**iOS系统中“隔空投送”取证溯源流程的**

**研究与设计**

摘　要：隔空投送是苹果公司从iOS7系统开始推出的在Apple设备之间进行快速、简便交换文件的功能，其高效、简便的优点便利用户的同时也为不良信息的传播提供了渠道，尤其是在人流密集场所如地铁、商场等。本文针对这一问题，研究了隔空投送传输原理、传输方式，以及隔空投送传输过程中的关键信息。采用日志提取技术和哈希值爆破原理编写脚本工具，具体阐释了取证溯源的原理并通过实验测试研究了取证工作中的可能出现的各种情况。研究结果提出了隔空投送取证溯源的全流程，为应对滥用隔空投送传播非法信息的行为提供了依据，也为防止类苹果隔空投送功能被非法应用提出了应对建议。

关键词：iOS取证；隔空投送；日志提取；低功耗蓝牙；安全散列算法

**Research and design of "AirDrop"**

**forensics traceability process in iOS system**

**Abstract:** AirDrop is a function launched by Apple from iOS7 to quickly and easily exchange files between Apple devices, and its efficient and simple advantages facilitate users and provide a channel for the spread of bad information, especially in crowded places such as subways, shopping malls, etc. In order to solve this problem, this paper studies the principle and mode of AirDrop transmission, as well as the key information in the process of AirDrop transmission. The principle of forensic traceability is explained by using log extraction technology and the principle of hash value blasting to write a scripting tool, and various situations that may occur in forensic work are studied through experimental tests. The research results put forward the whole process of AirDrop forensics and traceability, which provides a basis for dealing with the abuse of AirDrop to spread illegal information, and also puts forward countermeasures to prevent the illegal application of the apple-like AirDrop function.

**Keywords:** iOS Forensics; AirDrop; Log Extraction; Bluetooth Low Energy technology; Secure Hash Algorithm 256-bit

目　录

[1.绪 论 1](#_Toc88492301)

[1.1研究背景及意义 5](#_Toc88492303)

[1.2国内外研究现状 5](#_Toc88492303)

[1.3研究内容与研究方法](#_Toc88492303) 3

[2.隔空投送传输原理](#_Toc88492302) 5

[2.1低功耗蓝牙连续性协议](#_Toc88492303) 5

[2.2 Wi-Fi无线通信技术](#_Toc88492304) 5

[2.3取证思路的提出](#_Toc88492306) 6

[3.隔空投送取证分析模块设计](#_Toc88492305) 7

[3.1日志收集与提取模块](#_Toc88492304) 7

[3.2证据信息解析模块](#_Toc88492306) 9

[3.3总结](#_Toc88492306) 13

[4.实验过程与结果](#_Toc88492305) 14

[4.1日志分析实验 1](#_Toc88492306)4

[4.2哈希破解实验 14](#_Toc88492306)8

[4.3关于手机号码哈希值为空的研究](#_Toc88492306) 20

[4.4总结 2](#_Toc88492306)1

[5.取证溯源流程与应对建议的提出](#_Toc88492305) 23

[5.1取证溯源流程的提出](#_Toc88492306) 23

[5.2关于公民个人隐私的思考](#_Toc88492306) 24

5.[3防止隔空投送功能被非法应用的应对建议](#_Toc88492306) 25

[6.总结与展望 2](#_Toc88492305)7

[6.1总结](#_Toc88492306) 27

[6.2展望 2](#_Toc88492306)7

[致　谢 2](#_Toc88492308)9

[参考文献](#_Toc88492309) 30

# 绪 论

1.1 研究背景及意义

随着移动互联网的发展和智能手机的普及，iOS（iPhone Operating System，iPhone操作系统）系统中的隔空投送功能作为一种便捷的文件传输工具得到了广泛应用。

这种苹果设备之间快速、简便交换文件的功能，对于希望在苹果设备间分享照片、视频、联系人、文档等内容的用户来说，非常方便。然而，由于其便利性，隔空投送也可能被不法分子用作骚扰和传播非法信息的工具，这给公民的权益以及公共安全带来了潜在风险。隔空投送功能允许用户在无需连接同一局域网或添加通讯录联系人的情况下，通过蓝牙扫描和WI-FI（Wireless Fidelity，无线保真）传输范围内的设备列表进行文件传输。这种非接触式的交流方式既为用户提供了便捷，同时也存在一定的安全风险。恶意用户可能会利用此功能，在未经许可的情况下传输非法图片、视频、音频等文件，或在公共场所如地铁、公交、商城等向附近用户非法投送和传播不良信息。

