

结合 AI 规划和工作流的动态服务组合框架研究

方其庆^{1,3} 彭晓明^{1,2} 刘庆华¹ 胡亚慧¹

(空军雷达学院 武汉 430019)¹ (武汉大学计算机学院 武汉 430072)² (通信指挥学院 武汉 430010)³

摘 要 动态服务组合是构建面向服务、松耦合、集成化的应用系统的主要途径。现行的动态服务组合方法中,工业界倾向于采用 BPEL 建模服务流程,而语义网研究团体则吸取 AI 规划的思想以实现 Web 服务组合,以上两种方法都不同程度地存在各自的不足。在对上述方法进行对比分析的基础上,提出了一种结合 AI 规划和工作流技术的动态服务组合方法。实践表明,这种方法能有效克服采用单个方法所造成的不足,并可实现从需求到部署的全过程。

关键词 Web 服务,动态服务组合,工作流,AI 规划

中图分类号 TP393 **文献标识码** A

Study of Web Service Composition on Combining AI Planning with Workflow

FANG Qi-qing^{1,3} PENG Xiao-ming^{1,2} LIU Qing-hua¹ HU Ya-hui¹

(Air Force Radar Academy, Wuhan 430019, China)¹ (Computer School of Wuhan University, Wuhan 430072, China)²

(Commanding Communications Academy, Wuhan 430010, China)³

Abstract Dynamic service composition is a major route to build a service-oriented, loose coupling and integrated application system. And by now, the business world has adopted a approach in which Web service instances are composed into flows with a language like BPEL. Academia has propounded the AI approach. These approaches by themselves are piecemeal, and insufficient. Based on the contrastive analysis on these approaches, this paper proposed a dynamic Web service composition method which is integrated the AI planning with the workflow technique. The practice shows that it is an effective method to overcome the single method's insufficient and finally achieve the whole process from the requirement to the deployment.

Keywords Web service, Dynamic Web service composition, Workflow, AI planning

随着 Web 服务技术的日益成熟,越来越多的稳定易用 Web 服务共享在网络上。但单个的 Web 服务能够提供的功能有限,为了更加充分地利用共享的 Web 服务,有必要将共享的 Web 服务动态地组合起来,提供更为强大的服务功能,加快系统开发的速度,快速满足用户需求。

1 相关工作

当前,动态 Web 服务组合主要有基于工作流和基于 AI 规划两种主流方法。

工业界主张采用分布式的方法,用 WSDL(Web Service Definition Language)表述 Web 服务实例,利用诸如 BPEL(Business Process Execution Language)^[1]之类的语言将其组合成工作流,然后使用 SOAP 协议进行调用。其中,BPEL 是由 XLANG 和 WSFL 合并而成,是一种基于 XML 的,用来描写业务过程的编程语言。BPEL 由两个部分组成:执行流程模拟业务交互中一个参与者的实际行为;抽象流程则描述业务协议中各个参与方相互可见的消息交换行为。通过 BPEL 可以对可执行过程和抽象过程进行建模,完成 Web 服务调

用、操作数据、抛出故障或终止一个流程等工作的不同活动,从而创建出复杂的流程。

文献[2]分析了基于工作流的服务组合项目:METRO-RS。它首先采用 BPEL 作为工业标准,设计抽象的 Web 服务组合过程,包括使用 BPEL 提供的控制流结构来创建过程流、在过程中通过服务模板来表示每个服务需求、指定过程约束以达成优化等;其次对给定过程和服务模板,服务发现引擎将返回和该服务模板匹配的一个服务集合,根据约束关系选择最优的待组合服务;最后在运行时,将抽象的过程和服务模板转换成一个可执行的 Web 过程。另外,在文献[3]中,Wang 等人提出了基于动态工作流的设计阶段和运行阶段的服务组合,解决了运行时过程变化问题。

学术界提出用 AI 方法,以本体论形式化表达 Web 服务的功能,使用面向目标的推理技术规划 Web 服务的组合。OWL-S(Web Ontology Language for Services)^[4]的出现为基于 AI 规划的 Web 服务动态组合提供了可能。OWL-S 是一种具有显示语义的无歧义的机器可理解的标记语言,它为描述 Web 服务定义一套核心的扩展语言结构,并且推荐了一种

