第3章 审计证据收集与评价

信息系统审计活动的本质就是有目的、有计划地收集证据，进行分析评价，最后作出判断的过程。审计人员最终形成的任何结论和意见，都必须以合理、充分的证据为基础，因此，收集与评价审计证据是信息系统审计工作的中心环节。

审计证据有3个基本属性：内容、形式和功能，而电子审计证据则是一类特殊而重要的证据。在收集证据时需要考虑证据的充分性、适当性和可信性问题，电子审计证据在收集过程中除了要考虑上述因素以外，还要考虑到电子审计证据的特殊性，如内容显示具有间接性，存放介质具有多样性，内容修改或删除没有痕迹，内容复制具有无磨损性等，这些都增加了取证的难度。

根据信息系统审计的实践提出了观察、查询、函证、复核、黑盒法、白盒法、计算机取证等方法，提出了利用应用软件接口、审计软件工具、ODBC技术和XBRL技术等方法来获取电子审计证据。

审计证据是整个审计过程的基石，也是审计风险模型的基础。证据评价是分析证据与审计结论的关联性。在这一过程中包含大量的不确定因素，存在一定的审计风险。对审计风险的判断、分析和计量有助于审计人员更好地防范和控制风险。证据理论能够更好地表示审计人员基于证据的职业判断过程，并能区分不确定性和无知性。通过对审计证据的分类、分层，能够更好地区分不同证据类型和不同审计风险，提出了用审计证据计算总体审计风险的计算模型，通过量化了的审计证据来度量审计风险。

3.1 审计证据概述

3.1.1 审计证据的含义

信息系统审计证据是指用以说明审计的实际情况，形成鉴定结论的材料。审计证据有3个基本的属性：内容、形式和功能。

所谓内容是指真实、客观的情况。真实性和客观性是审计证据的基本要求。

所谓形式是指作为证据的事实必须符合法定形式。只有那些外在表现形式符合证据法规规定形式的事实才能成为证据。《审计机关审计证据准则》第3条就对审计证据的法律形式作了明确的规定，如书面证据、实物证据、口头证据等。凡是不符合法律形式的材料都不

能视为审计证据，这是证据形式合法的体现和要求。

所谓功能是指证据内容具有支撑结论的作用，即证明力。例如，审计证据概念中“用以说明审计的实际情况，形成鉴定结论的”要求，就是对审计证据功能的规定。

3.1.2　审计证据的种类

按审计证据与审计对象的关系可以分为直接证据和间接证据。按审计证据的来源可以分为内部证据、外部证据和审计人员自己获得的证据。按审计证据的形式可以分为实物证据、书面证据、电子证据、口头证据和环境证据。

(1) 实物证据

实物证据通常是证明企业实际情况与信息系统中数据所反映的是否一致的证据，以此确认信息系统数据的真实性。

(2) 书面证据

书面证据包括业务持续能力计划、业务恢复能力计划、各种制度与规定、软件开发文档、软件维护文档、软件更新文件、会议记录、合同等。

(3) 电子证据

电子证据包括存放在信息系统中的凭证、账簿、报表、表格、数据、软件、日志文件、字典、权限表、数字签名等。

(4) 口头证据

口头证据是被审计企业职员或其他有关人员对信息系统审计师的提问作口头答复所形成的一类证据。这类证据可靠性较差，证明力较弱，主要作用是发掘一些重要的线索，以搜集到更为可靠的证据。

