ICS 35.020
L09

GA

中华人民共和国公共安全行业标准

GA/T 711-2007

信息安全技术
应用软件系统安全等级保护通用技术指南

Information security technology-
Common technique guide of security classification protection for application software system

2007-08-13 发布　　2007-10-01 实施

中华人民共和国公安部　发布

目　　次

前　　言

（略）

引　　言

本标准为按照信息系统安全等级保护的要求设计和实现所需要的安全等级的应用软件系统提供指导，主要说明为实现 GB 17859—1999 所规定的每一个安全保护等级，应用软件系统应达到的安全技术要求。

应用软件系统是信息系统的重要组成部分，是信息系统中对应用业务进行处理的软件的总和。业务应用的安全需求，是信息系统安全需求的出发点和归宿。信息系统安全所采取的一切技术和管理措施，最终都是为确保业务应用的安全。这些安全措施，有的可以在应用软件系统中实现，有的需要在信息系统的其它组成部分实现。

本标准是对各个应用领域的应用软件系统普遍适用的安全技术要素的概括描述。不同应用领域的应用软件系统应根据需要选取不同的安全技术要素，以满足其各自业务应用的具体安全需求。

本标准第 4 章，应用软件系统基础安全技术要求，是对应用软件系统的每一个安全等级都适用的基础性安全技术要求的描述，包括：应用软件系统风险分析和安全需求，应用软件系统安全方案，应用软件系统环境安全，应用软件系统业务连续性，以及应用软件系统与相应信息系统安全等级划分等。

本标准第 5 章，应用软件系统安全技术分等级要求，以 GB 17859—1999 的五个安全等级的划分为基本依据，以 GB/T 20271-2006 关于信息系统通用安全技术要求的等级划分为基础，对每一个安全等级的应用软件系统的安全技术要求进行描述，包括：基础安全技术要求，安全功能技术要求，以及为实现上述安全技术要求应用软件系统安全子系统的自身保护、设计和实现及安全管理要求。其中，“**加粗宋体**”表示在较高等级中比上一级增加或增强的内容。

信息安全技术　应用软件系统安全等级保护通用技术指南

1　范围

本标准规定了按照 GB 17859-1999 的五个安全保护等级的划分对应用软件系统进行安全等级保护所涉及的通用技术要求。

本标准适用于按照 GB 17859-1999 的五个安全保护等级的划分对应用软件系统进行的安全等级保护的设计与实现。对于按照 GB 17859-1999 的五个安全保护等级的划分对应用软件系统进行的安全等级保护的测试、管理也可参照使用。

2　规范性引用文件

下列文件中的有关条款通过本标准的引用而成为本标准的条款。凡是注日期的引用文件，其随后所有的修改单（不包括勘误的内容）或修订版均不适用于本标准，然而，鼓励根据本标准达成协议的各方研究是否可使用这些文件的最新版本。凡是不注日期的引用文件，其最新版本适用于本标准。

GB 17859-1999　计算机信息系统安全保护等级划分准则
GB/T 20270-2006　信息安全技术　网络基础安全技术要求
GB/T 20271-2006　信息安全技术　信息系统通用安全技术要求
GB/T 20272-2006　信息安全技术　操作系统安全技术要求
GB/T 20273-2006　信息安全技术　数据库管理系统安全技术要求

3　术语、定义和缩略语

GB/T 20271—2006 确立的以及下列术语和定义适用于本标准。

3.1　术语和定义

3.1.1

应用软件系统　application software system
信息系统的重要组成部分，是指信息系统中对特定业务进行处理的软件系统。

3.1.2

应用软件系统安全技术　application software system security technology
为确保应用软件系统达到确定的安全性目标所采取的安全技术措施。

3.1.3

应用软件系统安全子系统（SSOASS）　security subsystem of application software system
应用软件系统中安全保护装置的总称。它建立了应用软件系统的一个基本安全保护环境，并提供安全应用软件系统要求的附加用户服务。按照 GB 17859-1999 对可信计算基（TCB）的定义，SSOASS 属于应用软件系统的 TCB。其中所需要的硬件和固件支持由低层的安全机制提供。

3.1.4

SSOASS 安全策略（SSP）　SSOASS security policy
对 SSOASS 中的资源进行管理、保护和分配的规则。一个 SSOASS 中可以有一种或多种安全策略。

3.1.5

安全功能策略（SFP）　security function policy
为实现 SSOASS 安全要素的功能所采用的安全策略。

3.1.6

安全要素　security element

本标准中各安全保护等级的安全技术要求所包含的安全内容的组成成份。

3.1.7

SSOASS 安全功能（SSF） SSOASS security function

正确实施 SSOASS 安全策略的全部硬件、固件、软件所提供的功能。每一种安全策略的实现，体现在 SSOASS 的某一个安全功能模块之中。一个 SSOASS 的所有安全功能模块共同组成该 SSOASS 的安全功能。

3.1.8

SSF 控制范围（SSC） SSF scope of control

SSOASS 的操作所涉及的主体和客体的范围。

3.1.9

用户公开数据 user published data

在应用软件系统中向所有用户公开的数据，该类数据的安全性受到破坏，将会对对公民、法人和其他组织的权益有一定影响，但不会危害国家安全、社会秩序、经济建设和公共利益。

3.1.10

用户一般数据 user general data

在应用软件系统中具有一般使用价值和保密程度，需要进行一定保护的单位内部的一般数据。该类数据的安全性受到破坏，将会对国家安全、社会秩序、经济建设和公共利益造成一定的损害。

3.1.11

用户重要数据 user important data

在应用软件系统中具有重要使用价值或保密程度，需要进行重点保护的单位的重要数据。该类数据的的安全性受到破坏，将会对国家安全、社会秩序、经济建设和公共利益造成较大损害。

3.1.12

用户关键数据 user chief data

在应用软件系统中具有很高使用价值或保密程度，需要进行特别保护的单位的关键数据。该类数据的的安全性受到破坏，将会对国家安全、社会秩序、经济建设和公共利益造成严重损害。

3.1.13

用户核心数据 user kernel data

在应用软件系统中具有最高使用价值或保密程度，需要进行绝对保护的单位的核心数据。该类数据的的安全性受到破坏，将会对国家安全、社会秩序、经济建设和公共利益造成特别严重损害。

3.2 缩略语

下列缩略语适用于本标准：

SSOASS 应用软件系统安全子系统 security subsystem of application software system

SSP SSOASS 安全策略 SSOASS security policy

SFP 安全功能策略 security function policy

SSF SSOASS 安全功能 SSOASS security function

SSC SSF 控制范围 SSF scope of control

4 应用软件系统基础安全技术要求

4.1 应用软件系统风险分析和安全需求

4.1.1 风险分析

应用软件系统的风险分析，包括系统设计前的风险分析和系统运行中的风险分析。风险分析应按照以下要求进行：

a) 从应用软件系统安全运行和信息安全保护出发，以应用软件系统相关的资产价值为基础,全面分析由于人为的和自然的原因对应用软件系统所造成的安全风险；
b) 通过对影响应用软件系统安全运行和信息安全保护的诸多因素的了解和分析,明确应用软件系统存在的风险，找出克服这些风险的办法；
c) 系统设计前和运行前应进行静态风险分析，以发现应用软件系统的潜在安全隐患；
d) 系统运行过程中应进行动态风险分析，收集、记录并分析系统运行中来自各种安全检测、监控机制的安全相关信息，以发现应用软件系统运行期的安全漏洞，并提供相应的系统脆弱性分析报告；
e) 采用基于模型的风险分析工具，通过收集数据、分析数据、输出数据，确定其面临威胁的严重性等级和可能性等，完成风险分析与评估，并确定相应的安全对策；
f) 按照不同用户数据的不同安全保护等级要求，在确定应用软件系统所存储、传输和处理的数据类型的基础上，进一步确定应用软件系统应具有的安全保护等级。

4.1.2 安全需求

应全面描述应用软件系统需要解决的全部安全问题。根据应用软件系统在信息系统中的地位和作用（参见附录 A），明确说明应用软件系统安全对整个信息系统安全的影响和作用。应根据风险分析所确定的风险情况，从以下方面考虑应用软件系统的安全需求：

a) 应用软件系统安全运行的环境条件需求；
b) 应用软件系统的整体安全需求；
c) 应用软件系统所实现的功能的安全需求；
d) 应用软件系统所需要的安全服务支持的强度/等级的需求；
e) 应用软件系统的其它安全需求。

4.2 应用软件系统安全方案

应用软件系统安全方案，是实现安全需求的具体方法的描述。安全方案的设计应满足以下要求：

a) 安全方案应对安全需求中的每一安全要求提供相应的安全策略和安全机制，并对其进行详细描述；
b) 安全方案应具有完备性，并以系统化方法进行安全设计,使所有安全策略和安全机制构成一个有机的整体；
c) 安全策略和安全机制的选择应充分考虑安全强度的一致性,尽量采用具有相同安全等级的安全技术；
d) 与密码支持相关的安全机制，无论是单独实现（如加密机）还是用综合的方法实现（如 CA 系统），都应具有相应一致的安全强度/等级；
e) 某一特定安全要求的安全策略和安全机制，可在应用软件系统中实现，也可在支持应用软件系统安全运行的信息系统的其它各组成部分（比如物理、操作系统层、网络层、数据库管理系统层等）的安全机制中实现,或者在系统各层分别实现。对此，安全方案应给出明确规定。

4.3 应用软件系统环境安全

应用软件系统安全是建立在其运行环境安全的基础之上的。按照关联/互补的原则，应用软件系统的安全要求可以在应用软件系统中实现，也可以在支持应用软件系统运行的低层环境中实现。应用软件系统的运行环境应满足以下安全要求：

a) 支持应用软件系统运行的网络系统应具有不低于应用软件系统的安全保护等级，具体要求见 GB/T 20270-2006 第 7 章；

b) 支持应用软件系统运行的操作系统应具有不低于应用软件系统的安全保护等级，具体要求见GB/T 20272-2006第4章；
c) 支持应用软件系统运行的数据库管理系统应具有不低于应用软件系统的安全保护等级，具体要求见GB/T 20273-2006第5章；
d) 支持应用软件系统运行的计算机和网络系统的物理安全应具有不低于应用软件系统的安全保护等级，具体要求见GB/T 20271-2006第6章。

4.4 应用软件系统业务连续性

应用软件系统的业务连续性要求是信息系统可用性要求的重要组成部分。业务连续性要求是一种相对独立的安全属性，需要通过对信息系统实行灾难备份与恢复来支持。实现不同的业务连续性要求应考虑以下因素：

a) 业务连续性要求通常以允许系统中断运行的时间间隔为依据进行等级划分；
b) 业务连续性要求还应包括中断后的恢复程度要求和安全系统更新前后的连续性要求；
c) 实现业务连续性的信息系统灾难备份与恢复是一种相对独立的信息系统安全机制；
d) 实现不同等级的业务连续性要求需要有相应等级的灾难备份与恢复系统支持；
e) 信息系统的安全保护等级与业务连续性等级具有相对的独立性；
f) 灾难备份与恢复系统需要对所处理的信息进行与该信息在相应信息系统中相同的安全保护。

4.5 应用软件系统及相应信息系统安全等级划分

应用软件系统的安全等级是相应信息系统安全等级划分的基本依据。一个应用软件系统可以是单一安全等级的系统，也可以是包含多个安全等级的系统。对于一个支持复杂业务的应用软件系统，根据其业务应用及数据信息所需要的不同的安全保护等级，应将应用软件系统及相应的信息系统划分为不同的安全域/子系统，实现不同等级的安全保护。应用软件系统及相应信息系统的安全等级分为单一安全等级和多安全等级，其安全等级的划分应满足以下要求：

a) 单一安全等级应用软件系统：按照业务应用确定的安全等级的要求，应用软件系统及相应信息系统划分为一个安全域，或者说整个信息系统具有相同的安全保护等级。在这种情况下，应用软件系统实现确定安全等级的安全功能要求和安全保证要求。同时，支持应用软件系统运行的计算机及网络的硬件系统、操作系统、数据库管理系统和网络软件等应提供相应安全等级的安全支持。
b) 多安全等级应用软件系统：按照不同业务应用的不同安全等级的要求，应用软件系统及相应信息系统划分为多个安全域/子系统，或者说，整个信息系统具有多个不同的安全保护等级。在这种情况下，各个安全域/子系统所属的应用软件系统实现其各自确定的安全功能要求和安全保证要求。同时，各个安全域/子系统中支持应用软件系统运行的计算机及网络的硬件系统、操作系统、数据库管理系统和网络软件系统等应提供相应安全等级的安全性支持。

5 应用软件系统安全技术分等级要求

5.1 第一级 用户自主保护级

5.1.1 基础安全技术要求

对第一级安全的应用软件系统应：

a) 按4.1的要求，进行风险分析并确定安全需求；
b) 按4.2的要求，设计应用软件系统安全方案；
c) 按4.3的要求，设计和实现应用软件系统的环境安全；
d) 按4.4的要求，确定应用软件系统的业务连续性要求及相应的灾难备份与恢复要求；
e) 按4.5的要求，划分应用软件系统及相应信息系统的安全等级。

