第 34 卷 第 2 期 计 算 机 工 程 2008 年 1 月
Vol.34 No.2 Computer Engineering January 2008
·软件技术与数据库· 文章编号：1000—3428(2008)02—0050—04 文献标识码：A 中图分类号：TP311

ChinaGrid 中 XML 解析方式设计

张海辉[1]，周兴社[1]，杨志义[1]，吴小钧[2]，王 涛[1]
(1. 西北工业大学计算机学院， 西安 710072；2. 长安大学信息工程学院，西安 710064)

摘 要：中国教育科研网(ChinaGrid)采用 XML 表示、交换和存储网格信息，具有种类多、信息量大等特点。常用解析方式（如 DOM、SAX 等）针对大型文档解析时，速度慢或占用大量系统资源。该文设计并实现了基于索引的解析方式，通过构造索引表加速信息查询和减少内存占用，具有快速、健壮以及占用系统资源少的特点，适应于大规模网格信息的解析。目前已应用在 ChinaGrid 的信息服务中，测试结果证明了其有效性。

关键词：信息服务；XML 解析方式；中国教育科研网格

Design of XML Parser Model in ChinaGrid

ZHANG Hai-hui[1], ZHOU Xing-she[1], YANG Zhi-yi[1], WU Xiao-jun[2], WANG Tao[1]
(1. College of Computer Science, Northwestern Polytechnical University, Xi'an 710072;
2. College of Information Engineering, Chang'an University, Xi'an 710064)

【Abstract】ChinaGrid uses XML to represent, exchange, and store grid information which is characteristic of diverse and tremendous. The common parser models, such as DOM and SAX, are inefficient or resource intensive for large XML documents. This paper presents an index-based parser model, which builds index tables to speed up information retrieve and reduce memory consumption. It is more suitable for mass grid information parsing with characters of speediness, robustness and low resource requirement. The model has been applies in information service of ChinaGrid, the results show it is effective.

【Key words】 information service; XML parser model; ChinaGrid

1 概述

中国教育科研网格[1](ChinaGrid)是中国最大的网格项目之一，2002 年由教育部实施。ChinaGrid整合分布在中国教育科研网内的各种资源，为科学研究提供高质量的服务。支撑平台CGSP2.0[2]为ChinaGrid网格应用开发者和具体网格应用提供一套开发工具。CGSP的信息服务(CGSP-IS)作为CGSP的关键服务是遵循OGSA和WSRF面向服务的体系，它需要对网格中成千上万的不同资源包括硬件、软件和数据，例如：计算机，网络，传感器，数据库和程序等进行统一管理。XML作为一套定义语义标记的规则，已成为数据交换的标准方式，具有平台和语言无关性。考虑到网格的开放性和信息的多样性，本文采用XML进行网格信息表示、交换和存储。

尽管XML有很多优点，但也有一个很大的缺点，即文档大小。XML并未将简洁性作为设计目标，其结果是XML文档可能是非标准文本或二进制格式的很多倍[3]。XML相关的压缩技术已经得到了很好研究，如XML Solutions的XMLZIP[4]，Liefke和Suciu's的XMILL[5]，以及无线访问协议标准组织的WBXML[6]和基于PPM的MHM[7]。这些研究关注了不丢失信息功能和语义的基础上减少XML文档的大小。然而在网格环境中，经常需要与一些采用原始XML表达的应用之间交换信息，因此压缩和优化并不能实用于所有场合。此外，加快XML信息访问的方法还存在优化路径表示算法[8]，增加节点标志和简化路径表示[9]，以及采用分割和动态加载技术[10]等。

Matthias[11]指出在使用XML作为数据库的项目中，XML解析器在XML应用中起了非常重要的作用，XML解析通常是主要的性能瓶颈。CGSP-IS要求提供快速的信息查询，设计过程中发现原有的DOM和SAX方式在信息量大时存在解析速度慢或占用资源多的缺点。

2 现有解析方式分析

XML 解析器是一个基本但又非常重要的工具。目前，广泛使用的解析器有：IBM 的 XML4J，微软的 MSXML，Oracle 的 XML Parser，Sun 的 JavaTM Project X 和一些开放源代码的解析器如 Expat，OpenXML，Xerces，SXP 等。大部分 XML 解析器实现了 SAX 或 DOM 接口。

2.1 文档对象模型(DOM)

