摘 要

网格通信是构建在互联网上的一组新兴技术，通过它可以把分散在不同地理位置的资源整合起来，实现各种资源的全面共享。网格资源分布的广泛性决定了通信安全是一个重要环节。作为网格通信安全的基础，安全认证密钥协商机制是研究的热点问题之一。

基于无证书的公钥密码体系（CL-PKC）结合了基于身份公钥密码体系（ID-PKC）和传统公钥密码体系（PKI）的优点，非常适合网格通信环境，能够满足网格安全的需要。在拾以娟所提的无证书认证密钥协商协议的基础上，本文给出了一种新的基于无证书的认证密钥协商协议方案，在网格用户和资源间进行双向认证，同时提供了简单实用的密钥管理方式，适用于网格安全结构（GSI）。与已存在的认证密钥协商协议相比，所给协议的安全性和运行效率更高。论文中分析了所给协议的运行效率和实用性，同时，证明了所给协议不但具有一般认证密钥协商的各种安全属性，如已知密钥安全性、密钥控制安全性，还具有完善的前向安全性、已知临时私钥的信息安全性，并且能够抵抗密钥复制攻击。

关键词：无证书　认证密钥协商　公钥密码体系　网格　双向认证

Abstract

Grid communication is a set of emerging technologies building on the Internet, which can integrate the resources in the different geographical regions to achieve the comprehensive sharing. The communication security is an important part because of the widely distributed grid resources. As the foundation of Grid communication security, research on the authentication and key agreement mechanism is one of the hot topics.

The certificateless public-key cryptography (CL-PKC), which combines the advantages of the identity-based cryptography (ID-PKC) and the public key infrastructure (PKI), has some attractive properties which seem to align well with the demands of grid communication. In this thesis, based on the Shiyijuan's certificateless authentication and key agreement protocol, a new certificateless authentication and key agreement (CL-AK) protocol for Grid is presented, which fits well with the Grid Security Infrastructure (GSI) and provides a more lightweight key management approach for grid users. It authenticates grid members and to secure resource allocation to these members. Compared with the existing protocol, this protocol is more efficient, and has a higher security. The efficiency and practicality of new protocol are analyzed in this thesis. It is proved that the presented protocol provides perfect forward secrecy and known session-specific temporary information security plus all the other security attributes of authenticated key agreement protocols such as known-key secrecy and no key control. Moreover, the presented protocol can resist Key replicating attack.

Keyword: Certificateless AKA PKC Grid Mutual Authentication

西安电子科技大学

学位论文独创性（或创新性）声明

秉承学校严谨的学风和优良的科学道德，本人声明所呈交的论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除了文中特别加以标注和致谢中所罗列的内容以外，论文中不包含其他人已经发表或撰写过的研究成果；也不包含为获得西安电子科技大学或其它教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中做了明确的说明并表示了谢意。

申请学位论文与资料若有不实之处，本人承担一切的法律责任。

本人签名：王永强　　日期 2010.3.15

西安电子科技大学

关于论文使用授权的说明

本人完全了解西安电子科技大学有关保留和使用学位论文的规定，即：研究生在校攻读学位期间论文工作的知识产权单位属西安电子科技大学。学校有权保留送交论文的复印件，允许查阅和借阅论文；学校可以公布论文的全部或部分内容，可以允许采用影印、缩印或其它复制手段保存论文。同时本人保证，毕业后结合学位论文研究课题再撰写的文章一律署名单位为西安电子科技大学。

（保密的论文在解密后遵守此规定）

本学位论文属于保密，在___年解密后适用本授权书。

本人签名：王永强　　日期 2010.3.15

导师签名：孙钦涛　　日期 2010.3.15

第一章 绪论

1.1 网格概述

1.1.1 网格的概念和特点

网格是利用计算机网络把地理上广泛分布的信息资源、计算资源、网络资源、存储资源、软件资源等连成一个逻辑整体，然后为科技人员和普通老百姓提供更多的资源、功能和一体化的信息应用服务[1]。网格的目标是把整个因特网整合成一台巨大的超级虚拟计算机，实现互联网上所有资源的互通互联，完成通信资源、计算资源、存储资源、信息资源、知识资源、软件、专家资源等的智能共享，消除信息孤岛。网格通信以其重要的战略意义以及广阔的应用前景成为当今吸引众多研究人员和巨大资金投入的研究热点。

网格通信技术被看成是未来的互联网通信技术。国外媒体常用“下一代Internet”，“Internet2”，“下一代 Web”等词语来称呼与网格相关的技术[2]。实际上，“下一代 Internet”(NGI)和“Internet2”是美国的两个具体科研项目的名字，它们与网格研究目标相交，但研究内容和重点有很大不同。中国科学院计算所所长李国杰院士认为，网格实际上是继传统因特网、Web 之后的第三个大浪潮，可以称之为第三代因特网。不同的是，传统因特网实现了计算机硬件的连通，Web 实现了网页的连通，而网格试图实现互联网上所有资源的全面连通，包括计算资源、存储资源、通信资源、软件资源、信息资源、知识资源等。

作为一个新出现的基础性设施，和传统的 Internet 等系统相比，网格通信具有一系列新的重要特点[3]，如下：

1. 分布与共享：网格的分布性首先指其资源是分布的。组成网格的计算能力不同的计算机，各种类型的数据库及电子图书馆，以及其他的各种设备与资源，是分布在地理位置互不相同的多个地方，而不是集中在一起的。分布的网格一般涉及的资源类型复杂、规模较大，跨越的地理范围较广。

网格的最终目的是实现资源的最大共享。因此可以说，没有共享便没有网格，解决分布资源的共享问题，是网格的核心内容。这里共享的含义是非常广泛的，不仅指一个地方的计算机可以用来完成其他地方的任务，还可以指中间结果、数据库、专业模型库以及人才资源等各方面的共享。

2. 自相似性：网格的局部和整体之间存在着一定的相似性，局部往往在许多地方具有全局的某些特征，而全局的特征在局部也有一定的体现。整体和部分间的相似性可以在多个阶段体现出来。

3. 动态性与多样性：网格的动态性包括动态增加和动态减少两个方面的含义。网格中原来拥有的资源或者功能，在下一时刻可能会出现故障或者不可用，而原来没有的资源或者功能，则有可能随着时间的推移不断地加入进来。

网格资源的动态增加需要提高网格的扩展性问题，也就是说在网格的设计与实现时，必须考虑到新的资源能否很自然地加入到网格中来，并且可以和原来的资源融合在一起，共同发挥作用。网格的扩展性要求体现在规模、能力、兼容性等几个方面。由于网格资源是异构和多样性的，所以网格系统必须能够解决这些不同结构、不同类别资源之间的通信和互操作问题。

4. 自治性与管理的多样性：网格上的资源，首先是属于某一个组织或者个人的，因此网格资源的拥有者对该资源具有最高级别的管理权限，网格应该允许资源拥有者对他的资源有自主的管理能力，这就是网格的自治性。但是网格资源也必须接受网格的统一管理，否则不同的资源就无法建立相互之间的联系，无法实现共享和互操作，无法作为一个整体为更多的用户提供方便的服务。因此网格的管理具有多重性，一方面它允许网格资源的拥有者对网格资源具有自主性的管理，另一方面又要求网格资源必须接受网格的统一管理。

1.1.2 网格的体系结构

网格体系结构就是关于如何构建网格的技术。它给出了网格的基本组成与功能，描述了网格各组成部分的关系以及它们集成的方式或方法，刻画了支持网格有效运转的机制。目前存在着两种最重要、极有影响的结构形式：五层沙漏结构和开放式网格服务结构[1]。

1. 五层沙漏结构

五层沙漏结构是 Globus 项目的创始人之一 Ian Foster 提出的一种影响十分广泛的结构，主要侧重于定性的描述而不是具体的协议定义，从整体上很容易理解。其基本思想是以“协议”为中心，也十分强调服务与 API(Application Programming Interfaces)和 SDK(Software Development Kits)的重要性。

五层沙漏结构和传统 TCP/IP 网络协议的对比结构如图 1.1 所示。

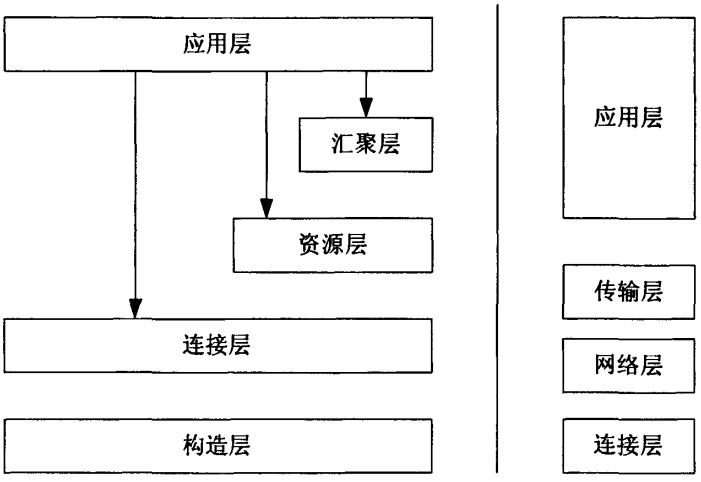

图 1.1 五层沙漏结构及其与 TCP/IP 网络协议的对比

构造层(Fabric)：它们是物理或逻辑实体局部控制的界面。它的功能是向上提供网格中可供共享的资源，常用的资源包括处理能力、存储系统、目录、网格资源、分布式文件系统、分布式计算机池、计算机集群等。

连接层(Connective)：支持便利安全的通信。它是网格中网络事务处理通信与授权控制的核心协议。各种资源间的数据交换、授权验证和安全控制都在这一层实现。在此协议中提供一次登录、委托授权、局域安全方案整合、基于用户的信任关系等功能。资源间的数据交换通过传输、路由及名字解析实现。

资源层(Resource)：共享单一资源。它的作用是对单个资源实施控制，与可用资源进行安全握手、对资源做初始化、监测资源运行状况、统计与付费有关的资源使用数据。

汇聚层(Collective)：协调各种资源。这层的作用是将资源层提交的受控资源汇集在一起，供虚拟组织的应用程序共享、调用。为了对来自应用的共享进行管理和控制，汇聚层提供目录服务、资源分配、日程安排、资源代理、资源监测诊断、网格启动、负荷控制、账户管理等多种功能。

应用层(Applications)：这层是网格上用户的应用程序。应用程序通过各层的 API 调用相应的服务，再通过服务调用网格上的资源来完成任务。

沙漏形状是五层结构的一个重要特点，其内在含义在于各部分协议的数量不同，最核心的部分要能够实现上层各种协议向核心协议的映射，同时实现核心协议向下层其他各种协议的映射，核心协议在所有支持网格计算的地点都应该得到支持，因此其数量不应太多，这样核心协议就成为协议层次结构中的一个瓶颈。在五层结构中，资源层和连接层共同组成这一核心的瓶颈部分，如图 1.2 所示。

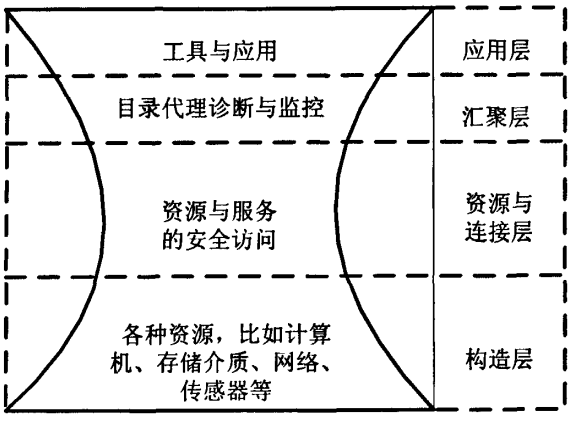

图 1.2 沙漏形状的五层结构

2．开放式网格服务体系结构：

开放式网格服务体系结构 OGSA(Open Grid Services Architecture)是一种以服务为中心的“服务结构”。在 OGSA 中，服务所指的概念广泛，包括各种计算资源、存储资源、网络、程序、数据库等。简言之，一切都是服务。为了使服务的思想更加明确和具体，OGSA 定义了“网格服务”(Service)的概念。网格服务是一种 Web Service，它提供一组接口，这些接口具有明确的定义并且遵守特定的惯例，解决服务发现、动态服务创建、生命周期管理、通知等问题。在 OGSA 中，将一切都看作是网格服务，因此网格就是可扩展的网格服务的集合，即网格={网格服务}。网格服务可以按不同的方式聚集起来满足虚拟组织(Virtual Organization)的需要，虚拟组织自身也可以部分地根据它们操作和共享的服务来定义。

Web Service 的定义是：Web Service 就是可以被 URI(Universal Resource Identifier：通用资源标志符)识别的软件应用，它的接口和绑定可以被 XML(eXtensible Markup Language)描述与发现，并且可以通过基于 Internet 的协议直接支持与其他基于 XML 消息的软件应用的交互。

以网格服务为中心的模型具有如下好处：(1)由于网格环境中所有的组件都是虚拟化的，因此，通过提供一组相对统一的核心接口，所有的网格服务都基于这些接口实现，就可以很容易地构造出具有层次结构的、更高级别的服务，这些服务可以跨越不同的抽象层次，以一种统一的方式来看待；(2)虚拟化也使得将多个逻辑资源实例映射到相同的物理资源上成为可能，在对服务进行组合时不必考虑具体的实现，可以以底层资源组成为基础，在虚拟组织中进行资源管理。通过网格服务的虚拟化，可以将通用的服务语义和行为，无缝地映射到本地平台的基础设施上。

基于网格服务的体系结构参见图 1.3 所示。

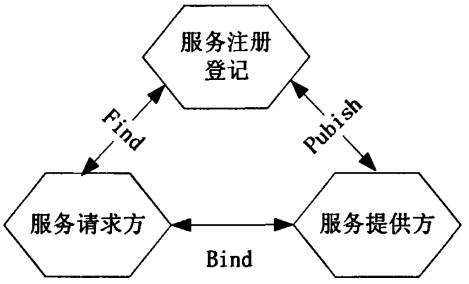

图 1.3 基于服务的开放式网格体系结构

不同平台上的网格服务通过 Publish，Find，Bind 等三个操作进行互通，为了实现这一点，就必须要有一个标准的协议栈来规范不同层次的功能互通。在该结构中，网格服务的基础是网络，一个网格服务必须可由 Requestor 通过网络进行访问；网络之上是消息管理，它包括信息表示及收发等，目前的 OGSA 采用与 Web Service 同样的消息表语言 XML；服务描述主要指服务的接口描述，包括其行为，数据等，服务描述主要是生成一个服务描述文档，其语言目前为 WSDL(Web Service Description Language：Web 服务描述语言)，它是基于 XML 的一个描述语言。至于服务的发布和发现则是一个规范工作流程。

1.2 认证技术在网格通信安全中的重要意义

网格通信技术虽然有着非常广泛的前景和发展空间，但是对它的研究还正处在起步阶段，还有很多关键技术问题需要解决。网格通信技术强大的功能在给科学研究和人们带来极大方便的同时，也不得不担心网上应用的安全性问题，从而使得我们在获取便利、快捷的功能和信息服务的同时不用担心通信信息的泄漏和被窃取。

由于网格的建立是以因特网作为通信支持的平台，而 Internet 是一个开发性、异构性极大的公共网络，这使得 Internet 上运行的网格通信作业面临各种各样的安全威胁[5]。如信息被截取、信息的内容被篡改和删除以及假冒合法用户和服务器等。

在网格通信环境中，就需要采取各种有效的安全措施来防止这样的情况发生，以确保通信信息安全。

作为网格安全的第一道防线，某种程度上也是最重要的一道防线，认证技术普遍受到关注[4]。认证技术提供了关于某个人或某个事物身份的保证，这意味着当某人（或某事）声称具有一个特别的身份（如某个特定的用户名称）时，认证技术将提供某种方法来证实这一声明是正确的，一般方法是输入个人信息，经特

