

# 普适计算中一种上下文集成方法

刘莉<sup>1</sup>, 黄海平<sup>2, 3, 4</sup>, 王汝传<sup>2, 3, 4</sup>, 蔡启旺<sup>2</sup>

(1. 南京人口管理干部学院 信息科学系, 江苏 南京 210042;

2. 南京邮电大学 计算机学院, 江苏 南京 210003;

3. 江苏省无线传感网高技术研究重点实验室, 江苏 南京 210003;

4. 宽带无线通信与传感网技术教育部重点实验室, 江苏 南京 210003)

**摘要:**普适计算作为一种全新的计算模式,目的是根据用户需要提供随时随地的服务。普适计算环境中存在大量异构的数据源,不利于上下文信息的处理和访问。文中提出了一种基于XML的上下文集成方法,在保证上下文精度和新鲜度的条件下,尽量减少上下文信息存储量,并向上提供统一的上下文表达形式,有效屏蔽了上下文信息的异构。实验测试表明,采用的文件内存保留和分片转存方法,进一步加快上下文的集成,从而有效地支持普适服务。

**关键词:**普适计算;上下文;可扩展标记语言;上下文集成

**中图分类号:**TP31

**文献标识码:**A

**文章编号:**1673-629X(2012)06-0130-05

## Context Integration Scheme in Pervasive Computing

LIU Li<sup>1</sup>, HUANG Hai-ping<sup>2, 3, 4</sup>, WANG Ru-chuan<sup>2, 3, 4</sup>, CAI Qi-wang<sup>2</sup>

(1. Dept. of Information Science, Nanjing College for Population Programme Management, Nanjing 210042, China;

2. College of Computer, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

3. Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing 210003, China;

4. Key Lab of Broadband Wireless Communication and Sensor Network Technology of Ministry of Education, Nanjing 210003, China)

**Abstract:** As a novel computing mode, pervasive computing provides services anytime, anywhere according to users' demands. In this environment, many heterogeneous data sources affect the efficiency of context processing and accessing. It presents a XML-based context integration method, which can reduce the amount of context storage, shield the heterogeneity, and provide a unified expression of context information on the premise of ensuring the precision and freshness. Furthermore, test results show that this method can speed up the integration by retaining a copy of document in memory and using file-partition strategy, and enable pervasive computing services more efficiently.

**Key words:** pervasive computing; context; XML; context integration

## 0 引言

普适计算(Ubiquitous Computing或Pervasive Computing)是建立一个充满计算机和通信能力的环境,使这个环境与人逐渐融合在一起,在其中,人们可以随时随地、透明地获得数字化服务<sup>[1]</sup>。为实现这样的服务,普适计算系统需要不断获取上下文信息。所

谓“上下文”,是所有能够描述用户与应用程序之间交互所涉及的实体(包括人、位置、物体等)状态的信息,其中包括用户和应用程序自身的信息<sup>[2]</sup>。通常,普适环境采集到的上下文,在种类、数据格式和精度等方面存在异构性,也可能存在大量冗余信息,如果直接交给上层应用处理,会造成很大的系统开销,也不利于实现普适计算的实时性服务<sup>[3]</sup>。如何减少上下文信息存储量又不失上下文的意义,且能实现透明访问上下文数据?解决此问题的关键是将各种异构上下文集成,提供统一的访问接口,使得应用程序从具体的操作细节中释放,仅专注于应用程序的逻辑。

## 1 研究现状及相关工作

以普适计算环境中的传感器网络为例,传感器节

收稿日期:2011-11-04;修回日期:2012-02-07

基金项目:国家自然科学基金(60973139,61003039);江苏省科技支撑计划(工业)项目(BE2010197, BE2010198);江苏省高校自然科学基金基础研究项目(10KJB520013, 10KJB520014);江苏省计算机信息处理技术重点实验室基金(KJS1022)

作者简介:刘莉(1980-),女,讲师,主要研究方向是无线传感器网络和普适计算技术;王汝传,教授,博士生导师,主要研究方向是计算机软件、计算机网络、信息安全、Agent和虚拟现实技术等。

