一种基于UPPAAL的Web服务组合模型检测方法

何亚丽[1] 戎 玫[2] 张广泉[1,3]
(苏州大学计算机科学与技术学院 苏州 215006)[1] (暨南大学深圳旅游学院 深圳 518053)[2]
(中国科学院软件研究所计算机科学国家重点实验室 北京 100080)[3]

摘 要 Web服务组合的正确性验证对提高软件开发效率、实现服务增值具有重要意义。为从高层抽象层次研究Web服务组合的正确性及其形式化验证方法,考虑到Web服务组合中的实时特征,在采用软件体系结构描述语言XYZ/ADL对Web服务组合进行描述的基础上,将其实时描述部分XYZ/RE转换至时间自动机模型,组合后系统应满足的性质用分支时序逻辑CTL公式表示,最后应用模型检测工具UPPAAL实现了Web服务组合正确性的自动化验证。

关键词 Web服务组合,模型检测,XYZ/ADL,XYZ/RE,UPPAAL
中图法分类号 TP311 **文献标识码** A

Model Checking of Web Service Composition Based on UPPAAL

HE Ya-li[1] RONG Mei[2] ZHANG Guang-quan[1,3]
(School of Computer Science and Technology, Soochow University, Suzhou 215006, China)[1]
(Shenzhen Tourism College, Jinan University, Shenzhen 518053, China)[2]
(State Key Laboratory of Computer Sciences, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China)[3]

Abstract Correctness verification of Web service composition plays an important role for improving software development efficiency and realizing service value-added. To study the correctness of Web service composition and their formal verification method from a high and abstract view, considering the real-time feature in Web service composition, based on the description of Web service composition using software architecture description language XYZ/ADL, the real-time description XYZ/RE was transformed to timed automata, the properties that the composite system should be satisfied were expressed in CTL formula, last, the correctness of Web service composition was automatically verified through a model checking tool—UPPAAL.

Keywords Web service composition, Model checking, XYZ/ADL, XYZ/RE, UPPAAL

1 引言

随着互联网及电子商务的迅速发展,Web服务以其良好的连接异构平台能力和实现互操作的优势受到工业界和学术界的一致关注。而单一Web服务提供的功能有限,通过组合Web服务则可实现服务的增值与重用。由于Web服务组合常用于描述跨平台、跨组织的高层业务逻辑,因此如何保证Web服务组合的正确性显得更为重要。

目前学术界多数学者主要是将Web服务组合描述语言(如BPEL,WS-CDL等)转换成已有的形式化模型(如自动机、Petri网、进程代数等),之后利用形式化方法的相关理论及工具验证Web服务组合的正确性。但从体系结构角度验证Web服务组合,则是一个新的研究方向。随着Web服务逐渐成为Internet环境下实现分布式应用的强有力工具,多数情形下的Web服务都具有实时特征。基于此,本文从软件体系结构角度出发,考虑到Web服务组合中的实时特征,在采用软件体系结构描述语言XYZ/ADL描述Web服务组合的基础上,将其实时描述部分XYZ/RE转换至时间自动机模型,组合后系统应满足的性质用分支时序逻辑CTL公式表示,最后应用模型检测工具UPPAAL来实现Web服务组合活性和安全性的自动化验证。

2 XYZ/E,XYZ/RE及XYZ/ADL简介

XYZ/E是中科院软件所唐稚松院士提出的一种可执行时序逻辑语言[1],其基本单元是条件元,有两种形式:

$$LB=y\wedge R\Rightarrow \$Ov=e\wedge \$OLB=z \quad (1)$$

$$LB=y\wedge R\Rightarrow @(Q\wedge LB=z) \quad (2)$$

其中,R,Q是一阶逻辑公式,R是条件部分,Q和$\$Ov=e$为动作部分,$LB$是控制变量,$y$和$z$分别是转入和转出控制标号,$\Rightarrow$表示逻辑蕴涵,@是下一时刻算子$\$O$或最终算子