(5) 环境证据

环境证据是指对被审计企业的信息系统产生影响的各种环境事实。包括以下几种：有关内部控制情况，软件、硬件等的运行状况，管理水平，人员素质等。

3.1.3　电子证据的特点

信息系统审计的一个显著特点是需要处理大量的电子证据，我们必须了解电子证据的性质，特别是它们与传统证据的区别。

(1) 内容显示具有间接性

电子证据实质上只是一堆按编码规则处理成的二进制信息，必须借助专门的设备和软件，以文字、图形、表格、数字、声音等形式显示或打印出来，供人们识读。

(2) 存放介质具有多样性

电子信息本身具有无形性，必须依附在有形的介质上，根据存储形式的不同(可以是电信号、光信号、磁信号等)，分别存储在光盘、磁带、磁盘、U 盘等各种存储介质上。

(3) 内容修改或删除没有痕迹

对电子信息的修改和删除操作可以不留下任何痕迹，无法通过差异比对等方法加以鉴别。

(4) 内容复制具有无磨损性

电子信息的复制过程是无磨损的,因此,没有正本、副本的区别。而且,复制及传播方便、快捷。

3.1.4 电子证据的形式

目前,对电子证据形式的定位问题的争论比较多,下面作一简要介绍。

(1) 电子证据属于视听资料

首先,在1982年的《民事诉讼法(试行)》中首次规定了视听资料这一新的证据种类,并把录音、录像、计算机存储资料等划归其中;其次,视听资料与电子证据在存在形式上有相似之处,即存储的视听资料及电子证据均需借助一定的工具或以一定的手段转化为其他形式后才能被人们直接感知;再次,两者的正本与副本均没有区别。

(2) 电子证据属于书证

书证是指以文字、图画、符号等表达的思想内容来证明事实的资料。它与电子证据的相同之处在于两者都以表达的思想内容来证明事实。我国《合同法》第11条规定:"书面形式是指合同书、信件及数据电文(包括电报、电传、传真、电子数据交换和电子邮件)等可以有形地表现所载内容的形式"。而且,电子证据通常通过打印到纸上或显示在屏幕上等,才能被人们看见和利用,因而具有书证的特点。

(3) 电子证据属于物证

奥恩·凯西(Eoghan Casey)在《数字证据与计算机犯罪》(Digital Evidence and Computer Crime)中提出:"数字证据是物证的一种。尽管数字证据不像其他形式的物证那样有形,但它仍然属于物证"。

(4) 电子证据属于混合证据

"混合证据说"认为电子证据是若干传统证据的组合,而非独立的一种新型证据,也非传统证据中的一种。

(5) 电子证据属于独立证据

鉴于电子证据种类划分的复杂性及其本身的特殊性,并参考国外的电子证据立法,有学者提出将电子证据作为一种独立的证据种类。由于电子证据具有区别于其他证据的显著特征,其外在表现形式也是多媒体的,几乎涵盖了所有的传统证据类型,因此把它放入哪一类传统证据都不合适。

3.1.5 审计证据的充分性

要对审计结果给出合理的结论,信息系统审计师必须获得充分的审计证据。

所谓充分性是指审计证据数量的最低要求。当审计证据相关与可靠程度较高时,所需审计证据数量较少,反之,所需数量较多。特别是单一证据,在一定数量基础上,各证据之间应通过逻辑推理方式形成有效的证据链。充分性并不是说证据数量越多越好,受审计成本限制,信息系统审计师应把需要足够数量的审计证据控制在最低限度。信息系统审计师应评估其审计期间所获得证据的充分性。

充分的审计证据是指：

(1) 如果审计证据支持所有关于审计目标和范围的实质问题，则可以被视为充分的证据。

(2) 审计证据应该客观、充分，使得一个有资格的独立方可以重复操作审计检查并获得同样的结果。审计证据应该与审计项目的实质性以及所涉及的风险相称。

(3) 充分性用来衡量审计证据的量，而正确性用来衡量审计证据的质，两者相互关联。在这种情况下，当信息系统审计师使用从机构获取的信息来进行审计工作时，审计师应要求并强调所获信息的正确性和完整性。

(4) 当信息系统审计师认为无法获取充分审计证据时，审计师应该用与表明审计结果相一致的方法来公开审计。

3.1.6　审计证据的适当性

信息系统审计师应评估其审计期间所获得证据的适当性。

所谓适当性是指证据与审计事项或审计目标之间有逻辑上的联系，能够证明审计事项的存在或不存在。与审计事项或审计目标相关程度越高，其证明力越强；相反，则证明力越弱，甚至不能作为审计证据。