DOM 是标准的基于树型 API 规范，由 W3C 创建，并且是该协会的正式建议书。使用 DOM 对 XML 文本文件进行操作时，它将文档中的元素、属性、注释、处理指令都看作节点(Node)，在内存中以树的形式展现了 XML 文档的逻辑结构。DOM 提供平台无关的接口，允许应用动态访问和更新文档内容和结构，DOM 允许应用执行树形操作。DOM 分析器的树形结构与 XML 文档的结构相吻合，得到了广泛使用。

因为DOM解析器需要构造XML内部树形结构，它占用的内存与文档大小成正比（一般是 2~5 倍），因此不适合大型文档[11]。同时，DOM解析器构造树时，它将考虑所有对象，如

基金项目：国家部委基金项目；国家自然科学基金资助项目(2001CG1101)；教育部中国教育科研网格计划 ChinaGrid 资助项目
作者简介：张海辉(1977－)，男，博士研究生，主研方向：分布计算及网格计算；周兴社，教授、博士生导师；杨志义，教授；吴小钧，讲师、博士；王 涛，博士研究生
收稿日期：2007-01-30 **E-mail：**zhanghh@mail.nwpu.edu.cn

元素、文本和属性，如果应用只关注其中一小部分时，其他从未使用或很少使用的对象将占用大量的资源，而且，对于结构复杂的树的遍历也是一项比较耗时的操作。

如何减少DOM的内存占用和提高访问效率最近得到了广泛关注和研究。延迟DOM解析器只有在实际访问文档树时，才构造相应部分，如果文档大部分都需要访问时，延迟DOM将比普通DOM方式更慢；DDOM[12]采用线性表的方式存储所有节点信息，与普通DOM实现方式相比，可以节省 30%~80%的内存占用；SEDOM[13]结合了压缩的方式，减少DOM的内存占用，并优化XML的访问效率，但带来了额外的压缩程序内存开销和性能影响。

2.2 XML 简单 API(SAX)

SAX(XML 简单应用程序接口)是由 XML_DEV 邮件列表中的成员根据应用的需求自发定义的一套对 XML 文档进行操作的接口规范。使用 SAX 分析器对 XML 文档进行分析时，会触发一系列事件，并激活相应的事件处理函数，从而完成对文档的访问。SAX 可以解析任意大小的文件，实现简单，占用内存少，当不需要改变文档的内容时，效率比较高。它可能不需要遍历整个文档就能获取相应数据。通常需要随机访问时采用 DOM 解析器，而顺序访问时 SAX 更优。

另一方面，SAX 存在的不足有：(1)和应用高度相关；(2)SAX 事件是无状态的，因此，需要查询分布在文档中的多个元素时，将重复遍历文档；(3)更重要的是事件仅仅是发现元素，而应用必须编写大量充满 IF/ELSE 结构的回调接口；(4)SAX 对文档只读，不能进行随机存取，难以实现复杂查询。

3 基于索引的解析方式 IBP

在 ChinaGrid 信息中心的设计中，发现描述网格信息的 XML 文档包含一个或者多个关键标签，类似于关系数据库中的索引。因此，在解析过程中引入索引机制，并以此加速文档的检索。对 XML 文档操作前，有一个初始化过程，这期间通过指定子树节点和索引节点，创建关键标签索引表和元素子树索引表。然后，IBP 允许应用程序基于这些表执行查询操作。IBP 类似于 DDOM，采用索引表的方式记录节点信息，但其不建立内存中树形结构，不记录所有节点信息，而是记录关键元素信息，查找时通过定位到元素子树后，通过遍历子树得到目标信息。

以下以 BookSet.xml 文档为例，说明 IBP 方式的基本处理过程，该文档符合 DTD 描述：

```
<?xml version="1.0" encoding="GB2312" ?>
<!ELEMENT BookSet (Book*)>
<!ELEMENT Book(ISBN, Name, Author+, Price*)>
<!ELEMENT ISBN(#PCDATA)>
<!ELEMENT Name(#PCDATA)>
<!ELEMENT Author(#PCDATA)>
<!ELEMENT Price(#PCDATA)>
<!ATTLIST Price currency (dollar | RMB | pound) 'Dollar'>
```