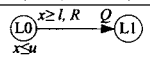
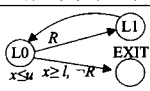
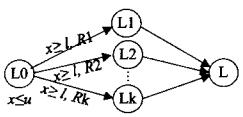
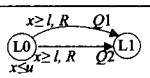
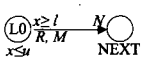
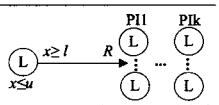
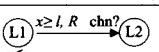
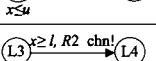
到稿日期:2009-12-10 返修日期:2010-03-07 本文受中国科学院计算机科学国家重点实验室开放课题(SYSKF0908),江苏省高校自然科学研究项目(08KJB520010)资助。

何亚丽(1983-),女,硕士生,主要研究方向为Web服务与形式化方法;戎 玫 博士,副教授,主要研究方向为软件工程和网络分布计算;张广泉 博士,教授,CCF高级会员,主要研究方向为网络化软件与服务计算、软件体系结构与形式化方法等。

<>。式(1)定义了相邻状态间的转换关系，式(2)表示程序抽象规范，它们的语义即为此时序逻辑公式的语义。

XYZ/E扩充了时序算子$O,<>,[],$U,$W,使之表示其实时下限(即 l)和实时上限(即 u)信息，得到 XYZ/RE。其中实时下限是指该算子从可能执行到该算子实际执行开始之间所需要的最短时间，实时上限是指该算子从可能执行到该算子全过程实际执行完毕之间所允许的最长时间，从而得到如下相应形式的实时算子：$O{l,u},<>{l,u},[]{l,u},$U{l,u},$W{l,u}。XYZ/RE 中相关句型的表达如表 1 所列。

表 1　XYZ/RE 句型、表达式及其对应的时间自动机

XYZ/RE 句型及表达式	对应时间自动机
基本条件元：LB=L0∧R⇒$O{l,u} (Q∧LB=L1)	
基本条件元：LB=L0∧R⇒@{l,u} (Q∧LB=L1)	同实时等待语句
循环语句：*{l,u}[LB=L0∧R⇒$OLB=NEXT\|$OLB=EXIT;LB=L1{|S|}$OLB=L0]	
实时情形语句：?{l,u}[LB=L0∧R1 ⇒$OLB=L1\|R2=>$OLB=L2 …… |~⇒$OLB=Lk; LB=L1{\|Q1\|}$OLB=EXIT; LB=L2{|Q2|}$OLB=EXIT; …… LB=Lk{\|Qk\|}$OLB=EXIT;]	
实时S条件语句：LB=L0∧R=>$O{l,u}(Q1∧LB=L1|Q2∧LB=L1)	
实时等待语句：LB=L∧R=> M$U{l,u}(N∧LB=NEXT)	
并发组合语句：||{l,u}[ProsInstaNmi(pari); …;ProsInstaNmk(park)]	
输入语句：LB=L1∧R1 => $O{l,u}chn? x1∧$OLB=L2	
输出语句：LB=L3∧R2 => $O{l,u}chn! x2∧$OLB=L4	

XYZ/ADL 是在 XYZ/E 的基础上扩展得到的一种体系结构描述语言[2]，支持软件体系结构中构件、连接件及配置的描述，其 XYZ/ADL 描述分别如下：

```
%COMPONENT COM==[
%PORT P:Record(%CHN A:DATATYPE1;
               %CHN B:DATATYPE2)
%FUNCTION==[…]
%COMPUTATION== […]]
%CONNECTOR CON== [
%ROLE SourceData==OUT(DT1,v1);
%ROLE SinkData= =IN(DT2,v2);
%GLUE==[…]]
%ATTACHMENTS== [
ComIns. Port # ConIns. Role;ComIns. Port # # Port;…]
```

其中，COMPUTATION 和 GLUE 都可以由 XYZ/E 表达式表示，直至精化为最基本的条件元表达式，对带有时间约束的系统而言，则可以由 XYZ/RE 表达式表示系统应满足的时间约束。

3　基于 UPPAAL 的 Web 服务组合模型检测

模型检测是一种验证并发有穷状态系统性质的技术，通常采用搜索状态空间的方法检查给定的计算模型是否满足某个特定性质，目前在网络协议、硬件等领域已有广泛应用。本部分将介绍模型检测工具 UPPAAL 用到的分支时序逻辑 CTL 及其输入模型时间自动机的相关概念，并给出 XYZ/RE 到时间自动机的映射规则。

3.1　分支时序逻辑 CTL 及时间自动机简介

UPPAAL 是 1995 年瑞典 Uppsala 大学与丹麦 Aalborg 大学联合开发的实时系统建模和验证工具[4]，具有可视化描述界面，包括编辑器、模拟器和验证器 3 个部分。