适当的审计证据是指：

(1) 审计证据包括审计师执行的程序，信息系统审计师操作程序的结果，原始文件(电子或文本形式)、档案记录和审计的相关佐证资料以及审计工作的发现和结果。

(2) 审计证据表明审计工作是在遵守现行法律、法规和政策的条件下开展的。

(3) 当信息系统审计师通过控制测试来获取审计证据时，应考虑审计证据的完整性是否能够支持控制风险的评估等级。

(4) 应正确识别审计证据，相互参照并将其分类。

(5) 当评估审计证据的可信度时，应考虑其诸如来源、性质(书面、口头、视觉、电子)以及真实性(数码和亲手签名、印章)等属性。

3.1.7　审计证据的可信性

信息系统审计师应评估其审计期间所获得证据的可信性。

所谓可信性是指审计证据反映审计事项客观现实的程度。审计证据可信性越强，其证明力越强。有的审计证据虽然有相同的客观属性，但不同形式、不同来源以及不同时间上的审计证据其可信程度则不同。一般认为，书面证据比经口头询问而获取的证据更可靠。书面证据中，国家机关、社会团体依职权制作的公文书证比其他书证更可靠；物证档案、鉴定结论、勘验笔录或经过公证、登记的书证比其他书证、视听资料和证人证言更可靠；外部取得的证据比从被审计单位内部获得的证据更可靠；原始证据比复制证据更可靠；直接证据比间接证据更可靠；信息系统审计师亲自取得的证据比被审计企业提供的证据更可靠；向独立的第三方获取的证据比向与被审计企业有利害关系者获取的证据更可靠；被审计企业内部控制较好时比该单位内部控制较差时提供的内部证据更可靠；不同渠道或不同性质的

审计证据能相互印证时，比来自单一渠道单一证据更可靠；越及时的证据越可靠；客观证据比主观证据可靠。可靠性具有高度的综合概括性，需要信息系统审计师针对具体情况运用专业判断对审计证据加以分析和比较。

可信的审计证据是指如下几方面。

(1) 一般来说，审计证据的可信度在以下情况下更大：

① 以书面形式而非口头表达。

② 取自独立来源。

③ 由信息系统审计师而非被审方获取。

④ 经独立第三方证实。

⑤ 被独立第三方保存。

(2) 信息系统审计师应考虑选取最经济而有效的方法收集必要的审计证据来达到审计的目标并规避风险。然而，困难和成本并非构成省略一个必要审计步骤的充分理由。

(3) 收集审计证据的程序应根据正在受审项目的内容(例如其性质、审计的时间安排、专业的判断)来变化。信息系统审计师应根据审计目标选择最合适的审计程序。

(4) 信息系统审计师可以通过如下方法获取审计证据：

① 检验。

② 观察。

③ 询问和确认。

④ 重复操作。

⑤ 验算。

⑥ 计算。

⑦ 分析步骤。

⑧ 其他可接受方法。

(5) 信息系统审计师应考虑其获取的任何信息的来源和性质，以便评估这些信息的可信度和进一步确认的需要。

3.2 审计证据收集方法

3.2.1 收集方法概述

在进行审计证据收集时要考虑成本因素和技术因素。

由于信息系统审计活动是在一定时间、一定范围内来活动的，证据收集必然要考虑到时间和成本，这与诉讼证据收集有所区别。在诉讼时取证成本负担对象不确定，有时为获得有价值的证据，取证成本并不很重要。这就是所谓的成本因素。

所谓技术因素是指许多审计证据的获取是依据信息系统审计师的专业手段，而且是由审计人员单方面取得的，这可能对审计证据的证明力产生一定的影响。

信息系统审计与控制协会标准管理委员会提出了信息系统审计师获取审计证据的方法，包括检验、观察、询问和确认、重复操作、验算、计算、分析步骤以及其他可接受的方法等。

我们根据信息系统审计的实践提出了观察法、查询法、函证法、复核法、黑盒法、白盒法、计算机取证技术等方法，下面分别加以介绍。

3.2.2　观察法

观察法是指审计人员到被审计单位的经营场所及其他有关场所进行实地察看，以证实审计事项的一种方法。通过观察业务操作流程和岗位之间相互制约程度以及检查内部制度的执行情况等手段，发现线索并直接获取证据。观察提供的审计证据仅限于观察发生的时间和地点，并且在相关人员已知被观察时，相关人员从事活动或执行程序可能与日常的做法不同，从而影响内部审计人员对真实情况的了解。