点不断地采集,如环境上下文(温度、气压、噪声等)、音/视频多媒体上下文、计算上下文(网络带宽、底层操作系统和硬件特性等)。这些异构上下文的大量涌现,增加了上下文的处理和访问难度。现有的一些技术,能对上下文进行精简,并用统一形式表达异构上下文或提供集成的访问接口,从而增强访问的透明性和效率。例如数据集成技术,它将多个相关联的分布式异构数据源集成到一起,使得用户以一种透明的方式统一访问这些数据源<sup>[4]</sup>,这为处理普适计算环境中的上下文提供了思路。目前,数据集成技术主要包括数据仓库技术、联邦数据库技术和中间件技术等。

基于数据仓库(Data Warehouse)的集成技术<sup>[5,6]</sup>按照条件将数据过滤后,以一个集中和统一的视图预先存储到同一数据库中。这种方法虽然可以减少对异构数据源的访问量,但数据在存储之前要经过一定的筛选,更新不及时,不适合提供普适计算的即时服务。

联邦数据库可以集成各种结构化数据和非结构化数据<sup>[7]</sup>。它系统结构简单,但所有成员数据库都添加彼此访问的接口,编程量大,因此不能适应普适计算中复杂多样的上下文数据。

中间件技术适用于数据源量大、结构多样、经常更新,且不能预知用户需求的查询情况。但其缺陷是对于动态增加数据源非常困难。还有许多中间件采用了基于Agent(本体)的结构,例如Agilla<sup>[8]</sup>,利用Agent之间的协作增加了数据融合和集成的灵活性,但实现难度大。

如果采用基于XML(eXtensible Markup Language,可扩展标记语言)的中间件技术完成异构数据的整合和交换<sup>[9]</sup>,将具有良好的移植性和功能扩展性,且易于实现,它们一般不改变底层数据源,异构信息集成发生在查询过程中。

现有的数据集成方法主要的操作对象是存储介质中的数据,往往在查询过程中完成异构信息的集成,并没有指明如何处理源数据。而普适计算系统不断从物理以及计算环境中获取上下文信息,数据量庞大且存在冗余,需要经过处理才能存储。因此,普适计算中上下文信息的集成应更靠近底层数据源,且要保障上下文的QoC<sup>[10]</sup>(Quality of Context,上下文质量),即上下文的精度(precision)和新鲜度(freshness);在集成上下文时,不失上下文语义的前提下,还应尽量减少上下文的存储量。

文中基于XML的中间件技术,提出分层集成的

思想,根据上下文在精度和新鲜度上的特点,使用不同的方法处理底层上下文数据源,并定义映射模式,最终将各种异构的上下文集成到XML文件中。

## 2 上下文转换和集成模型

文中分上下文转换和上下文集成两层处理上下文数据。上下文转换层说明如何处理上下文。上下文集成层定义了局部模式到全局模式的映射关系,并将转换层处理后的上下文信息集成到XML文件中。XML支持Web的各种不同应用,它可以和HTML一样使用HTTP进行传送,不需要对现存的网络设置进行改变。普适环境下的上下文多为半结构化数据,使用XML作为上下文的集成形式,可以方便上下文的扩展和访问,同时可以在此基础上开发更高层的应用,例如使用Agent(本体)进行推理和建模。

### 2.1 上下文转换

普适计算中上下文有多种分类标准,根据上下文信息的变化频率可分为两种类型:稳定型和即时变化型<sup>[1]</sup>,稳定型上下文一般需要手工输入后存储于关系数据库中,且长时间内不会改变;而即时变化型上下文具有实时性,需要利用某种上下文信息感知设备来实时接收,如实体位置、环境温度和噪声等。在保证上下文新鲜度的前提下,采用合并(Union)方法对具有相似属性的稳定型上下文做统一管理,即将相似上下文从数据库中提取出来,剔除无关数据后,集成到局部XML文件中。这样做的好处是,减少信息量,降低语义冲突,与即时变化型上下文分开处理,有利于提高即时变化型上下文的处理效率。采用组合(combination)、聚合(aggregation)、抽样(sampling)和泛化(generalization)<sup>[11]</sup>等方法处理即时变化型上下文,转换的基本框架如图1所示。