定的公式和算法运算后，将所得的结果与在数据库中存储的信息经公式和算法运算所得的结果进行比较，得出结论。

认证必须做到准确无误地将对方辨认出来，同时还应该提供双向的认证，即相互证明自己的身份。

1.3 认证密钥协商协议基本概念

通信的双方或多方生成一个会话密钥的过程称为密钥建立，密钥建立属于密钥管理的一部分。密钥建立有两种方法，一种是密钥传递，即通信中的一方（一般是一个其他用户都信任的权威机构）生成一个密钥，再将其发送给其他方，从而使通信中其他方与其共享一个会话密钥。另一种方法则是密钥协商，即通信中的双方或多方共同协商出一个共享的会话密钥。在协商的过程中，每一方都对会话密钥有所贡献，同时任何一方也不能控制密钥生成为某一个特定的值。

1976 年，Diffie 和 Hellman 提出了 Diffie-Hellman 协议[7]，简称 DH 协议。下面描述了两个协议参与者 Alice 和 Bob 利用 DH 协议协商会话密钥的过程：

Alice 随机选取一个一次性密钥 $x \in Z_q^*$，计算 $X = a^x$ 并发送给 *Bob*；

Bob 随机选取一个一次性密钥 $y \in Z_q^*$，计算 $Y = a^y$ 并发送给 *Alice*；

Alice 在接收到 Y 之后，计算 $K_{AB} = Y^x = a^{xy}$；

Bob 在接收到 X 之后，计算 $K_{BA} = X^y = a^{xy}$；

如此，Alice 和 Bob 协商得到了会话密钥 $K = a^{xy}$。该协议是建立在非对称密码体制上的，其安全性是基于离散对数分解困难性。但该协议并不能抵抗中间人攻击。下面对中间人攻击 DH 协议的过程进行一下简单的描述：

假设一个攻击者 Candy 在协议的参与者 Alice 和 Bob 之间实施中间人攻击。Alice 选取了一个一次性密钥 $x \in Z_q^*$，计算 $X = a^x$ 并发送给 Bob；Candy 截获了 Alice 发送的 X，并重新选取了一个一次性密钥 $x' \in Z_q^*$，计算 $X' = a^{x'}$ 并冒充 Alice 发送给 Bob；Bob 随机选取一个一次性密钥 $y \in Z_q^*$，计算 $Y = a^y$ 并发送给 Alice；Candy 也截获了 Bob 发送的 Y，并重新选取了一个一次性密钥 $y' \in Z_q^*$，计算 $Y' = a^{y'}$ 并冒充 Bob 发送给 Alice。这样 Alice 和 Candy 共享密钥 $K_{AC} = K_{CA} = a^{xy'}$，Bob 和 Candy

共享密钥 $K_{BC}=K_{CB}=a^{xy}$。而 Alice 却错误地认为他和 Bob 共享密钥，同样 Bob 也错误地认为他和 Alice 共享密钥。可见，Candy 成功地在 Alice 和 Bob 之间实施了中间人攻击。

Diffie-Hellman 协议是第一个双方的密钥协商协议，后来很多学者提出的密钥协商协议都是以该协议为基础扩展得到的。

一个密钥协商协议，无论是双方的还是多方的，都是处在不安全的网络环境当中的。需要抵抗各种各样的攻击。一般把攻击按照攻击方式分为被动攻击和主动攻击。在被动攻击中，攻击者只是截获和窃取信息，而不对信息进行篡改。在主动攻击中，攻击者除了进行被动攻击外，还会对信息进行篡改。

为了抵抗这些攻击，大量研究者希望在密钥协商协议中加入认证机制，提出了认证密钥协商协议。如果用户 A 能够确认除了用户 B 以外的其他用户都不能计算得到会话密钥，则认为协议向用户 A 提供了用户 B 的隐含密钥认证。如果一个密钥协商协议向所有的协议参与者提供了隐含密钥认证，则把该协议称为认证密钥协商协议，即 AKA（Authenticated Key Agreement）协议[7]。在 AK 协议中，任一用户只能够确定其他用户能够计算得到会话密钥，却不能确定其他用户是否已经通过计算得到了会话密钥。如果用户 A 能够确认用户 B 已经通过计算得到了会话密钥，则认为协议向用户 A 提供了用户 B 的密钥确认。如果一个认证密钥协商协议（AKA 协议）向所有的协议参与者提供了密钥确认，则称该协议为带密钥确认的认证密钥协商协议，即 AKC（Authenticated Key agreement with key Confirmation）。

认证密钥协商协议的研究对于保证网格通信过程中的会话密钥安全，从而保证信息的安全是至关重要的。这些信息中不仅包含日常生活中的个人信息，还包括一些重要的商业秘密，甚至还包括一些军事和国家机密。可见，该方面的研究对网格通信安全有着至关重要的作用。目前，关于认证密钥协商协议的研究已经有了大量的成果。但这些成果中，有些协议具有安全性的漏洞，不适合在安全性较差的网格环境中运行；有些协议所需的计算开销较大，不适合计算能力较小的终端使用。可见，网格环境中的认证密钥协商协议具有较高的研究必要性以及广阔的研究前景。

1.4 网格认证密钥协商技术的研究现状

目前的网格认证策略是建立在 Globus 项目中的网格安全基础设施（Grid Security Infrastructure，GSI）的基础之上。Globus 项目是目前国际上最具影响的网格项目之一。GSI 通过提出网格安全基础设施来提供在网格环境中的安全认证

和安全通信功能，并且是保证网格通信网安全性的核心[17]。GSI 是一种构建在传统公钥基础设施（Public Key Infrastructure，PKI）基础之上，通过安全套接层认证（Security Socket Lay Authentication，SAP）协议来实现实体间双向身份认证的网格认证体系，提供在网格通信网环境中的安全认证，支持网格环境中主体之间的安全通信，防止主体假冒和数据泄密，为网格通信提供机密性、完整性保护，以及为网格用户提供单点登陆和权限委托的能力。

目前存在的许多网格安全体系，诸如 GSI，OGSI 等几乎都使用基于传统 PKI 的认证框架，并通过安全套接字层认证协议（Security Socket Layer Authentication Protocol，SAP）实现网格通信环境中实体的交互认证。但是，由于基于传统 PKI 的认证框架依赖与证书及认证权威（Certificate Authority，CA），从而导致认证过程中证书处理过于频繁，并且需要交换验证通信双方的证书链，因而造成了巨大的通信开销。然而网格通信环境是非常复杂的网络环境，网格通信环境中各种资源的分散性以及网格用户请求资源频繁性，实体的动态性、多变性以及不确定性决定了网格通信环境必须以一种十分高效的策略进行任意实体和所需资源之间的身份认证。基于 PKI 的认证体系导致效率过低，虚拟组织扩展性较差，系统性能无法提高，成为网格通信网性能的瓶颈。

随着密码理论的发展，基于身份的密码学理论逐步应用于网格系统中，成为网格安全认证方面新的解决方案。1982 年 Shamir 提出基于身份的密码学理论（Identity-Based Cryptography，IBC）[6]，2001 年 Boneh 和 Franklin 提出了第一个实用的基于身份的加密（Identity-Based Encryption，IBE）方案[15]，这种方案使用了从椭圆曲线上构造得来的双线性对。在基于身份的密码系统（IBC）中，公钥可以通过任何随机串得到，如用户身份、电子邮箱地址等，信任权威（Trusted Authority，TA）利用系统主密钥和公钥为用户产生相应的私钥；同时基于身份密码系统享有传统 PKI 的功能，又免除了证书的使用，避免了由于使用证书带来的种种弊端，从而出现了大量的基于身份的认证密钥协商方案。这些新出现的认证密钥协商方案以其轻量、高效的优良性质，极大地满足了网格环境下实体认证的需求。

尽管基于 IBC 的网格通信认证密钥协商方案简化了密钥的管理，而且相比较传统的 PKI 具有很好的性质，但是基于身份的认证密钥协商协议方案中存在一些公开的问题：

1. 密钥托管：在 PKI 系统中，如果用户的密钥是由用户自己生成的，可信中心就不能获得用户的私钥，因此就不会出现密钥托管问题。而在基于身份的公钥密码中，用户的私钥是由 KGC 利用它的系统范围的主密钥来生成的，因而 KGC

能够生成所以用户的私钥，因此密钥托管是基于身份密码系统固有的性质。如果KGC有任何的不诚实行为或者主密钥的泄漏，都将直接导致用户私钥的泄漏，系统毫无保密性可言，这样攻击者就可以轻易地获取用户的信息和伪造用户的签名，包括用户已发生的基于此私钥的秘密通信信息都将直接暴露，如何克服基于身份的公钥系统中的密钥托管问题是一个值得研究的方向。

2. 密钥撤销：在基于身份的密码系统中，用户的公钥是由用户的身份信息获得的，撤销用户的公钥相当于撤销用户的身份，而用户的身份信息大多是固定不容易改变的，因此密钥撤销是ID-PKC面临的另一个难题。在基于证书的PKI体制中，证书撤销列表CRL（Certificate Revocation List）是一种最简单、最常用的证书撤销方法，CRL实质上是由颁发证书的CA定期签发的一个签名的数据结构，包含被该CA撤销的证书列表。CRL使用证书序列号来标识每一个被撤销的证书。使用证书的系统就可以通过查询“最近发布”的CRL来查询证书的状态。然而这些基于证书的撤销方案涉及到CA、目录服务和证书用户三方的交互通信，通信代价比较大。而基于身份的公钥密码最大的优点就是简化证书管理、非交互式通信所带来的低通信成本和计算成本，因此，一种简单而有效的密钥撤销方法也是基于身份密码系统实用化过程中的重要组成部分。

1.5 论文的主要内容

本文主要针对网格通信认证密钥协商方案进行研究，结合网格通信环境的安全问题及特点，围绕如何提高网格通信实体间认证密钥协商的安全性和运行效率展开工作，主要研究内容包括：

第一部分，研究分析了网格通信认证密钥协商协议方案及其改进方案，包括基于身份的认证密钥协商和基于无证书的认证密钥协商方案。首先，研究了Smart和Chen等的协议，分析他们的协议存在密钥托管问题和其他安全问题；其次，研究了Al-Riyami和Paterson的无证书认证密钥协商方案（AP-CL-AKA协议）及Shiyijuan所提的改进的SL-CL-AKA协议方案，并对协议进行了安全属性和性能分析。

第二部分，通过深入分析现有的认证密钥协商协议方案的缺陷以及难以解决的弊端。分析了基于无证书密码系统（CL-PKC）的优点，针对分布式的网格通信环境，在SL-CL-AKA协议的基础上，给出了一个适用于网格通信环境的基于无证书的认证密钥协商方案，所给出的方案不仅克服了基于身份认证密钥协商协议的密钥托管问题，具有完善的前向安全属性、已知临时私钥信息安全性和其他期望的安全属性，能够抵御密钥复制攻击，同时所给出的方案的性能相比AP-CL-AKA协议更加高效。

1.6 本文的组织结构

本文各章内容安排如下：

第一章介绍了本课题研究的背景、目的和意义。叙述了网格的概念、体系结构、网格的特点、认证技术在网格安全中的重要意义和研究现状。最后概括本文的研究内容及结构安排。

第二章介绍与认证技术相关的一些数学知识，包括概率、数论、困难性假设以及哈希函数，同时给出了认证密钥协商协议的安全属性及安全性证明模型。

第三章介绍基于身份的公钥密码系统，研究 Smart 的密钥协商协议和 Chen 等的密钥协商协议，分析了以上认证密钥协商协议的安全属性，发现以上协议都不具备完善的前向安全性，且都存在密钥托管问题和密钥撤销问题。

第四章研究基于身份的认证密钥协商协议，首先分析了现有的认证密钥协商协议的密钥托管问题及其安全性缺陷，下来针对密钥托管问题，介绍了无证书公钥密码的定义和安全模型，包括无证书加密方案和无证书签名方案。

第五章在第四章的基础上，研究了无证书可认证密钥协商协议。首先分别介绍了 S.S.Al-Riyami 和 K.G.Paterson 提出的无证书可认证密钥协商协议（AP-CL-AKA 协议）、Shiyijuan 所提的更加有效的无证书认证密钥协商协议（SL-CL-AKA 协议），下来介绍了无证书可认证密钥协商协议的安全模型，然后给出了一个应用于网格通信的新的无证书认证密钥协商协议，并对所给协议的安全性和性能进行了分析比较。

第六章对全文进行总结，并指出需要进一步研究的内容。

第二章 预备知识

2.1 椭圆曲线

2.1.1 椭圆曲线概念

人们对椭圆曲线的研究已有100多年的历史，而椭圆曲线是Neal Koblitz和Victorian Miller于1985年提出来的。目前，椭圆曲线密码已经成为除RSA密码之外呼声最高的公钥密码之一。

椭圆曲线指的是有维尔斯特拉斯（Weierstrass）方程$y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6$所确定的平面曲线[8]。若$F$是一个域，$a_i \in F, i=1,2,...,6$。满足该等式的数偶$(x,y)$成为$F$域（可以是有理数域或有限域$GF(P_r)$）上的椭圆曲线$E$的点。

在实数系中，椭圆曲线可定义成所有满足方程式$E:y^2=x^3+ax+b$的点(x,y)所构成的集合。若方程式x^3+ax+b没有重复的因式或$4a^3+27b^2 \neq 0$，则$E:y^2=x^3+ax+b$能成为群（Group）。例如，椭圆曲线$E:y^2=x^3-7x+3$的图形如图2.1所示。若$4a^3+27b^2=0$，则此曲线将会形成退化（某些数的反元素（inverse）将不存在）。

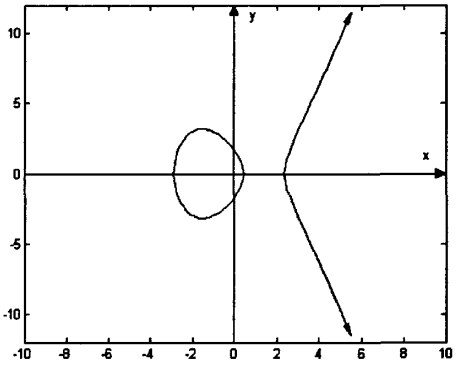

图2.1 椭圆曲线

椭圆曲线密码系统在模 p（或 F_p）下定义为椭圆曲线 $E: y^2 = x^3 + ax + b$，其中 $4a^3 + 27b^2 \neq 0$；模 F_{2^n} 下定义为椭圆曲线 $E: y^2 + xy = x^3 + ax + b$，其中 $b \neq 0$，此曲线称为 nonsuper-singular。椭圆曲线有一个特殊的点，记为 O，它并不在椭圆曲线 E 上，此点称为无限远的点（the point at infinity）。$E(K)$ 为在 K 之下椭圆曲线 E 上所有的点所构成的集合，点 $P=(x,y)$ 对 X 坐标轴反射的点为 $-P=(x,-y)$，而称 $-P$ 为点 P 的负点。若 $nP=O$ 且 n 为最小的正整数，则 n 为椭圆曲线 E 上点 P 的秩。除了无限远的点 O 之外，椭圆曲线 E 上任何可以生成所有点的点都可视为是 E 的生成数（generator），但并不是所有在 E 上的点都可视为生成数。

以下我们将介绍椭圆曲线运算在几何学上的定义。包括两个相异的点的相加与点的双倍运行，分部说明如下：

两个相异的点的相加：假设 P 和 Q 是椭圆曲线 $y^2 = x^3 - 3x + 3$ 上两个相异的点，而且 P 不等于 $-Q$。若 $P+Q=R$，则点 R 是经过 P、Q 两点的直线与椭圆曲线相交的唯一交点的负点。如图 2.2（a）所示。