IBP 解析器遍历 XML 数据，并将整个文档以指定元素为根拆分成许多元素子树。在这个文档中，最优的拆分元素为“Book”，标记所有起始标签(<Book>)和结束标签(</Book>)的位置形成元素子树索引表。如果知道哪个元素子树包含所需要的内容，只需要搜索特定子树。如何确定元素子树则是采用关键标签索引表。采用所需要的任意标签创建索引，推荐采用具有唯一值的元素。此例中，本文采用 ISBN 作为关键标签，并记录“<ISBN>”和“</ISBN>”之间的文本作为索引值。初始化过程以后，IBP 创建了元素子树索引表和关键标签索引表（如图 1）。

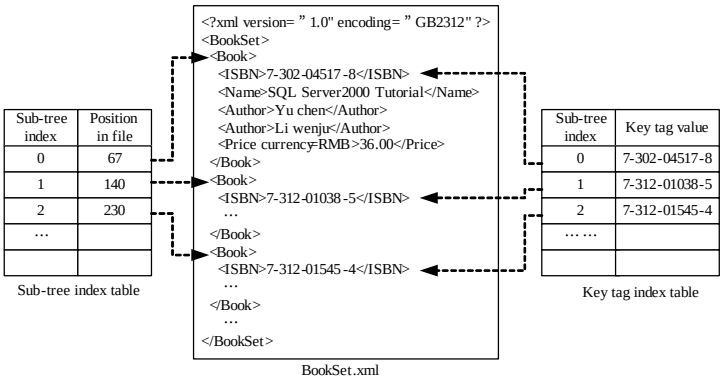


图 1 初始化后形成索引表

如果想查询书本 ISBN 号码为“7-302-04517-8”的作者姓名，只需要载入和搜索位置从 67 到 140，总共只有 73 个字符的文档。

在 ChinaGrid 中，根据网格信息的包含关系和层次结构，将其组织成一个树型的全局资源文档 GridInformation.xml，如图 2 所示。

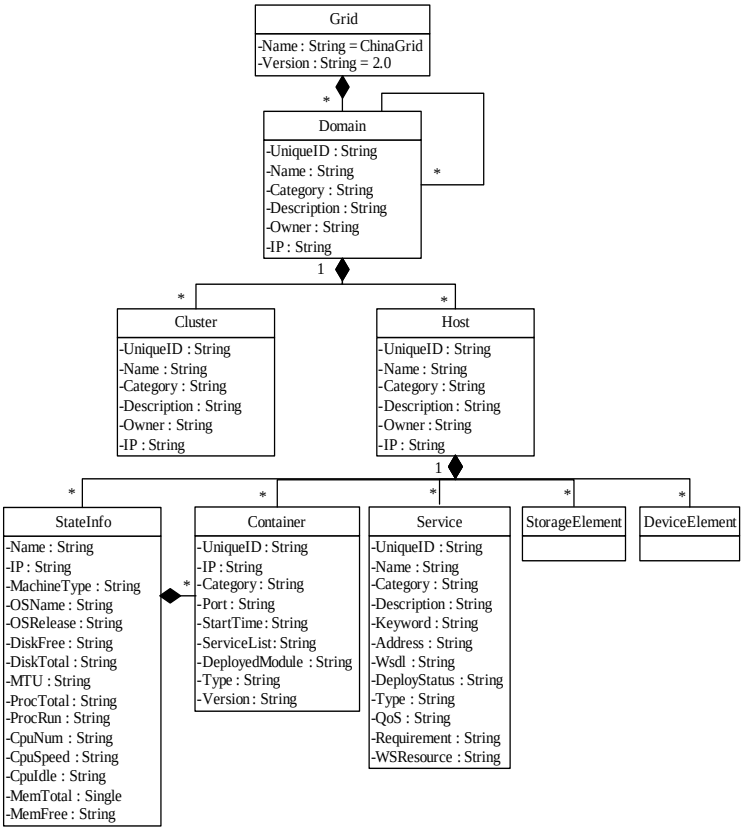
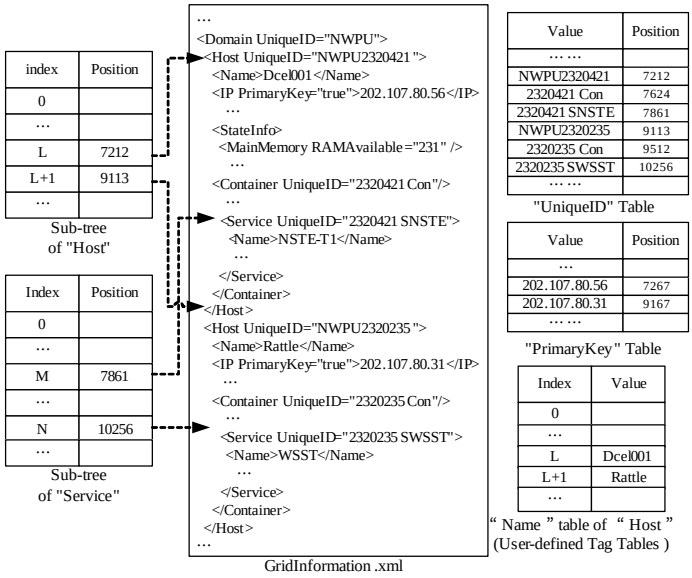


