Primer，http://www.w3.org/TR/rdf-primer/

[5] Peter Patel-Schneider，Jerome Simeon：The Yin/Yang Web：XML Syntax and RDF Semantics，http://www2002.org/CDROM/refereed/231/

[6] 张晓林主编，《元数据研究与应用》，北京图书馆出版社，2002 年 5 月第一版

[7] 张维明主编，肖卫东，黄凯歌，徐振宇等编著，《语义信息模型及应用》，电子工业出版社，2002 年 3 月第一版

[8] Chuch White，Lianm Quin，Linda Burman 著，《XML 从入门到精通 :黄金版 (美)》，电子工业出版社，2002

[9] Thinking XML :使用 RDF 开始知识管理，第六部分 :使用 Versa 的 RDF 查询 http://www-900.cn.ibm.com/developerWorks/cn/xml/rdf/part12/

一种基于扑克牌的手工序列密码*

张玉安,冯登国

(信息安全国家重点实验室(中国科学院研究生院),北京 100039)

Email:zyazs@163.com

摘要 我们设计了一种基于两副扑克牌的手工序列密码算法,命名为 Twopokers。为确保手工可操作性,算法的加解密过程比较简单。但是,算法的强度并不弱。该算法拥有较大的密钥空间,通过频繁变换扑克牌(密钥符)的位置产生很长的滚动密钥序列。初步分析表明,由 Twopokers 产生的滚动密钥序列拥有相当好的抗攻击性能。该算法容易软件实现。

关键词 手工序列密码,扑克牌加密算法,伪随机序列发生器,安全性分析

A Playing-Card-Based Manual Stream Cipher

Zhang Yu-an and Feng Deng-guo

State Key Laboratory of Information Security

(*Graduate School of Chinese Academy of Science*), *Beijing* 100039

Email: *zyazs@163.com*

Abstract: Twopokers, a manually operated cipher scheme, is a playing-card-based stream cipher. To ensure the operations be carried out manually, the encryption process is not complicated. However, it is not weak on security. It has a vast amount of key space. A long key stream sequence is generated by the frequent transposition of the cards which are pre-arranged by the key,. Our elementary cryptanalysis shows that the key stream sequence has pretty good features on resisting some general cryptanalysis attacks. Now Twopokers has been easily software-implemented.

Keywords: Manual Encryption, Playing-Card-Based Cipher Scheme, Pseudorandom Sequence Generator, Security Cryptanalysis

手工密码(包括一些用非专用器械完成加解密的器械密码)的操作一般比较简单,这是由手工可操作性决定的,这导致了(当算法泄露情况下)手工密码算法一般都很脆弱。因此,对密码设计者来说,设计真正可手工操作的安全加密算法并不比设计电子密码容易。

Solitaire[1]加密算法是著名的密码学家 Bruce Schneier 1998 年设计的一个序列密码(1999 年 5 月发表了 Solitaire 1.2 版),它可以用一副扑克牌实现手工加脱密,给那些不方便使用电子或机械密码产品而又想安全通信的人使用。Paul Crowley 受 Solitaire 算法的启发,也

* 国家杰出青年科学基金资助(60025205),信息安全国家重点实验室(中国科学院研究生院)开放课题基金资助,973 资助项目(G1999035802)。

设计了可以用一副扑克牌实现手工加密的密码算法 Mirdek[2]。

笔者了解了 Solitaire 和 Mirdek 加密算法后，感觉这种牌戏式的算法很有趣儿，因此，我们也设计了一种颇具独创性的扑克牌加密算法。由于该算法由两副扑克实现，故将其命名为 Twopokers。

1 算法描述

取两副扑克牌，将大、小王除外的每张牌与 1～26 之间的一个数字对应，亦即每张牌与 A～Z 之间的一个字母对应，对应关系见下表：

梅花或红桃牌	*A*	2	3	4	5	6	7	8	9	10	*J*	*Q*	*K*
对应的字母	A	B	C	D	E	F	G	H	I	J	K	L	M
对应的数字	1	2	3	4	5	6	7	8	9	10	11	12	13

方块或黑桃牌	*A*	2	3	4	5	6	7	8	9	10	*J*	*Q*	*K*
对应的字母	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
对应的数字	14	15	16	17	18	19	20	21	22	23	24	25	26