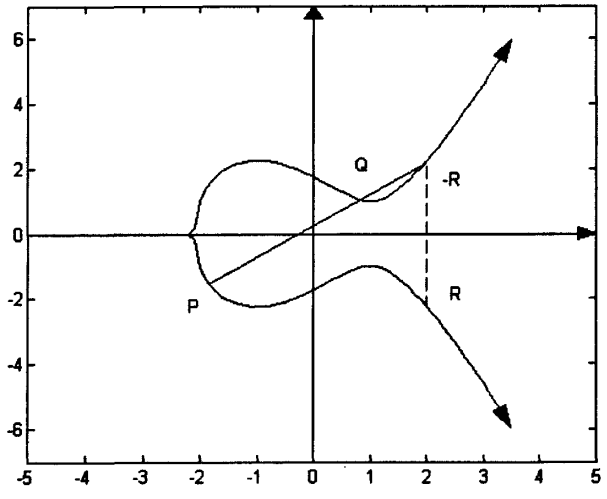

图 2.2（a）两个相异的点相加

P 的双倍的点：令 $P+P=2P$，则点 $2P$ 是经过 P 的切线与椭圆曲线相交的唯一交点的负点。如图 2.2（b）所示。

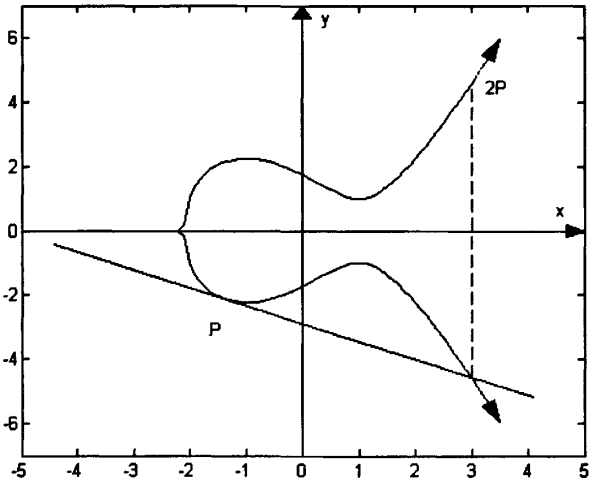

图 2.2（b）双倍的点

2.1.2 群的运算法则

自从近世纪代数学引入了群、环、域的概念，使得代数运算达到了高度的统一。数学家总结了普通加法的主要特征，提出了加群（也叫交换群、Abel 群）[18]。在加群的眼中，实数的加法和椭圆曲线上的加法没有什么区别。在椭圆曲线 E 上定义“+”运算，$(E,+)$ 就构成加群。

加法规则：

1. 无穷远点 O 是零元，对所有的点 $P\in E(F_p)$，有 $O+O=O$，$P+O=O+P=P$。

2. $P=(x_1,y_1)\in E(F_p)$ 的负元 $-P$ 是 $(x_1,-y_1)$，有 $P+(-P)=O$。

3. 令 $P=(x_1,y_1)\in E(F_p)$ 及 $Q=(x_2,y_2)\in E(F_p)$，且 $P\neq -Q$，则 $P+Q=(x_3,y_3)$，其中 $x_3=\lambda^2-x_1-x_2$，$y_3=\lambda(x_1-x_3)-y_1$。

$$\lambda=\begin{cases}\dfrac{y_2-y_1}{x_2-x_1} ifP\neq Q\\ \dfrac{3x_1^2+a}{2y_1} ifP=Q\end{cases}$$

4. 如果 $s,t\in F_p$，则对所有的点 $P\in E(F_p)$ 而言，$(s+t)P=sP+tP$。

乘法规则：

1. 如果$k \in F_p$，则对所有的点$P \in E(F_p)$而言，$kP = \overbrace{P + ... + P}^{k次}$。

2. 如果$s,t \in F_p$，则对所有的点$P \in E(F_p)$而言，$s(tP) = (st)P$。

2.2 双线性 CDH

2.2.1 双线性映射

设G_1、G_2是两个阶为q的群，其中q的一个大素数（例如 160 比特以上的素数）。G_1、G_2分别写作加法群和乘法群。G_1、G_2上的双线性映射$\hat{e}: G_1 \times G_1 \to G_2$是指满足如下性质的映射[18]：

双线性：$\hat{e}(aP,bQ) = \hat{e}(P,Q)^{ab}$，其中$P,Q \in G_1$，$a,b \in N$；

非退化性：如果$P,Q \in G_1$都不是G_1的单位元，则$\hat{e}(P,Q) \neq 1$；

可计算性：存在一个有效算法，对于任意的$P,Q \in G_1$，可有效地计算$\hat{e}(P,Q)$。

目前，能够用作加密系统的双线性映射，只找到了椭圆曲线和 Abel 代数簇上的 Weil 映射和 Tate 映射。使用这些映射时，椭圆曲线和代数簇的 MOV 次数要求相对较小，因此基于身份的密码系统的参数设置受到一定的制约，这一点与通常的椭圆曲线公钥密码的参数选择有很大的不同。

2.2.2 Bilinear Diffie-Hellman（BDH）问题[17]

设G_1，G_2为阶为素数q的两个群，$\hat{e}: G_1 \times G_1 \to G_2$为双线性映射，$P$为$G_1$的生成元则$<G_1,G_2,\hat{e}>$上的 BDH 如下：对任意的$a,b,c \in Z_q^*$，则$<P,aP,bP,cP>$计算$W = \hat{e}(P,P)^{abc} \in G^2$，称算法 A 求解$<G_1,G_2,\hat{e}>$上的 BDH 具有优势$\varepsilon$，如果：

$\Pr[A(P,aP,bP,cP) = \hat{e}(P,P)^{abc}] \geq \varepsilon$。

设随机算法IG称为 BDH 参数生成器，如果IG满足如下条件：

1. 输入为安全参数$k \geq 1$；

2. 运行时间为 k 的多项式；

3. IG 的输出阶数为 q，群 G_1，G_2 和双线性映射 $\hat{e}:G_1\times G_1\to G_2$，即 $<G_1,G_2,\hat{e}>\leftarrow IG(1^k)$。

2.2.3 Bilinear Diffie-Hellman(BDH)假设[23]

设 IG 为 BDH 参数生成器，称算法 A 求解以 IG 的输出作为参数的 BDH 具有优势 $\varepsilon(k)$，如果满足：对足够大的 k，

$$\begin{aligned} &Adv_{IG,A}(k)=\Pr[A(q,G_1,G_2,\hat{e},P,aP,bP,cP)|\\ &<q,G_1,G_2,P,\hat{e}>\leftarrow g(1^k),P\in G_1^*,a,b,c\leftarrow Z_q^*]\geq\varepsilon(k)\end{aligned} \tag{2-1}$$

若对任意随机多项式时间 $poly(k)$ 算法 A 和任意多项式函数 $f\in Z[x]$，$Adv_{IG,A}<\frac{1}{f(k)}$ 对足够大的 k 成立，则称 IG 为满足 BDH 假设。若 IG 满足 BDH 假设，则认为在由 IG 生成的群中 BDH 是困难的。

2.2.4 可计算 Diffie-Hellman（CDH）假设[23]

设 IG 为 BDH 参数生成器，设 G_1 为阶为素数 q 的群，P 为 G_1 的生成元，对任意的 $a,b,c\in Z_q^*$，则给定 $<G_1,P,aP,bP>$ 计算 abP，称算法 A 具有优势 $\varepsilon(k)$，如果：对任意的 k，

$$\begin{aligned} &Adv_{IG,A}(k)=\Pr[A(q,G_1,P,aP,bP)|\\ &<q,G_1,P>\leftarrow IG(1^k),P\in G_1^*,a,b\leftarrow Z_q^*]\geq\varepsilon(k)\end{aligned} \tag{2-2}$$

2.3 对（Paring）

椭圆曲线对是 2000 年被发现适合于设计基于身份的公钥密码系统的数学工具。这个概念的一般化是双线性映射，但目前找到的具有密码意义的双线性映射只有椭圆曲线和 Abel 代数簇上的 Weil 对和 Tate 对[19]。因此基于身份的密码系统的有效实现与椭圆曲线密码密切相关。比较 Weil 对和 Tate 对，由于 Weil 对的计算更费时间一些，因此通常选择 Tate 对作为实现身份加密系统的双线性映射。

2.3.1 Tate 对

定义 1 一个除子是曲线 E 上若干点的一种形式和

$$D=\sum_{P\in E} m_P P, m\in Z$$

这里只有有限个整数 m_p 非零，整数 $\sum_{P\in E} m_p$ 称为 D 的度（记为 $\deg D$），D 在 P 点的阶是 m_p，表示为 $ord_P(D)=m_p$。

设 $Div_0=\{D\in Div \mid \deg D=0\}$，那么 Div_0 是 Div 上的一个子群。

定义 2 设曲线 E 定义于域 K 上，$\bar{K}(E)$ 是 E 的函数域，一个除子 $D\in Div_0$ 称为一个主除子，如果存在某个非零有理数 $f\in \bar{K}(E)$ 使得 $D=div(f)$。

所有主除子的集合记为 $\Pr in$，它是 Div_0 的一个子群。

定义 3 设 Γ 表示商域 $Div_0/\Pr in$，P 称为曲线 E 的 jacobian 群。

对于与 K 相关的素数 m>0，令，

$$\Gamma[m]=\{[D]\in\Gamma : mD \text{是规则}\}$$

对于一个有理数 f 和一个除子 $E=\sum n_P P$ 并且 $(f)\cap E=\Phi$，令

$$f(E)=\Pi f(P)^{nP}\in K^* \tag{2-3}$$

Tate 对的定义：

$$\{-,-\}_m : \begin{matrix}\Gamma[m]\times\Gamma/m\Gamma\to K^*/K^{*m},\\ \{[D],[E]\}_m=f_D(E).\end{matrix} \tag{2-4}$$

对于椭圆曲线 E/k 我们有如下定义：

$E(k)\cong\Gamma$，$P\mapsto[P-O]$，对于一个椭圆曲线 E/k，和 $D=[P-O]$。有效计算 Tate 对的 $f_D(Q)$ 的算法是 Miller 算法。

2.3.2 Weil 对

给定函数 f 和除子 $A=\sum_P a_P(P)$，定义 $f(A)=\prod_P f(P)^{a_P}$，定义两点 $P,Q\in E[n]$ 的 Weil 对，其中 $E[n]=\{\bar{P}\in E \mid n\bar{P}=O\}$。令 A_P 是等于 $(P)-(O)$ 的除子，

我们知道nA_P是个主除子，等于$n(P)-n(O)$，因此存在函数f_P，$(f_P)=nA_P$，则Weil对定义[44]如下：

$$\hat{e}(P,Q)=\frac{f_P(A_Q)}{f_Q(A_P)} \tag{2-5}$$

Weil对具有以下性质：

1. 对所有的$P\in E[n]$，有$\hat{e}(P,P)=1$；

2. 双线性：$$\begin{aligned}&\hat{e}(P_1+P_2,Q)=\hat{e}(P_1,Q)\bullet\hat{e}(P_2,Q),\\&\hat{e}(P,Q_1+Q_2)=\hat{e}(P,Q_1)\bullet\hat{e}(P,Q_2)\end{aligned} \tag{2-6}$$

3. 当$P,Q\in E[n]$线性相关时，$\hat{e}(P,Q)=1$；即$\hat{e}(P,Q)=\hat{e}(Q,P)^{-1}$；

4. 对所有的$P,Q\in E[n]$，有$\hat{e}(P,Q)^n=1$，例如$\hat{e}(P,Q)=G_2$；

5. 非降阶：如果$P\in E[n]$，对于所有的$Q\in E[n]$，如果满足$\hat{e}(P,Q)=1$，则$P=Q$。

2.4 网格通信认证密钥协商协议的安全性

2.4.1 安全特性

一个安全的认证密钥协商协议应该具有以下安全特性[14]：

1. 隐式密钥认证（Implicit Authentication）：如果协议中的每一方（如UP）都确信共享密钥只有协议的参与者（如RP）才知道，攻击者无法获知，则称此协议提供了隐式密钥认证。隐式密钥认证并不需要协议的参与者确实计算出会话密钥。

2. 已知会话密钥的安全性（Known Session Key Security）：协议参与者（如UP或者RP）每执行一次密钥协商协议，会生成一个唯一的会话密钥，如果会话密钥的泄露不会导致被动攻击者得到其他的会话密钥或主动攻击者假冒协议参与者（如UP或RP），则称协议具有会话密钥的安全性。

3. 前向保密性（Forward Secrecy）：如果一个或多个协议参与者（如UP或者RP）的长期私钥（long-term private key）泄漏了，不会导致旧的会话密钥泄漏，则称该协议提供了部分前向保密（partial forward secrecy）；如果所以协议参与者（如UP和RP）的长期私钥泄漏，不会导致就的会话密钥泄漏，则称该协议有完善的前向保密性（perfect forward secrecy）。

4. KGC 前向安全性（KGC forward secrecy）：在基于身份的密码系统中，如果 KGC 的主密钥被攻破或被获取（所有用户的长期私钥都被泄露），所有用户之前产生的会话密钥都不会泄露，则称协议具有 KGC 前向安全性。在基于身份的密码系统中，KGC 前向安全性隐含了完善前向保密性。

5. 密钥泄漏的安全性（No Key-compromise Impersonation）：假设协议参与者 A（如 UP）的长期私钥（long-term private key）泄漏了，攻击者知道这个长期私钥只具有冒充 A 的能力，而不能在 A 面前伪造其他协议参与者（如 RP）与 A 生成会话密钥，称协议提供了密钥泄漏的安全性。

6. 密钥分享的知晓性（Unknown Key-share）：在协议参与者（如 UP）A 不知晓的情况下，A 是不能被强迫与协议参与者（如 RP）B 分享一个会话密钥，具有该性质的协议称协议提供了密钥分享的知晓性。

7. 密钥控制安全性（No Key Control）：协商会话密钥是任何一方（UP 或 RP）都不能控制会话密钥的结果，例如，将会话密钥限定在某个已知的小集合内。参与协议的任何一个用户都不能预先选定一个固定的值作为会话密钥。协商协议产生的会话密钥必须是根据协议的全部参与者的贡献值协商确定的。

8. 已知临时私钥的信息安全性（Known session-specific temporary information security）：在生成会话密钥时，一些随机的临时私钥信息作为会话密钥生成函数的输入。如果这些临时私钥信息的泄露不会导致会话密钥的泄露，则称协议具有已知临时私钥的信息安全性。

9. 消息独立性（Message independence）：协议产生的密码流是不相关的。

10. 密钥复制攻击（Key replicating attack）：

密钥复制攻击是中间人攻击的一种，通过在协议参与者之间截获、修改密钥协商时的交换信息，假扮协议参与者，控制会话密钥的生成，甚至获取协议参与者协商的会话密钥。如果允许攻击者向已经接受持有会话密钥的协议参与者发送 Reveal 询问，即攻击者要求 oracle 给出他接受持有的会话密钥，则将会影响认证密钥协商协议的密钥控制安全属性。

CL-AK 协议还应该具有比较高的带宽效率，即只需要在 UP 和 RP 之间传输很小量的数据就能够进行双向实体认证，并获得会话密钥。低计算开销和具有预计算能力是 CL-AK 协议的两个重要计算属性。

2.4.2 安全性证明模型

在密钥协商协议的研究以及其他密码学协议的相关研究中，经常会利用随机预言模型对协议的安全性进行证明。本文中利用 Bellare-Rogaway 模型对安全性进行证明。为了便于读者理解本文后面章节中介绍的协议的安全性的证明过程，在

此简要介绍一下 Bellare-Rogaway 模型[13]。

Bellare-Rogaway 模型（我们称为 BR 模型）是一种非标准化的计算模型，是一个随机预言模型。在该模型中，任何具体的对象被当作随机对象。人们可以规约参数到相应的计算，每一次新的查询，都能够得到一个随机的应答。假设存在一个敌手，让该敌手模拟协议的执行，得到一个结果。但该结果却与数学假设相矛盾，借助一些概率的概念，以此证明协议的安全性。尽管随机预言模型的有效性是存在争议的，但是该模型能够在一定程度上保证一个方案是没有缺陷的。因此该模型对于认证密钥协商协议还是很有用的。