5.1.2 安全功能技术要求

5.1.2.1 备份与故障恢复

应按 GB/T 20271-2006 中 6.1.2.4 的要求，设计和实现应用软件系统的备份与故障恢复功能。

5.1.2.2 用户身份鉴别

用户身份鉴别包括对用户的身份进行标识和鉴别。应按 GB/T 20271-2006 中 6.1.3.1 的要求，从以下方面设计和实现应用软件系统的身份鉴别功能：

a) 对应用软件系统的注册用户，按以下要求设计和实现标识功能：
   ——凡需进入应用软件系统的用户，应先进行标识（建立注册账号）；
   ——应用软件系统的用户标识一般使用用户名和用户标识符（UID）；
b) 对登录到应用软件系统的用户，应按以下要求进行身份的真实性鉴别：
   ——采用口令进行鉴别，并在每次用户登录系统时进行鉴别；
   ——口令应是不可见的，并在存储时有安全保护；
   ——通过对不成功的鉴别尝试的值（包括尝试次数和时间的阈值）进行预先定义，并明确规定达到该值时所应采取的动作等措施来实现鉴别失败的处理；
c) 对注册到应用软件系统的用户，应按以下要求设计和实现用户-主体绑定功能：
   ——将用户进程与所有者用户相关联，使用户进程的行为可以追溯到进程的所有者用户；
   ——将系统进程动态地与当前服务要求者用户相关联，使系统进程的行为可以追溯到当前服务要求者用户。

5.1.2.3 自主访问控制

应按 GB/T 20271-2006 中 6.1.3.2 的要求，从以下方面设计和实现应用软件系统的自主访问控制功能：

a) 命名用户以用户/用户组的身份规定并控制对客体的访问，并阻止非授权用户对客体的访问；
b) 提供用户按照确定的访问控制策略对自身创建的客体的访问进行控制的功能，包括：
   ——客体创建者有权以各种操作方式访问自身所创建的客体；
   ——客体创建者有权对其它用户进行“访问授权”，使其可对客体拥有者创建的指定客体能按授权的操作方式进行访问；
   ——客体创建者有权对其它用户进行“授权传播”，使其可以获得将该拥有者的指定客体的访问权限授予其它用户的权限；
   ——客体创建者有权收回其所授予其它用户的“访问授权”和“授权传播”；
   ——未经授权的用户不得以任何操作方式访问客体；
   ——授权用户不得以未授权的操作方式访问客体；
c) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.1.1.2 的要求，可实现文件级粒度的自主访问控制；
d) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.1.1.2 的要求，可实现对表级粒度的自主访问控制；
e) 在应用软件系统中，通过设置自主访问控制的安全机制，可实现文件级粒度的自主访问控制。

5.1.2.4 用户数据完整性保护

应按 GB/T 20271-2006 中 6.1.3.3 的要求，设计和实现用户公开数据的完整性保护功能。

5.1.3 SSOASS 自身保护要求

5.1.3.1 SSF 物理安全保护

应按 GB/T 20271-2006 中 6.1.4.1 的要求，设计和实现应用软件系统 SSF 的物理安全保护，通过对物理攻击的检测，发现以物理方式的攻击对 SSF 造成的威胁和破坏。

5.1.3.2 SSF 运行安全保护

应按 GB/T 20271-2006 中 6.1.4.2 的要求，从以下方面设计和实现应用软件系统 SSF 的运行安全保护：

a） 系统在设计时不应留有“后门”。即不应以维护、支持或操作需要为借口，设计有违反或绕过安全规则的任何类型的入口和文档中未说明的任何模式的入口；
b） 安全结构应是一个独立的、严格定义的系统软件的子集，并应防止外部干扰和破坏，如修改其代码或数据结构；
c） 提供设置和升级配置参数的机制。在初始化和对与安全有关的数据结构进行保护之前，应对用户和管理员的安全属性应进行定义；
d） 在 SSOASS 失败或中断后，应保护其以最小的损害得到恢复，并按照失败保护中所描述的内容，实现对 SSF 出现失败时的处理。

5.1.3.3 SSF 数据安全保护

应按 GB/T 20271-2006 中 6.1.4.3 的要求，对在 SSOASS 内传输的 SSF 数据，实现 SSOASS 内 SSF 数据传输的基本保护。

5.1.3.4 SSOASS 资源利用

应按 GB/T 20271-2006 中 6.1.4.4 的要求，从以下方面实现 SSOASS 的资源利用：

a） 通过一定措施确保当系统出现某些确定的故障时，SSF 也能维持正常运行；
b） 对主体使用 SSC 内某个资源子集，按有限服务优先级，进行 SSOASS 资源的管理和分配；
c） 按资源分配中最大限额的要求，进行 SSOASS 资源的管理和分配，确保用户和主体不会独占某种受控资源。

5.1.3.5 SSOASS 访问控制

应按 GB/T 20271-2006 中 6.1.4.5 的要求，从以下方面实现 SSOASS 的访问控制：

a） 按会话建立机制，对会话建立的管理进行设计；
b） 按可选属性范围限定的要求，从访问方法、访问地址和访问时间等方面，对用来建立会话的安全属性的范围进行限制；
c） 按多重并发会话限定中基本限定的要求，进行会话管理的设计。在基于基本标识的基础上，SSF 应限制系统的并发会话的最大次数，并就会话次数的限定数设置默认值。

5.1.4 SSOASS 设计和实现

5.1.4.1 配置管理

应按 GB/T 20271-2006 中 6.1.5.1 的要求，提供基本的配置管理能力，即要求开发者所使用的版本号与所表示的 SSOASS 样本完全对应。

5.1.4.2 分发和操作

应按 GB/T 20271-2006 中 6.1.5.2 的要求，从以下方面实现 SSOASS 的分发和操作：

a） 以文档形式提供对 SSOASS 安全地进行分发的过程，对安装、生成和启动并最终生成安全配置的过程进行说明。文档中所描述的内容应包括：
——分发的过程；
——安全启动和操作的过程；
b) 在交付过程中，应将系统的未授权修改风险控制到最低限度。包装及安全分送和安装过程中的安全性应由最终用户确认；
c) 所有软件应提供安全安装默认值，在客户不做选择时，使安全机制自动地发挥作用；
d) 随同系统交付的全部默认用户标识码，应在交付时处于非激活状态，并在使用前由管理员激活；
e) 用户文档应同交付的软件一起包装，并有相应的规程确保交付的软件是严格按照最新的版本制作的。

5.1.4.3 开发

应按 GB/T 20271-2006 中 6.1.5.3 的要求，从以下方面进行 SSOASS 的开发：

a) 按非形式化功能说明、描述性高层设计、SSF 子集实现、SSF 内部结构模块化、描述性低层设计和非形式化对应性说明的要求，进行 SSOASS 的设计；
b) 开发过程应保护数据的完整性，例如，检查数据更新的规则，多重输入的正确处理，返回状态的检查，中间结果的检查，合理值输入检查，事务处理更新的正确性检查等；
c) 通过对内部代码的检查，解决潜在的安全缺陷，关闭或取消所有的后门；
d) 对交付的软件和文档，应进行关于安全缺陷的定期的和书面的检查，并将检查结果告知用户；
e) 由系统控制的敏感数据，如口令、密钥等，不应在未受保护的程序或文档中以明文形式存储；
f) 应以书面形式提供给用户关于软件所有权法律保护的指南。

5.1.4.4 文档

应按 GB/T 20271-2006 中 6.1.5.4 的要求，从以下方面编制 SSOASS 的文档：

a) 用户文档应提供关于不同类型用户的可见的安全机制，并说明它们的用途和提供有关它们使用的指南；
b) 安全管理员文档应提供有关如何设置、维护和分析系统安全的详细说明，以及与安全有关的管理员功能的详细描述，包括增加和删除一个用户、改变主、客体的安全属性等；
c) 文档中不应提供任何一旦泄露将会危及本安全级范围内的系统安全的信息；
d) 有关安全的指令和文档根据权限应分别提供给用户、系统管理员和系统安全员；这些文档应为独立的文档，或作为独立的章条插入到管理员指南和用户指南中。

5.1.4.5 生存周期支持

应按 GB/T 20271-2006 中 6.1.5.5 的要求，从以下方面实现 SSOASS 的生存周期支持：

a) 按开发者定义生存周期模型进行 SSOASS 开发；
b) 文档应详细阐述安全启动和操作的过程，详细说明安全功能在启动、正常操作维护时是否能被撤消或修改，说明在故障或系统出错时如何恢复系统至安全状态。

5.1.4.6 测试

应按 GB/T 20271-2006 中 6.1.5.6 的要求，从以下方面对 SSOASS 进行测试：

a) 通过一般功能测试，相符性独立测试，确认 SSOASS 的功能与所要求功能的一致性；
b) 所有系统的安全特性，应被全面测试；
c) 所有发现的漏洞应被改正、消除或使其无效，并在消除漏洞后重新测试，以证实它们已被消除，且没有引出新的漏洞；
d) 应提供测试文档，详细描述测试计划、测试过程、测试结果。

5.1.5 SSOASS 安全管理

应根据本安全等级中安全功能技术要求所涉及的基础安全技术要求、安全功能技术要求和安全保证技术要求所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，按 GB/T 20271-2006 中 6.1.6 的要求，从以下方面实现 SSOASS 的安全管理：

a) 对安全保证措施所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，以及与一般的安装、配置等有关的功能，制定相应的操作、运行规程和行为规章制度；
b) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所实现的安全功能，实现 SSF 安全功能的管理。

5.2 第二级 系统审计保护级

5.2.1 基础安全技术要求

对**第二级安全**的应用软件系统应：

a) 按 4.1 的要求，进行风险分析并确定安全需求；
b) 按 4.2 的要求，设计应用软件系统安全方案；

c) 按 4.3 的要求，设计和实现应用软件系统的环境安全；
d) 按 4.4 的要求，确定应用软件系统的业务连续性要求及相应的灾难备份与恢复要求；
e) 按 4.5 的要求，划分应用软件系统及相应信息系统的安全等级。

5.2.2 安全功能技术要求

5.2.2.1 安全性检测分析

应按 GB/T 20271-2006 中 6.2.2.2 **的要求，检测分析应用软件系统的安全性，并结合计本级的的安全性要求加以改进。**

5.2.2.2 安全审计

应按 GB/T 20271-2006 中 6.2.2.3 的要求，从以下方面设计和实现应用软件系统的安全审计功能：

a) 安全审计功能的设计应与用户标识与鉴别、自主访问控制等安全功能的设计紧密结合；

b) 提供审计日志，潜在侵害分析，基本审计查阅和有限审计查阅，安全审计事件选择，以及受保护的审计踪迹存储等功能。

5.2.2.3 备份与故障恢复

应按 GB/T 20271-2006 中 6.2.2.5 的要求，设计和实现应用软件系统的备份与故障恢复功能。

5.2.2.4 用户身份鉴别

用户身份鉴别包括对用户的身份进行标识和鉴别。应按 GB/T 20271-2006 中 6.2.3.1 的要求，从以下方面设计和实现应用软件系统的用户身份鉴别功能：

a) 对应用软件系统的注册用户，按以下要求设计和实现标识功能：
——凡需进入应用软件系统的用户，应先进行标识（建立注册账号）；
——应用软件系统的用户标识一般使用用户名和用户标识符（UID），**并在应用软件系统的整个生存周期实现用户的唯一性标识，以及用户名或别名、UID 等之间的一致性；**

b) 对登录到应用软件系统的用户，应按以下要求进行身份的真实性鉴别：
——**采用强化管理的口令鉴别/基于令牌的动态口令鉴别/生物特征鉴别机制进行用户身份鉴别，**并在每次用户登录系统时进行鉴别；
——**鉴别信息应是不可见的，**并在存储**和传输**时进行安全保护；
——通过对不成功的鉴别尝试的值（包括尝试次数和时间的阈值）进行预先定义，并明确规定达到该值时所应采取的动作等措施来实现鉴别失败的处理；

c) 对注册到应用软件系统的用户，应按以下要求设计和实现用户-主体绑定功能：
——将用户进程与所有者用户相关联，使用户进程的行为可以追溯到进程的所有者用户；
——将系统进程动态地与当前服务要求者用户相关联，使系统进程的行为可以追溯到当前服务要求者用户。

5.2.2.5 自主访问控制

应按 GB/T 20271-2006 中 6.2.3.2 的要求，从以下方面设计和实现应用软件系统的自主访问控制功能：

a) 命名用户以**用户的身份**规定并控制对客体的访问，并阻止非授权用户对客体的访问；

b) 提供用户按照确定的访问控制策略对自身创建的客体的访问进行控制的功能，包括：
——客体创建者有权以各种操作方式访问自身所创建的客体；
——客体创建者有权对其它用户进行“访问授权”，使其可对客体拥有者创建的指定客体能按授权的操作方式进行访问；
——客体创建者有权对其它用户进行“授权传播”，使其可以获得将该拥有者的指定客体的访问权限授予其它用户的权限；
——客体创建者有权收回其所授予其它用户的“访问授权”和“授权传播”；
——未经授权的用户不得以任何操作方式访问客体；
——授权用户不得以未授权的操作方式访问客体；

c) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.2.1.2 的要求，可实现文件级粒度的自主访问控制；
d) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.2.1.2 的要求，可实现对**表级/记录、字段级**粒度的自主访问控制；
e) 在应用软件系统中，通过设置自主访问控制的安全机制，可实现文件级粒度的自主访问控制。