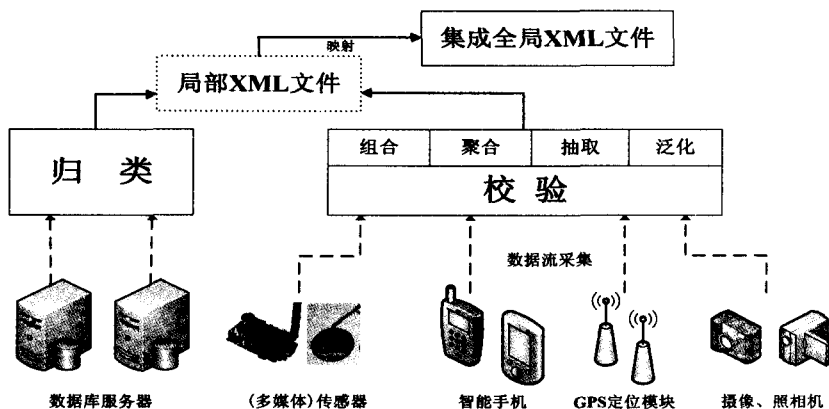


图1 上下文转换框架

由于即时变化型上下文往往因采集设备本身和环境因素产生失真,文中重点对其提出有效性校验。为保证上层服务不受影响,需要剔除这些失真信息。上

下文信息经过校验后,就可以采用组合、抽样、聚合和泛化等方法进一步处理<sup>[12]</sup>。

图 1 中的传感器节点可能采集人的体温、脉搏和血氧等上下文特征信息,在某一时刻  $i$ ,采集的信息可分别表示为  $tm_i, p_i, o_i$ ;也可能采集温度、湿度和光强等标量物理信息,分别表示为  $tr_i, h_i, l_i$ ;可能是 GPS 的定位信息,表示为  $\langle x_i, y_i \rangle$ ;还有多媒体信息,例如图像信息  $I_i$  ( $I_i$  为二进制编码表示的图像矩阵或子矩阵),视频信息流  $V$  中的某一帧  $VI_i$  ( $V = \langle VI_1, VI_2, \dots, VI_i, \dots, VI_n \rangle$ , 其中  $n$  表示视频流中图像帧的总数)。针对于某一时刻  $i$ ,混合的上下文数据流序列用  $D_i$  表示,  $D_i = \{tm_i, p_i, o_i, tr_i, h_i, l_i, \langle x_i, y_i \rangle, I_i, VI_i, \dots\}$ 。

方法 1:组合是对具有相似语义的上下文做简单的合并,并将组合后的上下文看成一个整体,这有利于上下文语义的表达。对于  $D_i$  组合以后可生成  $D_i' = \{[tm_i, p_i, o_i], [tr_i, h_i, l_i], [\langle x_i, y_i \rangle], [I_i], [VI_i], \dots\}$ 。

方法 2:抽样是从连续的上下文数据流中抽取某一时刻的上下文信息,作为当前上下文。抽样法使用的前提是牺牲上下文的实时性来换取上下文的高精度。这种场景常见于室内定位,一个被定位目标可能长时间停留在房间的某一处,就没有必要进行高频次的采样。特别的,对于某一时域的视频流信息  $V$ ,不能简单的在某一时刻抽取出某一图像帧。若  $V$  可表示为:

$$V = \begin{bmatrix} VI_{11} & VI_{12} & \dots & VI_{1i} & \dots & VI_{1n} \\ VI_{21} & VI_{22} & \dots & VI_{2i} & \dots & VI_{2n} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ VI_{j1} & VI_{j2} & \dots & VI_{ji} & \dots & VI_{jn} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ VI_{n1} & VI_{n2} & \dots & VI_{ni} & \dots & VI_{nn} \end{bmatrix}_{n \times n}$$

其中有  $V \cdot X = \lambda \cdot X$  ( $X$  为特征向量,  $\lambda$  为特征值)。对视频流信息的抽样表现为对其特征向量的抽样。