BR 模型包含了一组参与者 I，其中每个参与者抽象为一个预言机 oracle。攻击者可以通过发送询问访问这些语言机。预言机 $\prod_{i,j}^{n}$ 表示参与者 i 在他的 s 实例中与参与者 j 进行会话，模拟了用户 i 和用户 j 之间第 n 次通信，他保存了所有他们发送和接收的信息，以及相应的请求。

模型中还包括一个攻击者 E，该攻击者能够控制整个通信网络，控制者所有的协议参与者。他可以读取、插入、修改、延时、监听和删除协议中任何用户发送的消息，甚至能够发起和接收会话。一个攻击者 E 可以对 oracle 进行以下三种操作：

1. Send：攻击者可以向 oracle 发送任何消息。
2. Reveal：攻击者可以要求 oracle 给出他持有的会话密钥。
3. Corrupt：攻击者可以要求 oracle 给出其中用户的长期私钥。

可见，在该模型中很好的模拟了攻击者的各种行为。

一个 oracle 存在以下五种状态：

1. Accepted：接受持有一个会话密钥。
2. Rejected：拒绝持有一个会话密钥。
3. *：还没有决定好是接受还是拒绝。
4. Opened：给出了他持有的会话密钥。
5. Corrupted：给出了用户的长期私钥。

为了更好地介绍安全的认证密钥协商协议的定义，还需要介绍一下良性攻击者和匹配的会话的概念。如果一个攻击者在传递两个预言机之间的消息时，不进行任何篡改或故意遗漏消息，则把这个攻击者称为良性攻击者。如果两个 oracle 中的任何一方都只接收来自另一方的消息，则称这两个 oracle 之间有匹配的会话。用 $advantage^{E}(k)$ 表示攻击者 E 能够获得本次产生的会话密钥的优势。（其中 k 为安全参数）

安全的认证密钥协商协议的定义[31]：

1. 在只有良性攻击者存在的情况下，oracle $\prod_{i,j}^{s}$ 和 $\prod_{j,i}^{t}$ 在协议结束时都处于 accepted 状态，且持有相同的会话密钥。

2. 对于任意的攻击者，两个不处于 corrupted 状态的 oracle 之间有匹配的会话，那么他们都处于 accepted 状态，且持有相同的会话密钥。

3. 攻击者 E 能够获得本次产生的会话密钥的优势 $advantage^{E}(k)$ 是可忽略的。

如果 $\forall c>0,\exists k_c>0$ 满足 $\forall k>k_c,\exists \varepsilon(k)<k^{-c}$，则称函数 $\varepsilon(k)$ 是不可忽略的。如果一个函数不是不可忽略的，就称它是可忽略的。

2.5 网格通信认证密钥协商协议的性能分析

认证密钥协商协议的性能一般从以下几个方面进行考虑：

1. 最少的交互回合数，即协议中的信息交换次数应该尽可能少。

2. 最少的占用带宽，即协议中需要传输的信息的总位数（通信量）应该尽可能小。

3. 降低计算消耗时间，协议中需要执行的运算应该尽可能简单快速，即所需计算时间应该尽可能短，所需计算能力较低。

4. 其他性质：拥有预计算能力等。

2.6 小结

本章主要介绍了椭圆曲线密码体制、双线性映射，以及椭圆曲线对的基本知识和概念，这些都是论文所做的研究工作的理论基础。简述了认证密钥协商协议应该具有的安全特性，以及证明协议安全性所需要的 Bellare-Rogaway 模型，还有认证密钥协商协议的性能的分析方法。本文中后面部分叙述的认证密钥协商协议都是在椭圆曲线密码体制基础上的，并且文中后面章节中相关协议的安全性分析和证明，以及协议的性能分析都需要本章所讲述的内容。因此，介绍这些内容能够帮助读者更好的理解文中给出的协议和相关的安全性分析和证明过程，已及协议的性能分析。

第三章 基于身份的认证密钥协商协议

3.1 基于身份的密码系统

1984 年 Shamir 在[28]中提出了基于身份的密码系统（IBC）的概念。该系统是基于公钥体制的，在该系统中，有一对非对称的密钥即公钥和私钥。其中，公钥是一个表示身份的字符串。私钥则需要一个权威机构 TA 通过对公钥进行某些运算得到。这个权威机构 TA 掌握着系统中所有的加解密、签名以及认证所需的信息。该系统中的任意一对用户可以安全的通信。在该系统中的用户，不用交换公钥和私钥，不需要保留密钥目录，也无需通过第三方的帮助，就能够认证另一方的身份。跨国公司的各个管理者之间或是一家大银行的分行之间就可以应用该系统，此时这些机构的中央部门就可以作为权威机构 TA 为下属各个分支服务。此外，该系统也可以适应几百上千个 TA 和上亿个用户的需要。

Shamir 的 IBC 理论为认证密钥协商协议，乃至整个密码学都提供了一个新的思路，开辟了一个新的研究方向，在很多文献中，都运用了该密码理论。

3.2 Smart 的协议和 Chen 等的协议

3.2.1 Smart 的协议

Smart 提出了一个运用双线性对的基于身份的认证密钥协商协议[21]。该协议的执行过程如下：

用户 A 随机选择一个一次性密钥 $a \in Z_q^*$，计算 $T_A = aP$ 并发送给用户 B；

用户 B 随机选择一个一次性密钥 $b \in Z_q^*$，计算 $T_B = bP$ 并发送给用户 A；

用户 A 在接收到 W_B 之后，计算 $K_{AB} = \hat{e}(S_A, T_B)\hat{e}(aQ_B, P_s)$；

用户 B 在接收到 W_A 之后，计算 $K_{BA} = \hat{e}(S_B, T_A)\hat{e}(aQ_A, P_s)$；

这样，两个协议参与者 A 和 B 就共享了会话密钥 $K = \hat{e}(bQ_A + aQ_B, P_s)$。

Smart 的协议提供了隐性密钥认证，已知密钥安全性、部分前向安全性、密钥泄露安全性、未知密钥共享安全性和不完全的密钥控制安全性。计算量方面，

参与协议的每个用户需要执行两次椭圆曲线的点乘运行和两次双向性对运算。

3.2.2 Chen 等的协议

Chen 等对 Smart 的协议进行了部分修改，针对 Smart 协议的效率进行了一些改进，提出了一个新的运用了双线性对的基于身份的认证密钥协商协议[19]。该协议的执行过程如下：

用户 A 随机选择一个一次性密钥 $a \in Z_q^*$，计算 $W_A = aQ_A$ 并发送给用户 B；

用户 B 随机选择一个一次性密钥 $b \in Z_q^*$，计算 $W_B = bQ_B$ 并发送给用户 A；

用户 A 在接收到 W_B 之后，计算 $K_{AB} = \hat{e}(S_A, W_B + aQ_B)$；

用户 B 在接收到 W_A 之后，计算 $K_{BA} = \hat{e}(W_A + bQ_A, S_B)$；

协议执行完成之后，协议的两个参与者 A 和 B 就协商出了相同的会话密钥 $K = \hat{e}(Q_A, Q_B)^{s(a+b)}$。

Chen 等的协议与 Smart 的协议提供了相同的安全性，而协议的计算代价减少了，协议的效率提高了。在 Chen 等的协议中，参与者只需要执行两次椭圆曲线点乘运算，一次椭圆曲线点加法和一次双向性对运算。

Chen 等提出的协议和 Smart 提出的协议都只具备部分前向安全性，而不具备完善前向安全性。

3.3 基于身份密钥协商协议中的密钥托管问题

密钥托管问题：在 PKI 系统中，如果用户的密钥是由用户自己生成的，可信中心就不能获得用户的私钥，因此就不会出现密钥托管问题。而在基于身份的公钥密码中，用户的私钥是由 KGC 利用它的系统范围的主密钥来生成的，因而 KGC 能够生成所以用户的私钥，因此密钥托管是基于身份密码系统固有的性质。如果 KGC 有任何的不诚实行为或者主密钥的泄漏，都将直接导致用户私钥的泄漏，系统毫无保密性可言，这样攻击者就可以轻易地获取用户的信息和伪造用户的签名，包括用户已发生的基于此私钥的秘密通信信息都将直接暴露。

在分析认证密钥协商协议的安全性时，必须考虑密钥托管问题，以上协议都没有提供 KGC 私钥的前向保密性，即 KGC 能够托管认证密钥协商协议所产生的会话密钥。具体地说就是，一旦攻击者得到了 KGC 的私钥 s，他就可以通过计算恢复得到认证密钥协商协议所产生的会话密钥。可见，基于身份的认证密钥协商

协议存在密钥托管问题，如何克服基于身份的认证密钥协商协议中的密钥托管问题是一个值得研究的方向。

3.4 小结

在本章中，首先介绍了基于身份的密码系统以及 Smart 和 Chen 等提出的基于身份的认证密钥协商协议。分别描述了协议的具体执行过程，并对协议的计算代价及安全属性进行了分析。通过分析得到 Smart 和 Chen 等所提的认证密钥协商协议只具备部分前向安全性，不具备完善的前向安全性。同时，基于身份的密码系统与生俱来就存在密钥托管问题和密钥撤销问题，在下面的章节，我们分别给出基于身份的认证密钥协商协议所存在问题的解决方案。

第四章 无证书公钥密码

在2003年，A1-Riyami和Paterson[10]介绍了无证书的公钥密码系统(CL-PKC)，比较与传统公钥密码系统，无证书的公钥密码系统不要求认证中心的认证来保证公钥与持有人的合法关系，但是必须存在一个持有主要密钥的可信第三方为用户提供部分私钥。从这方面来看，无证书的公钥密码和基于身份的密码系统是很相似的。从另外一个角度来考虑，无证书的公钥密码系统又解决了基于身份的密码系统遗留的密钥托管问题。这样无证书的密码系统又可以看作是公钥密码的应用模型，所以无证书的公钥密码是一种介于传统公钥密码和基于身份的密码系统之间的一种系统。在无证书的公钥密码系统中，我们往往给某个用户 A 匹配一定的身份信息 IDA，此身份信息可以是任意的字符串。密钥生成中心(KGC)无法得到用户 A 的私钥，他只是知道根据用户的身份信息 IDA 和自己的主要密钥计算得到的用户的部分私钥，提供部分私钥给用户的过程应该是一个秘密的、认证的过程：私钥生成中心必须确保部分私钥被安全的送到合法的用户手里，在本章的第一节给出了目前的定义方法的详细介绍。

因为在无证书密码系统中，用户的公钥没有得到认证，所以在攻击模型中，必须允许攻击者有权利用他自己选择的不合法的公钥代替用户的公钥，而且，由于私钥生成中心知道主要密钥，从而能够计算所有用户的部分私钥，所以在许多攻击模型中，往往会考虑恶意的私钥生成中心。

4.1 无证书密码算法的定义方法

首先我们来了解一下最初的无证书的密码算法的定义，是由 Al-Riyami 和 Paterson 在[18]中第一次提出的。

定义 1 一个无证书的加密算法主要包括七个概率多项式时间的算法：

1. 系统建立：在算法中输入一个安全参数1^k，那么将返回一个主要密钥msk和系统公共参数mpk，这个算法是由私钥生成中心运行的，这样就建立了一个无证书的密码系统。

2. 部分私钥提取：在算法中输入系统公共参数mpk、主要私钥msk和用户的身份信息$ID \in \{0,1\}^*$，输出用户的部分私钥d_{ID}，同样此算法也是由私钥生成中心

运行的，针对每个用户运行一次，并把相应的部分私钥用一个比较安全的方式发送给用户。

3. 秘密值设定：在算法中输入系统公共参数 mpk 和用户的身份信息 ID，输出一个用户秘密值 $x_{ID} \in \varphi$。用户运行该算法一次，秘密值空间 φ 是由系统公共参数 mpk 和用户的身份 ID 决定的。

4. 私钥设定：在算法中输入系统公共参数 mpk，用户的部分私钥 d_{ID} 和用户的秘密值 x_{ID}，输出用户私钥 sk_{ID}。用户运行该算法一次。

5. 公钥设定：在算法中输入系统公共参数 mpk 和用户的秘密值 x_{ID}，输出用户公钥 $pk_{ID} \in PK$。用户运行完算法得到公钥后，把公钥发布，用户公钥空间 PK 是由系统公共参数 mpk 和用户的身份 ID 共同定义的。

6. 加密：在算法中输入系统公共参数 mpk，用户的身份 ID，用户公钥 $pk_{ID} \in PK$ 和要加密的信息 $m \in M$。输出密文 $C \in C$ 或者错误符合 $\perp$。其中信息空间 M 和密文空间 C 是由系统公共参数 mpk，用户公钥 pk_{ID} 和用户身份信息 ID 共同决定的。

7. 解密：在算法中输入系统公共参数 mpk，用户私钥 sk_{ID} 和密文 $C \in C$，返回信息 $m \in M$ 或者错误符合 $\perp$。

4.2 AP 的无证书公钥加密方案（AP-CL-PKE）

假设 k 是系统安全参数，IG 是一个 BDH 参数生成器。

AP-CL-PKE 方案由以下几个算法构成：

1. 系统建立算法 S：

输入参数 k，运行 IG 生成 BDH 参数 $<G_1, G_T, e>$，其中 G_1 是阶为 q 的加法循环群，G_T 是阶为 q 的乘法循环群，$\hat{e}: G_1 \times G_1 \to G_T$ 是 G_1、G_T 上的双线性映射。

随机选择一个 G_1 的生成元 $P \in G_1$。

随机选择一个数 $s \in Z_q^*$ 作为主密钥，并计算 $P_0 = sP$。

选择哈希函数：$H_1:\{0,1\}^* \to G_1^*$，$H_2:G_T \to \{0,1\}^n$。其中 n 是明文的位数。

系统参数 $params =< G_1, G_T, e, n, P, P_0, H_1, H_2 >$，主密钥是 $s \in Z_q^*$，消息空间是 $M = \{0,1\}^n$，密文空间是 $C = G_1 \times \{0,1\}^n$。

2. 部分私钥抽取算法 $PPKE$：

这个算法输入用户身份 $ID_A \in \{0,1\}^*$，$params$ 以及主密钥 s，返回用户的部分私钥 D_A。

计算 $Q_A = H_1(ID_A) \in G_1^*$；

输出用户 A 的部分私钥 $D_A = sQ_A \in G_1^*$。

用户 A 可以通过计算 $e(D_A, P) = e(Q_A, P_0)$ 来验证部分私钥抽取算法是否成功。

3. 密值设置算法 SSV：

该算法输入系统参数 $params$ 和用户身份 $ID_A \in \{0,1\}^*$，选择一个随机数 $x_A \in Z_q^*$ 作为用户 A 的密值并返回该密值。

4. 私钥生成算法 $SPRI$：

该算法输入给定的系统参数 $params$，用户 A 的部分私钥 D_A 以及秘密值 $x_A \in Z_q^*$，计算私钥 $S_A = x_A D_A = x_A s Q_A \in G_1^*$。

5. 公钥生成算法 $SPUB$：

该算法输入系统参数 $params$ 以及用户 A 的秘密值 $x_A \in Z_q^*$，返回用户的公钥 $P_A =< X_A, Y_A >$，其中 $X_A = x_A P$，$Y_A = x_A P_0 = x_A sP$。

6. 加密算法 E：

该算法输入系统参数 $params$，用户身份 $ID_A \in \{0,1\}^*$，公钥 $P_A =< X_A, Y_A >$ 以及消息 $m \in M$。

检查 X_A，Y_A 是否在群 G_1^* 中，同时验证等式 $e(X_A, P_0) = e(Y_A, P)$ 是否成立（检查公钥的有效性）。如果 X_A，Y_A 不在群 G_1^* 中或上面的等式不成立，则输出 $\perp$，停止加密算法；