UPPAAL 使用分支时序逻辑 CTL 的子集描述待满足的性质。CTL 公式中有时序算子和路径算子两种算子，每个时序算子前必须有一路径算子。时态算子用于描述沿着路径所具有的性质：X(下一状态)，F(将来某些状态)，G(总是)和 U(直到…为止)，路径算子分别是 A(对于所有的路径)和 E(存在某些路径)。若 AP 是原子命题，则状态公式和路径公式的构造分别如下：

(1)若 $p\in AP$，则 p 是一个状态公式。

(2)若 f 和 g 是状态公式，则 $\neg f$ 和 $f\wedge g$ 是状态公式。

(3)若 f 是路径公式，则 $E(f)$ 和 $A(f)$是状态公式。

(4)若 f 是状态公式，则 f 也是路径公式。

(5)若 f 和 g 是路径公式，则 $\neg f, f\wedge g, Xf, Ff, Gf$ 及 fUg 是路径公式。

下面给出时间自动机的相关语法及语义。

定义 1　对于一个时钟变量集 X，时钟约束 φ 的集合记为 $\varphi(X)$，φ 文法定义如下：

$$\varphi:=x\leqslant c\mid x\geqslant c\mid x<c\mid x>c\mid\neg\varphi\mid\varphi_1\wedge\varphi_2$$

其中，$x\in X$ 为时钟变量，$c\in\mathbb{R}^+$ 为非负实数常量，φ_1 和 φ_2 是时钟约束公式。

定义 2(时间自动机)　时间自动机(Timed- Automata，TA)是一个六元组$\langle L,L0,\Sigma,X,I,E\rangle$，其中

L：有穷状态的集合。

$L0$：开始状态集合，且 $L0\in L$。

Σ：有穷符号集合。

X：有穷时钟集合。

I：是一个映射，它给 L 中的每个状态指定 $\varphi(X)$中的一个时钟约束。

E：是一个转换集合，且 $E\subseteq L\times\varphi(X)\times\Sigma\times 2^X\times L$。$\langle l,\varphi,a,\gamma,l'\rangle$表示输入符号 a 时，从状态 l 到状态 l'的转换；φ 是 X 上的一个时钟约束，它在转换发生时被满足；$\gamma\subseteq X$ 是在该转换发生时复位零值的时钟集合。

时间自动机 TA 的语义可通过一个状态迁移系统 S_{TA} 定义。S_{TA}的一个状态是一个二元组(l,v)，l 是 TA 的一个状态，v 是满足 $I(l)$的一个时钟解释。如果 l 是 TA 的一个初始状态，且对于所有的时钟 x，$v(x)=0$，则状态 (l,v) 是一个初始状态。转换分为以下两种：

1)因时间流逝的转换。对一个状态(l,v)和一个实数值时间增量 $\delta\geqslant 0$，如果对于所有 $0\leqslant\delta'\leqslant\delta$，$v+\delta'$满足 $I(l)$，那么$(l,v)\rightarrow(l',v+\delta)$。

2)因状态改变的转换。对于一个状态(l,v)和一个转换$\langle l,\varphi,a,\gamma,l'\rangle$，如果 v 满足 φ，那么$(l,v)\rightarrow(l',v[\gamma=0])$。

3.2　XYZ/RE 到时间自动机的映射

目前已有 XYZ/RE 到时间自动机映射方面的相关研究，

文献[6]将时间自动机中一个位置上的时间局限在$\{0,u\}$内，因此无法对转移发生最短时间为l时的情形给出更具体的映射规则。l指算子从可能执行到实际执行开始之间所需要的最短时间，它在时间自动机中应该映射为转移发生的充分条件，而不是必要条件。这在实际应用中是有一定局限性的。本文对此工作做了进一步的补充，使时间自动机在其中一个位置上时间的允许范围限制在$\{l,u\}$内，而不局限在$\{0,u\}$内。基于此，本文给出规则映射，如表1所列。