另外，本文约定牌点是指 *A* 为 1 点、2 为 2 点、…、*K* 为 13 点。红牌是指方块和红桃，黑牌是指梅花和黑桃。在以下的所有操作中，总是牌面朝上。

生成密钥流序列的具体过程分为以下几步：

1) 取出第一副牌，先将大、小王拿出，然后将 52 张牌按照密钥指定的序排好后放在桌面的左侧，本文将这一堆牌称为候选牌组。将第二副牌的大、小王拿出，将剩下的 52 张牌也按照密钥指定的序进行排列。

2) 将第二副牌的前 13 张顺序取出，以平铺形式排列在桌面正中。这 13 张牌被称为明手牌。

3) 查看桌面上明手牌中部分牌张的牌点。设第一张牌的牌点是 n，则在剩下的 39 张牌中将一个大王插入在第 n 张之后。设第十三张牌的牌点是 m，则将另一个大王插入在倒数第 m 张之前。即第一个大王前边有 n 张牌，第二个大王后边有 m 张牌。接下来，设明手第二张牌的牌点是 p，则将一个小王插入在第 p 张之后。设第十二张牌的牌点是 q，则将另一个小王插入在倒数第 q 张之前。从而，使这组牌的张数达到 43 张，且第一个小王前边有 p 张牌，第二个小王后边有 q 张牌。本文把这 43 张牌称为控制牌组。

4) 将控制牌组 (43 张牌) 拿在手上。从手上开始一张一张地按顺序发牌。

a. 如果所发的牌不是大王或小王，设牌点是 n，则将桌面上明手牌的第 n 张牌拿到桌面的右侧牌堆上 (这就输出了一个滚动密钥符，即一个 1 — 26 的数)，从左侧候选牌中拿一张牌补到明手 (第 n 个位置) 出现的空位上，如果补牌后发现桌上的 13 张牌中出现两张相同的牌，则把相同牌中早先在桌上的那一张 (亦即跟当前第 n 张牌相同的那一张牌) 与刚刚拿到右侧的那张牌交换 (注意 :牌被换回来了，但刚才的输出值不变)。

b. 如果所发的牌是大王，则把大王放到桌面右侧的牌堆上 (把大王放到输出牌之中，但不产生滚动密钥符)，并将明手 13 张牌首尾调换，即第 1 与第 13、第 2 与第 12、第 3 与第 11、……、第 6 与第 8 交换位置。

c. 如果所发的牌是小王，则把小王放到桌面右侧的牌堆上 (把小王放到输出牌之中，与大王一样也不产生滚动密钥符)，并将明手 13 张牌相邻两两换位，即第 1 与第 2、第 3 与第 4、……、第 11 与第 12 交换位置。

5) 发完 43 张牌后，产生了 39 个 1 至 26 间的数。把已经用了一遍的、倒了序的原控制牌组的 39 张牌 (4 个王已被拿出) 重新拿到手上，用它改造桌面右侧的 (43 张) 输出牌。从手上按顺序一张一张地发牌，每发一张，也从 (43 张) 输出牌中按顺序确定一张不是王的牌 (遇到王时把王跳过) 与之对应。该对应牌与手上发的牌比色。如果同是红牌或黑牌就将这张对应牌与它后边第 n 张牌交换 (约定第 43 张后边接第 1 张牌)，否则，将这张牌与它前边第 n 张牌交换 (约定第 1 张前边是第 43 张牌)，其中 n 是手上所发牌的牌点。

6) 做完 39 次比色和换位后，把刚才所发

的、又倒了一遍序的原控制牌组的 39 张牌依序补接在桌面左侧剩下的 13 候选牌之后，构成新的候选牌组。把被换了位的 43 张输出牌拿到手上，用它作为新的控制牌组。

7) 重复第 4) 至 6) 步，每次产生 39 个 1 至 26 间的数。最终产生相当长的输出序列。

附录 1 中给出了一个例子。

如果用 Twopokers 算法输出的滚动密钥序列加密由 26 个字母构成的报文，加解密可采用模 26 加减法 (约定 0 和 26 都对应字母 z)，即

密文 = 明文 + 滚动密钥符　　mod 26

明文 = 密文 − 滚动密钥符　　mod 26。

需注意的是加密完成后，一定要把牌彻底洗乱，否则，会被人反推出明文和密钥，因为当攻击者有了某一时刻的牌序状态情况下可以通过猜设明文由后往前逆推出滚动密钥序列。

2　密钥设置与安全性分析

Twopokers 的密钥是去除王情况下两副 (各 52 张) 牌的排列序，变化量高达 $(52!)^2 \approx 6.5 \times 10^{135}$。不过，这样的密钥设置会导致 Twopokers 有下述三个小小的缺陷：