到稿日期:2008-10-31 返修日期:2009-05-22 本文受 863 国家重点基金项目(2007AA01Z138)资助。

方其庆(1979—),男,博士生,讲师,主要研究方向为动态 Web 服务组合和军事电子信息系统应用集成等,E-mail:fangqiqing@yeah.net;彭晓明(1965—),男,博士生,副教授,主要研究方向为面向服务技术和面向对象技术等;刘庆华(1964—),男,硕士,副教授,主要研究方向为军事装备管理等;胡亚慧(1980—),女,硕士,讲师,主要研究方向为 Web 个性化。

本体框架用于定义 Web 服务。定义这种模型有两种主要的 W3C 标准:资源描述框架(RDF)和 Web 本体语言(OWL)。其中,RDF 用于描述信息和知识交换,OWL 用于定义本体,支持 Web 搜索和知识管理。该框架中,每个服务由 3 部分描述:Service Profile, Service Model 和 Service Grounding 所组成。Service Profile 是服务基本信息,说明了本服务能做什么。它主要描述了那些用于判断服务是否满足需求的信息的类型,并主要提供给服务搜索引擎服务。Service Model 是服务模型,主要描述了服务如何进行工作,以及当服务执行时,哪些行为会被执行。Service Grounding 是实现模型,描述了该服务如何被访问和执行。通过 OWL-S 进行动态服务组合的基本思想是:将 Web 服务看成 AI 中的动作,通过输入输出参数、前提和结果等来描述 Web 服务,在服务组合时,则是将 Web 服务的这些描述映射为动作形式化描述,在 Web 服务空间中以构造 Web 服务组合为目标,通过形式化的推理来得出 Web 服务的组合序列,动态形成服务组合方案。

文献[5]于 2002 年提出了一个形式化的方法,将 OWL-S 转换到情境演算,将 Web 服务组合问题转换为一个满足目标属性要求的程序执行问题。文献[6]于 2003 年提出了一种从高层次声明描述产生组合服务的技术,使用组合规则来决定两个服务是否可以组合。Liu 等人在 2005 年提出一种最小执行代价的基于规划的服务组合方法^[7];采用规则对 Web 服务进行建模,使用参数本体来消除语义冲突,定义并使用一个演绎网络和一个后向演绎方法以发现所有的组合规划并选择一个最优规划。文献[8]通过分层任务网络规划(Hierarchical Task Network, HTN)来实现服务的自动组合,利用 SHOP2 规划器进行求解。文献[9]给出了一种基于领域本体的服务动态组合方法,该方法利用领域本体及其推理能力,生成一个优化的服务组合图,然后基于该图给出算法找出满足要求的路径,即一个满足服务请求的服务组合。

上述两类方法,前者主要关注 Web 服务组合的执行方面,而较少关注需求的捕获和服务组合的自动性;其服务组合的描述主要是针对语法层面,缺少语义支持;主要侧重于设计的灵活性及执行期间的可管理性,服务组合自动性不够。而后者主要关注基于语义的 Web 服务组合的可行性,但并不关心其输出是否可以直接部署到运行环境中。针对上述方法的不足,本文提出一种结合 AI 规划和工作流技术的动态服务组合新方法。

2 结合 AI 规划和工作流的动态服务组合

2.1 问题定义

为了降低问题的复杂度,将 Web 服务类型与 Web 服务实例区别开来,其中,服务类型是指具备相似功能的一组 Web 服务;服务实例是指具有明确 URL 地址的、实际的、可被调用的 Web 服务,分别记为:

$T = \{T_1, T_2, \dots, T_\alpha\}$, 由 α 个 Web 服务类型组成的集合;

$I = \{I_1, I_2, \dots, I_\beta\}$, 注册在服务注册中心中的 β 个 Web 服务实例组成的集合;

很显然, T 和 I 之间是一种满射的关系,即:假设每个服务类型 T_i 包含 X 个 Web 服务实例,则 $\beta = X \times \alpha$ 。

另外,对组合服务的需求通常分为两类:功能性需求和非功能性需求。功能性需求是指组合服务是否满足用户期望的

功能要求;非功能性需求则包含性能、可用性、响应时间等信息。

为了尽可能使得组合过程自动化,结合本体,使用 OWL-S 以 IOPE 形式来表达需求。IOPE 指的是 Input, Output 以及 Preconditions 和 Effect,即输入、输出、前提、效果。另外,由于 BPEL 标准广泛受到厂商的支持,并有着 Active BPEL^[10], IBM 的 Websphere Business Integration^[11] 等比较成熟的执行引擎,因此采用 BPEL 语言描述的工作流作为服务组合的输出。