3.3 基于UPPAAL的Web服务组合模型检测方法

本文从软件体系结构角度出发，考虑到Web服务组合中的实时特征，在使用软件体系结构描述语言XYZ/ADL描述Web服务组合的基础上，将其中的实时描述部分XYZ/RE转换至时间自动机，Web服务组合系统的行为描述转换至多个时间自动机，将此多个时间自动机作为模型检测工具UPPAAL的一个输入。组合后系统应满足的性质由CTL公式表示，从而实现对运行之前Web服务组合正确性的验证。基本思想如图1所示。

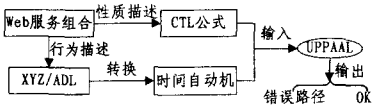


图1 SAS系统各端交互图

4 实例分析

本部分以一个股票分析服务(Stock Analysis Service, SAS)为例。该服务组合中包括3个交互端，分别是Investor, StockBroker和ResearchDept，各交互端交互流程如图2所示。具体如下：首先，Investor发送一个regist请求给StockBroker，其中包含一组stockID以及股票价格stockprice等信息。若StockBroker拒绝该请求，则发送reject消息给Investor；若接受，则返回accept消息，并发送request请求给ResearchDept，其中包含一个stockID。ResearchDept接收到request请求后，将该stockID股票的分析结果report反馈给Investor。Investor收到report后，决定是继续还是取消该服务。若继续，则发送一个ack确认消息给StockBroker。否则发送cancel消息给StockBroker，StockBroker给ResearchDept发送一个terminate信息，结束整个服务。不断重复此过程，直至所有stockID都被处理结束。StockBroker若收到ack消息则发送一个账单bill消息给Investor，并且发送request消息给ResearchDept，请求继续分析的服务。StockBroker收到cancel消息后发送terminate消息给ResearchDept，结束整个服务。

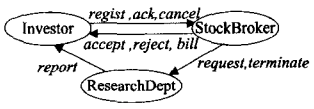


图2 SAS系统各端交互图

上述交互中的时间约束如下：Investor在该服务启动5个时间单位内发送regist请求给StockBroker，StockBroker在接到该请求10个时间单位内决定是接受或拒绝该请求。ResearchDept接收到request 7个时间单位内分析该股票并将report发送给Investor。Investor收到report 3个时间单位内决定是继续或取消该服务。StockBroker接收到ack消息请求后在5个时间单位内反馈一个bill消息给Investor。若接收到cancel消息，则仍是在5个时间单位内发送一个terminate消息给ResearchDept，结束整个服务。

以下使用XYZ/ADL对上述交互行为描述如下：

%COMPONENT Investor= =[
LB=I0∧regist! ⇒{0,5}$OLB=I1;
LB=I1∧reject? ⇒OLB=I0;
LB=I1∧accept? ⇒$OLB=I2;
LB=I2∧report? ⇒$O(LB=I3∧x=0);
LB=I3∧ack! ⇒{0,3} $O LB=I4;
LB=I3∧cancel! ⇒{0,3}$O LB=I5;
LB=I4∧bill! ⇒$O LB=I0;
LB=I5⇒$O LB=I0;]
%COMPONENT StockBroker= =[
LB=S0∧regist? ⇒$O LB=S1;
LB=S1∧accept! ⇒{0,10} $O LB=S2;
LB=S2∧request! ⇒$O LB=S3;
LB=S3∧ack? ⇒$O (LB=S4∧y=0);
LB=S4∧bill! ⇒{0,5}$O LB=S0;
LB=S3∧cancel? ⇒$O (LB=S5∧y=0);
LB=S5∧terminate! ⇒{0,5} $O LB=S0;]
%COMPONENT ResearchDept= =[
LB=R0∧request? ⇒$O (LB=R1∧z=0);
LB=R0∧terminate? ⇒$O LB=R0;
LB=R1∧report! ⇒{0,7}$O LB=R0;]

根据第3部分的映射规则，将上述XYZ/RE部分进行转换得到的时间自动机如图3所示。UPPAAL中死锁、安全性、活性的验证如图4所示，分别用CTL公式表达如下。

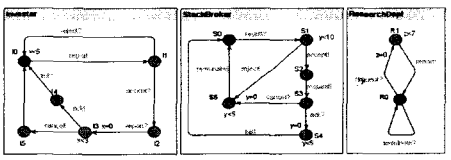


图3 SAS服务系统各交互端的时间自动机模型

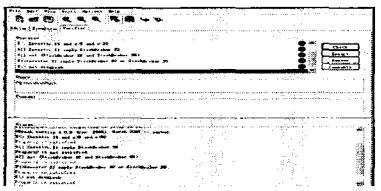


图4 SAS服务组合系统验证结果

(1)死锁：$E<>$not deadlock，验证结果为真，即该组合系统可能不存在死锁。

(2)安全性：$A[]$ not (StockBroker. $S2$ and StockBroker. $S5$)，验证结果为真，即StockBroker不可能同时到达接受和拒绝regist请求的状态。