5.2.2.5 用户数据完整性保护

应按 GB/T 20271-2006 中 6.2.3.3 的要求，设计和实现**用户一般数据**的完整性保护功能。

5.2.2.6 用户数据保密性保护

应按 GB/T 20271-2006 中 6.2.3.4 的要求，设计和实现用户一般数据的保密性保护功能。

5.2.3 SSOASS 自身保护

5.2.3.1 SSF 物理安全保护

应按 **GB/T 20271-2006 中 6.2.4.1** 的要求，实现应用软件系统 SSF 的物理安全保护，通过对物理攻击的检测，发现以物理方式的攻击对 SSF 造成的威胁和破坏。

5.2.3.2 SSF 运行安全保护

应按 GB/T 20271-2006 中 6.2.4.2 的要求，从以下方面实现应用软件系统 SSF 的运行安全保护：
a) 系统在设计时不应留有“后门”。即不应以维护、支持或操作需要为借口，设计有违反或绕过安全规则的任何类型的入口和文档中未说明的任何模式的入口；
b) 安全结构应是一个独立的、严格定义的系统软件的子集，并应防止外部干扰和破坏，如修改其代码或数据结构；
c) 提供设置和升级配置参数的机制。在初始化和对与安全有关的数据结构进行保护之前，应对用户和管理员的安全属性应进行定义；
d) 当应用软件系统安装完成后，在普通用户访问之前，系统应配置好初始用户和管理员职责、审计参数、系统审计设置以及对客体的合适的访问控制；
e) 在 SSOASS 失败或中断后，应保护其以最小的损害得到恢复，并按照失败保护中所描述的内容，实现对 SSF 出现失败时的处理。

5.2.3.3 SSF 数据安全保护

应按 GB/T 20271-2006 中 6.2.4.3 的要求，对在 SSOASS 内传输的 SSF 数据进行以下安全保护：
a) 实现 SSOASS 内 SSF 数据传输的基本保护；
b) SSOASS 内 SSF 数据复制的一致性保护。

5.2.3.4 SSOASS 资源利用

应按 GB/T 20271-2006 中 6.2.4.4 的要求，从以下方面实现 SSOASS 的资源利用：
a) 通过一定措施确保当系统出现某些确定的故障时，SSF 也能维持正常运行；
b) 对 SSC 内某个资源子集，按有限服务优先级，进行 SSOASS 资源的管理和分配；
c) 按资源分配中最大限额的要求，进行 SSOASS 资源的管理和分配，确保用户和主体不会独占某种受控资源；
d) 确保在被授权的主体发出请求时，资源能被访问和利用；
e) 当系统资源的服务水平降低到预先规定的最小值时，应能检测和报警。

5.2.3.5 SSOASS 访问控制

应按 GB/T 20271-2006 中 6.2.4.5 的要求，从以下方面实现 SSOASS 的访问控制：
a) 按会话建立机制的要求，设计会话建立的管理；**在建立 SSOASS 会话之前，应鉴别用户的身份，并不允许鉴别机制本身被旁路；**
b) 按可选属性范围限定的要求，从访问方法、访问地址和访问时间等方面，对用来建立会话的

安全属性的范围进行限制；

c) 按多重并发会话限定中基本限定的要求，进行会话管理的设计；在基于基本标识的基础上，SSF 应限制系统的并发会话的最大次数，并就会话次数的限定数设置默认值。

d) 在用户成功登录系统后，SSOASS 应记录并向用户显示以下数据：

——日期、时间、来源和上次成功登录系统的情况；

——上次成功访问系统以来用户身份鉴别失败的情况；

——应显示口令到期的天数；

——成功或不成功的事件次数的显示可以用整数计数、时间戳列表等表述方法。

5.2.4 SSOASS 设计和实现

5.2.4.1 配置管理

应按 GB/T 20271-2006 中 6.2.5.1 的要求，从以下方面实现 SSOASS 的配置管理：

a) 在配置管理能力方面，实现对版本号、**配置项、授权控制**等方面的管理；

b) **配置管理范围方面，将 SSOASS 的实现表示、设计文档、测试文档、用户文档、管理员文档以及配置管理文档等置于配置管理之下；**

c) 在系统的整个生存期，即在它的开发、测试和运行维护期间，只有被授权的代码和代码修改才允许被加进已交付的源码的基本部分；所有改变应被记载和检查，以确保不危及系统的安全；通过技术、物理和规章方面的结合，充分保护生成系统所用到的源码免遭未授权的修改和毁坏；

d) 在软件配置管理系统中，应包含以下方面的工具：

——从源码产生出系统新版本；

——鉴定新生成的系统版本；

——保护源码免遭未授权修改。

5.2.4.2 分发和操作

应按 GB/T 20271-2006 中 6.2.5.2 的要求，从以下方面实现 SSOASS 的分发和操作：

a) 以文档形式提供对 SSOASS 安全地进行分发的过程，对安装、生成和启动并最终生成安全配置的过程进行说明。文档中所描述的内容应包括：

——提供分发的过程；

——安全启动和操作的过程；

——建立日志的过程；

b) 在交付过程中，应将系统的未授权修改风险控制到最低限度。包装及安全分送和安装过程中的安全性应由最终用户确认；

c) 所有软件应提供安全安装默认值，在客户不做选择时，使安全机制自动地发挥作用；

d) 随同系统交付的全部默认用户标识码，应在交付时处于非激活状态，并在使用前由管理员激活；

e) 用户文档应同交付的软件一起包装，并有相应的规程确保交付的软件是严格按照最新的版本制作的。

5.2.4.3 开发

应按 GB/T 20271-2006 中 6.2.5.3 的要求，从以下方面进行 SSOASS 的开发：

a) 按**非形式化安全策略模型、完全定义的外部接口、**描述性高层设计、SSF 子集实现、SSF 内部结构**层次化**、描述性低层设计、非形式化对应性说明的要求，进行 SSOASS 的设计；

b) 系统的设计和开发应保护数据的完整性，例如，检查数据更新的规则，多重输入的正确处理，返回状态的检查，中间结果的检查，合理值输入检查，事务处理更新的正确性检查等；

c) 在内部代码检查时，应解决潜在的安全缺陷，关闭或取消所有的后门；

d) 交付的软件和文档，应进行关于安全缺陷的定期的和书面的检查，并将检查结果告知用户；

e) 由系统控制的敏感数据，如口令、密钥等，不应在未受保护的程序或文档中以明文形式存储；

f) 应以书面形式提供给用户关于软件所有权法律保护的指南。

5.2.4.4 文档

应按 GB/T 20271-2006 中 6.2.5.4 的要求，从以下方面编制 SSOASS 的文档：

a) 用户文档应提供关于不同类型用户的可见的安全机制，并说明它们的用途和提供有关它们使用的指南；
b) 安全管理员文档应提供有关如何设置、维护和分析系统安全的详细说明，**包括当运行一个安全设备时，需要控制的有关功能和特权的警告，**以及与安全有关的管理员功能的详细描述，包括增加和删除一个用户、改变主、客体的安全属性等；
c) 文档中不应提供任何一旦泄露将会危及**本安全级范围内的**系统安全的信息；
d) 有关安全的指令和文档根据权限应分别提供给用户、系统管理员和系统安全员；这些文档应为独立的文档，或作为独立的章插入到管理员指南和用户指南中；
e) 提供关于所有审计工具的文档，包括为检查和保持审计文件所推荐的过程、针对每种审计事件的详细审计记录、为周期性备份和删除审计记录所推荐的过程等；
f) 提供如何进行系统自我评估（带有网络管理、口令要求、拨号访问控制、意外事故计划的安全报告）和为灾害恢复计划所做的建议，以及描述普通入侵技术、其它威胁及其检查和阻止的方法。

5.2.4.5 生存周期支持

应按 GB/T 20271-2006 中 6.2.5.5 的要求，从以下方面实现 SSOASS 的生存周期支持：

a) 按开发者定义生存周期模型**明确定义开发工具的要求**进行 SSOASS 开发，并提供**开发过程中的安全措施说明；**
b) 文档应详细阐述安全启动和操作的过程，详细说明安全功能在启动、正常操作维护时是否能被撤消或修改，说明在故障或系统出错时如何恢复系统至安全状态；
c) **如果系统含有加强安全性的硬件，那么管理员、终端用户或自动的诊断测试，应能在各自的操作环境中运行它并详细说明操作过程。**

5.2.4.6 测试

应按 GB/T 20271-2006 中 6.2.5.6 的要求，从以下方面对 SSOASS 进行测试：

a) 通过**范围证据和范围分析，高层设计的测试**，一般功能测试和相符独立性测试，确认 SSOASS 的功能与所要求功能的一致性；
b) 所有系统的安全特性，应被全面测试，**包括查找漏洞，如允许违反系统访问控制要求、允许违反资源访问控制要求、允许拒绝服务、允许验证数据进行未授权访问等；**
c) 所有发现的漏洞应被改正、消除或使其无效，并在消除漏洞后重新测试，以证实它们已被消除，且没有引出新的漏洞；
d) 应提供测试文档，详细描述测试计划、测试过程、测试结果。

5.2.4.7 脆弱性评定

应按 GB/T 20271-2006 中 6.2.5.7 的要求，从以下方面对 SSOASS 进行脆弱性评定：

a) 对防止误用的评定，通过对文档的检查，查找 SSOASS 以不安全的方式进行使用或配置而不为人们所察觉的情况；
b) 对 SSOASS 安全功能强度评估，通过对安全机制的安全行为的合格性或统计结果的分析，证明其达到或超过安全目标要求所定义的最低强度；
c) 开发者脆弱性分析，通过确定明显的安全脆弱性的存在，并确认在所期望的环境中所存在的脆弱性不会被利用。

5.2.5 SSOASS 安全管理

应根据本安全等级中安全功能技术要求所涉及的基础安全技术要求、安全功能技术要求和安全保证技术要求所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，按 GB/T 20271-2006 中 6.2.6 的要求，从以下方面实现 SSOASS 的安全管理：

a) 对安全保证措施所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，以及与一般的安

装、配置等有关的功能，制定相应的操作、运行规程和行为规章制度；

b) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所实现的安全功能，实现 SSF 安全功能的管理；

c) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全属性，从管理安全属性、安全的安全属性、静态属性初始化、安全属性终止和安全属性撤消等方面，实现 SSF 安全属性的安全管理；

d) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全数据，从管理 SSF 数据和 SSF 数据界限的管理等方面，实现 SSF 安全数据的安全管理。

5.3 第三级 安全标记保护级

5.3.1 基础安全技术要求

对**第三级安全**的应用软件系统应：

a) 按 4.1 的要求，进行风险分析并确定安全需求；

b) 按 4.2 的要求，设计应用软件系统安全方案；

c) 按 4.3 的要求，设计和实现应用软件系统的环境安全；

d) 按 4.4 的要求，确定应用软件系统的业务连续性要求及相应的灾难备份与恢复要求；

e) 按 4.5 的要求，划分应用软件系统及相应信息系统的安全等级。

5.3.2 安全功能技术要求

5.3.2.1 安全性检测分析

应按 GB/T 20271-2006 中 6.3.2.2 **的要求，**检测分析应用软件系统的安全性，并结合计本级的的安全性要求加以改进。

5.3.2.2 安全审计

应按 GB/T 20271-2006 中 6.3.2.4 的要求，从以下方面设计和实现应用软件系统的安全审计功能：

a) 安全审计功能的设计应与用户标识与鉴别、自主访问控制、**标记与强制访问控制**等安全功能的设计紧密结合；

b) 提供审计日志、**实时报警生成**，潜在侵害分析、**基于异常检测**，基本审计查阅、有限审计查阅和**可选审计查阅**，安全审计事件选择，以及受保护的审计踪迹存储和**审计数据的可用性确保**等功能；

c) **对与标识及强制访问控制等安全机制有关的内容，如敏感标记的操作等进行审计；**

d) **对网络环境下运行的应用软件系统，应建立统一管理和控制的安全审计机制**。

5.3.2.3 备份与故障恢复

应按 GB/T 20271-2006 中 6.3.2.6 的要求，设计和实现应用软件系统的备份与故障恢复功能。

5.3.2.4 用户身份鉴别

用户身份鉴别包括对用户的身份进行标识和鉴别。应按 GB/T 20271-2006 中 6.3.3.1 的要求，从以下方面设计和实现应用软件系统的用户身份鉴别功能：

a) 对应用软件系统的注册用户，按以下要求设计和实现标识功能：

——凡需进入应用软件系统的用户，应先进行标识（建立注册账号）；

——应用软件系统的用户标识一般使用用户名和用户标识符（UID），并在应用软件系统的整个生存周期实现用户的唯一性标识，以及用户名或别名、UID 等之间的一致性；

b) 对登录到应用软件系统的用户，应按以下要求进行身份的真实性鉴别：

——采用强化管理的口令鉴别/基于令牌的动态口令鉴别/生物特征鉴别/**数字证书鉴别**进行用户身份鉴别，并在每次用户登录系统时进行鉴别；

——鉴别信息应是不可见的，并**在存储和传输时应按 GB/T 20271-2006 中 6.3.3.9 的要求，用加密方法进行安全保护；**

——通过对不成功的鉴别尝试的值（包括尝试次数和时间的阈值）进行预先定义，并明确规

定达到该值时所应采取的动作等措施来实现鉴别失败的处理；

c) 对注册到应用软件系统的用户，应按以下要求设计和实现用户-主体绑定功能：

——将用户进程与所有者用户相关联，使用户进程的行为可以追溯到进程的所有者用户；

——将系统进程动态地与当前服务要求者用户相关联，使系统进程的行为可以追溯到当前服务要求者用户。

5.3.2.5 抗抵赖

a) 抗原发抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.3.3.2 a）的要求，通过提供选择性原发证据，实现抗原发抵赖功能；

b) 抗接收抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.3.3.2 b）的要求，通过提供选择性接收证据，实现抗接收抵赖功能。