3.2.3　查询法

查阅资料是指审计人员查阅被审计单位的有关资料、技术文档等，对业务和技术资料、文件、档案材料等从形式到内容进行认真阅读。阅读中不能断章取义，片面理解，要在全面分析、客观公正的基础上，寻求相关的审计证据。对审计发现的问题，在做好笔录的同时，必须获得相关材料的影印件，这样收集的证据才是有价值的证据。

在审计过程中，由于审计人员不可能对被审单位的业务活动进行全面检查，但又必须完成审计项目。因此，审计人员只能根据能够反映被审单位经济活动的相关资料来进行抽查，从而获得必要的审计证据。所以，所要抽查的资料必须是有用的、具有代表性和公允性的资料。

问卷调查是指审计人员以书面或口头方式，向被审计单位内部或外部的知情人员获取审计线索和证据的方法，并对答复进行评价的过程。知情人员对询问的答复可能为审计人员提供尚未获悉的信息或线索，也可能提供与已获悉信息存在重大差异的信息。询问本身不足以认定相关问题，也不足以测试内部控制运行的有效性，审计人员还应当实施其他审计程序以获取充分、适当的审计证据。

审计人员可以根据询问结果考虑修改审计程序或实施追加的审计程序。

3.2.4　函证法

函证法是指审计人员向有关单位或个人发函以证明某一审计事项的一种方法。它通过直接来自第三方对有关信息和现存状况的声明来获取和评价审计证据。例如，为证实应收、应付款项的真实性、准确性，就常采用这种方法。而电子商务交易的真实性也必须通过函证方法加以核实。

3.2.5　复核法

复核法是指审计人员利用已获得的技术文档资料，如数据字典、数据流程图、控制流程图、权限表、科目编码表、报表取数公式等，对信息系统的流程进行复核，以此来获取审计线

索或证据。

3.2.6 黑盒法

黑盒法(black box testing)也称为功能测试或数据驱动测试。信息系统审计师在已知企业信息系统功能的条件下,通过测试来检测每个功能是否都能正常使用。

在测试时,把企业信息系统看作一个不能打开的黑盒子,在完全不考虑程序内部结构和内部特性的情况下,利用测试用例来对系统进行测试,以检查程序功能是否按照需求规格说明书的规定正常使用,程序是否能够适当地接收输入数据而产生正确的输出信息,并且保持外部信息(如数据库或文件)的完整性。

黑盒法着眼于信息系统外部功能,不考虑内部逻辑结构,针对企业信息系统的界面和功能进行测试,信息系统审计师在应用黑盒法时,手头只需有程序的功能说明书就足够了。

黑盒法主要有等价类划分、边界值分析、因果图分析、错误推测等方法。

(1) 等价类划分

等价类划分法的基本思想是:如果将输入数据的可能值分成若干个"等价类",就可以合理地假定每一类的一个代表性值在测试中的作用等价于这一类中的其他值,即如果某一类中的一个测试用例发现了错误,这一等价类中的其他测试用例也能发现同样的错误;反之,如果某一类中的一个测试用例没有发现错误,则这一类中的其他测试用例也不会查出错误(除非等价类中的某些测试用例又属于另一等价类,因为几个等价类有可能是相交的)。

设计等价类的测试用例一般分为以下两步:

① 划分等价类并给出定义。

② 选择测试用例。

选择测试用例的原则是:有效等价类的测试用例尽量公用,以期进一步减少测试的次数;无效等价类必须每类一例,以防漏掉本来可能发现的错误。

划分等价类时,需要研究程序的功能说明,以确定输入数据的有效等价类和无效等价类。在确定输入数据的等价类时常常还需要分析输出数据的等价类,以便根据输出数据的等价类导出对应的输入数据等价类。

(2) 边界值分析

经验表明,处理边界情况时程序最容易发生错误。例如,许多程序错误出现在下标、纯量、数据结构和循环等边界附近。因此,设计使程序运行在边界情况附近的测试方案,暴露出错误的可能性更大一些。