(3)活性：$E<>$ Investor. $I1$ imply StockBroker. $S2$ or StockBroker. $S5$，验证结果为真，即Investor若发出regist请求则StockBroker可能接受该请求，也可能拒绝该请求。

$E<>$ Investor. $I5$ and $x>5$ and $x<20$，验证结果为真，即Investor可能在5～20个单位时间内完成从发出regist请求到取消结束服务的操作。

5 相关工作

Web服务组合是Web服务领域的研究热点。虽然已有一些验证Web服务组合方面的相关工作，但从体系结构角度

验证 Web 服务组合，则是一个新的研究方向。文献[7]指出在 Web 服务体系结构方面国际上仍然缺乏相关的研究；文献[8]提出需要从软件体系结构的角度研究 Web 服务组合及其验证的观点，并指出了组合服务体系结构的基本元素；文献[9]提出一种专门描述 Web 服务组合系统的体系结构描述语言 WSC/ADL，但与已有的软件体系结构描述语言相比，该语言还不够成熟，也未给出如何验证 Web 服务组合的工作。

本文从软件体系结构角度出发，并考虑到 Web 服务组合的实时特征，最后应用一种模型检测工具 UPPAAL 来实现对运行之前的 Web 服务组合的正确性验证。

结束语 由于 Web 服务处于一种分布式异构环境下，依靠实际运行发现其中错误的代价会较高，因此有必要从形式化角度验证 Web 服务组合的正确性。模型检测作为形式验证的一种重要技术，因其自动化程度高并且在不满足系统性质时能够给出反例，故应用越来越广泛。本文研究是模型检测在 Web 服务组合中的一个初步应用。本文未考虑到 Web 服务中涉及到多种复杂机制，如异常处理及补偿机制等以及如何缓解模型检测的 Web 服务组合状态空间爆炸问题，下一步工作将考虑解决这些问题。

参 考 文 献

[1] 唐稚松，等. 时序逻辑程序设计与软件工程[M]. 北京：科学出版社，2002

[2] 朱雪阳，唐稚松. 基于时序逻辑的软件体系结构描述语言 XYZ/ADL [J]. 软件学报，2003，14(4)：714-720

[3] 魏慧，戎枚，张广泉. 一种基于体系结构的 Web 服务组合描述方法[J]. 计算机工程与科学，2008，30(12)：5-8

[4] Behrmann G, David A, Larsen K G. A Tutorial on UPPAAL [C]// Int'l School on Formal Methods for the Design of Computer, Communication, and Software Systems. SFM-RT 2004, LNCS 3185, Springer-Heidelberg, 2004: 200-236

[5] Alur R, Dill D L. A Theory of Timed Automata[J]. Theoretical Computer Science, 1994, 126(2): 183-235

[6] 刘珊艳，张广泉. 实时时序逻辑语言 XYZ/RE 到时间自动机的映射[J]. 微计算机应用，2008，29(6)：69-75

[7] Vinoski S. WS-Nonexistent Standards [J]. IEEE Internet Computing, 2004, 8(6): 94-96

[8] Milanovic N, Malek M. Architectural Support for Automatic Service Composition[C]// Proc. IEEE Int'l Conf. Service Computing (SCC 2005). Washington, DC, 2005: 133-140

[9] 杨鑫，陈俊亮. WSC/ADL：Web Services 组合系统体系结构描述语言[J]. 软件学报，2006，17(5)：1182-1194

(上接第 98 页)