5.3.2.6 自主访问控制

应按 GB/T 20271-2006 中 6.3.3.3 的要求，从以下方面设计和实现应用软件系统的自主访问控制功能：

a) 命名用户以用户的身份规定并控制对客体的访问，并阻止非授权用户对客体的访问；

b) 提供用户按照确定的访问控制策略对自身创建的客体的访问进行控制的功能，包括：

——客体创建者有权以各种操作方式访问自身所创建的客体；

——客体创建者有权对其它用户进行“访问授权”，使其可对客体拥有者创建的指定客体能按授权的操作方式进行访问；

——客体创建者有权对其它用户进行“授权传播”，使其可以获得将该拥有者的指定客体的访问权限授予其它用户的权限；

——客体创建者有权收回其所授予其它用户的“访问授权”和“授权传播”，**并对授权传播进行限制，对不可传播的授权进行明确定义，由系统自动检查并限制这些授权的传播**；

——未经授权的用户不得以任何操作方式访问客体；

——授权用户不得以未授权的操作方式访问客体；

c) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.3.1.2 的要求，可实现文件级粒度的自主访问控制；

d) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.3.1.2 的要求，可实现对表级/记录、字段级粒度的自主访问控制；

e) 在应用软件系统中，通过设置自主访问控制安全机制，可实现文件级粒度的自主访问控制。

5.3.2.7 标记

应按 GB/T 20271-2006 中 6.3.3.4 的要求，从以下方面设计和实现主、客体标记功能：

a) 用户的敏感标记，应在用户建立注册账户后由系统安全员通过 SSOASS 所提供的安全员界面操作进行标记；

b) 客体的敏感标记，应在数据输入到由 SSOASS 安全功能的控制范围内时，以默认方式生成或由安全员通过操作界面进行标记。

5.3.2.8 强制访问控制

应按 GB/T 20271-2006 中 6.3.3.5 的要求，从以下方面设计和实现应用软件系统的强制访问控制功能：

a) 按确定的强制访问控制安全策略，设计和实现相应的强制访问控制功能；

b) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.3.1.4 的要求，可实现文件级粒度的强制访问控制；

c) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.3.1.4 的要求，可实现表/记录、字段级粒度的强制访问控制；

d) 在应用软件系统中，在 PMI（授权管理基础设施）支持下，可实现文件级粒度的强制访问控制；

e) 将强制访问控制的范围应限定在所定义的主体与客体；
f) 将系统的常规管理、与安全有关的管理以及审计管理，分别由系统管理员、系统安全员和系统审计员来承担，按最小授权原则分别授予它们各自为完成自己所承担任务所需的最小权限，并在它们之间形成相互制约的关系。

5.3.2.9 用户数据完整性保护

应按 GB/T 20271-2006 中 6.3.3.7 的要求，设计和实现用户重要数据的完整性保护功能。

5.3.2.10 用户数据保密性保护

应按 GB/T 20271-2006 中 6.3.3.8 的要求，设计和实现用户重要数据的保密性保护功能。

5.3.3 SSOASS 自身保护

5.3.3.1 SSF 物理安全保护

应按 GB/T 20271-2006 中 6.3.4.1 的要求，实现应用软件系统 SSF 的物理安全保护，通过对物理攻击的检测和自动报告，及时发现以物理方式的攻击对 SSF 造成的威胁和破坏。

5.3.3.2 SSF 运行安全保护

应按 GB/T 20271-2006 中 6.3.4.2 的要求，从以下方面实现应用软件系统 SSF 的运行安全保护：

a) 系统在设计时不应留有“后门”。即不应以维护、支持或操作需要为借口，设计有违反或绕过安全规则的任何类型的入口和文档中未说明的任何模式的入口；
b) 安全结构应是一个独立的、严格定义的系统软件的一个子集，并应防止外部干扰和破坏，如修改其代码或数据结构；
c) 应提供设置和升级配置参数的安装机制，在初始化和对与安全有关的数据结构进行保护之前，应对用户和管理员的安全策略属性应进行定义；
d) 当数据库管理系统安装完成后，在普通用户访问之前，系统应配置好初始用户和管理员职责、审计参数、系统审计跟踪设置以及对客体的合适的访问控制；
e) 在 SSOASS 失败或中断后，应保护其以最小的损害得到恢复。并按照失败保护中所描述的内容，实现对 SSF 出现失败时的处理；
f) 系统应为数据库系统安全管理人员提供一种机制，来产生安全参数值的详细报告。

5.3.3.3 SSF 数据安全保护

应按 GB/T 20271-2006 中 6.3.4.3 的要求，对在 SSOASS 内传输的 SSF 数据，从以下方面实现安全保护：

a) 实现对输出 SSF 数据可用性、保密性、和完整性保护；
b) 实现 SSOASS 内 SSF 数据传输的基本保护、数据分离传输、数据完整性保护；
c) 实现 SSF 间的 SSF 数据的一致性和 SSOASS 内 SSF 数据复制的一致性保护。

5.3.3.4 SSOASS 资源利用

应按 GB/T 20271-2006 中 6.3.4.4 的要求，从以下方面实现 SSOASS 的资源利用：

a) 通过一定措施确保当系统出现某些确定的故障时，SSF 也能维持正常运行；
b) 对 SSC 内某个资源子集，按有限服务优先级，进行资源的管理和分配；
c) 按资源分配中最大限额的要求，进行 SSOASS 资源的管理和分配，确保用户和主体不会独占某种受控资源；
d) 确保在被授权的主体发出请求时，资源能被访问和利用；
e) 当系统资源的服务水平降低到预先规定的最小值时，应能检测和报警；
f) 系统应提供软件及数据备份和恢复的机制；
g) 系统应能提供命名的或用户可访问的系统资源的修改历史记录。

5.3.3.5 SSOASS 访问控制

应按 GB/T 20271-2006 中 6.3.4.5 的要求，从以下方面实现 SSOASS 的访问控制：

a) 按会话建立机制的要求，对会话建立的管理进行设计。在建立 SSOASS 会话之前，应鉴别用户的身份，不允许鉴别机制本身被旁路；
b) 按可选属性范围限定的要求，从访问方法、访问地址和访问时间等方面，对用来建立会话的安全属性的范围进行限制；
c) 按多重并发会话限定中基本限定的要求，进行会话管理的设计；在基于基本标识的基础上，SSF 应限制系统的并发会话的最大次数，并就会话次数的限定数设置默认值。
d) 在用户成功登录系统后，SSOASS 应记录并向用户显示以下数据：
——日期、时间、来源和上次成功登录系统的情况；
——上次成功访问系统以来用户身份鉴别失败的情况；
——应显示口令到期的天数；
——成功或不成功的事件次数的显示可以用整数计数、时间戳列表等表述方法；
e) 当用户鉴别过程不正确的次数达到系统规定的次数时，系统应退出登录过程并终止与用户的交互；
f) 系统应提供一种机制，能按时间、进入方式、地点、网络地址或端口等条件规定哪些用户能进入系统；
g) 在规定的未使用时限后，系统应断开会话或重新鉴别用户，系统应提供时限的默认值。

5.3.4 SSOASS 设计和实现

5.3.4.1 配置管理

应按 **GB/T 20271-2006 中 6.3.5.1** 的要求，从以下方面实现 SSOASS 的配置管理：
a) 在配置管理能力方面，实现对版本号、配置项、授权控制等方面的管理；
b) 在配置管理自动化方面，实现部分的配置管理自动化；
c) 配置管理范围方面，将 SSOASS 的实现表示、设计文档、测试文档、用户文档、管理员文档以及配置管理文档等置于配置管理之下，**实现对配置管理范围内安全缺陷问题的跟踪；**
d) 在系统的整个生存期，即在它的开发、测试和运行维护期间，应有一个软件配置管理系统处于保持对改变源码和文件的控制状态，确保只有被授权的代码和代码修改才允许被加进已交付的源码的基本部分；所有改变应被记载和检查，以确保不危及系统的安全；通过技术、物理和规章方面的结合，充分保护生成系统所用到的源码免遭未授权的修改和毁坏；
e) 在软件配置管理系统中，应包含以下方面的工具规程：
——从源码产生出系统新版本；
——鉴定新生成的系统版本；
——保护源码免遭未授权修改。

5.3.4.2 分发和操作

应按 **GB/T 20271-2006 中 6.3.5.2** 的要求，从以下方面实现 SSOASS 的分发和操作：
a) 以文档形式提供对 SSOASS 安全地进行分发的过程，并对**修改检测及**最终生成安全配置的过程进行说明。文档中所描述的内容应包括：
——分发过程、安全启动和操作过程、建立日志过程**及修改检测内容的说明；**
——对任何安全加强功能在启动、正常操作维护时能被撤消或修改的说明；
——在故障或硬件、软件出错后恢复系统至安全状态的规程说明；
——对含有加强安全性的硬件，说明用户或自动诊断测试的操作环境和使用方法；
——对所有加强安全性的硬件部件的诊断测试过程，提供例证的结果；
——在启动和操作时产生审计踪迹输出的例证；
b) 对系统的未授权修改的风险，应在交付时控制到最低限度。在包装及安全分送和安装过程中，这种控制应采取软件控制的方式，安全性由末端用户确认，所有安全机制都应以功能状态交付；
c) 所有软件应提供安全安装默认值，在客户不做选择时，默认值应使安全机制有效地发挥作用；
d) 随同系统交付的全部默认用户标识码，应在交付时处于非激活状态，并在使用前由管理员激活；
e) 用户文档应同交付的软件一起包装，并应有一套规程确保当前送给用户的软件是严格按照最新

的版本制作的；

f） **以安全方式开发并交付系统后，仍应提供对产品的长期维护和评估的支持，及时以书面形式向用户通告新的全问题；**

g） **对已知的可能出现的所有的安全问题，均应描述其特点，并被作为主要问题对待，直到它被解决或在用户同意下降级使用；**

h） **安全漏洞应及时修改；安全功能的增加和改进应独立于系统版本的升级；**

i） **新版本不应违反最初的安全策略和设想，应避免在维护、增加或功能升级中引入安全漏洞。所有功能的改变和安全结构设置的默认值都应在提交用户的文档中说明。**

5.3.4.3 开发

应按 **GB/T 20271-2006 中 6.3.5.3** 的要求，从以下方面进行 SSOASS 的开发：

a） 应按非形式化安全策略模型、非形式化功能说明、完全定义的外部接口、**安全加强的高层设计、SSF 完全实现、**SSF 内部结构层次化、描述性低层设计、非形式化对应性说明的要求，进行 SSOASS 的设计；

b） 系统的设计和开发应保护数据的完整性，例如，检查数据更新的规则，多重输入的正确处理，返回状态的检查，中间结果的检查，合理值输入检查，事务处理更新的正确性检查等；

c） 在内部代码检查时，应解决潜在的安全缺陷，关闭或取消所有的后门；

d） 交付的软件和文档，应进行关于安全缺陷的定期的和书面的检查，并将检查结果告知用户；

e） 系统控制数据，如口令、密钥，不应在未受保护的程序或文档中以明文形式存储，应以书面形式提供给用户关于软件所有权法律保护的指南；

f） **SSOASS 的开发过程应保持一个安全环境，该安全环境要求：**

——系统开发所使用的计算机系统的安全使用和维护情况的安全策略和措施应有书面记载，并可供检查；

——系统开发过程中使用的所有计算机系统应接受定期的和有书面记载的内部安全审查，描述审查过程的文件和真实的审查报告应可供检查；

——除授权的分发机构外，不应在开发环境外部复制或分发内部文档；

——系统开发环境的计算机系统使用的所有软件应当合法地从确定的渠道获得；

——系统开发者个人独自开发的软件，应在被开发管理者审核后才能用于开发的系统。

5.3.4.4 文档

应按 **GB/T 20271-2006 中 6.3.5.4** 的要求，从以下方面编制 SSOASS 的文档：

a） 用户文档应提供关于不同类型用户的可见的安全机制，并说明它们的用途和提供有关它们使用的指南；

b） 安全管理员文档应提供有关如何设置、维护和分析系统安全的详细说明，包括当运行一个安全设备时，需要控制的有关功能和特权的警告，以及与安全有关的管理员功能的详细描述，包括增加和删除一个用户、改变主、客体的安全属性等；