图 2 全局资源文档结构

初始化后，其形成了如图 3 所示的索引表。

图 3 ChinaGrid 信息索引表

解析器根据是否支持 DTD 或 Schema 验证，分为支持验



证和非验证。非验证的解析器只检查 XML 文档是否合乎语法(well-formed)，而支持验证的解析器可以检查 XML 文档是否符合 DTD 约束。

非验证 IBP 方式(IBP-nv)工作过程如上所述，XML 文档是只读的。它适合于 XML 数据库查询，搜索工具和过滤器等 SAX 使用场合。但其比 SAX 更有效率，因为通过初始化形成元素子树后，可快速定位到特定元素子树，并进行小范围查找，而 SAX 通常需要从文件开始处遍历文档。初始化过程后，可以关闭 XML 文档，在需要的时候重新打开文件，根据元素子树索引表读入指定区域的内容。IBP-nv 占用很少的内存，而且并不随着文件变大而增加。

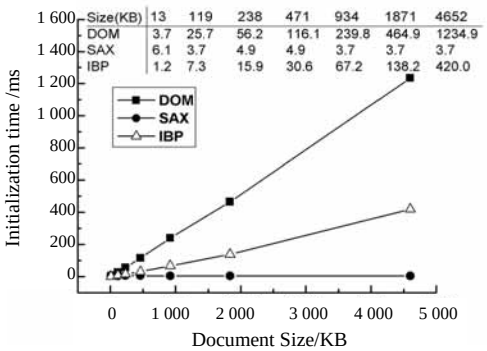
验证的 IBP 方式(IBP-v)允许应用执行更新操作，如节点增加、修改、转换和删除，该模式下，XML 文档保存在内存中。初始化过程后，XML 文档分配了一块连续的内存空间，如果一个元素需要改变，元素子树的内容先必须拷贝到一块重新分配的内存空间，然后将元素子树的内存指针赋值新的地址，并在元素更新后将“Reallocated”标志改为“真”。这个模式的关键在于元素子树被当成一个独立分配空间的存储单位，内存指针永远指向最新的内存区域。关闭 XML 文档时，所有的元素子树将依次写入文件。

以 XML 文档是否需要更新来决定使用 IBP-nv 或 IBP-v。如果只是遍历和搜索文档，IBP-nv 具有更好的性能。对 IBP-v 而言，由于文档更新导致重新遍历和验证的开销相对较大。通过试验，发现以验证方式即使遍历一个很小的 XML 文档，也将使 CPU 开销增加 2~3 倍或更多。

4 XML 解析器性能分析

解析器性能通常取决于文档特征，如标签和数据比率，属性使用程度[11]，元素子树数目以及平均元素子树大小等。本文暂时不对这些特性进行量化。在项目中，笔者设计和实现了IBP解析器。微软MSXML在SQL Server、IE等应用中得到了广泛应用，并同时支持SAX和DOM方式，将两者的性能进行比较。

以 GridInformation.xml 为例，创建 7 个 XML 文档，大小分别为 13 KB, 119 KB, 238 KB, 471 KB, 934 KB, 1 871 KB, 4 652 KB，它们包含了不同数量的元素。然后量化比较采用 MSXML4.0 和 IBP 进行初始化和解析的时间开销。