把签名发给 V_i。投票者 V_i 对签名解盲从而获得签名 $S_{SK_{AC}}(v_i)$。

投票阶段：V_i 选择两个随机数 r_i, y_i，并用 SC 的公钥 h 对他填好的选票 m_i 进行加密，并计算 $b_i = v_i // s_i // (g^{y_i}, m_i h^{y_i})$，$V_i$ 通过一个投票站将 b_i 发送给 TC。获得 b_i 后，TC 可以获取 v_i, s_i 和 $(g^{y_i}, m_i h^{y_i})$。TC 验证 v_i 的签名；它不允许假名 v_i 投两次票。TC 将 v_i 存入到只有 TC 可以编辑的一个网页中；这个网页对所有的投票者公开，投票者可以通过访问这个网页查看假名以确认自己的选票是否发给 TC。同时，TC 将 $(g^{y_i}, m_i h^{y_i})$ 重加密为 $(g^{z_i}, m_1 h^{z_i})$，并把这些 $(g^{z_i}, m_1 h^{z_i})$ 存入到它的被保护的数据库中。

计票阶段：投票阶段一结束，TC 把这些重加密后的密文 $(g^{z_1}, m_1 h^{z_1}), (g^{z_2}, m_2 h^{z_2}), \cdots, (g^{z_n}, m_n h^{z_n})$ 以随机顺序发给 SC。对于这些密文，首先，从 k 个政党 $SC_1, SC_2, \cdots, SC_k$ 中取出 t 个政党恢复私钥 x，接着利用 x 去解密每一个密文以获得选票。最后 SC 统计所有选票并公布此次投票结果。

4.3 安全性分析

下面说明我们的方案满足引言中所列的安全要求。

正确性：因为使用了 SSL，在我们的方案中选票在通过因特网传输时，不会被修改、复制和删除。TC 的可信与选票的门限解密保证了每一张选票会被正确地统计。

公平性：因为使用了 SSL，在选举结果出来前每一张选票都是被保密的。没有人能在选举中途知道选举结果。

合格性和唯一性：通过验证假名的签名，这两个要求可以被满足，只有合格的投票者才能投票，每个合格的投票者只能投一次。

不可强迫性：投票者只能出示从 TC 维护的网页中获得的他选取的假名，而 SC 只解密从 TC 处获得的打乱的重加密的密文，因此，投票者不能向别人证明他投的是什么票。

匿名性：在认证阶段我们的方案使用了盲签名，从而可以确保 AC 不能将假名与投票者联系起来；在投票阶段，选票是通过投票站投的，也可以提供匿名支持；在计票阶段，因为我们使用了重加密，所以 SC 不能将一个选票与投票者联系起来。

可验证性：投票者可通过访问 TC 管理的网页查看自己的假名是否在这个网页上以确认自己的选票是否被准确地统计。

结束语 在本文中，我们指出了 CJC 电子选举方案的安全漏洞，给出了一个修正方案，修正后的方案克服了 CJC 方案的安全漏洞。最后指出在我们的修正方案中可以用 Mixnet[1] 代替投票站以提高灵活性。

参 考 文 献

[1] Chaum D. Untraceable electronic mail, return addresses and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84-88

[2] Fujioka A, Okamoto T, Ohta K. A practical secret voting scheme for large scale elections [C] // Proceeding of Cryptology-AUCRYPT'92. Springer, 1993: 15-19

[3] Okamoto T. Receipt-free electronic voting schemes for large scale elections [C] // Proceeding of Security Protocols Workshop. Springer, 1997: 25-35

[4] Dini G. A secure and available electronic voting service for a large-scale distributed system [J]. Future Generation Comput Syst, 2003, 19(1): 69-85

[5] Liaw H. A secure electronic voting protocol for general elections [J]. Computer & security, 2004, 23(2): 107-119

[6] Chen Y, Jan J, Chen C. The design of a secure anonymous internet voting system[J]. Computer & security, 2004, 23(4): 330-337

[7] Cao F, Cao Z. A secure anonymous internet electronic voting scheme based on the polynomial[J]. Wuhan University Journal of Natural Sciences, 2006, 11(6): 1777-1780

[8] Bellare M, Rogaway P. Optimal asymmetric encryption [C]// Proceeding of EUROCRYPT'94. Springer, 1995: 92-111

[9] Park C, Itoh K, Kurosawa K. Efficient anonymous channel and all/nothing election scheme[C]// Proceeding of EUROCRYPT'88. Springer, 1988: 177-182

[10] Pedersen T P. A threshold cryptosystem without a trusted party (extended abstract) [C] // Proceeding of EUROCRYPT'91. Springer, 1991: 522-526