c） 文档中不应提供任何一旦泄露将会危及**本安全级范围内的**系统安全的信息；

d） 有关安全的指令和文档根据权限应分别提供给用户、系统管理员和系统安全员；这些文档应为独立的文档，或作为独立的章条插入到管理员指南和用户指南中；

e） 文档也可为硬拷贝、电子文档或联机文档，如果是联机文档应控制对其的访问；

f） 应提供关于所有审计工具的文档，包括为检查和保持审计文件所推荐的过程、针对每种审计事件的详细审计记录、为周期性备份和删除审计记录所推荐的过程等；

g） 提供如何进行系统自我评估（如：带有网络管理、口令要求、拨号访问控制、意外事故计划的安全报告）和为灾害恢复计划所做的建议，以及描述普通入侵技术、其它威胁及其检查和阻止的方法；

h） **安全管理员文档应提供安全管理员如何以安全的方式管理系统，除了给出一般的安全忠告，还要明确：**

——在系统用安全的方法安装时，围绕用户、用户账户、用户组成员关系、主体和客体的属性等，以及如何安装或终止安装；

——在系统的生存周期内，如何用安全的方法维护系统，包括为了防止系统被破坏而进行的每天、每周、每月的常规备份等；
——如何用安全的方法重建部分 SSOASS（如内核）的方法（如果允许在系统上重建 SSOASS）；
——说明安全审计机制，使授权用户可以有效地使用安全审计来检查安全策略；
——必要时，如何调整系统的安全默认配置。

5.3.4.5 生存周期支持

应按 **GB/T 20271-2006 中 6.3.5.5** 的要求，从以下方面实现 SSOASS 的生存周期支持：

a) **按标准的生存周期模型和**明确定义开发工具的要求**进行安全系统的开发，提供安全措施说明和基本的缺陷纠正；**
b) 文档应详细阐述安全启动和操作的过程，详细说明安全功能在启动、正常操作维护时是否能被撤消或修改，说明在故障或系统出错时如何恢复系统至安全状态；
c) 如果系统含有加强安全性的硬件，那么管理员、终端用户或自动的诊断测试，应能在各自的操作环境中运行它并详细说明操作过程。

5.3.4.6 测试

应按 **GB/T 20271-2006 中 6.3.5.6** 的要求，从以下方面对 SSOASS 进行测试：

a) 通过范围证据和范围分析，高层设计测试和**低层设计测试，顺序的功能测试，**相符独立性测试和**抽样独立性测试等**，确认 SSOASS 的功能与所要求的功能相一致；
b) 所有系统的安全特性，应被全面测试，包括查找漏洞，如允许违反系统访问控制要求、允许违反资源访问控制要求、允许拒绝服务、允许验证数据进行未授权访问等；
c) 所有发现的漏洞应被改正、消除或使其无效，并在消除漏洞后重新测试，以证实它们已被消除，且没有引出新的漏洞；
d) 应提供测试文档，详细描述测试计划、测试过程、测试结果。

5.3.4.7 脆弱性评定

应按 **GB/T 20271-2006 中 6.3.5.7** 的要求，从以下方面对 SSOASS 进行脆弱性评定：

a) 对防止误用的评定，通过对文档的检查和**分析确认**，查找 SSOASS 以不安全的方式进行使用或配置而不为人们所察觉的情况；
b) 对 SSOASS 安全功能强度评估，通过对安全机制的安全行为的合格性或统计结果的分析，证明其达到或超过安全目标要求所定义的最低强度；
c) 独立脆弱性分析，应通过独立穿透测试，确定 SSOASS 可以抵御的低攻击能力攻击者发起的攻击。

5.3.5 SSOASS 安全管理

应根据本安全等级中安全功能技术要求所涉及的基础安全技术要求、安全功能技术要求和安全保证技术要求所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，按 **GB/T 20271-2006 中 6.3.6** 的要求，从以下方面实现 SSOASS 的安全管理：

a) 对安全保证措施所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，以及与一般的安装、配置等有关的功能，制定相应的操作、运行规程和行为规章制度；
b) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所实现的安全功能，实现 SSF 安全功能的管理；
c) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全属性，从管理安全属性、安全的安全属性、静态属性初始化、安全属性终止和安全属性撤消等方面，实现 SSF 安全属性的安全管理；
d) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全数据，从管理 SSF 数据、SSF 数据界限的管理和**安全的 SSF 数据**等方面，实现 SSF 安全数据的安全管理；
e) **将应用软件系统管理员、安全员和审计员等重要安全角色分别设置专人担任，并按最小授权**

原则分别授予他们各自为完成自身任务所需的最小权限，并形成相互制约的关系；

f) **对网络环境运行的应用软件系统，实现 SSOASS 安全机制的集中管理。**

5.4 第四级 结构化保护级

5.4.1 基础安全技术要求

对**第四级安全**的应用软件系统应：

a) 按 4.1 的要求，进行风险分析并确定安全需求；
b) 按 4.2 的要求，设计应用软件系统安全方案；
c) 按 4.3 的要求，设计和实现应用软件系统的环境安全；
d) 按 4.4 的要求，确定应用软件系统的业务连续性要求及相应的灾难备份与恢复要求；
e) 按 4.5 的要求，划分应用软件系统及相应信息系统的安全等级。

5.4.2 安全功能技术要求

5.4.2.1 安全性检测分析

应按 GB/T 20271-2006 中 6.4.2.2 的要求，检测分析应用软件系统的安全性，并结合计本级的的安全性要求加以改进。

5.4.2.2 安全审计

应按 **GB/T 20271-2006 中 6.4.2.4** 的要求，从以下方面设计和实现应用软件系统的安全审计功能：

a) 安全审计功能的设计应与用户标识与鉴别、自主访问控制、标记与强制访问控制等安全功能的设计紧密结合；
b) 提供审计日志、实时报警生成和**违例进程终止**，潜在侵害分析、基于异常检测和**简单攻击探测**，基本审计查阅、有限审计查阅和可选审计查阅，安全审计事件选择，以及受保护的审计踪迹存储、审计数据的可用性确保和**防止审计数据丢失的措施**等功能；
c) 对与标识及强制访问控制等安全机制有关的内容，如敏感标记的操作等进行审计；
d) 对网络环境下运行的应用软件系统，应建立统一管理和控制的安全审计机制。

5.4.2.3 备份与故障恢复

应按 GB/T 20271-2006 中 6.4.2.6 的要求，设计和实现应用软件系统的备份与故障恢复功能。

5.4.2.4 用户身份鉴别

用户身份鉴别包括对用户的身份进行标识和鉴别。应按 **GB/T 20271-2006 中 6.4.3.1** 的要求，从以下方面设计和实现应用软件系统的用户身份鉴别功能：

a) 对应用软件系统的注册用户，按以下要求设计和实现标识功能：
——凡需进入应用软件系统的用户，应先进行标识（建立注册账号）；
——应用软件系统的用户标识一般使用用户名和用户标识符（UID），并在应用软件系统的整个生存周期实现用户的唯一性标识，以及用户名或别名、UID 等之间的一致性；
b) 对登录到应用软件系统的用户，应按以下要求进行身份的真实性鉴别：
——采用**强化管理的口令和/或基于令牌的动态口令和/或生物特征鉴别和/或数字证书等相结合的方式，采用多鉴别机制，**进行用户的身份鉴别，并在每次用户登录系统时和**重新连接时**进行鉴别；
——鉴别信息应是不可见的，并在存储和传输时应按 **GB/T 20271-2006 中 6.4.3.10** 的要求，用加密方法进行安全保护；
——通过对不成功的鉴别尝试的值（包括尝试次数和时间的阈值）进行预先定义，并明确规定达到该值时所应采取的动作等措施来实现鉴别失败的处理；
c) 对注册到应用软件系统的用户，应按以下要求设计和实现用户-主体绑定功能：
——将用户进程与所有者用户相关联，使用户进程的行为可以追溯到进程的所有者用户；
——将系统进程动态地与当前服务要求者用户相关联，使系统进程的行为可以追溯到当前服务要求者用户。

5.4.2.5 抗抵赖

a) 抗原发抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.4.3.2 a)的要求，通过提供**强制性原发证明**，实现抗原发抵赖功能；
b) 抗接收抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.4.3.2 b)的要求，通过提供**强制性接收证明**，实现抗接收抵赖功能。

5.4.2.6 自主访问控制

应按 GB/T 20271-2006 中 6.4.3.3 的要求，从以下方面设计和实现应用软件系统的自主访问控制功能：

a) 命名用户以用户的身份规定并控制对客体的访问，并阻止非授权用户对客体的访问；
b) 提供用户按照确定的访问控制策略对自身创建的客体的访问进行控制的功能，包括：
——客体创建者有权以各种操作方式访问自身所创建的客体；
——客体创建者有权对其它用户进行“访问授权”，使其可对客体拥有者创建的指定客体能按授权的操作方式进行访问；
——客体创建者有权对其它用户进行“授权传播”，使其可以获得将该拥有者的指定客体的访问权限授予其它用户的权限；
——客体创建者有权收回其所授予其它用户的“访问授权”和“授权传播”，并对授权传播进行限制，对不可传播的授权进行明确定义，由系统自动检查并限制这些授权的传播；
——未经授权的用户不得以任何操作方式访问客体；
——授权用户不得以未授权的操作方式访问客体；
c) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.4.1.2 的要求，可实现文件级粒度的自主访问控制；
d) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.4.1.2 的要求，可实现表级/记录、字段级粒度的自主访问控制；
e) 在应用软件系统中，通过设置自主访问控制安全机制，可实现文件级粒度的自主访问控制。

5.4.2.7 标记

应按 GB/T 20271-2006 中 6.4.3.4 的要求，从以下方面设计和实现主、客体标记功能：

a) 用户的敏感标记，应在用户建立注册账户后由系统安全员通过 SSOASS 所提供的安全员界面操作进行标记；
b) 客体的敏感标记，应在数据输入到由 SSOASS 安全功能的控制范围内时，以默认方式生成或由安全员通过操作界面进行标记；
c) **将标记扩展到应用软件系统中的所有主体与客体；对于从 SSOASS 控制范围外输入的未标记数据，应进行默认标记或由系统安全员进行标记；对于输出到 SSOASS 控制范围以外的数据，如打印输出的数据，应明显地标明该数据的安全标记。**

5.4.2.8 强制访问控制

应按 GB/T 20271-2006 中 6.4.3.5 的要求，从以下方面设计和实现应用软件系统的强制访问控制功能：

a) 按确定的强制访问控制安全策略，设计和实现相应的强制访问控制功能；
b) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.4.1.4 的要求，可实现文件级粒度的强制访问控制；
c) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，**按** GB/T 20273-2006 中 5.4.1.4 的要求，可实现表级/记录、字段级粒度的强制访问控制；
d) 在应用软件系统中，在 PMI（授权管理基础设施）的支持下，可实现文件级粒度的强制访问控制；
e) 将强制访问控制的范围应扩展到应用软件系统的所有主体与客体；
f) 将系统的常规管理、与安全有关的管理以及审计管理，分别由系统管理员、系统安全员和系统审计员来承担，按最小授权原则分别授予它们各自为完成自己所承担任务所需的最小权限，

并在它们之间形成相互制约的关系。

5.4.2.9 用户数据完整性保护

应按 GB/T 20271-2006 中 6.4.3.7 的要求，设计和实现用户关键数据的完整性保护功能。

5.4.2.10 用户数据保密性保护

应按 GB/T 20271-2006 中 6.4.3.8 的要求，设计和实现用户关键数据的保密性保护功能。

5.4.2.11 可信路径

对用户进行初始登录和鉴别或用户与 SSOASS 间进行数据传送，应按 GB/T 20271-2006 中 6.4.3.9 的要求，设计和实现应用软件系统的可信路径。

5.4.3 SSOASS 自身保护

5.4.3.1 SSF 物理安全保护

应按 **GB/T 20271-2006 中 6.4.4.1** 的要求，实现应用 SSF 的物理安全保护，通过对物理攻击的检测、自动报告和**抵抗，防止**以物理方式的攻击对 SSF 造成的威胁和破坏。