计算 $Q_A = H_1(ID_A) \in G_1^*$；

选择随机数 $r \in Z_q^*$；

计算并输出密文：$C = < rP, m \oplus H_2(e(Q_A, Y_A)^r) >$。

7. 解密算法 D：

该算法输入系统参数 $params$，用户身份 $ID_A \in \{0,1\}^*$ 及其私钥 S_A 和密文 $C = < U, V >$，返回明文消息：

$m = V \oplus H_2(e(S_A, U))$

加密方案的一致性证明为：

$$\begin{aligned} V \oplus H_2(e(S_A, U)) &= V \oplus H_2(e(x_A s Q_A, rP)) \\ &= V \oplus H_2(e(Q_A, x_A sP)^r) \\ &= V \oplus H_2(e(Q_A, Y_A)^r) \\ &= m \end{aligned} \tag{4-1}$$

4.3 AP 的无证书公钥签名方案（AP-CL-PKS）

通常，AP-CL-PKS 方案由以下七个算法构成：

1. 系统建立算法 S：

输入参数 k，运行 IG 生成 BDH 参数 $< G_1, G_T, e >$，其中 G_1 是阶为 q 的加法循环群，G_T 是阶为 q 的乘法循环群，$\hat{e}: G_1 \times G_1 \to G_T$ 是 G_1、G_T 上的双线性映射。

随机选择一个 G_1 的生成元 $P \in G_1$。

随机选择一个数 $s \in Z_q^*$ 作为主密钥，并计算 $P_0 = sP$。

选择哈希函数：$H: \{0,1\}^* \times G_T \to Z_q^*$。

系统参数 $params = < G_1, G_T, e, n, P, P_0, H >$，主密钥是 $s \in Z_q^*$，签名空间是 $S = G_1 \times Z_q^*$。

2. 部分私钥抽取算法 $PPKE$：

这个算法输入用户身份 $ID_A \in \{0,1\}^*$，$params$ 以及主密钥 s，返回用户的部分

私钥D_A。

计算$Q_A = H_1(ID_A) \in G_1^*$；

输出用户 A 的部分私钥$D_A = sQ_A \in G_1^*$。

用户 A 可以通过计算$e(D_A, P) = e(Q_A, P_0)$来验证部分私钥抽取算法是否成功。

3. 密值设置算法*SSV*：

该算法输入系统参数 *params* 和用户身份$ID_A \in \{0,1\}^*$，选择一个随机数$x_A \in Z_q^*$作为用户 A 的密值并返回该密值。

4. 私钥生成算法*SPRI*：

该算法输入给定的系统参数 *params*，用户 A 的部分私钥D_A以及秘密值$x_A \in Z_q^*$，计算私钥$S_A = x_A D_A = x_A sQ_A \in G_1^*$。

5. 公钥生成算法*SPUB*：

该算法输入系统参数 *params* 以及用户 A 的秘密值$x_A \in Z_q^*$，返回用户的公钥$P_A = < X_A, Y_A >$，其中$X_A = x_A P$，$Y_A = x_A P_0 = x_A sP$。

6. 签名S：

该算法输入系统参数 *params*，用户私钥S_A和待签名的消息$m \in M$，执行如下：

选择随机数$a \in Z_q^*$；

计算$r = e(aP, P) \in G_T$；

计算$v = H(M, r) \in Z_q^*$；

计算$U = vS_A + aP \in G_1$；

输出签名$< U, v >$。

7. 验证V：

该算法输入系统参数 *params*，用户ID_A的消息$m \in M$的签名$< U, v >$和公钥$P_A = < X_A, Y_A >$，验证算法执行如下：

检查等式$e(X_A,P_0)=e(Y_A,P)$是否成立，用来检查用户 A 公钥的合法性。如果不成立，则输出⊥，终止签名验证算法；

计算$r'=e(U,P)\cdot e(Q_A,-Y_A)^v$；

验证等式$v=H(M,r')$是否成立，如果成立，则输出有效，否则输出无效。

加密方案的一致性证明为：

$$\begin{aligned} r' &= e(U,P)\cdot e(Q_A,-Y_A)^v \\ &= e(vS_A+aP,P)\cdot e(Q_A,-Y_A)^v \\ &= e(vS_A,P)\cdot e(aP,P)\cdot e(vQ_A,Y_A)^{-1} \\ &= e(aP,P)\cdot e(vx_AsQ_A,P)\cdot e(vQ_A,x_AsP)^{-1} \\ &= e(aP,P)\cdot e(Q_A,P)^{vx_As}\cdot e(Q_A,P)^{-vx_As} \\ &= e(aP,P) \\ &= r \end{aligned} \tag{4-2}$$

Al-Riyami 和 Paterson 提出的无证书公钥密码 CL-PKC 仍然需要一个可信中心 KGC，和 ID-PKC 中的 KGC 相比，CL-PKC 中的 KGC 不知道用户的私钥。实际上，KGC 只能生成用户的部分私钥，用户把利用自己的秘密和 KGC 生成的部分私钥结合起来生成完整的私钥，因此 KGC 并不知道用户完整的私钥，因而 KGC 不能解密用户的密文和伪造用户的签名。Al-Riyami 和 Paterson 给出的具体的无证书加密方案（AP-CL-PKE）基于 BF-IBE 方案，KGC 生成用户 ID 的部分私钥$D_{ID}=sQ_{ID}$，其中$Q_{ID}=H_1(ID)$，用户自己选择一个随机数$x_{ID}\in Z_q^*$作为自己的另一部分私钥，用户 ID 完整的私钥$S_{ID}=x_{ID}D_{ID}$，其公钥为$P_{ID}=<X_{ID},Y_{ID}>$，其中$X_{ID}=x_{ID}P$，$Y_{ID}=x_{ID}P_{pub}$。加密一个消息 M，先通过检验方程$\hat{e}(X_{ID},P_{pub})=\hat{e}(Y_{ID},P)$是否满足来检查公钥的有效性。然后随机选择一个数$r\in Z_q^*$并计算密文$C=(rP,M\oplus H_2(\hat{e}(Q_{ID},Y_{ID})^r))$。为了解密密文$C=(U,V)$，用户 ID 计算$M=V\oplus H_2(\hat{e}(S_{ID},U))$。可以看出 CL-PKC 像 ID-PKC 一样不需要利用证书来实现

认证，同时又可以克服 ID-PKC 中的密钥托管问题。该方案提供了对用户最终密钥的保护，即使是不诚实的 KGC 也不能够获得用户完整的私钥，因此不能冒充用户解密密文或者伪造签名。而且在 KGC 和用户之间也不需要安全的密钥分发信道，部分密钥的丢失不会影响系统地机密性。CL-PKC 是介于 PKI 和 ID-PKC 之间的公钥密码体制，继承了二者的优点。

但是 AP-CL-PKE 方案存在两个不足，一是公钥占用两个群点带宽，二是公钥的验证需要额外的两个对计算。Shiyijuan 受到 SK-IBE 的密钥构造方法的启示，在充分利用 Weil 对性质的基础上提出了 SL-CL-PKE 加密方案，该方案继承了 SK-IBE 的优点，同时，它采用一种并行的工作模式，因而在效率上也有所提高，下一节介绍该 SL-CL-PKE 方案。

4.4 Shiyijuan 的无证书加密方案（SL-CL-PKE）

SL-CL-PKE 方案[23]充分利用了 Weil 对的属性，方案由以下几个算法构成：

1. 系统建立算法 S：该算法输入 1^k（k 是系统安全参数），执行以下步骤：

(1). 生成一个 Weil 对 $e: E[q]\times E[q]\to G_T$，其中 $E[q]=G_1\oplus G_2$，φ 是 G_2 到 G_1 的映射。随机选择一个 G_2 上的生成元 $P_2\in G_2^*$ 并设置 $P_1=\varphi(P_2)$。

(2). 随机选择一个数 $s\in Z_q^*$ 作为主密钥，并且计算 $P_{pub}=sP_1$。

(3). 计算 $g=\hat{e}(P_1,P_2)$。

(4). 选择哈希函数：$H_1:\{0,1\}^*\to Z_q^*$，$H_2:G_T\to\{0,1\}^n$。

消息空间是 $M=\{0,1\}^n$，密文空间是 $C=E[q]\times\{0,1\}^n$。KGC 发布公开系统参数 $params=<q,G_1,G_2,G_T,e,n,P_1,P_2,g,P_{pub},H_1,H_2>$，主密钥是 s。

2. 部分私钥提取算法 $PPKE$：这个算法输入用户身份 $ID\in\{0,1\}^*$，$params$ 以及主密钥 s，返回用户的私钥 $D_{ID}=\dfrac{1}{H_1(ID)+s}P_2\in G_2^*$。

3. 密值设定算法 SSV：该算法输入系统参数 $params$ 和用户身份 $ID\in\{0,1\}^*$，选择一个随机数 $x_{ID}\in Z_q^*$ 作为用户的密值并返回该密值。

4. 私钥生成算法 $SPRI$：该算法输入给定系统参数 $params$，用户 ID 的部分私钥 D_{ID} 以及密值 x_{ID}，返回用户私钥为 $S_{ID}=(D_{ID},x_{ID})$。

公钥生成算法 $SPUB$：该算法输入系统参数 $params$ 以及用户 ID 的密值 x_{ID}，返回用户的公钥 $P_{ID}=x_{ID}P_2$。

5. 加密算法 E：该算法输入系统参数 $params$，用户 ID 以及消息 $m\in M$，执行如下步骤：

(1). 检查公钥 P_{ID} 是否在群 G_2^* 中，如果不在，则停止。这用来检查用户公钥的有效性。

(2). 计算 $Q_{ID}=H_1(ID)P_1+P_{pub}\in G_1^*$。

(3). 选择随机数 $r_1,r_2\in Z_q^*$ 并计算密文 $c=<r_1Q_{ID}+r_2P_{ID},m\oplus H_2(g^{(r_1+r_2)})>$。

解密算法 D：该算法输入系统参数 $params$，用户 ID 及其私钥 $S_{ID}=(D_{ID},x_{ID})$ 和密文 $c=<U,V>$，返回明文消息：

$$m=V\oplus H_2(\hat{e}(U,D_{ID}-\frac{1}{x_{ID}}P_1))$$

根据 Weil 对的单位元性和交换性，即 $\hat{e}(P_1,P_1)=1$，$\hat{e}(P_2,P_2)=1$，$\hat{e}(P_2,-P_1)=\hat{e}(P_1,P_2)$，该方案的一致性证明为：

$$\begin{aligned}\hat{e}(U,D_{ID}-\frac{1}{x_{ID}}P_1)&=\hat{e}(r_1Q_{ID}+r_2P_{ID},D_{ID}-\frac{1}{x_{ID}}P)\\&=\hat{e}(r_1(H_1(ID)+s)P_1+r_2x_{ID}P_2,\frac{1}{H_1(ID)+s}P_2-\frac{1}{x_{ID}}P_1)\\&=\hat{e}(r_1(H_1(ID)+s)P_1,\frac{1}{H_1(ID)+s}P_2)\hat{e}(r_2x_{ID}P_2,-\frac{1}{x_{ID}}P_1)\\&=\hat{e}(P_1,P_2)^{r_1}\hat{e}(P_1,P_2)^{r_2}\\&=g^{r_1+r_2}\end{aligned}\qquad(4\text{-}3)$$

SL-CL-PKE 方案中，主要用到四种运算操作：双线性对计算（p），标乘计算

（s），指数运算（e）和哈希运算（h），其中双线性对运算是最繁重的[29]。

在 S.S.Al-Riyami 和 K.G..Paterson 的方案（AP 方案）[10]中，用户的公钥包含 G_1 中的两个元素，而且公钥的验证需要两个双线性对运算。在 Cheng 等人的方案（CC 方案）[27]中也存在同样的问题。而 SL-CL-PKC 方案中公钥只包含 G_1 中的一个元素，公钥的验证只需要判断用户的公钥 P_{ID} 是否满足 $P_{ID} \in G_1$，整个方案只需要进行一次双线性对计算，因而在计算效率上有了很大的提高。

SL-CL-PKC 方案继承了 SK-IBE 的优点，具有以下优势：

1. 在 AP 方案和 CC 方案中都需要一个特殊的哈希函数，该函数把用户的身份信息映射为椭圆曲线上的一个点，而 SL-CL-PKC 方案只需要一个简单的普通哈希函数把用户的身份信息映射为 Z_q^* 中的一个元素。

2. 在 SL-CL-PKC 方案中，加密过程不需要双线性对计算，虽然该方案比以前的方案多了两个标乘计算，但是文献[29]中表明，一个双线性对运算需要的时间相当于二十个标乘运算，因此 SL-CL-PKC 方案中的加密过程是非常有效的。

3. AP 方案和 CC 方案的密文空间都为 $C = G_1 \times \{0,1\}^n \times \{0,1\}^n$，SL-CL-PKC 方案中的密文空间为 $C = E[q] \times \{0,1\}^n$，因而 SL-CL-PKC 方案中的密文更紧凑，适合带宽有限的环境应用。

4. 在 SL-CL-PKC 方案的解密过程中，接受者可以预先计算自己的公钥，提高解密效率。

4.5 小结

在本章中，针对上一章基于身份的认证密钥协商协议中的密钥托管问题，首先，介绍了无证书密码算法的原理、特点及定义方法；其次，介绍了 A1-Riyami 和 Paterson 的具体的无证书加密方案和无证书签名方案，并分析了 AP 方案的优点，即克服了基于身份密码系统的密钥托管问题，同时分析了 AP 方案在性能方面的不足；最后，介绍了性能比较优越的 Shiyijuan 的无证书加密方案，并对其在性能方面进行了分析比较。

第五章 网格通信认证密钥协商协议研究

5.1 引言

网格通信网为用户提供更多的信息资源共享和计算资源接入。网格通信网是一个分布式的网络系统，网格由大量的信息、计算资源站点组成，这些站点通过网格通信网形成各种虚拟组织（VO, Virtual Organization），这些虚拟组织（VO）为用户提供信息共享、计算应用等高性能服务[17]。网格通信网中的用户、成员、资源可以随时加入这种虚拟组织。各个虚拟组织拥有的各种资源都可以被虚拟组织中的成员共享访问。

不同的资源站点形成不同的信任域（Trust Domains），用户为了安全接入并访问这些虚拟组织中的资源站点，用户代理（UP, User Proxy）必须和管理这些资源站点的资源代理（RP, Resource Proxy）进行双向认证（Mutual Authentication）。

无证书公钥密码的引入克服了基于身份的公钥密码体制中存在的密钥托管限制，它结合了基于身份的公钥密码体制和传统的 PKI 体制的优点。最近许多无证书公钥加密和签名算法相继提出， S.S.Al-Riyami 和 K.G.Paterson 在提出无证书加密概念的同时给出了一个无证书可认证密钥协商方案（AP-CL-AKA 协议）[10]。近来 Benoit Libert 和 Jean-Jacques Quisquater 在构造无证书加密方案中提出了一种新的密钥生成方法，这种新的密钥生成算法构造的无证书加密方案比现有的方案更加有效。受这种密钥生成算法的启发，Shiyijuan 给出了一个无证书可认证的密钥协商协议方案（SL-CL-AKA 方案）[23]，提高了运行效率。

本文在 SL-CL-AKA 方案的基础上，给出了一个新的网格通信认证密钥协商协议，在网格通信网的用户代理（UP）和资源代理（RP）之间提供双向认证，并为随后的安全通信协商出即时的会话密钥。同时，对所给出的方案的安全性和性能进行了分析证明。