方法 3:聚合是针对数值型上下文所做的操作,主要是分析短期内的历史数据,并对其做变换,最终用某个值替代这些历史数据。使用该方法的前提是上下文变化平缓,且上下文采集是规律的周期性的进行,不至于破坏上下文的新鲜度,聚合的优点是可以尽量减少上下文的存储量。在一段时域  $T[1, 2, \dots, i]$  采集的体温信息依次为  $tm_1, tm_2, \dots, tm_i$ , 可将  $Tm\_Average$  作为简单的聚合结果。其中

$$Tm\_Average = \frac{1}{i} \sum_{j=1}^i tm_j$$

方法 4:泛化是将上下文源数据映射为具有一定

语义的容易理解的概念。例如, GPS 采集到的实体位置信息有经度和纬度两个值,对于一般的定位跟踪,需要知道的是实体当前所处的地点名字,故将经纬度转换为具体地点名称再存储,既可以减少存储量,也便于向上层提供服务(如根据地理位置提供天气预报服务)。泛化法不影响上下文的新鲜度,但会破坏上下文的精度,对精度要求很高的应用,不宜采用泛化法。

根据所采集上下文的特点,可以单独或者联合使用组合、抽样、聚合和泛化等方法,在保证上下文 QoC 的前提下,进一步减少上下文的信息量。经过以上几种方法处理后,上下文信息将存入每个实体的局部 XML 文件中。对于每个局部 XML 文件,则需要预先定义 DTD (Document Type Definition, 文档定义类型)。DTD 用来说明文档中元素、属性和实体,以及以上这些内容之间的相互关系。因 DTD 只能检验 XML 是否合法,而不能对其校正,因此在 DTD 中没有对元素类型做限制,数据的有效性应在上下文转换中完成。基于图 1, DTD 片段如图 2 所示。

```
<? xml version="1.0" encoding="UTF-8"? >
<! ELEMENT Person (Name, Sex, Address, Photo)>
<! ATTLIST Person PersonID ID #REQUIRED>
<! ELEMENT Name (#PCDATA)>
<! ELEMENT Sex (#PCDATA)>
<! ELEMENT Address (#PCDATA)>
<! ELEMENT Photo (#PCDATA)>
<? xml version="1.0" encoding="UTF-8"? >
<! ELEMENT Person (Location, Temperature, Pulse)>
<! ATTLIST Person UserID ID #REQUIRED>
<! ELEMENT Location (#PCDATA)>
<! ELEMENT Temperature (#PCDATA)>
<! ELEMENT Pulse (#PCDATA)>
```

图 2 基于图 1 的 DTD 片段

不同的应用对上下文的新鲜度要求不同,当应用对某种上下文的新鲜度不敏感时,可以进一步降低该种上下文的集成次数。针对这一情况,文中除了使用之前介绍的方法处理源上下文数据外,在生成局部 XML 文件后,还对上层的数据集成做了延迟处理。如此一来,可以减少文件的映射操作,而设置不同的局部集成次数,会使得不同上下文以不同的时间间隔访问映射模式文件和全局 XML 文件,从而尽量避免同时访问全局 XML 文件所产生的冲突。

为了加快局部单类型上下文的集成、提高上下文处理效率、配合上下文集成次数,采用了文件内存保留策略,每集成一次信息后,在内存中保留该实体的 XML 文件 DOM 树,当下次采集到的仍为该实体信息时,就直接更新内存 DOM 树,而省去 I/O 操作和创建 DOM 树的开销。由于需要不断更新 XML 文件,故使用 DOM 接口。需要说明的是,局部 XML 文件均以“实

体的标识 & 字符串”命名,这样不同实体通过实体的标识区别,同一实体的不同上下文,使用不同的字符串加以区分,这样便于上下文的集成和管理。内存保留的处理流程如下:

```
processData() {
    收到新的上下文数据;
    if (与上一数据属同一用户)
        更新 DOM 树;
    else {
        if (新用户)
            重新创建 DOM 树;
        else
            读取硬盘 XML 文件,创建并更新 DOM 树;
    }
}
```

2.2 上下文集成

上下文集成层主要设计局部上下文数据到全局数据的映射模式、全局 XML 模式文件和信息的存储策略。映射模式提供局部上下文信息到全局 XML 中元素(上下文名称)和内容(上下文数据)的映射关系。映射文件的形式如下所示:

```
<? xml version="1.0" encoding="UTF-8"? >
<root>
  <Name>      PName      </Name>
  <Sex>       PSex       </Sex>
  <Location>  PLocation  </Location>
  <Photo>     PPhoto     </Photo>
  .....
</root>
```