5.4.3.2 SSF 运行安全保护

应按 **GB/T 20271-2006 中 6.4.4.2** 的要求，从以下方面实现 SSF 的运行安全保护：

a) 系统在设计时不应留有“后门”。即不应以维护、支持或操作需要为借口，设计有违反或绕过安全规则的任何类型的入口和文档中未说明的任何模式的入口；
b) 安全结构应是一个独立的、严格定义的系统软件的一个子集，并应防止外部干扰和破坏，如修改其代码或数据结构；
c) 应提供设置和升级配置参数的安装机制，在初始化和对与安全有关的数据结构进行保护之前，应对用户和管理员的安全策略属性应进行定义；
d) 当应用软件系统安装完成后，在普通用户访问之前，系统应配置好初始用户和管理员职责、审计参数、系统审计跟踪设置以及对客体的合适的访问控制；
e) 在 SSOASS 失败或中断后，应保护其以最小的损害得到恢复。并按照失败保护中所描述的内容，实现对 SSF 出现失败时的处理；
f) 系统应为应用软件系统安全管理人员提供一种机制，来产生安全参数值的详细报告。

5.4.3.3 SSF 数据安全保护

应按 **GB/T 20271-2006 中 6.4.4.3** 的要求，对在 SSOASS 内传输的 SSF 数据，从以下方面进行安全保护：

a) 实现对输出 SSF 数据可用性、保密性、和完整性保护；
b) 实现 SSOASS 内 SSF 数据传输的基本传输保护、数据分离传输、数据完检测和**改正**等；
c) 实现 SSF 间的 SSF 数据的一致性和 SSOASS 内 SSF 数据复制的一致性保护；
d) **实现用户与 SSF 间的可信路径。**

5.4.3.4 SSOASS 资源利用

应按 **GB/T 20271-2006 中 6.4.4.4** 的要求，从以下方面实现 SSOASS 的资源利用：

a) 通过一定措施确保当系统出现某些确定的故障时，SSF 也能维持正常运行；
b) 对 SSC 内某个资源子集，按有限服务优先级，进行资源的管理和分配；
c) 按资源分配中最大限额的要求，进行 SSOASS 资源的管理和分配，确保用户和主体不会独占某种受控资源；
d) 确保在被授权的主体发出请求时，资源能被访问和利用；
e) 当系统资源的服务水平降低到预先规定的最小值时，应能检测和报警；
f) 系统应提供软件及数据备份和恢复的机制；
g) 系统应能提供命名的或用户可访问的系统资源的修改历史记录。

5.4.3.5 SSOASS 访问控制

应按 **GB/T 20271-2006 中 6.4.4.5** 的要求，从以下方面实现 SSOASS 的访问控制：

a) 按会话建立机制的要求，对会话建立的管理进行设计。在建立 SSOASS 会话之前，应鉴别用户的身份，不允许鉴别机制本身被旁路；

b) 按可选属性范围限定的要求，从访问方法、访问地址和访问时间等方面，对用来建立会话的安全属性的范围进行限制；

c) 按多重并发会话限定中基本限定的要求，进行会话管理的设计；在基于基本标识的基础上，SSF 应限制系统的并发会话的最大次数，并就会话次数的限定数设置默认值；

d) 在用户成功登录系统后，SSOASS 应记录并向用户显示以下数据：
——日期、时间、来源和上次成功登录系统的情况；
——上次成功访问系统以来用户身份鉴别失败的情况；
——应显示口令到期的天数；
——成功或不成功的事件次数的显示可以用整数计数、时间戳列表等表述方法；

e) 当用户鉴别过程不正确的次数达到系统规定的次数时，系统应退出登录过程并终止与用户的交互；

f) 系统应提供一种机制，能按时间、进入方式、地点、网络地址或端口等条件规定哪些用户能进入系统；

g) 在规定的未使用时限后，系统应断开会话或重新鉴别用户，系统应提供时限的默认值。

5.4.4 SSOASS 设计和实现

5.4.4.1 配置管理

应按 **GB/T 20271-2006 中 6.4.5.1** 的要求，从以下方面实现 SSOASS 的配置管理：

a) 在配置管理能力方面，实现**生成支持和验收过程的要求**；

b) 在配置管理自动化方面，实现部分的配置管理自动化；

c) 在配置管理范围方面，将 SSOASS 的实现表示、设计文档、测试文档、用户文档、管理员文档以及配置管理文档等置于配置管理之下，**实现对开发工具配置管理范围的管理；**

d) 在系统的整个生存期，即在它的开发、测试和运行维护期间，应有一个软件配置管理系统处于保持对改变源码和文件的控制状态，确保只有被授权的代码和代码修改才允许被加进已交付的源码的基本部分；所有改变应被记载和检查，以确保不危及系统的安全；通过技术、物理和规章方面的结合，充分保护生成系统所用到的源码免遭未授权的修改和毁坏；

e) 在软件配置管理系统中，应包含以下方面的工具规程：
——从源码产生出系统新版本；
——鉴定新生成的系统版本；
——保护源码免遭未授权修改。

5.4.4.2 分发和操作

应按 **GB/T 20271-2006 中 6.4.5.2** 的要求，从以下方面实现 SSOASS 的分发和操作：

a) 以文档形式提供对 SSOASS 安全地进行分发的过程，并对**防止修改及**最终生成安全配置的过程进行说明。文档中所描述的内容应包括：
——分发过程、安全启动和操作过程、建立日志过程及修改检测内容的说明；
——对任何安全加强功能在启动、正常操作维护时能被撤消或修改的说明；
——在故障或硬件、软件出错后恢复系统至安全状态的规程说明；
——对含有加强安全性的硬件，说明用户或自动诊断测试的操作环境和使用方法；
——对所有加强安全性的硬件部件的诊断测试过程，提供例证的结果；
——在启动和操作时产生审计踪迹输出的例证；

b) 对系统的未授权修改的风险，应在交付时控制到最低限度。在包装及安全分送和安装过程中，这种控制应采取软件控制的方式，安全性由末端用户确认，所有安全机制都应以功能状态交付；

c) 所有软件应提供安全安装默认值，在客户不做选择时，默认值应使安全机制有效地发挥作用；
d) 随同系统交付的全部默认用户标识码，应在交付时处于非激活状态，并在使用前由管理员激活；
e) 用户文档应同交付的软件一起包装，并应有一套规程确保当前送给用户的软件是严格按照最新的版本制作的；
f) 以安全方式开发并交付系统后，仍应提供对产品的长期维护和评估的支持，及时以书面形式向用户通告新的全问题；
g) 对已知的可能出现的所有的安全问题，均应描述其特点，并被作为主要问题对待，直到它被解决或在用户同意下降级使用；
h) 安全漏洞应及时修改；安全功能的增加和改进应独立于系统版本的升级；
i) 新版本不应违反最初的安全策略和设想，应避免在维护、增加或功能升级中引入安全漏洞。所有功能的改变和安全结构设置的默认值都应在提交用户的文档中说明。

5.4.4.3 开发

应按 **GB/T 20271-2006 中 6.4.5.3** 的要求，从以下方面进行 SSOASS 的开发：

a) 按**半形式化的 SSOASS 安全策略模型、半形式化功能说明、半形式化高层设计、SSF 的结构化实现、SSF 内部结构复杂度最小化、半形式化低层设计、半形式化对应性说明**的要求，进行 SSOASS 的设计；
b) 系统的设计和开发应保护数据的完整性，例如，检查数据更新的规则，多重输入的正确处理，返回状态的检查，中间结果的检查，合理值输入检查，事务处理更新的正确性检查等；
c) 在内部代码检查时，应解决潜在的安全缺陷，关闭或取消所有的后门；
d) 交付的软件和文档，应进行关于安全缺陷的定期的和书面的检查，并将检查结果告知用户；
e) 系统控制数据，如口令、密钥，不应在未受保护的程序或文档中以明文形式存储，应以书面形式提供给用户关于软件所有权法律保护的指南；
f) SSOASS 的开发过程应保持一个安全环境，该安全环境要求：
——系统开发所使用的计算机系统的安全使用和维护情况的安全策略和措施应有书面记载，并可供检查；
——系统开发过程中使用的所有计算机系统应接受定期的和有书面记载的内部安全审查，描述审查过程的文件和真实的审查报告应可供检查；
——除授权的分发机构外，不应在开发环境外部复制或分发内部文档；
——系统开发环境的计算机系统使用的所有软件应当合法地从确定的渠道获得；
——系统开发者个人独自开发的软件，应在被开发管理者审核后才能用于开发的系统。

5.4.4.4 文档

应按 **GB/T 20271-2006 中 6.4.5.4** 的要求，从以下方面编制 SSOASS 的文档：

a) 用户文档应提供关于不同类型用户的可见的安全机制，并说明它们的用途和提供有关它们使用的指南；
b) 安全管理员文档应提供有关如何设置、维护和分析系统安全的详细说明，包括当运行一个安全设备时，需要控制的有关功能和特权的警告，以及与安全有关的管理员功能的详细描述，包括增加和删除一个用户、改变主、客体的安全属性等；
c) 文档中不应提供任何一旦泄露将会危及**本安全级范围内的**系统安全的信息；
d) 有关安全的指令和文档根据权限应分别提供给用户、系统管理员和系统安全员；这些文档应为独立的文档，或作为独立的章条插入到管理员指南和用户指南中；
e) 文档也可为硬拷贝、电子文档或联机文档，如果是联机文档应控制对其的访问；
f) 应提供关于所有审计工具的文档，包括为检查和保持审计文件所推荐的过程、针对每种审计事件的详细审计记录、为周期性备份和删除审计记录所推荐的过程等；
g) 提供如何进行系统自我评估（如：带有网络管理、口令要求、拨号访问控制、意外事故计划的安全报告）和为灾害恢复计划所做的建议，以及描述普通入侵技术、其它威胁及其检查和阻止的方法；

h) 安全管理员文档应提供安全管理员如何以安全的方式管理系统，除了给出一般的安全忠告，还要明确：
——在系统用安全的方法安装时，围绕用户、用户账户、用户组成员关系、主体和客体的属性等，以及如何安装或终止安装；
——在系统的生存周期内，如何用安全的方法维护系统，包括为了防止系统被破坏而进行的每天、每周、每月的常规备份等；
——如何用安全的方法重建部分 SSOASS（如内核）的方法（如果允许在系统上重建 SSOASS）；
——说明安全审计机制，使授权用户可以有效地使用安全审计来检查安全策略；
——必要时，如何调整系统的安全默认配置。

5.4.4.5 生存周期支持

应按 **GB/T 20271-2006 中 6.4.5.5** 的要求，从以下方面实现 SSOASS 的生存周期支持：

a) 按标准的生存周期模型和**遵照实现标准-应用部分的工具和技术的要求**进行开发，并**提供充分的安全措施和缺陷报告；**
b) 文档应详细阐述安全启动和操作的过程，详细说明安全功能在启动、正常操作维护时是否能被撤消或修改，说明在故障或系统出错时如何恢复系统至安全状态；
c) 如果系统含有加强安全性的硬件，那么管理员、终端用户或自动的诊断测试，应能在各自的操作环境中运行它并详细说明操作过程。

5.4.4.6 测试

应按 **GB/T 20271-2006 中 6.4.5.6** 的要求，从以下方面对 SSOASS 进行测试：

a) 通过范围证据和**严格的范围分析，**高层设计测试、低层设计测试**和实现表示测试，**顺序的功能测试，相符独立性测试和抽样独立性测试等，确认 SSOASS 的功能与所要求的功能相一致；
b) 所有系统的安全特性，应被全面测试，包括查找漏洞，如允许违反系统访问控制要求、允许违反资源访问控制要求、允许拒绝服务、允许验证数据进行未授权访问等；
c) 所有发现的漏洞应被改正、消除或使其无效，并在消除漏洞后重新测试，以证实它们已被消除，且没有引出新的漏洞；
d) 应提供测试文档，详细描述测试计划、测试过程、测试结果。

5.4.4.7 脆弱性评定

应按 **GB/T 20271-2006 中 6.4.5.7** 的要求，从以下方面对所开发的 SSOASS 进行脆弱性评定：

a) **通过一般性的隐蔽信道分析，对隐蔽存储信道进行搜索，标识出可识别的隐蔽存储信道；**
b) 对防止误用的评定，通过对文档的检查和分析确认，查找 SSOASS 以不安全的方式进行使用或配置而不为人们所察觉的情况；
c) 对 SSOASS 安全功能强度评估，通过对安全机制的安全行为的合格性或统计结果的分析，证明其达到或超过安全目标要求所定义的最低强度；
d) **中抵抗力分析，**通过独立穿透测试和对**脆弱性的系统化搜索**，确定 SSOASS 可以抵御**中攻击能力攻击者发起的穿透性攻击**。

5.4.5 SSOASS 安全管理

应根据本安全等级中安全功能技术要求所涉及的基础安全技术要求、安全功能技术要求和安全保证技术要求所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，按 **GB/T 20271-2006 中 6.4.6** 的要求，从以下方面实现 SSOASS 的安全管理：

a) 对安全保证措施所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，以及与一般的安装、配置等有关的功能，制定相应的操作、运行规程和行为规章制度；
b) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所实现的安全功能，实现 SSF 安全功能的管理；
c) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全属性，从管理安全属性、安全的安全属性、静态属性初始化、安全属性终止和安全属性撤消等方面，