1) 产生前 39 个滚动密钥符时，密钥的变化量远远小于 $(52!)^2$，因为控制牌组中起作用的仅仅是牌点，与花色无关。也就是说，在控制牌组中把点数相同的两张牌换位后至少不会影响前 39 个滚动密钥符的值。

2) 当两组密钥非常相似时，在起初的若干拍内，表现在滚动密钥序列上的差异可能非常小。即算法易受相关密钥攻击。

3) 密钥不宜分离设置。比如，将候选牌组的编排顺序 (变化量为 $52!$) 作为基本密钥 (Base Key)，用户一次选定后长时间固定不变，将用于预置明手牌和控制牌组的 52！种变化作为报文密钥 (Message Key)，当攻击者拥有数万条滚动密钥序列情况下 (只需每条序列的前 20 个左右滚动密钥符)，可以区分出候选牌组，进而，可以求出全部密钥。当然，作为手工加密，攻击者一般不可能积累到如此多的已知明文。

Twopokers 的输出序列具有相当平衡的统计特性。尽管状态变化是非线性的，由于当占有密钥情况下，滚动密钥序列的生成过程可以由后往前逆推，说明该算法的非线性状态变换是非奇异的，进而该算法的输出序列不容易进入较小的反复周期，其平均长度理论上约为 $(52!)^2/2 \approx 3.3 \times 10^{135}$。

在抗攻击方面，Twopokers 有相当好的性能，比如在抗相关分析、分割分析和统计分析等方面均有较好性能。目前，由于它的密钥空间比较巨大，相关密钥攻击远缺乏适用性。因此，我们还没有找到由滚动密钥序列求取控制牌组或候选牌组的可行途径。

在当前的实际应用中，序列电子密码的用户密钥多为 256 或 128 比特。例如在 RC4 算法中，256 元置换 S 盒由 128 比特密钥生成，使 256！种 S 盒中的极少一部分 S 盒有机会被选用。如果 Twopokers 也采用类似 RC4 的密钥生成方案，比如通过变换，使 2^{256} 或 2^{128} 种密钥态与两副扑克牌的部分排列序相对应 (附录 2 中例示了一种由 256 比特密钥产生两副扑克牌序的方法)，这通常可以很好地消除对算法的相关密钥攻击，因为用户密钥的 1 比特改变将大大影响两副扑克牌的排列序。但是，添加了这类复杂的密钥转换逻辑后，不便手工操作，这将违背扑克牌密码算法的设计初衷。

Twopokers 算法的核心逻辑可以被看作是搅拌器型带记忆逻辑的巧妙运用和推广。该算法的真正意义并不在于它是一个实用的加密算法。它给出了一种仅仅依赖换位变换而产生大周期伪随机序列的简便方法。

3　结束语

第一和第二次世界大战期间，密码破译改变或影响了一些战役的结果和进程，进而它影响了人类历史的发展进程。然而，当时的很多密码破译成果来自于手工密码。随着微电子技术和电子计算机应用浪潮的迅猛到来，手工加密时代只能成为前辈们的一种美好回忆和遐想。随着互联网和笔记本、掌上电脑等电子产

品的广泛应用,对身居它乡的间谍们来说,手工加密时代将永久地成为历史。

Twopokers 算法不仅仅是个手工序列密码,它也易于在计算机上通过软件编程实现。如果对 Twopokers 稍作拓宽,将每副扑克 54 张牌改为 4N + 2 张,取 N = 8、16、32、64 或 128 等形如 2 的方幂的数字时,比较适合二进制操作习惯,这时,每种花色的牌点由 1—13 改为 1—N,将明手牌 13 张改为 N 张,候选牌组由 52 张改为 4N 张,控制牌组由 43 张改为 3N + 4 张,输出值由 1—26 变为 1—2N 或 0—2N - 1,这样,就产生了一个安全性相当不错的仿 Twopokers 序列电子密码(实际应用中若从字符输出中每次只选用 1 比特,安全性会更好)。将 Twopokers 软件编程实现时,最初生产的 3N 个滚动密钥符最好弃而不用,因为这时无需吝惜是否增加了一点点预备性操作。