映射文件中,标签名(如 Name)是局部 XML 文件中所使用的名称,标签内容(如 PName)是全局 XML 文件所使用的名称,使用 XML 的访问接口可以很容易完成字段的映射。在进行映射匹配时,只需读取 XML 文件,故使用 SAX 接口。从映射文件可以看出,没有 PersonID 和 UserID 的映射关系,这种实体标识的映射是通过局部 XML 文件名和全局 XML 文件名的映射实现的,假定有存储实体“Y0810”的身份上下文文件“Y0810&Social\_Context.xml”以及体征上文文件“Y0810&Physical\_Context.xml”,最终这两个文件中的上下文信息都将映射到全局上下文文件“Y0810.xml”中。

全局 XML 文件模式的定义使用 schema 完成,全局模式说明了全局 XML 文件中上下文应遵守的数据类型以及逻辑结构,并验证其正确性。全局模式片段和映射形式如图 3 所示。

通过以上方式易于实现底层各种异构上下文的集成。虽然在上下文转换时已经对特定类型的上下文信

息做了压缩,但随着时间的推移,全局 XML 文件仍会变得十分庞大。

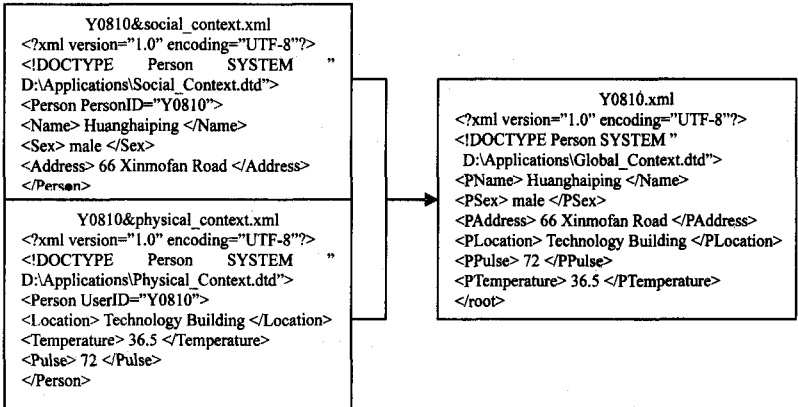


图 3 全局 XML 文件模式片段及映射形式

为进一步加快全局 XML 文件的访问速度和信息集成,在文件内存保留策略的基础上采用了分布式数据库分片的思想,对全局 XML 文件进行分割转存,分割后的分片可存放在本地目录或网络服务器上。具体流程为:当 XML 全局文件大小超出设定的阈值时,就将文件转存,形成分片。分片后的文件使用“实体标识#数字”的形式命名,如此可快速检索某一实体的全部上下文信息。分片的组织形式如图 4 所示。

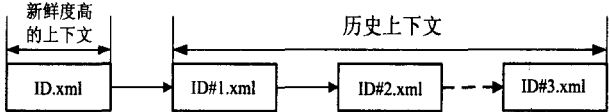


图 4 分片后文件组织形式

文件分片转存机制的优点在于:

- 1) 保证上下文信息的新鲜度。没被转存的分片(如图 4 中 ID.xml)的上下文信息整体新鲜度大,能更好地支持普适服务,如定位跟踪。
- 2) 有利于历史上下文信息的管理。使用“实体标识#数字”的形式命名分片文件,便于对历史上下文感兴趣的应用实现服务,例如根据历史上下文生成旅游日志。
- 3) 提高集成效率,进一步减少访问冲突。

3 实验测试

文中采用的软硬件测试环境是真实的无线多媒体图像传感器网络、无线医疗传感器网络和 Java 开发平台。

测试的场景一假定为对感知区域内的用户基本生理状况和位置做实时跟踪,处理的是数值型信息。通过传感器节点实时采集用户的体温、血氧和脉搏信息, GPS 定位器则采集用户所处的经纬度值。传感器节点采用唯一的编号并作为用户的身份标识。为了节约测试成本,在一定通信范围内实际采集 20 个用户的源上