实现 SSF 安全属性的安全管理；

d) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全数据，从管理 SSF 数据、SSF 数据界限的管理和安全的 SSF 数据等方面，实现 SSF 安全数据的安全管理；

e) 将应用软件系统管理员、安全员和审计员等重要安全角色分别设置专人担任，并按最小授权原则分别授予他们各自为完成自身任务所需的最小权限，并形成相互制约的关系；

f) 对网络环境运行的应用软件系统，实现 SSOASS 安全机制的集中管理。

5.5 第五级 访问验证保护级

5.5.1 基础安全技术要求

对**第五级安全**的应用软件系统应：

a) 按 4.1 的要求，进行风险分析并确定安全需求；

b) 按 4.2 的要求，设计应用软件系统安全方案；

c) 按 4.3 的要求，设计和实现应用软件系统的环境安全；

d) 按 4.4 的要求，确定应用软件系统的业务连续性要求及相应的灾难备份与恢复要求；

e) 按 4.5 的要求，划分应用软件系统及相应信息系统的安全等级。

5.5.2 安全功能技术要求

5.5.2.1 安全性检测分析

应按 GB/T 20271-2006 中 6.5.2.2 **的要求，**检测分析应用软件系统的安全性，并结合计本级的的安全性要求加以改进。

5.5.2.2 安全审计

应按 **GB/T 20271-2006 中 6.5.2.4** 的要求，从以下方面设计和实现应用软件系统的安全审计功能：

a) 安全审计功能的设计应与用户标识与鉴别、自主访问控制、标记与强制访问控制等安全功能的设计紧密结合；

b) 提供审计日志、实时报警生成、违例进程终止、**服务取消和用户帐号断开与失效**，潜在侵害分析、基于异常检测和**复杂攻击探测**，基本审计查阅、有限审计查阅和可选审计查阅，安全审计事件选择，以及受保护的审计踪迹存储、审计数据的可用性确保和防止审计数据丢失的措施等功能；

c) 对与标识及强制访问控制等安全机制有关的内容，如敏感标记的操作等进行审计；

d) 对网络环境下运行的应用软件系统，应建立统一管理和控制的安全审计机制。

5.5.2.3 备份与故障恢复

应按 GB/T 20271-2006 中 6.5.2.6 的要求，设计和实现应用软件系统的备份与故障恢复功能。

5.5.2.4 用户身份鉴别

用户身份鉴别包括对用户的身份进行标识和鉴别。应按 **GB/T 20271-2006 中 6.5.3.1** 的要求，从以下方面设计和实现应用软件系统的用户身份鉴别功能：

a) 对应用软件系统的注册用户，按以下要求设计和实现标识功能：

——凡需进入应用软件系统的用户，应先进行标识（建立注册账号）；

——应用软件系统的用户标识一般使用用户名和用户标识符（UID），并在应用软件系统的整个生存周期实现用户的唯一性标识，以及用户名或别名、UID 等之间的一致性；

b) 对登录到应用软件系统的用户，应按以下要求进行身份的真实性鉴别：

——采用强化管理的口令和/或基于令牌的动态口令和/或生物特征鉴别和/或数字证书和/或**以协议形式化分析为基础的鉴别**等相结合的方式，采用多鉴别机制，进行用户的身份鉴别，并在每次用户登录系统时和重新连接时进行鉴别；

——鉴别信息应是不可见的，并在存储和传输时应按 **GB/T 20271-2006 中 6.5.3.10** 的要求，用加密方法进行安全保护；

——通过对不成功的鉴别尝试的值（包括尝试次数和时间的阈值）进行预先定义，并明确规定达到该值时所应采取的动作等措施来实现鉴别失败的处理；

c) 对注册到应用软件系统的用户，应按以下要求设计和实现用户-主体绑定功能：
——将用户进程与所有者用户相关联，使用户进程的行为可以追溯到进程的所有者用户；
——将系统进程动态地与当前服务要求者用户相关联，使系统进程的行为可以追溯到当前服务要求者用户。

5.5.2.5 抗抵赖

a) 抗原发抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.5.3.2 a)的要求，通过提供强制性原发证明，设计和实现抗原发抵赖功能；
b) 抗接收抵赖：对于在网络环境进行数据交换的情况，应按 GB/T 20271-2006 中 6.5.3.2 b)的要求，通过提供强制性接收证明，设计和实现抗接收抵赖功能。

5.5.2.6 自主访问控制

应按 GB/T 20271-2006 中 6.5.3.3 的要求，从以下方面设计和实现应用软件系统的自主访问控制功能：

a) 命名用户以用户的身份规定并控制对客体的访问，并阻止非授权用户对客体的访问；
b) 提供用户按照确定的访问控制策略对自身创建的客体的访问进行控制的功能，包括：
——客体创建者有权以各种操作方式访问自身所创建的客体；
——客体创建者有权对其它用户进行“访问授权”，使其可对客体拥有者创建的指定客体能按授权的操作方式进行访问；
——客体创建者有权对其它用户进行“授权传播”，使其可以获得将该拥有者的指定客体的访问权限授予其它用户的权限；
——客体创建者有权收回其所授予其它用户的“访问授权”和“授权传播”，并对授权传播进行限制，对不可传播的授权进行明确定义，由系统自动检查并限制这些授权的传播；
——未经授权的用户不得以任何操作方式访问客体；
——授权用户不得以未授权的操作方式访问客体；
c) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.5.1.2 的要求，可实现文件级粒度的自主访问控制；
d) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.5.1.2 的要求，可实现表级/记录、字段/元素级粒度的自主访问控制；
e) 在应用软件系统中，通过设置自主访问控制安全机制，可实现文件级粒度的自主访问控制。

5.5.2.7 标记

应按 GB/T 20271-2006 中 6.5.3.4 的要求，从以下方面设计和实现主、客体标记功能：

a) 用户的敏感标记，应在用户建立注册账户后由系统安全员通过 SSOASS 所提供的安全员界面操作进行标记；
b) 客体的敏感标记，应在数据输入到由 SSOASS 安全功能的控制范围内时，以默认方式生成或由安全员通过操作界面进行标记；
c) 将标记扩展到信息系统中的所有主体与客体；对于从 SSOASS 控制范围外输入的未标记数据，应进行默认标记或由系统安全员进行标记；对于输出到 SSOASS 控制范围以外的数据，如打印输出的数据，应明显地标明该数据的安全标记。

5.5.2.8 强制访问控制

应按 GB/T 20271-2006 中 6.5.3.5 的要求，从以下方面设计和实现应用软件系统的强制访问控制功能：

a) 按确定的强制访问控制安全策略，设计和实现相应的强制访问控制功能；
b) 以文件形式存储和操作的用户数据，在操作系统的支持下，按 GB/T 20272-2006 中 4.5.1.4 的要求，可实现文件级粒度的强制访问控制；

c) 以数据库形式存储和操作的用户数据，在数据库管理系统的支持下，按 GB/T 20273-2006 中 5.5.1.4 的要求，可实现表级/记录、字段/元素级粒度的强制访问控制；
d) 在应用软件系统中，在 PMI（授权管理基础设施）支持下，可实现文件级粒度的强制访问控制；
e) 将强制访问控制的范围应扩展到应用软件系统的所有主体与客体；
f) 将系统的常规管理、与安全有关的管理以及审计管理，分别由系统管理员、系统安全员和系统审计员来承担，按最小授权原则分别授予它们各自为完成自己所承担任务所需的最小权限，并在它们之间形成相互制约的关系。

5.5.2.9 用户数据完整性保护

应按 GB/T 20271-2006 中 6.5.3.7 的要求，设计和实现用户核心数据的完整性保护功能。

5.5.2.10 用户数据保密性保护

应按 GB/T 20271-2006 中 6.5.3.8 的要求，设计和实现用户核心数据的保密性保护功能。

5.5.2.11 可信路径

对用户进行初始登录和鉴别或用户与 SSOASS 间进行数据传送，应按 **GB/T 20271-2006 中 6.5.3.9** 的要求，设计和实现应用软件系统的可信路径。

5.5.3 SSOASS 自身保护

5.5.3.1 SSF 物理安全保护

应按 **GB/T 20271-2006 中 6.5.4.1** 的要求，实现 SSF 的物理安全保护，通过对物理攻击的检测、自动报告和抵抗，防止以物理方式的攻击对 SSF 造成的威胁和破坏。

5.5.3.2 SSF 运行安全保护

应按 **GB/T 20271-2006 中 6.5.4.2** 的要求，从以下方面实现 SSF 的运行安全保护：
a) 系统在设计时不应留有“后门”。即不应以维护、支持或操作需要为借口，设计有违反或绕过安全规则的任何类型的入口和文档中未说明的任何模式的入口；
b) 安全结构应是一个独立的、严格定义的系统软件的一个子集，并应防止外部干扰和破坏，如修改其代码或数据结构；
c) 应提供设置和升级配置参数的安装机制，在初始化和对与安全有关的数据结构进行保护之前，应对用户和管理员的安全策略属性应进行定义；
d) 当应用软件系统安装完成后，在普通用户访问之前，系统应配置好初始用户和管理员职责、审计参数、系统审计跟踪设置以及对客体的合适的访问控制；
e) 在 SSOASS 失败或中断后，应保护其以最小的损害得到恢复。并按照失败保护中所描述的内容，实现对 SSF 出现失败时的处理；
f) **应在确定不减弱保护的情况下启动 SSOASS，并在 SSF 运行中断后能在不减弱 SSP 保护的情况下以手动或自动方式恢复运行；**
g) 系统应为应用软件系统安全管理人员提供一种机制，来产生安全参数值的详细报告。

5.5.3.3 SSF 数据安全保护

应按 **GB/T 20271-2006 中 6.5.4.3** 的要求，对在 SSOASS 内传输的 SSF 数据，从以下方面进行安全保护：
a) 实现对输出 SSF 数据可用性、保密性、和完整性保护；
b) 实现 SSOASS 内 SSF 数据传输的基本传输保护、数据分离传输、数据完检测和改正等；
c) 实现 SSF 间的 SSF 数据的一致性和 SSOASS 内 SSF 数据复制的一致性保护；
d) 实现用户与 SSF 间及 SSF 间的可信路径。

5.5.3.4 SSOASS 资源利用

应按 **GB/T 20271-2006 中 6.5.4.4** 的要求，从以下方面实现 SSOASS 的资源利用：

a) 通过一定措施确保当系统出现某些确定的故障时，SSF 也能维持正常运行；
b) 对 SSC 内某个资源子集，按有限服务优先级，进行资源的管理和分配；
c) 按资源分配中最大限额的要求，进行 SSOASS 资源的管理和分配，确保用户和主体不会独占某种受控资源；
d) 确保在被授权的主体发出请求时，资源能被访问和利用；
e) 当系统资源的服务水平降低到预先规定的最小值时，应能检测和报警；
f) 系统应提供软件及数据备份和恢复的机制；
g) 系统应能提供命名的或用户可访问的系统资源的修改历史记录。

5.5.3.5 SSOASS 访问控制

应按 **GB/T 20271-2006 中 6.5.4.5** 的要求，从以下方面实现 SSOASS 的访问控制：

a) 按会话建立机制的要求，对会话建立的管理进行设计。在建立 SSOASS 会话之前，应鉴别用户的身份，不允许鉴别机制本身被旁路；
b) 按可选属性范围限定的要求，从访问方法、访问地址和访问时间等方面，对用来建立会话的安全属性的范围进行限制；
c) 按多重并发会话限定中基本限定的要求，进行会话管理的设计；在基于基本标识的基础上，SSF 应限制系统的并发会话的最大次数，并就会话次数的限定数设置默认值；
d) 在用户成功登录系统后，SSOASS 应记录并向用户显示以下数据：
——日期、时间、来源和上次成功登录系统的情况；
——上次成功访问系统以来用户身份鉴别失败的情况；
——应显示口令到期的天数；
——成功或不成功的事件次数的显示可以用整数计数、时间戳列表等表述方法；
e) 当用户鉴别过程不正确的次数达到系统规定的次数时，系统应退出登录过程并终止与用户的交互；
f) 系统应提供一种机制，能按时间、进入方式、地点、网络地址或端口等条件规定哪些用户能进入系统；
g) 在规定的未使用时限后，系统应断开会话或重新鉴别用户，系统应提供时限的默认值。

5.5.4 SSOASS 设计和实现

5.5.4.1 配置管理

应按 **GB/T 20271-2006 中 6.5.5.1** 的要求，从以下方面实现 SSOASS 的配置管理：

a) 在配置管理能力方面，实现生成支持和验收过程及**进一步支持**的要求；
b) 在配置管理自动化方面，实现**完全的配置管理自动化**；
c) 在配置管理范围方面，将 SSOASS 的实现表示、设计文档、测试文档、用户文档、管理员文档以及配置管理文档等置于配置管理之下，实现对开发工具配置管理范围的管理；
d) 在系统的整个生存期，即在它的开发、测试和运行维护期间，应有一个软件配置管理系统处于保持对改变源码和文件的控制状态，确保只有被授权的代码和代码修改才允许被加进已交付的源码的基本部分；所有改变应被记载和检查，以确保不危及系统的安全；通过技术、物理和规章方面的结合，充分保护生成系统所用到的源码免遭未授权的修改和毁坏；
e) 在软件配置管理系统中，应包含以下方面的工具规程：
——从源码产生出系统新版本；
——鉴定新生成的系统版本；
——保护源码免遭未授权修改。