Twopokers 是一个富有创意的加密算法。它与传统的移位寄存器序列密码的设计思想截然不同。狭义地看,它没有乱源序列发生器,也没有线性或非线性合成函数,但是它能产生非常好的伪随机序列。该算法的设计思想可能对现代序列密码算法设计具有借鉴作用。目前,我们还不能准确地评估这类算法做为电子密码时的硬件实现效率,因为这将取决于算法实现方案的可行性、资源的可用性和工程实现人员的实际操作技能等多个方面。

参考文献

[1] Bruce Schneier, "The Solitaire Encryption Algorithm"; Available from http://packetstormsecurity.nl/crypt/misc/solitaire.html。

[2] Paul Crowley, "Mirdek: a card cipher inspired by Solitaire", Available from http://www.ciphergoth.org/crypto/mirdek/description.html。

附录 1 滚动密钥序列生成过程举例

为了便于书写,我们用 S 表示黑桃、H 表示红桃、D 表示方块、C 表示梅花,例如黑桃 A 写为 SA,、红桃 10 写为 H10,梅花 Q 写为 CQ 等等。将大王写为 Ja,小王写为 Jb。设密钥确定的 52 张候选牌组的牌序为:

D2,S10,H10,C9,SK,H2,SJ,HJ,D10,H8,H9,D7,S5,
H3,C6,S6,S7,H5,S8,C3,S2,DJ,DQ,S3,S4,HK,
SA,SQ,DK,CQ,H4,CK,H6,D8,C7,D5,C4,C8,D3,
CA,HA,C10,CJ,D4,D9,C2,D6,DA,C5,S9,H7,HQ

设密钥确定的第二副牌 52 张的牌序为:

S3,HQ,S7,DK,DJ,C4,H10,H8,D7,C6,H5,DQ,H9,
S10,C2,CA,H3,S8,S6,H6,S4,H4,S5,C10,C9,D10,
S2,SA,CK,D5,CQ,H2,D4,CJ,D9,D6,D3,D8,HA,
SK,D2,DA,SQ,SJ,HJ,C8,C7,H7,C5,S9,C3,HK

加入大、小王后的控制牌组为:

S10,C2,CA,**Ja**,H3,S8,S6,H6,S4,H4,S5,C10,**Jb**,C9,
D10,S2,SA,CK,D5,CQ,H2,D4,CJ,D9,D6,D3,D8,
HA,SK,D2,**Jb**,DA,SQ,**Ja**,SJ,HJ,C8,C7,H7,C5,S9,
C3,HK

产生滚动密钥序列的过程如下表一:

输出了 39 个滚动密钥符后,43 张输出牌(由上到下)的序为:

DK, H5, DQ, S5, D8, S4, S7, H4, C3, Ja, S2, SQ,
Jb, S8, S6, C6, D10, S3, H8, D7, S10, C9, H3, SJ,
S7, H10, DQ, DK, DJ, H9, Jb, H9, D7, HJ, D2,
H2, H8, C4, H5, Ja, H10, HQ, C6

接下来对上述输出牌比色换位。首先 HK 与 DK 同色,DK 与其后的第 13 张牌 S8 互换。C3 与 H5 不同色,H5 与其前的第 3 张牌 HQ 互换。S9 与 DQ 不同色,DQ 与其前的第 9 张牌 H8 互换。C5 与 S5 同色,S5 与其后的第 5 张牌 C3 互换。H7 与 D8 同色,D8 与其后的第 7 张牌 SQ 互换。C7 与 S4 同色,S4 与其后的第 7 张牌 JB 互换。C8 与 S7 同色,S7 与其后的第 8 张牌 S6 互换。HJ 与 H4 同色,H4 与其后的第 11 张牌 H8 互换。SJ 与 S5 同色,S5 与其后的第 11 张牌 D7 互换。SQ 与 JA 不比较,比下一张,SQ 与 S2 同色,S2 与其后的第 12 张牌 H3 互换……。

附录 2 密钥扩展方案举例

下面给出一种由 256 比特(32 字节)密钥产生两组 H = 4N 元排列序的方案,其中 N 通常可