5.2 无证书可认证双方密钥协商协议的两类攻击模型

假设攻击者能够获得系统的主密钥，这类攻击者用来模仿恶意的 KGC，但是这类攻击者不允许更换用户的公钥。因此在此模型中定义两种类型的攻击者：

第一类攻击者 A_I：这类攻击者 A_I 不能够访问主密钥，但是他能够请求并替换

用户的公钥，获取用户的部分私钥和完整私钥。但是他受到如下的限制：

1. A_I 任何时候都不能够提取用户 ID_{ch} 的私钥；

2. 如果 A_I 已经替换了一个用户的公钥，那么他就不能请求该用户的私钥；

3. A_I 不能同时在挑战阶段前替换挑战用户 ID_{ch} 的公钥，在任何阶段获取用户 ID_{ch} 的部分私钥 D_{ID}；

4. 在第二阶段，A_I 不能进行对 (ID_{ch}, P_{ch}) 上挑战密文 C^* 的解密询问。

第二类攻击者 A_{II}：这类攻击者拥有主密钥，但是不能替换用户的公钥。A_{II} 由于知道主密钥，因此能够计算用户的部分私钥。A_{II} 也能进行公钥请求、私钥提取以及解密询问。这类攻击者有如下的限制：

1. A_{II} 不能替换用户的公钥；

2. A_{II} 不能获取 ID_{ch} 的私钥；

3. 在第二阶段，A_{II} 不能进行对 (ID_{ch}, P_{ch}) 上挑战密文 C^* 的解密询问。

5.3 AP-CL-AKA 协议方案

AP-CL-AKA 方案[10]包含以下几个算法：

1. 系统建立算法 S：

输入参数 k，运行 IG 生成 BDH 参数 $<G_1, G_T, e>$，其中 G_1、G_T 是两个阶为 q 的群，$\hat{e}: G_1 \times G_1 \to G_T$ 是 G_1、G_T 上的双线性映射。

随机选择一个 G_1 的生成元 $P \in G_1$。

随机选择一个数 $s \in Z_q^*$ 作为主密钥，并计算 $P_0 = sP$。

选择哈希函数：$H_1: \{0,1\}^* \to G_1^*$，$H_2: G_T \to \{0,1\}^n$。其中 n 是明文的位数。

系统参数 $params = <G_1, G_T, e, n, P, P_0, H_1, H_2>$，主密钥是 $s \in Z_q^*$，消息空间是 $M = \{0,1\}^n$，密文空间是 $C = G_1 \times \{0,1\}^n$。

2. 部分私钥抽取算法 $PPKE$：

这个算法输入用户身份 $ID_A \in \{0,1\}^*$， $params$ 以及主密钥 s，返回用户的部分私钥 D_A。

计算 $Q_A = H_1(ID_A) \in G_1^*$；

输出用户 A 的部分私钥 $D_A = sQ_A \in G_1^*$。

用户 A 可以通过计算 $e(D_A, P) = e(Q_A, P_0)$ 来验证部分私钥抽取算法是否成功。

3. 密值设置算法 SSV：

该算法输入系统参数 $params$ 和用户身份 $ID_A \in \{0,1\}^*$，选择一个随机数 $x_A \in Z_q^*$ 作为用户 A 的密值并返回该密值。

4. 私钥生成算法 $SPRI$：

该算法输入给定的系统参数 $params$，用户 A 的部分私钥 D_A 以及秘密值 $x_A \in Z_q^*$，计算私钥 $S_A = x_A D_A = x_A s Q_A \in G_1^*$。

5. 公钥生成算法 $SPUB$：

该算法输入系统参数 $params$ 以及用户 A 的秘密值 $x_A \in Z_q^*$，返回用户的公钥 $P_A = < X_A, Y_A >$，其中 $X_A = x_A P$， $Y_A = x_A P_0 = x_A sP$。

6. 密钥协商算法 AKA：

假设 A 和 B 想要协商一个会话密钥。

给定以上的初始化，A 和 B 分别选择随机数 $a \in Z_q^*$ 和 $b \in Z_q^*$，然后交换以下信息：

$$A \to B: \quad T_A = aP, \ < X_A, Y_A > \tag{5-1}$$

$$B \to A: \quad T_B = bP, \ < X_B, Y_B > \tag{5-2}$$

收到以上的交换信息，A 和 B 首先互相检验对方公钥的有效性（A 检验是否满足 $e(X_B, P_0) = e(Y_B, P)$，B 检验是否满足 $e(X_A, P_0) = e(Y_A, P)$），然后 A 计算 $K_A = \hat{e}(Q_B, Y_B)^a \cdot \hat{e}(S_A, T_B)$，B 计算 $K_B = \hat{e}(Q_A, Y_A)^b \cdot \hat{e}(S_B, T_A)$。

该协议的一致性由以下等式保证：

$$K_A = \hat{e}(Q_B, Y_B)^a \cdot \hat{e}(S_A, T_B)$$

$$= \hat{e}(Q_B, x_B P_0)^a \bullet \hat{e}(x_A s Q_A, bP)$$

$$= \hat{e}(Q_B, P_0)^{ax_B} \bullet \hat{e}(Q_A, sP)^{bx_A}$$

$$= \hat{e}(Q_B, P_0)^{ax_B} \bullet \hat{e}(Q_A, P_0)^{bx_A} \tag{5-3}$$

$$K_B = \hat{e}(Q_A, Y_A)^b \bullet \hat{e}(S_B, T_A)$$

$$= \hat{e}(Q_A, x_A P_0)^b \bullet \hat{e}(x_B s Q_B, aP)$$

$$= \hat{e}(Q_A, P_0)^{bx_A} \bullet \hat{e}(Q_B, sP)^{ax_B}$$

$$= \hat{e}(Q_A, P_0)^{bx_A} \bullet \hat{e}(Q_B, P_0)^{ax_B}$$

$$= \hat{e}(Q_B, P_0)^{ax_B} \bullet \hat{e}(Q_A, P_0)^{bx_A}$$

$$= K_A \tag{5-4}$$

因此他们共享的秘密会话密钥是 $K = K_A = K_B$，为了保证前向保密性，A 和 B 之间使用秘密会话密钥 $H_2(K \| abP)$，其中 H 是一个合适的哈希函数。

在 AP-CL-AKA 方案中，用户 A 和 B 只进行两次交互，所以频带利用率较高。在一次密钥协商过程中，每一方需要进行四次双线性对运算，运算次数较多，运行时间会相对较长，有待进一步减少双线性对运算的次数，从而减少运行时间，提高运行效率。

5.4 Shiyijuan 的无证书认证密钥协商协议（SL-CL-AKA 协议）

5.4.1 SL-CL-AKA 协议方案

Shiyijuan 的无证书可认证的双方密钥协商协议[23]包含以下算法：

1. 建立算法 S：给定安全参数 1^t，KGC 生成系统参数 $params = < q, G_1, G_T, \hat{e}, P, P_{pub}, g, H_1, H_2 >$。其中 G_1 是阶为 q 的加法循环群，G_T 是阶为 q 的乘法循环群，。P 是 G_1 的生成元，$g = \hat{e}(P, P)$。KGC 随机选择一个数 $s \in Z_q^*$ 作为他的私钥，并计算 $P_{pub} = sP$。$\hat{e}: G_1 \times G_1 \to G_T$ 是一个双线性对，

$H_1:\{0,1\}^* \to Z_q^*$，$H_2:G_T \to \{0,1\}^n$ 是密码哈希函数。

2. 部分私钥提取算法 *PPKE*：这个算法输入用户身份 $ID \in \{0,1\}^*$，*params* 以及主密钥 s，返回用户的私钥 $D_{ID} = \frac{1}{H_1(ID)+s}P$。

3. 密值设置算法 *SSV*：

该算法输入系统参数 *params* 和用户身份 $ID \in \{0,1\}^*$，选择一个随机数 $x_{ID} \in Z_q^*$ 做用户的密值并返回该密值。

4. 私钥生成算法 *SPRI*：

该算法输入给定系统参数 *params*，用户 *ID* 的部分私钥 D_{ID} 以及密值 x_{ID}，返回用户私钥 $S_{ID} = x_{ID}D_{ID}$。

5. 公钥生成算法 *SPUB*：

该算法输入系统参数 *params* 以及用户 *ID* 的密值 x_{ID}，返回用户的公钥 $P_{ID} = g^{x_{ID}}$。

6. 密钥协商算法 *KA*：

假设 A 和 B 想要协商一个会话密钥，他们的长期公私钥如下：

A 的公钥是 $P_A = g^{x_A}$，私钥是 $S_A = x_A \bullet D_A$

B 的公钥是 $P_B = g^{x_B}$，私钥是 $S_B = x_B \bullet D_B$

给定以上的初始化，A 和 B 分别随机地选择临时私钥 $a \in Z_q^*$ 和 $b \in Z_q^*$，然后交换以下信息：

$$A \to B: \quad <T_A, P_A> \text{其中} T_A = a(H_1(ID_B)P + P_{pub}) = aQ_B \tag{5-5}$$

$$B \to A: \quad <T_B, P_B> \text{其中} T_B = a(H_1(ID_A)P + P_{pub}) = aQ_A \tag{5-6}$$

收到以上的交换信息，A 和 B 首先互相检验对方公钥的有效性（Alice 检验是否满足 $P_B \in G_2$，Bob 检验是否满足 $P_A \in G_2$），然后 Alice 计算 $K_A = \hat{e}(T_B, S_A) \bullet P_B^a$，Bob 计算 $K_B = \hat{e}(T_A, S_B) \bullet P_A^b$。该协议的一致性由以下等式保证：

$$K_A = \hat{e}(T_B, S_A) \bullet P_B^a = g^{bx_A} g^{ax_B} = g^{bx_A + ax_B} \tag{5-7}$$

$$K_B = \hat{e}(T_A, S_B) \bullet P_A^b = g^{ax_B} g^{bx_A} = g^{ax_B + bx_A} \tag{5-8}$$

因此他们共享的秘密会话密钥是 $K = H_2(K_A) = H_2(K_B) = H_2(g^{ax_B+bx_A})$

5.4.2 SL-CL-AKA 的安全属性分析

可认证密钥协商协议应该具有已知会话密钥安全性、前向保密性、密钥泄露安全性和密钥分享的知晓性，下面给出具体的分析。

1. 已知会话密钥的安全性（Known Session Key Security）：

根据不同的临时私钥 a,b，协议参与者每执行一次密钥协商，会生成不同的会话密钥。

2. 前向保密性（Forward Secrecy）：

在所提协议中，其中一方的长期私钥泄露不会影响旧的会话密钥的保密性。假如攻击者知道用户 A 的私钥 S_A。对于第一类攻击者 A_{I}，利用用户 A 的私钥 S_A，他能够计算 $\hat{e}(T_B,S_A)=g^{bx_A}$，但是由于不知道 B 的私钥 S_B 和 a，他不能计算 g^{ax_B}。对于第二类攻击者 A_{II}，他知道主密钥 s。利用用户 A 的私钥 S_A，A_{II} 能够计算出 $\hat{e}(T_B,S_A)=g^{bx_A}$，并且利用主密钥 s 他能够计算 $g^a=\hat{e}(T_A,\frac{1}{H_1(ID_B+s)}P)$，但是他不能计算出 g^{ax_B}。给定 g^a 和 $P_B=g^{x_B}$，计算 g^{ax_B} 将面临 CDH 难题。因此，所提协议提供部分前向保密性。但是如果协议双方的长期私钥都泄露了，攻击者就能够计算出旧的会话密钥。假定攻击者知道了双方的长期私钥 S_A 和 S_B，那么他就能给计算出会话密钥 $\hat{e}(T_B,S_A)\hat{e}(T_A,S_B)=g^{bx_A}g^{ax_B}$，因此，该协议只提供部分前向保密性。

3. 密钥泄露的安全性（No Key-compromise Impersonation）：

如果第一类攻击者 A_{I} 知道用户 A 的长期私钥 S_A，他想在协议中冒充 B。利用 S_A，A_{I} 能够计算 $\hat{e}(T_B,S_A)$，但是他不能够计算出 P_B^a。甚至对于第二类攻击者 A_{II}，由于拥有主密钥 s 他能够计算 $g^a=\hat{e}(T_A,\frac{1}{H_1(ID_B+s)}P)$，但是他不能够计算出 P_B^a。给定 g^a 和 $P_B=g^{x_B}$，在 CDHP 假设下很难计算出 g^{ax_B}。

4. 密钥分享的知晓性（Unknown Key-share）：

要想实现这样的攻击，攻击者需要知道用户的长期私钥，因此这种攻击很难

实现。实际上，隐身密钥认证的安全属性隐含了密钥分享的知晓性，因此，所提协议满足密钥分享的知晓性。

5. 已知临时私钥的信息安全性（Known session-specific temporary information security）：

会话中，用户临时私钥的泄露将会导致会话密钥的泄露。在 SL-CL-AKA 协议中，如果攻击者在每次会话中截获 A 和 B 的临时私钥$<a,b>$，攻击者就能够计算出会话密钥$K=(P_A)^b\cdot(P_B)^a=g^{bx_A}\cdot g^{ax_B}=g^{bx_A+ax_B}$，因此，SL-CL-AKA 协议不具有已知临时私钥的信息安全性。

6. 密钥复制攻击（Key replicating attack）：

密钥复制攻击是中间人攻击的一种。一个攻击者可以截获并适当的修改协议参与者协商会话密钥时的交换信息，使得协议双方接受持有一个相同的会话密钥，而该会话密钥并不是协议双方真正自己想要得到的会话密钥。

对 SL-CL-AKA 协议的密钥复制攻击如下：

(1). $A\rightarrow B$：$<T_A,P_A>$

攻击者 E（Eve）截获并篡改 A 发往 B 的消息，随机选取$c\in Z_q^*$，计算$T_A'=cT_A$和$P_A'=(P_A)^c$，并假扮 A，将$<T_A',P_A'>$发送给 B；

(1). $Eve\rightarrow B$：$<T_A',P_A'>$

(2). $B\rightarrow A$：$<T_B,P_B>$

攻击者 E（Eve）截获并篡改 B 发往 A 的消息，随机选取和上面相同的$c\in Z_q^*$，计算$T_B'=cT_B$和$P_B'=(P_B)^c$，并假扮 B，将$<T_B',P_B'>$发送给 A；

(2). $Eve\rightarrow A$：$<T_B',P_B'>$

A 和 B 分别计算会话密钥K_A和K_B：

A：$K_A=\hat{e}(T_B',S_A)\cdot(P_B')^a=\hat{e}(cT_B,S_A)\cdot(P_B^c)^a=g^{cbx_A}g^{cax_B}=g^{(bx_A+ax_B)c}$； (5-9)

B：$K_B=\hat{e}(T_A',S_B)\cdot(P_A')^b=\hat{e}(cT_A,S_B)\cdot(P_A^c)^b=g^{cax_B}g^{cbx_A}=g^{(bx_A+ax_B)c}$； (5-10)

显然，从上面可以得出$K_A=K_B$，表明 A 和 B 计算并接受持有了一个相同的会话密钥$g^{(bx_A+ax_B)c}$。而实际上，A 和 B 希望协商得到的会话密钥应该是$g^{bx_A+ax_B}$。

最后，E（Eve）向 B 发送一个 Reveal 询问，RP 将会把协商所得的会话密钥返回给 E（Eve）：

$Eve \rightarrow B$：Reveal

$B \rightarrow Eve$：$g^{(bx_A+ax_B)c}$

在以上的密钥复制攻击中，A 和 B 都接受持有了一个相同的会话密钥，然而，A 和 B 都不是正常的密钥协商伙伴，因为他们都没有进行匹配会话，只是和攻击者进行了会话。攻击者 E（Eve）通过向 A 或者 B 发送一个 Reveal 询问，就能够得到最新生成的会话密钥。所以，由以上分析可知，SL-CL-AKA 协议方案不能够抵抗密钥复制攻击。

5.4.3 SL-CL-AKA 的性能分析

将 SL-CL-AKA 协议与 S.S.Al-Riyami 和 K.G.Paterson 提出的无证书可认证双方密钥协商协议（AP-CL-AKA 协议）的性能进行比较。和 AP 协议相比，SL-CL-AKA 协议具有以下优点：