5.5.4.2 分发和操作

应按 **GB/T 20271-2006 中 6.5.5.2** 的要求，从以下方面实现 SSOASS 的分发和操作：

a) 以文档形式提供对 SSOASS 安全地进行分发的过程，并对防止修改及最终生成安全配置的过程进行说明。文档中所描述的内容应包括：
——分发过程、安全启动和操作过程、建立日志过程及修改检测内容的说明；

——对任何安全加强功能在启动、正常操作维护时能被撤消或修改的说明；
——在故障或硬件、软件出错后恢复系统至安全状态的规程说明；
——对含有加强安全性的硬件，说明用户或自动诊断测试的操作环境和使用方法；
——对所有加强安全性的硬件部件的诊断测试过程，提供例证的结果；
——在启动和操作时产生审计踪迹输出的例证；

b) 对系统的未授权修改的风险，应在交付时控制到最低限度。在包装及安全分送和安装过程中，这种控制应采取软件控制的方式，安全性由末端用户确认，所有安全机制都应以功能状态交付；
c) 所有软件应提供安全安装默认值，在客户不做选择时，默认值应使安全机制有效地发挥作用；
d) 随同系统交付的全部默认用户标识码，应在交付时处于非激活状态，并在使用前由管理员激活；
e) 用户文档应同交付的软件一起包装，并应有一套规程确保当前送给用户的软件是严格按照最新的版本制作的；
f) 以安全方式开发并交付系统后，仍应提供对产品的长期维护和评估的支持，及时以书面形式向用户通告新的全问题；
g) 对已知的可能出现的所有的安全问题，均应描述其特点，并被作为主要问题对待，直到它被解决或在用户同意下降级使用；
h) 安全漏洞应及时修改；安全功能的增加和改进应独立于系统版本的升级；
i) 新版本不应违反最初的安全策略和设想，应避免在维护、增加或功能升级中引入安全漏洞。所有功能的改变和安全结构设置的默认值都应在提交用户的文档中说明。

5.5.4.3 开发

应按 **GB/T 20271-2006 中 6.5.5.3** 的要求，从以下方面进行 SSOASS 的开发：

a) 按**形式化的 SSOASS 安全策略模型、形式化功能说明、形式化高层设计、**SSF 的结构化实现、SSF 内部结构复杂度最小化、**形式化低层设计、形式化对应性说明**的要求，进行 SSOASS 的设计；
b) 系统的设计和开发应保护数据的完整性，例如，检查数据更新的规则，多重输入的正确处理，返回状态的检查，中间结果的检查，合理值输入检查，事务处理更新的正确性检查等；
c) 在内部代码检查时，应解决潜在的安全缺陷，关闭或取消所有的后门；
d) 交付的软件和文档，应进行关于安全缺陷的定期的和书面的检查，并将检查结果告知用户；
e) 系统控制数据，如口令、密钥，不应在未受保护的程序或文档中以明文形式存储，应以书面形式提供给用户关于软件所有权法律保护的指南；
f) SSOASS 的开发过程应保持一个安全环境，该安全环境要求：
——系统开发所使用的计算机系统的安全使用和维护情况的安全策略和措施应有书面记载，并可供检查；
——系统开发过程中使用的所有计算机系统应接受定期的和有书面记载的内部安全审查，描述审查过程的文件和真实的审查报告应可供检查；
——除授权的分发机构外，不应在开发环境外部复制或分发内部文档；
——系统开发环境的计算机系统使用的所有软件应当合法地从确定的渠道获得；
——系统开发者个人独自开发的软件，应在被开发管理者审核后才能用于开发的系统。

5.5.4.4 文档

应按 **GB/T 20271-2006 中 6.5.5.4** 的要求，从以下方面编制 SSOASS 的文档：

a) 用户文档应提供关于不同类型用户的可见的安全机制，并说明它们的用途和提供有关它们使用的指南；
b) 安全管理员文档应提供有关如何设置、维护和分析系统安全的详细说明，包括当运行一个安全设备时，需要控制的有关功能和特权的警告，以及与安全有关的管理员功能的详细描述，包括增加和删除一个用户、改变主、客体的安全属性等；
c) 文档中不应提供任何一旦泄露将会危及**本安全级范围内的**系统安全的信息；

d) 有关安全的指令和文档根据权限应分别提供给用户、系统管理员和系统安全员；这些文档应为独立的文档，或作为独立的章条插入到管理员指南和用户指南中；
e) 文档也可为硬拷贝、电子文档或联机文档，如果是联机文档应控制对其的访问；
f) 应提供关于所有审计工具的文档，包括为检查和保持审计文件所推荐的过程、针对每种审计事件的详细审计记录、为周期性备份和删除审计记录所推荐的过程等；
g) 提供如何进行系统自我评估（如：带有网络管理、口令要求、拨号访问控制、意外事故计划的安全报告）和为灾害恢复计划所做的建议，以及描述普通入侵技术、其它威胁及其检查和阻止的方法；
h) 安全管理员文档应提供安全管理员如何以安全的方式管理系统，除了给出一般的安全忠告，还要明确：
——在系统用安全的方法安装时，围绕用户、用户账户、用户组成员关系、主体和客体的属性等，以及如何安装或终止安装；
——在系统的生存周期内，如何用安全的方法维护系统，包括为了防止系统被破坏而进行的每天、每周、每月的常规备份等；
——如何用安全的方法重建部分 SSOASS（如内核）的方法（如果允许在系统上重建 SSOASS）；
——说明安全审计机制，使授权用户可以有效地使用安全审计来检查安全策略；
——必要时，如何调整系统的安全默认配置。

5.5.4.5 生存周期支持

应按 **GB/T 20271-2006 中 6.5.5.5** 的要求，从以下方面实现 SSOASS 的生存周期支持：
a) 按**可测量的生存周期模型和遵照实现标准-所有部分的工具和技术的要求**进行 SSOASS 的开发，并提供充分的安全措施**和有组织的缺陷纠正；**
b) 文档应详细阐述安全启动和操作的过程，详细说明安全功能在启动、正常操作维护时是否能被撤消或修改，说明在故障或系统出错时如何恢复系统至安全状态；
c) 如果系统含有加强安全性的硬件，那么管理员、终端用户或自动的诊断测试，应能在各自的操作环境中运行它并详细说明操作过程。

5.5.4.6 测试

应按 **GB/T 20271-2006 中 6.5.5.6** 的要求，从以下方面对 SSOASS 进行测试：
a) 通过范围证据和严格的范围分析，高层设计测试、低层设计测试和实现表示测试，顺序的功能测试，相符独立性测试和**完全独立性测试**等，确认 SSOASS 的功能与所要求的功能相一致；
b) 所有系统的安全特性，应被全面测试，包括查找漏洞，如允许违反系统访问控制要求、允许违反资源访问控制要求、允许拒绝服务、允许验证数据进行未授权访问等；
c) 所有发现的漏洞应被改正、消除或使其无效，并在消除漏洞后重新测试，以证实它们已被消除，且没有引出新的漏洞；
d) 提供测试文档，详细描述测试计划、测试过程、测试结果。

5.5.4.7 脆弱性评定

应按 **GB/T 20271-2006 中 6.5.5.7** 的要求，从以下方面对 SSOASS 进行脆弱性评定：
a) 通过**严格的隐蔽信道分析，对隐蔽信道进行严格搜索，标识出可识别的隐蔽信道；**
b) 对防止误用的评定，应通过对文档的检查和分析确认，查找 SSOASS 以不安全的方式进行使用或配置而不为人们所察觉的情况；
c) 对 SSOASS 安全功能强度评估，应通过对安全机制的安全行为的合格性或统计结果的分析，证明其达到或超过安全目标要求所定义的最低强度；
d) **高抵抗力分析，**应通过独立穿透测试和对脆弱性的系统化搜索和完备性分析，确定 SSOASS 可以抵御**高攻击能力攻击者发起的穿透性攻击**。

5.5.5 SSOASS 安全管理

应根据本安全等级中安全功能技术要求所涉及的基础安全技术要求、安全功能技术要求和安全保

证技术要求所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，按 **GB/T 20271-2006 中 6.5.6** 的要求，从以下方面实现 SSOASS 的安全管理：

a) 对安全保证措施所涉及的 SSOASS 自身保护、SSOASS 设计和实现等有关内容，以及与一般的安装、配置等有关的功能，制定相应的操作、运行规程和行为规章制度；
b) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所实现的安全功能，实现 SSF 安全功能的管理；
c) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全属性，从管理安全属性、安全的安全属性、静态属性初始化、安全属性终止和安全属性撤消等方面，实现 SSF 安全属性的安全管理；
d) 对 SSOASS 中的每个安全功能模块，根据安全功能技术和安全保证技术所涉及的安全数据，从管理 SSF 数据、SSF 数据界限的管理和安全的 SSF 数据等方面，实现 SSF 安全数据的安全管理；
e) 将应用软件系统管理员、安全员和审计员等重要安全角色分别设置专人担任，并按最小授权原则分别授予他们各自为完成自身任务所需的最小权限，并形成相互制约的关系；
f) 对网络环境运行的应用软件系统，实现 SSOASS 安全机制的集中管理。

附录 A
（资料性附录）
应用软件系统安全的有关概念说明

A.1　应用软件系统在信息系统中的位置

应用软件系统位于信息系统最上层，与用户直接打交道。应用软件系统是在信息系统的硬件系统、操作系统、网络系统、数据库管理系统的支持下运行的，是构成信息系统的最重要部分，是信息系统中直接为用户提供服务的部分。上述其它系统都是为应用软件系统的运行提供支持和服务的。应用软件系统在信息系统中的位置如图A.1所示。

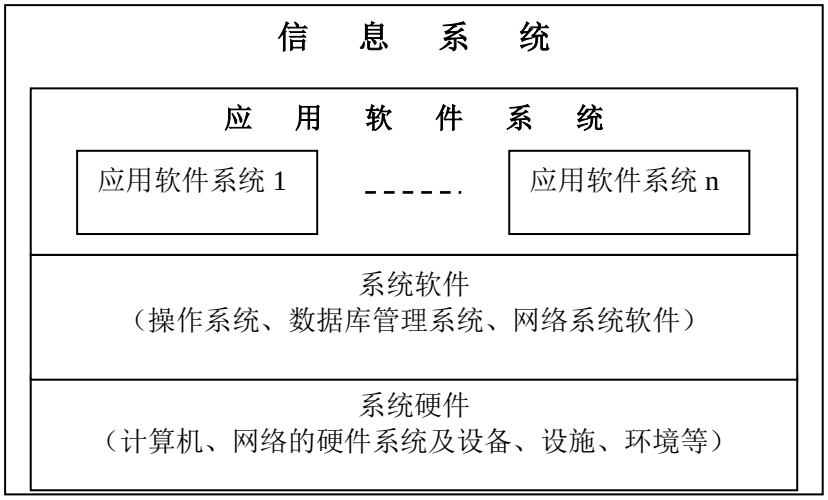

图 A.1　应用系统在信息系统中的位置

A.2　应用软件系统安全在信息系统安全中的作用

应用软件系统的安全是信息系统安全的重要组成部分。应用软件系统的安全需求是信息系统安全需求的来源和基础。为了实现应用软件系统的安全，需要有支持应用软件系统运行的硬件系统、操作系统、网络系统、数据库管理系统等各层安全的支持。应用软件系统的安全需求，根据具体情况，可以在应用软件系统层实现，也可以在支持应用软件系统运行的各层的支持下实现。

A.3　关于应用软件系统的业务连续性

应用软件系统的业务连续性是信息系统所承载的业务应用的连续性的表征，是信息系统安全运行的重要组成部分，通过应用软件系统的连续运行来支持。为对抗信息系统发生灾难性故障（比如：水灾、火灾、地震或严重的外部攻击等），使信息系统发生灾难性故障时能在限定的时间范围内恢复运行，业务连续性需要通过灾难备份与恢复来确保。与一般的安全性概念不同的是，对灾难备份与恢复的要求仅仅与信息系统所承载的业务的连续性要求有关，不同的业务应用有不同的业务连续性要求，从而有不同的灾难备份与恢复要求。其中的两个重要因素是数据备份的间隔时间和业务中断的时间。数据备份的时间间隔与允许数据丢失的程度有关，允许数据丢失的程度越小，数据备份的时间间隔就应越小；允许业务中断的时间间隔与业务停止运转所造成的损失有关，业务中断所造成的损失越大，允许业务中断的时间间隔就越小。

需要指出的是，业务连续性要求的分级与信息系统的安全等级并没有严格的对应关系。因为两者的依据是不一样的。但是，一般来讲，低等级安全要求的信息系统，业务连续性要求往往较低，所以在灾难备份与恢复方面的要求也就比较低，而高等级安全要求的信息系统，业务连续性要求往往较高，所以灾难备份与恢复方面的要求也就较高。