使用边界值分析方法设计测试用例首先应该确定边界情况,这需要经验和创造性。通常,输入等价类和输出等价类的边界就是应该着重测试的程序边界情况。选取的测试数据应该刚好等于、刚刚小于和刚刚大于边界值。也就是说,根据边界值分析法,应该选取刚好等于、稍小于和稍大于等价类边界值的数据作为测试数据,而不是选取每个等价类内的典型值作为测试数据。

(3) 因果图分析

因果图分析是为了解决边界值分析和等价划分的一个弱点,即未对输入条件的组合进

行分析。而因果图恰恰有助于用一个系统的方法选择出此类高效的测试用例集，并且可以指出规格说明的不完整性和不明确之处。

因果图是一种形式语言(有严格语法限制的语言)，可将自然语言描述的规格说明转换为因果图。实质上，它是一种数字逻辑电路(一个组合的逻辑网络)，但没有使用标准的电子学符号，而是使用了稍微简单点的符号。借助因果图列出输入数据的各种组合与程序对应动作效果之间的阶段联系，构造判定表，由此设计测试用例是生成测试用例的有效方法。

(4) 错误推测法

人们也可以通过经验或直觉来推测程序中可能存在的各种错误，从而有针对性地编写检查这些错误的例子，这就是错误推测法。

错误推测法在很大程度上依靠直觉和经验进行。其基本想法是列举出程序中可能有的错误和容易发生错误的特殊情况，并据此选择测试用例。对于程序中容易出错的情况也有一些经验总结出来，例如，输入数据为 0 或输出数据为 0 往往容易发生错误；如果输入或输出的数目允许变化(例如，被检索的或生成的表的项数)，则输入或输出的数目为 0 和 1 的情况(例如，表为空或只有一项)是容易出错的情况。还应仔细分析程序规格说明书，注意找出其中遗漏或省略的部分，以便设计相应的测试用例，检测程序员对这些部分的处理是否正确。

3.2.7　白盒法

白盒法(white box testing)也称为结构测试或逻辑驱动测试，它是已知产品内部工作过程，可通过测试来检测产品内部动作是否按照规格说明书的规定正常进行，按照程序内部的结构测试程序，检验程序中的每条通路是否都能按预定要求正确工作，而不顾及它的功能。白盒测试的主要方法有逻辑驱动、基路测试等，主要用于软件验证。

白盒法需要全面了解企业信息系统的内部流程。白盒法是穷举路径测试。在使用这一方案时，测试者必须检查程序的内部结构，从检查程序的逻辑着手，得出测试数据。贯穿程序的独立路径数是一个天文数字。但即使每条路径都测试了，也仍然可能有错误。第一，穷举路径测试绝不能查出程序违反了设计规范，即程序本身是一个错误的程序；第二，穷举路径测试不可能查出程序中因遗漏路径而出错；第三，穷举路径测试可能发现不了一些与数据相关的错误。

白盒测试法的覆盖标准有逻辑覆盖、循环覆盖和基本路径测试。其中，逻辑覆盖包括语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖和路径覆盖。

(1) 语句覆盖

为了暴露程序中的错误，至少每条语句应该执行 1 次。语句覆盖的含义是选择足够多的测试数据，使被测试程序中的每条语句至少执行 1 次。

例如，图 3.1 是一个被测模块的流程图。其源程序(用 Pascal 语言书写)如下：

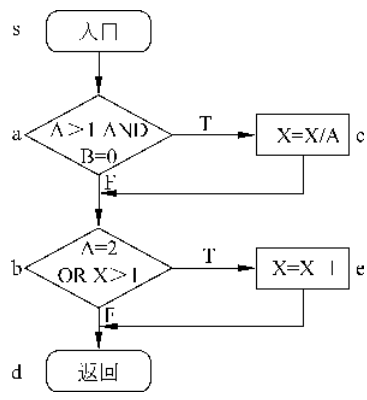

图 3.1　某段程序的流程图

```
PROCEDURE EXAMPLE (A,B: REAL; VAR X: REAL)
    BEGIN
        IF (A>1)AND (B=0)
            THEN X := X/A
        IF (A=2) OR (X>1)
            THEN X := X+1
    END;
```