我们已经知道标乘和指数运算要比双线性对计算快的多，所提协议每个用户只需要一个对计算，公钥的验证只需要一个简单的群元素检查。而 AP 协议中的每个用户需要进行四个双线性对计算，因此在计算量上，SL-CL-AKA 协议占有很大的优势。

AP 的 CL-AKA 协议和他们的 CL-PKE 方案一样都需要一个特殊的函数 MaptoPoint，该函数将用户的身份信息映射为椭圆曲线上的一个点，其计算消耗比较大，而 SL-CL-AKA 协议只需要一个简单的普通哈希函数把用户的身份信息映射为 Z_q^* 中的一个元素。

在 128bit 的安全等级条件下，取 k=12 时，G_1 的单个元素为 256bits，G_T 的单个元素为 3072bits[34]。

SL-CL-AKA 协议的公钥只包含了 G_T 中的一个元素，而在 AP 协议中的公钥包含了 G_1 中的两个元素。

SL-CL-AKA 协议的带宽，即信令开销，包含 G_1 中的一个元素和 G_T 中的一个元素，而 AP 协议中的带宽包括了三个 G_1 中的元素，因此，SL-CL-AKA 协议在信令开销方面还有待进一步改善。

由以上的安全属性和性能分析可知，虽然 SL-CL-AKA 协议在安全和性能方面相对于 AP-CL-AKA 协议有了很大的提升，但是 SL-CL-AKA 协议只提供部分前向安全性，不具有完善的前向安全属性，当攻击者同时知道双方的长期私钥时，就能计算出会话密钥；同时，SL-CL-AKA 协议不能够抵抗密钥复制攻击，当中间攻击者截获并修改交换信息时，攻击者能够控制会话密钥的生成，对协议的密钥控制安全属性构成威胁；SL-CL-AKA 协议还不具有已知临时私钥的信息安全性，当每次的临时会话私钥泄露时，攻击者能够计算得到会话密钥。针对这些不足，我们在 SL-CL-AKA 协议的基础上，给出了一个新的基于无证书的网格通信认证密钥协商协议。

5.5 改进的基于无证书的网格通信认证密钥协商协议

基于 Shiyijuan 所提的无证书双方认证密钥协商协议（SL-CL-AK 协议），我们给出一个应用于网格通信的改进的无证书双方认证密钥协商协议方案，在网格通信网的用户代理（UP）和资源代理（RP）之间提供双向认证，并为随后的安全通信协商出即时的会话密钥。

5.5.1 所给协议的方案

为了达到对网格通信系统中用户与资源的分散、自治管理，要求网格通信系统包含多个信任域，在每一个信任域中设置一个网格密钥生成中心（Grid Key Generate Center，GKGC），作为该信任域内的网格权威机构，用于注册网格实体身份、系统初始化、生成部分用户私钥等。

改进的协议方案由以下三部分组成：

一. 系统建立

系统建立算法S：该算法由网格密钥生成中心（Grid Key Generate Center，GKGC）完成。输入1^k（k 是系统安全参数），执行以下步骤：

1. 生成一个 Weil 对$e: E[q]\times E[q]\to G_T$，其中$E[q]=G_1\oplus G_2$，$\varphi$是$G_2$到$G_1$的映射。随机选择一个$G_2$上的生成元$P_2\in G_2^*$并设置$P_1=\varphi(P_2)$。

2. 随机选择一个数$s\in Z_q^*$作为主密钥，并且计算$P_{pub}=sP_1$。

3. 计算$g=\hat{e}(P_1,P_2)$。

4. 选择哈希函数：$H_1:\{0,1\}^*\to Z_q^*$，$H_2:G_T\to\{0,1\}^n$，$H:\{0,1\}^*\to\{0,1\}^{k_1}$其

中，$k_l = |sk|$，sk 是协议最终生成的网格用户代理 UP 和资源代理 RP 之间的会话密钥。

消息空间是 $M = \{0,1\}^n$，密文空间是 $C = E[q] \times \{0,1\}^n$。KGC 发布公开系统参数 $params = < q, G_1, G_2, G_T, e, n, P_1, P_2, g, P_{pub}, H_1, H_2 >$，主密钥是 s。

二．密钥对生成

若身份为 ID 的实体 A 想要成为网格通信中的用户，实体 A 必须在网格密钥生成中心（Grid Key Generate Center，GKGC）处注册身份 ID，然后按照以下算法生成用户公钥和私钥。

1. 部分私钥提取算法 $PPKE$：这个算法输入用户身份 $ID \in \{0,1\}^*$，$params$ 以及主密钥 s，返回用户的部分私钥 $d_{ID} = \frac{1}{H_1(ID)+s} P_2 \in G_2^*$，该算法由 GKGC 完成。

GKGC 生成用户的部分私钥后，将部分私钥 d_{ID} 发送给身份为 ID 的用户。

2. 密值设定算法 SSV：该算法输入系统参数 $params$ 和用户身份 $ID \in \{0,1\}^*$，选择一个随机数 $x_{ID} \in Z_q^*$ 作为用户的密值并返回该密值，该算法在网格用户处实现。

3. 私钥生成算法 $SPRI$：该算法输入给定系统参数 $params$，用户 ID 的部分私钥 d_{ID} 以及密值 x_{ID}，返回用户私钥为 $S_{ID} = (d_{ID}, x_{ID})$，该算法在网格用户处实现。

4. 公钥生成算法 $SPUB$：该算法输入系统参数 $params$ 以及用户 ID 的密值 x_{ID}，返回用户的公钥 $P_{ID} = x_{ID} P_2$，该算法在网格用户处实现。

三．相互认证及密钥协商

一旦两个实体在 GKGC 处注册之后，双方知道对方的 ID，即如果用户代理知道资源代理的 ID-RP，资源代理也知道了用户代理的 ID-UP。在网格用户代理 UP 和网格资源代理 RP 进行安全通信前，双方就可以按照下面的算法进行相互认证并计算出通信的即时会话密钥。

密钥协商算法 KA：

假设网格通信网中用户代理 UP 和资源代理 RP 需要进行认证密钥协商，它们各自的长期公钥和长期私钥分别为：

$$UP : < P_{UP}, S_{UP} > = < x_{UP} P_2, d_{UP} : x_{UP} >; \tag{5-11}$$

$$RP:< P_{RP}, S_{RP} >=< x_{RP}P_2, d_{RP} : x_{RP} >. \tag{5-12}$$

给定以上的初始化，计算$Q_{ID} = H_1(ID)P_1 + P_{pub} \in G_1^*$，UP 和 RP 分别随机地选择临时私钥$r_{UP1}, r_{UP2} \in Z_q^*$和$r_{RP1}, r_{RP2} \in Z_q^*$，然后交换以下信息：

$$UP \rightarrow RP: \quad < T_{UP1}, P_{UP} > \tag{5-13}$$

其中$T_{UP1} = T_{UP11} \| T_{UP12} \| T_{UP13} = r_{UP1}Q_{RP} \| r_{UP2}P_{RP} \| r_{UP2}P_{UP}$；

$$RP \rightarrow UP: \quad < T_{RP1}, P_{UP} > \tag{5-14}$$

其中$T_{RP1} = T_{RP11} \| T_{RP12} \| T_{RP13} = r_{RP1}Q_{UP} \| r_{RP2}P_{UP} \| r_{RP2}P_{RP}$；

收到以上的交换信息，UP 和 RP 首先互相检验对方公钥的有效性（UP 检验是否满足$P_{RP} \in G_2$，RP 检验是否满足$P_{UP} \in G_2$），然后 UP 和 RP 分别进行以下计算：

UP：$T_{UP2} = \hat{e}((r_{RP1}Q_{UP} + r_{RP2}P_{UP}), d_{UP} - \frac{1}{x_{UP}}P_1) \cdot g^{(r_{UP1}+r_{UP2})}$

$$= g^{(r_{RP1}+r_{RP2})} \cdot g^{(r_{UP1}+r_{UP2})}$$

$$= g^{(r_{UP1}+r_{UP2}+r_{RP1}+r_{RP2})} \tag{5-15}$$

$$T_{UP3} = x_{UP}r_{UP2}T_{RP13} = x_{UP}r_{UP2}r_{RP2}x_{RP}P_2 \tag{5-16}$$

$$K_{UP} = H(T_{UP1} \| T_{RP1} \| T_{UP2} \| T_{UP3}) \tag{5-17}$$

RP：$T_{RP2} = \hat{e}((r_{UP1}Q_{RP} + r_{UP2}P_{RP}), d_{RP} - \frac{1}{x_{RP}}P_1) \cdot g^{(r_{RP1}+r_{RP2})}$

$$= g^{(r_{UP1}+r_{UP2})} \cdot g^{(r_{RP1}+r_{RP2})}$$

$$= g^{(r_{UP1}+r_{UP2}+r_{RP1}+r_{RP2})} \tag{5-18}$$

$$T_{RP3} = x_{RP}r_{RP2}T_{UP13} = x_{UP}r_{UP2}r_{RP2}x_{RP}P_2 \tag{5-19}$$

$$K_{RP} = H(T_{UP1} \| T_{RP1} \| T_{RP2} \| T_{RP3}) \tag{5-20}$$

该协议的一致性：

由上面式(5-15)和式(5-18)可知$T_{UP2} = T_{RP2}$，由式(5-16)和式(5-19)可知

$T_{UP3}=T_{RP3}$。所以共享会话密钥 sk 为：

$$sk=K_{UP}=K_{RP}=H(T_{UP1}\|T_{RP1}\|g^{(r_{UP1}+r_{UP2}+r_{RP1}+r_{RP2})}\|x_{UP}x_{RP}r_{UP2}r_{RP2}P_2) \tag{5-21}$$

5.5.2 所给协议的安全属性分析

这一节，我们分析所给协议的安全属性，分析说明所给新协议几乎具有所有的认证密钥协商协议期望的安全属性，相比 SL-CL-AKA 方案，该方案提供了完善的前向保密安全性、已知临时私钥的信息安全性。同时，我们从攻击方面分析了所给协议的安全性，它能够抵挡一些已知的攻击，如密钥复制攻击和公钥替换攻击。

可认证密钥协商协议应该具有已知会话密钥安全性、前向保密性、密钥泄露安全性和密钥分享的知晓性，下面给出具体的分析。

1. 隐式密钥认证（Implicit Authentication）：

在所给协议中，协议参与者（UP 或 RP）在计算会话密钥时，都要用到自己的私钥，将私钥融入到会话密钥的计算过程中。而私钥被认为是只有用户（UP 或 RP）自己才知道的。因此，只有合法的用户才能够计算出会话密钥，也就是说协议提供了隐式密钥认证。

2. 已知会话密钥安全性（Known Session Key Security）：

根据 UP 和 RP 随机选择的不同的临时私钥 $r_{UP1},r_{UP2}\in Z_q^*$ 和 $r_{RP1},r_{RP2}\in Z_q^*$，协议参与者每执行一次密钥协商，会生成不同的会话密钥。

3. 前向保密性（Forward Secrecy）：

在所给出的协议中，任何一方（UP 或 RP）的长期私钥泄露或者双方（UP 和 RP）的长期私钥泄露都不会影响旧的会话密钥的保密性。假定攻击者知道了双方（UP 和 RP）的长期私钥 S_{UP} 和 S_{RP}，如果不知道临时私钥 $<r_{UP1},r_{UP2}>$（或者 $<r_{RP1},r_{RP2}>$），很明显他不能够计算出 T_{UP2} 和 T_{UP3}（或者 T_{RP2} 和 T_{RP3}），所以该协议具有完善的前向保密性。

4. KGC 前向安全性（KGC forward secrecy）：

不存在会话密钥托管是基于无证书公钥方案的固有属性。KGC 的主密钥 s 的泄露不会导致以前建立的会话密钥的泄露。在该方案中，尽管攻击者可能得到 KGC 产生的部分私钥，但是要计算出会话密钥，还必须获取用户的临时私钥和部分长期私钥。

5. 密钥泄露的安全性（No Key-compromise Impersonation）：

所给协议具有密钥泄露安全属性，可以抵抗密钥泄露攻击。假设攻击者获取了 UP 的私钥，攻击者想要假扮 RP 和 UP 进行密钥协商，虽然攻击者能够替换 B 的公钥使得$T_{UP13}=T_{RP13}$，但是，如果攻击者不知道 RP 的私钥，就不能够计算出相应的T_{RP2}和T_{RP3}，使得$T_{UP2}=T_{RP2}$和$T_{UP3}=T_{RP3}$，所以攻击者假扮 RP 和 UP 进行密钥协商失败，所给协议具有密钥泄露安全性。

6. 密钥分享的知晓性（Unknown Key-share）：

在协议的执行过程中，任意一方（UP）都是将对方（RP）的公钥嵌入到了会话密钥的计算当中的。而每一个参与者的公钥都是由代表其身份的字符串经过哈希函数运算得来的。可见，攻击者是不能够冒充参与者（RP）与参与者（UP）协商会话密钥的。即该协议可以抵抗未知密钥共享攻击，提供了密钥分享的知晓性。

7. 密钥控制安全性（No Key Control）：

在产生会话密钥的过程中，协议双方（UP 和 RP）独立选择随机的短私钥，因此任何一方（UP 或 RP）都不能最终决定会话密钥。要想实现这样的攻击，攻击者需要知道用户的长期私钥，因此这种攻击很难实现。实际上，隐身密钥认证的安全属性隐含了密钥分享的知晓性，因此，所给出的协议满足密钥分享的知晓性。

8. 已知临时私钥的信息安全性（Known session-specific temporary information security）：

会话中，用户临时私钥的泄露不会导致会话密钥的泄露。在该协议中，即使在每次会话中截获 UP 和 RP 的临时私钥$<r_{UP1},r_{UP2},r_{RP1},r_{RP2}>$，攻击者也不能够计算出$T_{UP2}$和$T_{UP3}$（或者$T_{RP2}$和$T_{RP3}$），因此，该协议具有已知临时私钥的信息安全性。

9. 消息独立性（Message independence）：

在该协议中，因为 UP 和 RP 独立选取临时私钥$<r_{UP1},r_{UP2}>$和$<r_{RP1},r_{RP2}>$，所以协商的会话密钥流是不相关的。

10. 密钥复制攻击（Key replicating attack）：

密钥复制攻击是中间人攻击的一种，如果允许攻击者向已经接受持有会话密钥的协议参与者发送 Reveal 询问，即攻击者要求 oracle 给出他接受持有的会话密钥，则将会影响认证密钥协商协议的密钥控制安全属性。在所给的认证密钥协商方案中，攻击者对密钥协商过程中交换信息的任何修改都将导致协议双方生成不同的会话密钥，即最终协议参与者不能够接受持有一个相同的会话密钥。在协议

中，外部攻击者可以通过将T_{UP1}和T_{RP1}替换为T'_{UP1}和T'_{RP1}，使得$T_{UP3}=T_{RP3}$，但是此时将导致$T_{UP2}\neq T_{RP2}$，从而，协议双方不会生成相同的会话密钥，所以，所给的认证密钥协商方案可以抵抗密钥复制攻击。

在安全属性方面，将所给协议与 AP-CL-AKA 方案及 SL-CL-AKA 方案进行比较，其中，PFS 表示完善的前向安全性，KCI-R 表示密钥泄露安全性，UKS-R 表示密钥分享的知晓性，KSSTIS 表示已知临时私钥的信息安全性，KRA-R 表示密钥复制攻击安全性。如表 5.1 所示，其中“√”表示协议具有某安全属性，“×”表示协议不具有安全属性：

表 5.1 协议安全属性比较

认证密钥协商协议	安全属性				
	PFS	KCI-R	UKS-R	KSSTLS	KRA-R
AP-CL-AKA 协议	√	√	√	×	×
SL-CL-AKA 协议	×	√	√	×	×
所给的 CL-AKA 协议	√	√	√	√	√