下文信息,并据此拷贝出 100 个用户的原始上下文信息,存入 100 个以 1,2,3,...,100 命名的文件中,相应的用户身份标识由 1 到 100 编号。

基于 XML 模式来存储这些文件,分为三种情形:

(1) 不采用组合、聚合、抽样等方法,也不采用文件内存保留策略和数据分片方法,直接存储这 100 个文件;

(2) 不采用组合、聚合、抽样等方法,但采用文件内存保留策略和数据分片方法来存储这 100 个文件;

(3) 采用组合、聚合、抽样等方法,同时采用文件内存保留策略和数据分片方法来存储这 100 个文件。

测试程序利用随机函数每秒产生一个 1 到 100 之间的随机数,然后根据产生的随机数读入用户原始上下文文件,这样使得每次访问的用户可能各不相同。对于在上下文转换时提到的集成次数,本实验中设定为  $N$ ,即图 5 中的横坐标值。三种方式对于不同的集成次数有一个执行时间开销的比较,图 5 中纵坐标的值表示时间开销,均减去了 3000 秒,并且忽略了节点间传送数据的延迟。

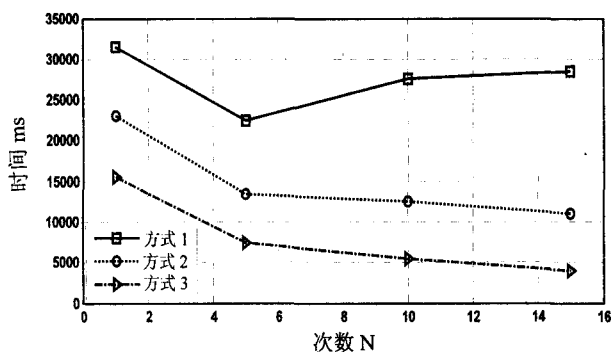


图 5 针对数值型数据三种方式的时间开销比较

从图 5 的测试结果可以看出,集成方法的执行时间随  $N$  的增大而逐渐降低,也就是如果某一用户的上下文信息被连续的采集和集成,将表现出更好的性能,但  $N$  不能无限增大,因为还要保证上下文的新鲜度。同时,采用了数据处理,或者文件内存保留策略和数据分片方法的方式要比没有采用的更加高效。这种性能的提高是使用集成方法的整体结果,并不能说明某一单一方法,如聚合方法的性能,因为就聚合方法本身的特点而言,可能需要额外的运算开销,但它换来的是上下文存储量的降低。

测试场景二利用无线多媒体图像传感器节点来采集 30 个用户的照片信息,体现在 XML 文件中的 Photo 项。对于图像信息的采集均采用了抽样的方法,图 6 对比了未采用文件内存保留策略和数据分片方法的方式 1 和采用了以上策略的方式 2 之间的比较(纵坐标的处理同图 5)。测试的结果和结论等同于图 5 的结论说明。

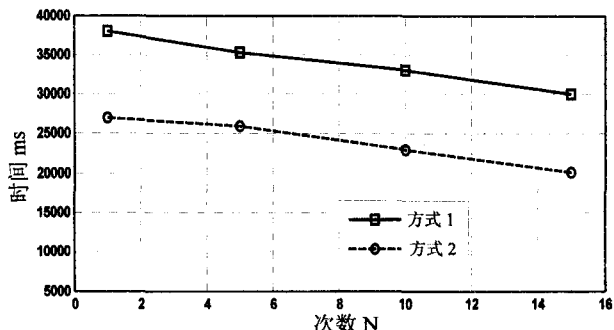


图 6 针对多媒体数据两种方式的时间开销比较

## 4 结束语

提出了一种普适计算环境上下文的转换和集成方法,直接作用于上下文源文件,采用 XML 文件存储上下文信息,解决了上下文异构且信息量庞大所造成的上层应用访问困难的问题。基于 XML 中间件形式,集成各种异构上下文,其统一的访问接口无需加载驱动即可操作 XML 文件,屏蔽了平台和语言的相关性;提出文件内存保留和分片策略,可加快处理 XML 文件的作业时间,更有效地提供实时的普适服务。该方法还有待改进,如可将全局 XML 文件作为数据源,使用 Agent(本体)进一步解决上下文在语义上的冲突,这将是进一步的研究工作。