对于组合服务的问题,定义为:给定一套 Web 服务的类型及该类型下 Web 服务的实例以及对组合服务的以 OWL-S 为形式的需求描述,通过对现有的 Web 服务进行组合,创建一个满足设计功能性需求及非功能性需求的可执行工作流。

2.2 动态服务组合框架

结合 AI 规划和工作流的动态服务组合框架如图 1 所示,该框架中的服务组合大致划分为两个阶段:抽象服务规划阶段和服务实例选取阶段。

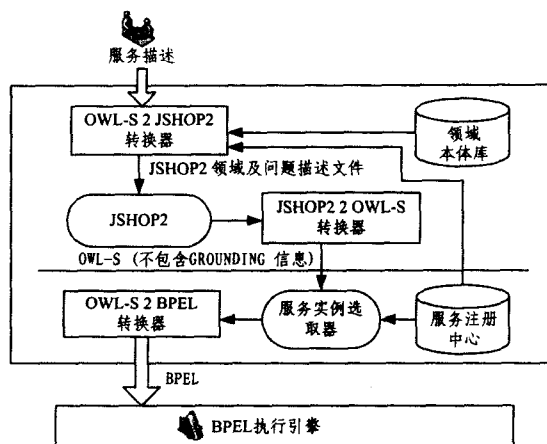


图 1 动态服务组合框架

抽象服务规划主要选取较为成熟的 AI 规划方法及工具,从服务注册中心中选取满足服务功能性需求的服务类型,通过 AI 规划生成 Web 服务类型的组合序列(不包含具体的服务实例)。

在抽象服务规划工具的选取上,本文选用成熟的 HTN 方法及 JSHOP 工具。HTN 规划是一种采用分层分解的方法将复杂问题分层分解,将高层行动分解为一个低层的偏序级的规划方法。SHOP2^[12] 是一种利用层次规划法的领域无关的规划系统,在 2002 年的 International Planning Competition 取得第一名,而其优于其它层次规划以及规划系统的是,它可以插入领域知识,了解当前的状态,可以使用逻辑推理、外接程序进行预处理和估计等。而 JSHOP2 是 SHOP2 采用 JAVA 进行编译的版本。

服务实例匹配则是将抽象 Web 服务规划中的每一个 Web 服务类型实例化,最终生成以 BPEL 为描述语言的可部署的工作流。通常来讲,Web 服务可以分为改变世界状态的 Web 服务(World-altering service)和提供信息的 Web 服务(Information-providing service),前者在满足前提条件的情况下,通过运行服务,使得事物(世界)的状态产生迁移,如学籍注册服务。后者,未对事物(世界)状态产生影响,但却可为服

务请求者提供所需信息,如航班查询服务。抽象服务规划采用的是 AI 规划方法,主要针对的是改变世界状态的 Web 服务,利用前提条件和效果信息对 Web 服务进行组合,而对于 Web 服务的输入、输出是否满足用户的需求,则未作考虑。为了补充抽象服务规划阶段的这一不足,本文在服务实例匹配过程中,除了对 Web 服务实例的非功能性指标即 QoS 性能进行排序外(以下简称 QoS 匹配),还综合考虑用户对 Web 服务输入、输出参数即接口类型的需求(以下简称接口匹配)。

该动态服务组合框架中包含的核心组件有:

OWL-S 2 JSHOP2 转换器:将 OWL-S 格式的需求转换为 JSHOP2 规划器可理解的领域文件和问题描述文件。

领域本体库:存储组合服务相关领域的本体信息。

JSHOP2 规划器:通过领域文件描述的领域知识,对服务组合问题运用 HTN 规划技术,推理出抽象服务规划。

JSHOP2 2 OWL-S 转换器:将 JSHOP2 产生的问题解转化为 OWL-S。

服务实例选取器:按照用户对组合服务的接口类型和 QoS 性能需求,为抽象服务规划结果中的每个服务类型选取相应的服务实例,即生成包含 Grounding 信息的 OWL-S。