5.5.3 所给协议的性能分析

将所给协议与 AP-CL-AKA 协议及 SL-CL-AKA 协议的性能进行比较。和以上两个协议方案相比，所给出的协议方案具有以下优点：

标乘运算和指数运算要比双线性对运算快的多，所给协议每个用户只需要一个双线性对计算，公钥的验证只需要一个简单的群元素检查，和 SL-CL-AKA 协议相同。而 AP 协议中的每个用户需要进行四个双线性对计算，因此在计算量上，所提协议占有很大的优势。

AP 的 CL-AKA 协议和他们的 CL-PKE 方案一样都需要一个特殊的函数 MaptoPoint，该函数将用户的身份信息映射为椭圆曲线上的一个点，其计算消耗比较大，而所给协议基于 SL-CL-AKA 协议，都是只需要一个简单的普通哈希函数把用户的身份信息映射为Z_q^*中的一个元素。

所给协议的公钥只包含了G_2中的一个元素，SL 协议中公钥包含G_T中的一个元素，而在 AP 协议中的公钥包含了G_1中的两个元素。

所给协议的带宽包括G_1中的一个元素和G_2中的三个元素，而 SL 协议包含G_1中的一个元素和G_T中的一个元素，AP 协议中的带宽包括了三个G_1中的元素，所

以所给出的协议在带宽上不具有优势。

在这三个协议中，一方的计算量和通信复杂度比较如下表所示，其中，P 表示双线性对运算，S 表示点乘运算，H 表示 MaptoPoint hash 运算，E 表示指数运算。

所给出的 CL-AKA 协议方案性能比较如表 5.2 所示：

表 5.2 协议性能比较

认证密钥协商协议	计算量				公钥	带宽
	P	S	H	E		
AP-CL-AKA 协议	4	2	1	1	G_1 中 2 个元素	G_1 中 3 个元素
SL-CL-AKA 协议	1	2	0	1	G_T 中 1 个元素	G_1 中 1 个元素和 G_T 中 1 个元素
所给的 CL-AKA 协议	1	4	0	1	G_2 中 1 个元素	G_1 中 1 个元素和 G_2 中 3 个元素

在协议参与方（UP 或者 RP）为 Pentium IV 2.1GHz 处理器的条件下，文献[34, 35]给出了类似处理器上密码操作的运行时间。表 5.3 列出了不同密码操作的运行时间：

表 5.3 密码操作运算时间（毫秒）

	点乘	G_T 中的指数	G_1 中的哈希	双线性对
协议参与方	3.8	9.7	3.0	79.8

基于上述实验结果，采样估算的方法表示不同协议的运行时间。在所给方案中，协议参与方（UP 或者 RP）进行一次认证密钥协商，需要进行 4 次点乘运算、1 次 G_T 中的指数运算和 1 次双线性对运算，运算时间为 $3.8\times4+2.7\times1+79.8\times1=97.7$ ms。

信令开销用协议方案中需要传输消息的总长度表示。在 128bit 的安全等级条件下，取 k=12 时，G_1、G_2 和 G_T 的单个元素分别为 256bits、512bits 和 3072bits[34]。在所给方案中，协议参与者（UP 或者 RP）在认证密钥协商过程中交互的消息包括 G_1 中的 1 个元素和 G_2 中的 3 个元素，因此相应的信令开销为 $256\times1+512\times3=1792bits$。

采样相同的方法估计 AP-CL-AKA 方案和 SL-CL-AKA 方案的运算时间及信令开销，见表 5.4。

表 5.4 协议性能比较

	AP-CL-AKA 协议	SL-CL-AKA 协议	所给协议方案
运算时间（ms）	339.5	97.1	104.7
信令开销（bit）	768	3328	1792

在性能方面，考虑到双线性对运算、MaptoPoint Hash 运算及指数运算的计算复杂度大约分别是点乘运算的 20 倍、1 倍及 3 倍[33]，由表 5.4 可知：所给出的 CL-AKA 协议方案的计算效率比 AP-CL-AKA 方案提高了 69.2%，比 SL-CL-AKA 方案降低了 7.82%；在信令开销方面，所给协议比 SL-CL-AKA 方案减小了 46.1%。

与 AP-CL-AKA 协议方案相比，所给协议方案的计算效率有了很大提高。虽然计算效率比 SL-CL-AKA 协议方案降低了 7.82%，但是信令开销减小了 46.1%，同时，在安全性方面，所给出的协议方案能够提供良好的安全属性，与 AP-CL-AKA 协议方案和 SL-CL-AKA 协议方案相比，不但具有完善的前向安全性、已知临时私钥的信息安全性，而且能够抵抗密钥复制攻击。

5.6 小结

无证书公钥密码克服了基于身份的公钥密码体制中存在的密钥托管问题，我们在 Shiyijuan 的无证书双方可认证密钥协商协议（SL-CL-AKA）的基础上，给出了一种改进的基于无证书的网格通信双方认证密钥协商协议，继承了无证书加密系统（SL-CL-PKE）和 SL-CL-AKA 协议的优点，同时，具有良好的安全属性和计算性能，所给协议不但具有完善的前向保密安全性、已知临时私钥的信息安全性和其他认证密钥协商协议所期望的安全属性，而且能够抵御密钥复制攻击。虽然相对于 SL-CL-AKA 协议，在开销上略有不足，但是与现有的协议相比，该协议方案更加安全有效。

第六章 总结与展望

6.1 论文总结

网格是下一代网络，它能够把分散在网络中不相同的、分布式的各种软件和硬件资源组合起来，有效地组织成为一个虚拟组织，实现对分散资源的共享以及系统工作，进而为科学和工程技术人员提供一个计算环境，为整个网格环境中的用户提供以往网络环境无法提供的资源共享和服务。由此可以看到网格通信网组成复杂，并且其中必定涉及到巨大的经济利益，这样网格通信中的安全体制必须成为网格环境中一个不可缺少的重要保证，而安全认证是安全服务的基础，其中，作为网格安全系统的基础之一，认证密钥协商就十分重要了。

本文从目前网格通信安全的现状入手，分析了当前存在的一些网格通信安全认证问题，并着重介绍了网格环境下的认证密钥协商技术。通过对目前网格通信认证密钥协商研究现状及存在问题的分析，针对所存在的问题进行了深入的研究。

首先，介绍了网格的概念和特点、网格的体系结构，认证密钥协商技术的概念，以及在网格安全中的重要意义和研究现状。

其次，对基于身份的认证密钥协商协议进行了研究，包括 Smart 的认证密钥协商协议和 Chen 等的认证密钥协商协议，通过对它们进行认证密钥协商协议的安全属性分析，发现以上协议都不具备完善的前向安全性，且都存在密钥托管问题和密钥撤销问题。

最后，重点研究了适合于网格通信环境的无证书密码系统，介绍了 Al-Riyami 和 Paterson 的基于无证书的认证密钥协商方案（AP-CL-AKA）和 Shiyijuan 的无证书认证密钥协商方案（SL-CL-AKA），并对这些方案进行了安全属性和性能分析，基于 SL-CL-AKA 协议方案给出了网格通信认证密钥协商协议，并将所给协议方案与 AP-CL-AKA、SL-CL-AKA 进行了比较，所给方案更加安全有效。

6.2 展望

本文虽然做了一些工作，但由于一些时间和能力方面的限制，仍有一些问题需要一步的研究和探索：

1. 本文研究的认证密钥协商协议方案都是双方密钥协商协议，网格通信中的多方认证密钥协商协议方案（认证组密钥协商协议）是下一步研究的方向。

2. 基于无证书的认证密钥协商协议还存在密钥撤销问题。用户的公钥是由用户的身份 ID 参与生成的。比如，若用户 B 的身份 ID 发送了变化，用户 A 还是使用 B 原来公钥进行会话密钥协商，那么 B 还可以用原来的身份得到会话密钥，即密钥撤销问题。认证密钥协商协议中的密钥撤销问题有待于进一步研究。

致 谢

衷心地感谢我的导师孙献璞教授，在我攻读硕士学位期间给予我学习上的指导和生活上的照顾。孙老师所营造的良好的学习气氛和工作环境使我受益匪浅，他以渊博的知识、严谨的治学态度、敏锐的观察力给了我莫大的帮助和启发；孙老师谦和近人、孜孜以求的作风，是永远值得我学习的。非常感谢孙老师给我的学习和生活上的无私的关心和帮助，在此表示我最诚挚的敬意和最衷心的感谢！

衷心感谢我的实验室同学邹琮、杨飞、苏鉴、张杰群、吕智杰、张超、孙金霞、刘鸽、汪宁、董宏伟，与他们在学术上的探讨和交流使我开阔了眼界和视野，增进了我对知识的理解，同时感谢他们在生活上对我的关心和鼓励，和他们一起度过了美好宝贵的研究生时光。

感谢师兄王军、赵晓雷、张玉山、杨锐、赵雨、郭森宝、陆巍炜，师姐李璇、王馨莉，感谢他们对我的帮助和鼓励。

感谢师弟黄宇、魏忠山、张华伟，师妹明彩叶、张怡，感谢他们在各方面对我的支持。

感谢我所有的朋友孔宪庆、王军、袁亮、杜磊等，他们和我一起度过了这朴实而难忘的研究生岁月。他们的指导、支持与帮助使我的生活挫折中具有美丽，平淡中更见芬芳。

衷心地感谢我的父母和亲人，感谢他们把我养育成人，感谢他们为我的成长付出的巨大艰辛。我在人生路途上的每一步历程都凝聚着他们的心血和嘱咐。我将以此为激励，继续努力，在将来的工作和生活中争取更大的进步！

参考文献

[1] 桂小林编著，网格技术导论，北京：北京邮电大学出版社，2005；

[2] Joshy Joseph, Craig Fellenstein 著，战晓苏，张少华译，网格计算，北京：清华大学出版社，2005；

[3] Francine Berman, Geoffrey C.Fox, Tony Hey 编著，都志辉等译，网格计算：支持全球化资源共享与协作的关键技术，武汉：华中理工大学出版社，2005；

[4] Charlie Kaufman，Radia Perlman，Mike Special 著，许剑卓，左英男等译，网络安全：公众世界中的秘密通信，电子工业出版社，2004；

[5] Marty Humphrey, Mary R. Thompson and Keith R. Jackson，Security for Grids，PROCEEDINGS OF THE IEEE, VOL.93, NO.3, MARCH 2005.

[6] A. Shamir，Identity-Based Cryptosystems and Signature，Advanced in Cryptology-CRYPTO1984，LNCS196，1985：47-53.

[7] Chen L, Kudla C. Identity based authenticated key agreement protocols from pairings. CSFW'03.UK:IEEE,2003,219-236.

[8] Li Hongwei, Sun Shixin and Yang Haomiao, Identity-based authentication protocol for grid, Journal of Systems Engineering and Electronics, Vol. 19, No. 4, 2008, pp.860-864.

[9] Wenbo Mao, An Identity-based Non-interactive Authentication Framework for Computational Grids, HPL-2004-96, June 3, 2004.

[10] Sattam S. Al-Riyami and Kenneth G. Paterson, Certificateless Public Key Cryptography, C.S.Laih(Ed.):ASIACRYPT 2003, LNCS 2894, PP.452-473, 2003.

[11] Craig Gentry and Alice Silverberg, Hierarchical ID-Based Cryptography, ASIACRYPT 2002,LNCS 2501, pp. 548-566, 2002.

[12] Joonsang Baek, Reihaneh Safavi-Naini and Willy Susilo, Certificateless Public Key Encryption Without Pairing, ISC2005, LNCS 3650, pp.134-148, 2005.

[13] Tarjei K. Mandt and Chik How Tan, Certificateless Authenticated Two-Party Key Agreement Protocols, M.Okada and I.Satoh (Eds.): ASIAN 2006, LNCS 4435, pp.37-44,2007.

[14] Blake-Wilson S, and Menezes A, Authenticated Diffie-Hellman Key Agreement Protocols. In:Tavares,S., Meijer H.(eds.) SAC 1998. LNCS, vol.1556, pp.339-361. Springer, Heidelberg(1998).

[15] Boneh, D., Franklin, M.: Identity-Based Encryption from the Weil Pairing. In:

Kilian, J. (ed.) CRYPTO 2001. LNCS, vol. 2139, pp. 213–229. Springer, Heidelberg(2001).

[16] Mitchell, C.J., Ward, M., Wilson, P.: Key Control in Key Agreement Protocols. Electronics Letters 34, 980–981 (1998).

[17] Shengbao Wang, Zhenfu Cao, and Haiyong Bao: Efficient Certificateless Authentication and Key Agreement(CL-AK) for Grid Computing, International Journal of Network Security, Vol.7,No.3,PP.342-347,Nov.2008.

[18] S. S. Al-Riyami and K. G. Paterson. "CBE from CL-PKE: A generic construction and efficient schemes." Proceedings PKC 2005, LNCS 3386, pp. 398-415,Springer-Verlag, Berlin, 2005.

[19] L. Chen and C. Kudla, "Identity based key agree-ment protocols from pairings," Proceedings of the 16th IEEE Computer Security Foundations Work-shop, pp. 219-213, IEEE Computer Society, 2002.

[20] I. Foster, C. Kesselman, G. Tsudik, and S. Tuecke,"A security architecture for Computational Grids," Proceedings of 5th ACM Conference on Computer and Communications Security, pp. 83-92, 1998.

[21] Smart N P. An Identity based authenticated key agreement protocol based on the Weil pairing. Electronics Letters, 2002,38:630-632.

[22] D. H. Yum, and P. J. Lee, "Generic construction of certificateless encryption," Proceedings of ICCSA 2004, LNCS 3043, pp. 802-811, Springer-Verlag, 2004.

[23] Shi Yijuan and Li Jianhua: Two-Party Authenticated Key Agreement in Certificateless Public Key Cryptography, Wuhan University Journal of Natural Sciences Vol.12 No.1 2007.

[24] Cheng Z H, Comley R. Efficient Certificateless Public Key Encryption [EB/OL]. 2005-01-19.

[25] D. Boneh, M. Franklin, Identity-based Encryption from the Weil Pairing, Advances in Cryptology-Crypto'2001,Lecture Notes on Computer Science 2139, Springer-verlag(2001), pp.213-229.

[26] B. Libert and JJ. Quisquater, On constructing Certificateless Cryptosystems from Identity Based Encryption, PKC 2006, LNCS3958, pp.474-490, 2006.

[27] Z. H. Cheng, R. Comley, and L. Vasiu, Remove Key Escrow from The Identity-Based Encryption System, lin Foundations of Information Technology in the Era of Network and Mobile Computing. France, August 2004.

[28] Shamir A. Identity-based cryptosystems and signature schemes. Advances in Cryptology-CRYPTO'84, LNCS196. Berin: Springer-Verlag, 1984, 47-53.

[29] P. Barreto, H. Kim, B. Lynn and M. Scott, Efficient Algorithms for Pairing-based Cryptosystems, Advances in Cryptology-Crypto 2002, LNCS2442, pp.354-368, Spring-Verlag, 2002.

[30] Diffie W, Hellman M. New directions in cryptography. IEEE Trans. Inform. Theory, 1976, IT-22(6): 44-654.

[31] W. Mao, An Identity-Based Non-Interactive Authentication Framework For Computational Grids, HP Laboratories Bristol, HPL-TR-2004-96, 2004.

[32] Hoon Wei Lim and Kenneth G.Paterson, Identity-Based Cryptography for Grid Security, Proceeding of the First International Conference on e-Science and Grid Computing (e-Science'05).

[33] Z. Cheng, "Pairing-based cryptosystems and key exchange protocols," Ph.D Thesis of Middleser University, 2007.

[34] W. Dai, "Crypto++ 5.2.1 Benchmarks," http://www.eskimo.com/ weidai/ bench-marks.html, 2004.

[35] Y. Zhang, W. Liu, Y. Fang, "Securing mobile Ad Hoc networks with certificateless public keys," IEEE Trans. Depend. Secure. Compute. 3(4):386-399, 2006.