## 参考文献:

- [1] 徐光祐,史元春,谢伟凯. 普适计算[J]. 计算机学报,2003,26(9):1042-1050.
- [2] Dey A K, Salber D, Abowd G D. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-aware Applications[J]. Human-computer Interaction,2001,16(2):97-166.
- [3] 余平,王汝传,孙力娟. 基于无线传感器网络的普适计算模型研究[J]. 计算机技术与发展,2006,16(4):1-4.
- [4] Lezerini M. Data integration; a theoretical perspective[C]//Proceedings of the 21st ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database System. Madison, Wisconsin, USA:[s. n.],2002:233-246.
- [5] Lin Wenyang, Wu Chin-Ang, Wu Chuanchu. An object-relational data warehouse modeling for complex data[C]//Proceeding of the Joint Conference on Information Sciences 2006. Kaohsiung, Taiwan:[s. n.],2006:257-260.
- [6] 朱益霞,孙道清,沈展. 一种普适计算下的访问控制策略[J]. 计算机技术与发展,2010,20(8):91-95.
- [7] Heimbigner D, Mcleod D. A Federated Architecture for Information Management[J]. ACM Transactions on Office Information System,1985,3(3):253-278.
- [8] Washington University of USA. Agilla: Intelligent Mobile Agents in Wireless Sensor Networks[EB/OL]. 2004-10. http:

(下转第 155 页)

式,因  $R_2^4$  为已知,故可求  $W_2^4$ 。

$$R_1^5 = S( L_1( W_{1L}^4 || W_{2H}^4 ))$$

$$R_2^5 = S( L_2( W_{2L}^4 || W_{1H}^4 ))$$

$W_1^4$  和  $W_2^4$  已由前面公式求出,将其代入上述公式即可求  $R_1^5$  和  $R_2^5$ 。

$$S_{16}^4 = 2^{15} S_{15}^4 + 2^{17} S_{13}^4 + 2^{21} S_{10}^4 + 2^{20} S_4^4 + (1 + 2^8) S_0^4 \bmod (2^{31} - 1)$$

$$\text{If } S_{16}^4 = 0, \text{ then set } S_{16}^4 = 2^{31} - 1$$

预猜测  $S_{10}^4$  的高 8 位,因  $S_{10}^4$  的其余位和  $S_{15}^4, S_{13}^4, S_4^4$  均已知,  $S_0^4$  已求出,故可求  $S_{16}^4$ 。

至此在  $t=4$  时刻,已猜测  $S_{10}^4$  的高 8 位,共 8 比特。推导出来的有  $S_0^4, S_{16}^4, R_1^5, R_2^5$ 。因此  $t=4$  时刻的全部已知量为  $S_0^4, S_1^4, S_2^4, S_3^4, S_4^4, S_5^4, S_6^4, S_7^4, S_8^4, S_9^4, S_{10}^4, S_{11}^4, S_{12}^4, S_{13}^4, S_{14}^4, S_{15}^4, S_{16}^4$ 。

在  $t=5$  时刻,由  $(S_1^4, S_2^4, \dots, S_{15}^4, S_{16}^4) \rightarrow (S_0^5, S_1^5, \dots, S_{14}^5, S_{15}^5)$  可知该时刻的已知量为  $S_0^5, S_1^5, S_2^5, S_3^5, S_4^5, S_5^5, S_6^5, S_7^5, S_8^5, S_9^5, S_{10}^5, S_{11}^5, S_{12}^5, S_{13}^5, S_{14}^5, S_{15}^5$ , 此外还有  $R_1^5$  和  $R_2^5$ 。因此  $t=5$  时刻的全部内部单元均已找到。之后即可用所求内部单元生成密钥来检测与原密钥是否相同,若相同则说明猜的是对的,否则需要重新搜索。