4.1 初始化时间分析

XML 文档分别采用 DOM、SAX 和 IBP 初始化 100 次，平均时间如图 4。

图 4 DOM，SAX 和 IBP 初始化时间比较

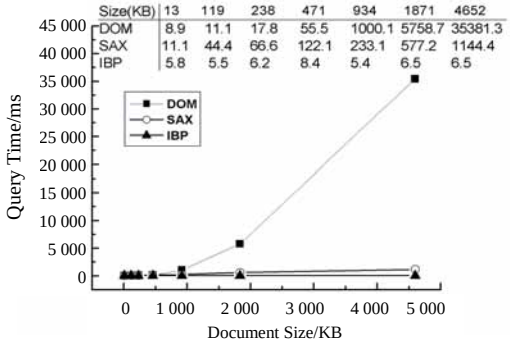
在初始化过程中，DOM 需要遍历和解析 XML 文档形成一个内存中的文档树，初始化时间与 XML 文档的大小成正比。SAX 只是创建了打开文件的句柄，但并不真正读取数据，因此，其开销是最少的。IBP 需要解析和建立索引表，需要一定的时间开销。

4.2 解析开销

7 个 XML 文档在不进行语法缓存的情况下，随机查找 1 000 次，平均解析时间反映了解析器整体性能，如图 5 所示。

图 5 DOM, SAX 和 IBP 解析时间比较

DOM 解析器在小文档解析时具有良好的性能，但针对大文档的开销远远超出了预计。SAX 解析器解析性能优于 DOM



方式，仍然效率低下。进一步的研究发现 DOM 和 SAX 针对 GridInformation.xml 文档结构，当文档为 740 KB 时，具有近似的性能。在 IBP 初始化过程中，创建了元素子树表和关键标签索引表，解析过程包含 2 个部分：查找索引表和在特定元素子树中进行元素匹配。IBP 采用哈希函数优化索引表，查找索引表时间几乎是不变的，搜索元素子树的时间只与子树的大小相关。因此，IBP 解析器在大规模的 XML 文档下也能有很好的性能。

在最坏情况，IBP 需要以新的索引标签重新初始化索引表。将 DOM 和 SAX 解析时间与 IBP 初始化和解析时间之和进行比较，IBP 性能仍大大优于 DOM 和 SAX 方式，见表 1。

表 1 DOM 和 SAX 解析时间与 IBP 初始化和解析时间之和的比较

Document size/KB	DOM	SAX	IBP
13	8.9	11.1	7.0
119	11.1	44.4	12.8
238	17.8	66.6	22.1
471	55.5	122.1	39.0
934	1 000.1	233.1	72.6
1 871	5 758.7	577.2	144.7
4 652	3 5381.3	1 144.4	426.5

5 结束语

本文讨论了通用的 XML 解析器模型 DOM 和 SAX，发现它们不符合大型 XML 文档的需求。针对网格计算系统要求，提出了基于索引的快速解析方式。它具有资源占用少、解析速度快的优点。对于大型 XML 文档，IBP 的解析速度远远快于 DOM 和 SAX。通过提供通用接口，该方式能广泛应用于各种 XML 文档的解析，为 XML 文本分析提供了一种新的思路。

后续的工作将进一步在理论和实践方面完善 IBP，开发符合 XML 规范的完整 IBP 应用程序接口，继续完善多重索引和混合索引，将 IBP 应用于 XML 数据库。设计和实现基于 IBP 的 XPath 或 XQuery 查询引擎，增强应用的灵活性。

参考文献

[1] Jin H. ChinaGrid: Making Grid Computing a Reality[C]//Proc. of the 7th International Conference of Asian Digital Libraries. Shanghai, China: [s. n.], 2004: 13-24.

[2] ChinaGrid Support Platform[EB/OL]. (2006-12-30). http://www.chinagrid.edu.cn/CGSP.

[3] Cheney J. Compressing XML with Multiplexed Hierarchical PPM Models[EB/OL]. (2000-11-20). http://www.cs.cornell.edu/People/jcheney/xmlppm/paper/paper.html.

[4] XML Solutions[EB/OL]. (2006-11-20). http: //www.xmls.com/.

[5] Liefke H, Suciu D. An Efficient Compressor for XML Data[C]// Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data. New York, NY, USA: [s. n.], 2000: 153-164.

[6] WAP Binary XML Content Format, W3C Note[Z]. (1999-06-24). http://www.w3.org/TR/wbxml/.