为了使每条语句都执行1次，程序的执行路径应该是sacbed，为此只需输入下面的测试数据（实际上，X可以是任意实数）：

A=2，B=0，X=4

语句覆盖对程序的逻辑覆盖很少，在例子中，两个判定条件都只测试了条件为真的情况，如果条件为假时处理有错误，显然不能发现。此外，语句覆盖只关心判定表达式的值，而没有分别测试判定表达式中每个条件取不同值时的情况。在上面的例子中，为了执行sacbed路径，以测试每条语句，只需两个判定表达式（A>1）AND（B=0）和（A=2）OR（X>1）都取真值，因此，使用上述一组测试数据就足够了。但是，如果程序中把第1个判定表达式中的逻辑运算符AND错写成OR，或把第2个判定式中的条件“X>1”误写成“X<1”，则使用上面的测试数据并不能查出这些错误。

综上所述可以看出，语句覆盖是很弱的逻辑覆盖标准，为了更充分地测试程序，可以采用以下的逻辑覆盖标准。

（2）判定覆盖

判定覆盖的含义是，不仅每条语句必须至少执行1次，而且每个判定的可能的结果都应该至少执行1次，也就是说，每个判定的每个分支都至少执行1次。

对于上述例子来说，能够分别覆盖路径sacbed和sabd的两组测试数据，或者可以分别覆盖路径sacbd和sabed的两组测试数据，都满足判定覆盖标准。例如，用下面两组测试数据就可以做到判定覆盖：

① A=3，B=0，X=3（覆盖sacbd）

② A=2，B=1，X=1（覆盖sabed）

判定覆盖比语句覆盖强，但是对程序逻辑的覆盖程度仍然不高，例如，上面的测试数据只覆盖了程序全部路径的一半。

（3）条件覆盖

条件覆盖的含义是，不仅每条语句至少执行1次，而且是判定表达式中的每个条件都取到各种可能的结果。

在图3.1的例子中共有两个判定表达式，每个表达式中有两个条件，为了做到条件覆盖，应选取测试数据使得在a点有下述各种结果出现：

A>1，A≤1，B=0，B≠0

在b点有下述各种结果出现：

A=2，A≠2，X>1，X≤1

只需要使用下面两组测试数据就可以达到上述覆盖标准：

① A=2,B=0,X=4（满足 A>1,B=0,A=2 和 X>1 的条件,执行路径 sacbed)。

② A=1,B=1,X=1（满足 A≤1,B≠0,A≠2 和 X≤1 的条件,执行路径 sabd)。

条件覆盖通常比判定覆盖强,因为它使判定表达式中每个条件都取到了两个不同的结果,判定覆盖却只关心整个判定表达式的值。例如,上面两组测试数据也同时满足判定覆盖标准。但是,也可能有相反的情况,虽然每个条件都取得了两个不同的结果,判定表达式却始终只取一个值。例如,如果使用下面两组测试数据,则只满足条件覆盖标准并不满足判定覆盖标准(第 2 个判定表达式的值总为真):

① A=2,B=0,X=1(满足 A>1,B=0,A=2 和 X≤1 的条件,执行路径 sacbed)。

② A=1,B=1,X=2（满足 A≤1,B≠0,A≠2 和 X>1 的条件,执行路径 sabed)。

(4) 判定/条件覆盖

既然判定覆盖不一定包含条件覆盖,条件覆盖也不一定包含判定覆盖,那么自然会提出一种能够同时满足这两种覆盖标准的逻辑覆盖,这就是判定/条件覆盖,它的含义是选取足够多的测试数据,使得判定表达式中的每个条件都取到各种可能的值,而且每个判定表达式也都取到各种可能的结果。

对于图 3.1 所示例子而言,下述两组测试数据满足判定/条件覆盖标准:

① A=2,B=0,X=4。

② A=1,B=2,X=1。

但是,这两组测试数据也就是为了满足条件覆盖标准最初选取的两组数据,因此,有时判定/条件覆盖也并不比条件覆盖更强。

3.2.8　计算机取证技术

通过软件工具直接从企业信息系统中获取证据。计算机取证技术包括利用应用软件接口功能、审计软件工具、ODBC 技术以及 XRBRL 技术等方式获取审计证据。

1. 利用应用软件接口功能获取审计证据

企业的信息系统功能虽然大相径庭,但仍然可以利用其中所提供的一些功能、模块、工具等收集审计证据。如数据转存功能、数据导入导出功能、取数函数接口功能等,还可以利用数据库管理系统(DBMS)获取审计证据。