服务注册中心:存储 Web 服务的类型、Web 服务的实例以及实例与类型之间对应关系等信息。

OWL-S 2 BPEL 转换器:将包含 Grounding 信息的 OWL-S 文件转换为可部署至执行引擎的 BPEL。

2.3 抽象服务规划

本文通过 JSHOP2 进行抽象服务规划的流程为:

步骤 1 将 OWL-S 格式的服务需求描述及相关领域本体信息通过 OWL-S 2 JSHOP2 转换器,转换为 JSHOP2 可理解的领域文件和问题描述文件;

步骤 2 通过 JSHOP2 实施规划,并从服务注册中心选取相关的服务类型构成 Web 服务的组合序列(抽象的组合服务,不包含具体的服务实例);

步骤 3 将 JSHOP2 产生的结果,通过 JSHOP2 2 OWL-S 转换器转换为 OWL-S。

其中,JSHOP2 的知识库包含算子(Operator)和方法(Method),算子用来描述如何完成一个简单任务(Primitive Task),方法用来描述如何将一个复合任务(Compound Task)进行分解成为一套部分有序的子任务。

定义 1(算子) JSHOP2 算子的表达式为 $(h(\vec{v}), Pre, Del, Add)$ 。

其中, $h(\vec{v})$ 是一个拥有输入参数 $\vec{v}$ 的简单任务;

Pre 表示算子的前提条件(PRECONDITIONS);

Del 表示在算子执行后应该从当前状态中删除的状态列表;

Add 表示在算子执行后应该向当前状态中增加的状态列表。

定义 2(方法) JSHOP2 方法的表达式为 $(h(\vec{v}), Pre_1, T_1, Pre_2, T_2, \dots)$,其中

$h(\vec{v})$ 是一个拥有输入参数 $\vec{v}$ 的复合任务;

Pre_i 表示前提条件;

T_i 表示有序的部分子任务。

为此,一个规划问题可以表示为:

定义 3(规划问题) JSHOP2 的规划问题表示为一个三

元组 (S, T, D) ,其中 S 代表初始状态, T 代表任务列表, D 是领域描述,JSHOP2 将 (S, T, D) 作为输入,返回一个规划 $P = (p_1, p_2, \dots, p_n)$ 。

文献[8]还介绍了利用 SHOP2 及 HTN 进行 Web 服务组合以及组合过程中 OWL-S 向 SHOP2 转换的方法,以下,仅介绍它的一个运用于 JSHOP2 实施规划过程的简化算法,如图 2 所示。

Algorithm JSHOP2(s, T, D)

If T is empty Then return empty;

Let t be the first task in T ;

If t is a primitive task Then {

Find an operator $o = (h, Pre, Del, Add)$
such that h unifies with t and s
satisfies Pre ;

If no such o exists Then return failure;

Let s' be s after deleting Del and adding Add ;

Let T' be T after removing t ;

Return $[o, JSHOP(s', T', D)]$;

}

Else if t is a composite task {

Find a method $(h, Pre_1, T_1, Pre_2, T_2, \dots)$ in
 D such that h unifies with t

Find the task list T_i such that

s satisfies Pre_1 and does not satisfy
 $Pre_k, k < i$;

If no such T_i exists then Return failure;

Let T' be T after removing t

and adding all the elements in T_i at the
beginning

Return $JSHOP(s', T', D)$

}

End if

End JSHOP2

图 2 JSHOP2 规划过程

2.4 服务实例匹配

服务实例匹配的基本步骤为:

步骤 1 通过服务实例选取器为抽象服务规划结果中的每个服务类型选取相应的服务实例。其选取的过程为首先筛选出满足用户接口要求的 Web 服务,再进一步对筛选出的结果按照 QoS 指标进行排序。

步骤 2 通过 OWL-S 2 BPEL 转换器,将包含 Grounding 信息的 OWL-S 文件转换为可部署至执行引擎的 BPEL。

步骤 3 将 BPEL 部署至 BPEL workflow 执行引擎上。

2.4.1 接口匹配

接口匹配必须满足以下两方面条件:1)服务的输出可以满足用户请求的输出。即用户想要得到的输出必须能够通过服务调用获得。2)用户请求的输入可以满足服务的输入。即服务正常运转所需要的输入必须有用户请求提供。