[7] Cheney J. Compressing XML with Multiplexed Hierarchical Models[C]//Proceedings of the IEEE Data Compression Conference. Snowbird, Utah: [s. n.], 2001: 163-172.

[8] Li Quanzhong, Moon B. Indexing and Querying XML Data for Regular Path Expressions[C]//Proceedings of the 27th International Conference on Very Large Databases. Roma, Italy: [s. n.], 2001: 361-370.

[9] Bremer J. Gertz M. An Efficient XML Node Identification and Indexing Scheme[D]. California: Department of Computer Science University of California, 2003.

[10] 孙 静, 宋 扬, 胡金星, 等. 大型 XML 文件的分割和动态加载研究[J]. 计算机工程与应用, 2003, 39(16): 107-108,125.

[11] Nicola M, John J. XML Parsing: A Threat to Database Performance[C]//Proc. of the 12th Intl Conference on Information and Knowledge Management. New Orleans: [s. n.], 2003-11.

[12] DDOM Project[Z]. DDOM API Reference. (2004-10-20). http://www.cis.strath.ac.uk/~mathias/ddom.

[13] Wang Fangju, Li Jing, Homayounfar H. A Space Efficient XML DOM Parser[J]. Data & Knowledge Engineering, 2007, 60(1): 185-207.

(上接第 36 页)

综上所述，可以得到重新设计后的协议如下：

M1 $A \rightarrow S : A, B, N_a$

M2 $S \rightarrow B : A$

M3 $B \rightarrow S : N_b^0$

M4 $S \rightarrow A : \{N_a, B, K_{ab}\}_{K_{as}}$

M5 $S \rightarrow B : \{K_{ab}, A, N_b^0\}_{K_{bs}}$

M6 $B \rightarrow A : \{N_b, K_{ab}\}_{K_{ab}}$

M7 $A \rightarrow B : \{K_{ab}, N_b - 1\}_{K_{ab}}$

可以用本文的逻辑来分析该协议，在分析时，不需要初始条件$(A12)$ $B \mid\equiv \#(K_{ab})$ 就可以得出该协议能满足需要的目标。具体分析过程从略。

6 结束语

本文提出了一种分析和设计认证协议的新逻辑，可以在同一个逻辑框架中对认证协议进行分析和设计，消除了用一种方法来设计认证协议而用另一种方法来分析协议的不一致性。通过新逻辑的运用，可以将认证协议的初始条件直接运用到认证协议的分析和设计中，减少了初始条件的形式化过程，提高了工作效率。另外，还提出了一序列的合成规则，使协议设计者在设计协议时可用一种系统化的方法来构造满足需要的协议，使协议设计者可以方便地进行协议的形式化设计。

参考文献

[1] Needham R M, Schroeder M D. Using Encryption for Authentication of Large Networks of Computers[J]. Communication of the ACM, 1978, 21(12): 993-999.

[2] Denning D E, Sacco G M. Timestamps in Key Distribution Protocols[J]. Communications of the ACM, 1981, 24(8): 198-208.

[3] Dolev D, Yao A. On the Security of Public Key Protocols[J]. IEEE Transactions on Information Theory, 1983, 29(2): 198-208.

[4] Burrows M, Abadi M, and Needham R. A logic of Authentication[J]. ACM Transactions on Computer Systems, 1990, 8(1): 18-36.

[5] Heintze N, Tygar J D. A Model for Secure Protocols and Their Composition[J]. IEEE Transaction on Software Engineering, 1996, 22(16): 100-112.

[6] Li Gong, Syverson P. Fail-stop Protocols: An Approach to Designing Secure Protocols[C]//Proc. of the 5th International Working Conference on Dependable Computing for Critical Applications. Santa Barbara, USA: [s. n.], 1995.

[7] Buttyan L, Staamann S, Wilhelm U. A Simple Logic for Authentication Protocol Design[C]//Proceedings of the IEEE Computer Security Foundations Workshop XI. [S. l.]: IEEE Press, 1998.

[8] Rudolph C. A Formal Model for Systematic Design of Key Establishment Protocols[C]//Proceedings of ACISP'98. Koyto, Japan: [s. n.], 1998.

[9] Guttman J D, Thayer F J. Authentication Tests[C]//Proceedings of the 2000 IEEE Symposium on Security and Privacy. Los Alamitos: IEEE Computer Society Press, 2000: 150-164.