(1) 利用被审计单位信息系统的数据转出功能。利用此功能的前提是转出数据格式符合审计人员的需求,或者转出数据经过处理后能够符合审计人员的需求。例如,利用财务系统的凭证和余额的查询功能将数据查询出来,然后利用其“文件”菜单下的“数据转出”功能(大部分财务软件都有该功能)可将查询出的数据转存为 xls 文件、dbf 文件或文本文件格式,然后可以直接打开分析。

(2) 利用被审计单位信息系统所使用的数据库系统的转出功能。例如,利用 SQL 数据库自身的“导入和导出数据”功能可以将 SQL 数据库中的数据表全部或有选择地转存为文本文件和 doc 文件格式。

(3) 根据被审计单位信息系统的情况,直接将可利用的数据文件复制到审计人员的计算机中。例如,用友财务软件的数据库为 access,则可以找到其文件存放位置,直接将 mdb

文件复制出来，再运用专用审计软件导出进行数据分析处理。

2. 利用审计软件工具获取审计证据

市面上的一些通用审计软件都有数据采集功能。例如“审计数据采集分析 2.0”软件具有强大的数据采集功能，能用来采集各种类型的被审计数据。它所适应的数据库有 Access，dBASE，FoxPro，Paradox 等微机数据库，Excel，文本文件以及可以用 ODBC 方式访问大中型数据库，如 Oracle，Sybase，SQL Server，Informix，DB2 等。

3. 利用 ODBC 技术获取审计证据

企业信息系统中的数据大部分都存放在数据库中，但商用的数据库系统之间互不连通。1991 年，微软公司提出了开放式数据库互连技术（open database connectivity，ODBC），1992 年又推出了 ODBC SDK 2.0 版。由于 ODBC 思想上的先进性，推出后仅仅二三年就受到了众多厂家与用户的青睐，成为一种广为接受的标准。目前，已经有 130 多家独立厂商宣布了对 ODBC 的支持，常见的 DBMS 都提供了 ODBC 的驱动接口，这些厂商包括 Oracle，Sybase，Informix，Ingres，IBM（DB/2），DEC（RDB），HP（ALLBASE/SQL），Gupta，Borland（Paradox）等。

ODBC 是一个数据库访问库，它包含访问不同数据库所要求的 ODBC 驱动程序，如要访问 Sybase，就用 Sybase 的 ODBC 驱动程序；要访问 DB2 数据库，就用 DB2 的 ODBC 驱动程序。总之，应用程序要访问不同类型的数据库，只要调用 ODBC 所支持的函数，动态链接到相应的驱动程序上即可。所以，运用 ODBC 技术可以实现跨系统、跨平台的数据采集。

4. XBRL 获取审计证据

XBRL（extensible business reporting language，可扩展商业报告语言），是 XML（extensible markup language，可扩展的置标语言）于财务报告信息交换的一种应用。

XBRL 自其 1998 年诞生起，在国际上已经得到了迅速发展，而且研究表明，XBRL 技术增加了公司财务报告披露的透明度。倡导 XBRL 国际化的 XBRL 成立于 1999 年，由美国注册会计师协会与 EDGAR 在线、微软、普华永道等 12 家公司共同组建。目前，世界各国已有 250 多个机构参加了该组织。通过 XBRL 获取的各种财务数据可以直接作为审计证据使用，而且，通过 XBRL 还可以很方便地转换成 PDF 文件、HTML 页面或者其他相应的文件格式。

3.3 审计证据评价模型

3.3.1 审计风险的度量

审计过程本质上是一个以审计证据为基础的决策、判断的过程，在这一过程中包含大量的不确定因素，存在着审计风险。审计人员对取得的审计证据进行分析、判断和归纳。按照审计事项分类，按照审计证据与审计事项相关程度排序；对审计证据进行比较判断，决定取舍，剔除与审计事项无关、无效、重复、冗余的证据；对审计证据进行汇总和分析，确定审计