本文的服务接口匹配的算法(以输出匹配为例)如图 3 所示。

degreeOfMatch(outR, outA);

if outA=outR then return exact

if outR subclassOf outA then return exact

otherwise fail

图 3 接口匹配算法

其中,outR 表示用户请求的输出,outA 表示服务提供的输出,subclassOf 表示子类关系。

通过以上分支语句,可以很容易地看出,当用户请求输出与服务输出相同时,为完全匹配,当用户请求输出是服务输出的直接子类时,也为完全匹配;否则,为匹配失败。

2.4.2 QoS 匹配

Web 服务的 QoS 包括很多方面的因素,为了描述方便,仅以服务费用、服务响应时间、服务可用性 3 个方面的性能指标为例(根据需要可进行灵活的扩展),同时,假设服务注册中心中,为每个服务实例储存了这 3 个方面的信息,并分别记作 QC_i, QR_i, QA_i ,运用加权平均法将得到该服务实例的总体 QoS 指标 $Q_i = W_c \times QC_i + W_R \times QR_i + W_A \times QA_i$,其中 $W_c, W_R, W_A \in [0, 1]$,并且 $W_c + W_R + W_A = 1$ 。该权重指标可由用户进行设置。

服务实例的匹配方法如图 4 所示。

```
Algorithm WebServicesMatch(cusIORequirement, service Type)
/* 输入: 用户接口需求 cusIORequirement; 服务类型 service-
Type */
/* 输出: 基于匹配度的服务实例排序 */
1)从服务注册中心中选取与 serviceType 匹配的服务实例集合
serviceInstances
2)for(int i=0;i<|serviceInstances|;i++){
    2-1) 计算 cusIORequirement 和 serviceInstances [i]的接口
    匹配度 interface[i]
    2-2) 提取 serviceInstances [i]中的 QoS 信息
    2-3)计算其 QoS 指标
}
3)首先,基于 interface [i]排序,因为 interface [i]只有 fail 和
exact 两种可能值,exact 在前,fail 在后;
4)其次,分别对 interface [i]为 exact 和为 fail 的两个服务序列
基于 Qi 排序
5)返回排序好的服务集合
```

图 4 服务实例匹配算法

3 示例

现以参加 WWW 2007 学术会议的旅行规划作为实际案例。通常异地参加学术会议的流程包括:会议注册,预订出发机票,预订返程机票,预订酒店。以上 4 类服务中,会议注册有着明确的服务提供者即 WWW 2009,而对于预订机票及酒店等行程安排则是由第三方来提供,同时,在网络上又存在着许多的预订机票及酒店的 Web 服务,用户需要通过人工搜索及筛选,才能得到适合自己的行程安排,比如,当查询到满足合适日期的机票服务后,发现当天没有合适的酒店,为此,又得更改出发日期并重新预订机票和酒店。本文的研究目的,则是为用户提供一种动态的服务组合方法,以用户提供的约束条件(如:会议开始时间、会议结束时间)为输入,通过 AI 规划形成满足用户约束条件的抽象服务规划,进而,进行服务实例匹配,得到实际可运行的工作流,完成流程中每项任务即可出发旅行,而不用再担心其中的旅行相关问题。

其中,用户的输入主要包括个人信息及其所参加会议的相关信息,如表 1 所列。

表 1 用户问题描述文件中的主体信息

输入	属性	属性描述
Person	Name	与会者姓名
	hasRegistration	是否注册

Conference	register_conference	会议名称
	startDat	会议开始日期
	endDate	会议结束日期
	city	会议所在城市

服务注册中心中包含的服务类型主要如表 2 所列。

表 2 服务注册中心中包含的服务类型

服务类型	描述
bookFlight	预订机票
bookHotel	预订旅馆
bookRoundTrip	预订往返旅程
register	注册会议
bookRoundTrip	预订往返旅程
bookRTOneDayLater	旅程提前一天
bookRTOneDayEarlierAndLater	旅程延期一天
.....	