### 3 攻击的时间复杂度

截止到  $t=4$  时刻  $S_0^5 \sim S_{15}^5, R_1^5$  和  $R_2^5$  都求出来了。计算过程中用到了  $Z^2, Z^3, Z^4$ 。猜测的位置是  $t=0$  时刻的  $S_9^0, S_5^0$  的高 8 位,共 16 比特; $t=1$  时刻的  $S_9^1, S_5^1$  的高 8 位,共 16 比特; $t=2$  时刻的  $S_9^2, S_5^2, S_2^2, S_{10}^2, S_{13}^2$  的高 8 位和  $S_{15}^2$  的低 15 位,共 55 比特; $t=3$  时刻的  $S_3^3, S_{10}^3$  的高 8 位和  $S_{13}^3$  的高 15 位,共 31 比特; $t=4$  时刻的  $S_{10}^4$  的高 8 位,共 8 比特。因此总共猜测  $16+16+55+31+8=126$  比特。此时搜索复杂度为  $O(2^{126})$ ,而其穷尽搜索复杂度为  $O(2^{128})$ ,所以提高了搜索效率。

### 4 结束语

文中提出了对 ZUC 算法的 Guess and Determine 攻击,先猜一部分内部单元,然后去推导剩余的,从而得到全部的内部单元。其搜索复杂度为  $O(2^{126})$ ,而穷尽搜索的复杂度为  $O(2^{128})$ ,因此提高了搜索效率。

#### 参考文献:

- [1] 中国通信标准化协会. ZUC 算法公开评估[S/OL]. 2011. <http://www.ccsa.org.cn/zuc.php>.
- [2] CCSA. 3GPP Confidentiality and Integrity Algorithms 128-EEA3 & 128-EIA3[S]. 2011.
- [3] Feng Xiutao, Liu Jun, Zhou Zhaocun, et al. A Byte-based Guess and Determine Attack on SOSEMANUK[M]//Asia-crypt. [s. l.]:[s. n.], 2010:146-157.
- [4] Hastad J, Naslund M. The Stream Cipher Polar Bear[R/OL]. 2005. <http://www.ecrypt.eu.org/stream>.
- [5] Mattsson J. A Guess-and-Determine Attack on the Stream Cipher Polar Bear[EB/OL]. 2006. <http://www.ecrypt.eu.org/stream/polarbear.html>.
- [6] Hasanzadeh M, Shakour E, Khazaei S. Improved Cryptanalysis of Polar Bear[EB/OL]. 2006. <http://www.ecrypt.eu.org/stream>.
- [7] Hawkes P, Rose G. Guess and Determine Attacks on SNOW[C]//SAC, 2002, LNCS 2595. [s. l.]:[s. n.], 2002:37-46.
- [8] 张海霞. 流密码算法 SOSEMANUK 的安全性分析[D]. 西安:西安电子科技大学, 2011.
- [9] 刘树凯, 关杰, 常亚勤. 针对流密码 K2 算法的猜测决定攻击[J]. 计算机工程, 2011(7):168-170.
- [10] 冯登国, 裴定. 密码学导引[M]. 北京:科学出版社, 1999.
- [11] 龙冬阳. 应用编码与计算机密码学[M]. 北京:清华大学出版社, 2005.
- [12] 丁存生, 肖国镇. 流密码学及其应用[M]. 北京:国防工业出版社, 1994.

(上接第 134 页)

[//www.cs.wustl.edu/mobilab/projects/agilla/index.html](http://www.cs.wustl.edu/mobilab/projects/agilla/index.html).

- [9] 丁月华, 杨敏, 文贵华, 等. 基于 XML 的异构数据源集成与交换的实现[J]. 计算机应用与软件, 2006, 23(10):34-38.
- [10] Kamran S, Maarten W, van Sinderen. Middleware support for quality of context in pervasive context-aware systems[C]//Proceedings of the Fifth IEEE International Conference on Pervasive Computing and Communications Workshops. New

York, USA:IEEE Computer Society, 2007:461-466.

- [11] Rasheed F, Lee Young-Koo, Lee Sung-Young. Applying Context Summarization Techniques in Pervasive Computing System[C]//3rd Workshop on Software Technologies for Future Embedded & Ubiquitous Systems (SEUS 2006). Gyeongju, Korea:IEEE Computer Society, 2006:107-112.
- [12] 宋欢, 邬家炜, 成永常. 基于普适计算的学习框架的研究[J]. 计算机技术与研究, 2010, 20(4):117-123.