其中,预订机票服务类型的 OWL-S 描述如图 5 所示。

```
<process:AtomicProcess rdf:about="http://www.whu.edu.cn/travel#bookFlight">
  <process:hasInput>
    <process:Input rdf:about="http://www.whu.edu.cn/travel#person">
      <process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.whu.edu.cn/travel#Person</process:parameterType>
    </process:Input>
  </process:hasInput>
  <process:hasInput>
    <process:Input rdf:about="http://www.whu.edu.cn/travel#flight">
      <process:parameterType rdf:datatype="http://www.w3.org/2001/XMLSchema#anyURI">http://www.whu.edu.cn/travel#Flight</process:parameterType>
    </process:Input>
  </process:hasInput>
</process:AtomicProcess>
```

图 5 预订机票服务类型的 OWL-S 描述主体部分

最终,通过 AI 规划及实例匹配后生成的结果如图 6 所示。

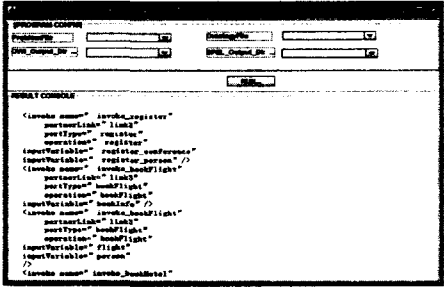


图 6 程序运行结果

结束语 本文提出了一种结合 AI 规划和工作流的动态服务组合方法,该方法首先将 Web 服务类型与 Web 服务实例区别开来,并将动态服务组合划分为抽象服务规划和服务实例匹配两个阶段,并最终生成可部署至 BPEL 工作流引擎的 BPEL 文档。

其中,将动态服务组合划分为两个阶段,有效地综合了工业界和学术界各自的长处,在充分运用语义信息的同时,实现了 Web 组合服务的可部署及可运行,并克服了单独采用 AI 规划方法不能有效考虑提供信息的 Web 服务的问题。而将 Web 服务划分为 Web 服务类型和 Web 服务实例,有效地降

低了 Web 服务匹配的搜索空间。

下一步,将进一步对该方法进行完善,其中对于抽象服务规划,由于 HTN 方法必须基于封闭世界假设(closed world assumption),即实现动态服务组合的前提是给出完备的语义信息,当语义信息不够完备时,规划结果将不尽人意。而描述逻辑^[13]在推理过程中是基于开放世界假设(open world assumption),可以有效解决信息不完备的问题,并成为当前的一大研究热点,现拟采用描述逻辑对该部分进行进一步的研究;另外,服务接口匹配中,为了提高匹配效率,目前采用的接口匹配方法也稍显简单。为此,我们也将对该部分的方法进一步地进行完善。

参考文献

- [1] OASIS . Web Services Business Process Execution Language Version 2.0 [EB/OL]. [2007-4-11]. <http://docs.oasis-open.org/wsbpel/2.0/OS/wsbpel-v2.0-OS.html>
- [2] Aggarwal R, Verma K, et al. Dynamic Web Service Composition in METEOR-S [EB/OL]. [2004]. <http://lsdis.cs.uga.edu/lib/download/IEEE-SCC-2004.pdf>
- [3] Wang S, Sheng W, Hao Q. Agent based workflow ontology for dynamic business process composition [C] // Proceedings of the Ninth International Conference on Computer Supported Cooperative Work in Design, 2005. New York: IEEE Press, 2005: 452-457
- [4] Martin D, Burstein M, Lassila O, et al. OWL-S: Semantic Markup for Web Services [EB/OL]. [2004 -11]. <http://www.daml.org/services/owl-s/1.1/overview/>

- [5] Narayanan S, McIlraith S. A Simulation, verification and automated composition of Web services [C] // Proceedings of the 11th Int'l World Wide Web Conference. Honolulu, ACM, 2002: 77-88
- [6] Medjahed B, Bouguettaya A, Elmagarmid A K. Composing Web Services on the Semantic Web [J]. The VLDB Journal-The International journal on Very Large Data Bases, 2003, 12(4): 333-351
- [7] Liu Jiamao, Fan Chenhui, Gu Ning. Web services automatic composition with minimal execution price Web Services [C] // Proceedings of the IEEE International Conference on Web Services. Washington DC: IEEE Computer Society, 2005: 302-309
- [8] Sirin E, Parsia B, Wu D, et al. HTN planning for Web service composition using SHOP2 [J]. Journal of Web Semantics, 2004, 1(4): 377-396
- [9] Li M, Wang D Z, Du X Y, et al. Dynamic composition of Web services based on domain ontology [J]. Chinese Journal of Computers, 2005, 28(4): 644-650
- [10] ActiveBPEL, LLC. ActiveBPEL-BPEL Execution Engine [EB/OL]. [2007-3]. <http://www.activebpel.org>
- [11] Martin K, Jonathan C, et al. BPEL4WS Business Processes with WebSphere Business Integration: Understanding, Modeling, Migrating [EB/OL]. [2004-12-22]. <http://www.redbooks.ibm.com/redbooks/pdfs/sg246381.pdf>
- [12] Nau D, Au T, Ilghami O, et al. Applications of SHOP and SHOP2 [J]. IEEE Intelligent systems, 2005, 20(2): 34-41
- [13] 王杰生, 李舟军, 李梦君. 用描述逻辑进行语义 Web 服务组合 [J]. 软件学报, 2008, 19(4): 967-980

(上接第 102 页)

```
MOV R1, #0xBA ; f(R15, K16) 地址送入 R1 寄存器
XRL A, @R1 ; 计算 L15 ⊕ f(R15, K16)
MOV @R0, A
```

通过分析指令的执行时间发现,图 5A 中 $|\rho_{\max}|$ 尖峰出现在第 4 条异或指令 XRL 处。该指令中 A 寄存器保存有 L_{15} 的值, R1 寄存器指向的存储单元保存有 $f(R_{15}, K_{16})$ 的计算结果,异或运算的结果送给 A 寄存器,因此 A 寄存器的内容在异或运算前后发生了变化。 $|\rho_{\max}|$ 出现明显尖峰说明寄存器 A 中数据变化同电磁辐射能量之间存在相关性,验证了电磁信息泄漏汉明距离模型 $E=aH(D \oplus D')+b$ 能够正确反映出这种线性关系,同时也说明了 XRL 指令存在电磁信息泄漏。

在与差分攻击比较中发现,相关分析使用 2500 组数据就可全部猜测正确。而使用 1 位差分攻击虽然也能攻击成功,但需要约 3000 组才能出现尖峰,用时约 70min。此外,由于实验中相关分析算法每次同时攻击 6 位子密钥,因此在实验结果中没有发现奇异峰现象。而在 1 位差分攻击中, D 函数以 1 比特位作为攻击目标位,因此单独判读某位出现的尖峰容易造成误判。为了避免这种奇异峰值造成的误判,需要在 $f(R_{15}^0, K_{16}^0)$ 中的 4 位同时出现尖峰时才能判断子密钥猜测正确,这是 1 位差分攻击本身的缺陷造成的。

结束语 本文实验结果表明,大规模数字集成电路在工作过程中的瞬态脉冲电流可产生电磁辐射信号。寄存器中的数据变化与电磁辐射之间存在着线性关系,可用汉明距离描述。利用这种线性关系,通过相关性分析可以获得集成电路

中的敏感数据。此外,与差分攻击相比,相关分析对集成电路密码系统进行旁路攻击需要的样本数量较少,破译更为高效和精确。而对于 DES 密码算法,每一轮的异或操作都可能成为密钥攻击点,在设计和实现中必须加以注意。

参考文献

- [1] Kocher P, Jaffe J, Jun B. Differential Power Analysis [C] // Proceeding of the Advances in Cryptology (CRYPTO'99). Santa Barbara, USA, 1999
- [2] Brier E, Clavier C, Olivier F. Correlation Power Analysis with a Leakage Model [C] // Proceedings of the 5th. International Workshop on Cryptographic Hardware and Embedded Systems (CHES-2004). Boston, USA, 2004
- [3] Messergers T S, Dabbish E A, Sloan R H. Investigation of power analysis attacks on smartcards [C] // Proceedings of the USENIX Workshop on Smartcard Technology. Chicago, Illinois, USA, 1999
- [4] Agrawal D, Archambeault B, Rao J R, et al. The EM side-channel(s): Attacks and assessment methodologies [C] // Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES-2003). Cologne, Germany, 2003
- [5] Kang Sung-Mo, Leblebici Y. CMOS 数字集成电路分析与设计 (第 3 版) [M]. 王志功, 窦建华, 译. 北京: 电子工业出版社, 2005
- [6] 孟昭敦. 电磁场导论 [M]. 北京: 中国电力出版社, 2007
- [7] Menezes A J, van Oorschot P C, et al. 应用密码学手册 [M]. 胡磊, 王鹏, 李学俊, 译. 北京: 电子工业出版社, 2005