由于隔空投送过程不依赖于常规网络，因此无法通过常规网络监测手段进行有效监管。为了打击骚扰行为，为滥用行为提供可靠依据，本文深入研究隔空投送的工作原理和传输方式；以期实现用于iOS系统中隔空投送取证溯源的全流程示例，捕获、分析和记录隔空投送传输过程中的关键信息；提高侦查取证工作中对隔空投送滥用行为的识别和应对能力；提高公民对隔空投送行为的识别和防范能力；为相关领域的研究和实践提供参考和借鉴。

1.2 国内外研究现状

在iOS7之前的版本中，主要的文件传送方式通常涉及使用传统的方法，例如通过电子邮件、信息传输协议FTP（File Transfer Protocol，文件传输协议）、蓝牙或第三方应用程序来实现。这些方式在一定程度上限制了文件传输的便捷性和速度，尤其是对于大文件或多个文件的传送，操作可能相对繁琐或耗时。对于传统文件传送方式，涉及网络通信的取证，侦查机关可能会采用网络数据捕获技术，监视和记录网络流量，以获取目标设备或用户的网络活动信息。但对于iOS7以后的隔空投送传输方式引发的陌生人之间的社交活动，由于不涉及网络的参与，当发生骚扰案件以及扰乱社会公共秩序的事件发生时，我们很难利用传统的方式进行取证。

近年来，一些其他操作系统和平台同样提供了与iOS隔空投送类似的文件传送功能，其中Google推出的平台级文件传输服务名为“附近分享”（Nearby Share）。鉴于隔空投送曾被滥用为骚扰工具，Google在设计附近分享功能时采取了更为谨慎的策略。它将文件分享功能限定于Google账号及联系人的熟人网络内。若用户不希望受到该功能的打扰，可在可见性设置中限制发送者范围为特定联系人，或直接关闭可见性。对于偶尔的文件传输需求，用户可利用快速设置开关进行灵活应对。这些文件传输功能主要依赖于蓝牙或Wi-Fi直连技术，实现了设备间的快速数据传输，同时采用了身份验证和加密机制，以确保文件传输的安全性和用户隐私的保护。

1.2.1 国外研究现状

关于iOS系统中“隔空投送”的取证和溯源，虽然这一领域的研究尚处于起步阶段，但已有一些国外研究团队和安全专家对此进行探讨和研究。这些研究主要集中在如何从iOS设备中提取隔空投送的相关数据、理解隔空投送的工作机制以及如何用这些信息进行溯源和取证[1]。[Kinga Kieczkowska](https://kieczkowska.wordpress.com/)的研究包括隔空投送的工作原理分析、安全漏洞挖掘，通过分析隔空投送的技术原理和安全机制，探讨其在数据传输过程中的的工作原理和安全特性[2]。另有匿名的信息安全工作者通过统一日志平台分析发送方和接收方的日志，发布博客介绍隔空投送的相关知识[3]。

在隔空投送的取证研究方面，研究者致力于探讨如何从设备中提取隔空投送的活动记录，这包括传输的文件、传输时间、以及设备之间的互动信息等。例如通过传输日志和数据库分析iOS系统文件结构，特别是某些日志文件和数据库，可能包含隔空投送的活动记录。研究者尝试从这些数据存储中提取有用信息[3]。另外还涉及网络通讯分析，通过捕获并分析隔空投送过程中的网络通讯数据（如Wi-Fi和蓝牙通讯包），来尝试理解隔空投送的通讯细节和安全特性。

除了取证外，一些研究还聚焦于隔空投送可能带来的安全和隐私问题，例如如何防止恶意设备“发现”和接收未授权的文件，以及用户的联系信息（如电话号码和邮箱地址）可能在未经用户同意的情况下被暴露[4]。

1.2.2 国内研究现状

在国内，随着智能手机的普及和社会信息化程度的提升，iOS设备在刑事调查和电子取证中的重要性逐渐凸显[8]。国内研究人员针对传统的iOS设备取证技术进行了深入探索和研究[11]。研究内容涵盖了iOS设备的构成、运行机制、数据提取方法、取证工作流程以及应对取证挑战的策略[5]。在方法上，研究涉及到备份、逻辑、物理和非常规数据获取方式，并对不同取证软件进行了比较和分析，以确保取证结果符合司法程序和证据要求[6]。他们开发了针对iOS设备的取证工具和技术，实现了对短消息、通讯录、地理定位信息和音频数据等的有效提取和分析[10]。部分研究还涉及到在线取证系统的设计和开发，使得取证过程更加便捷高效。这些成果在国内刑事侦查和司法实践中得到了广泛应用，为案件调查提供了有力支持[7]。更有公安部鉴定中心对国内市场上主流产品按照不同种类进行调研归纳，详细分析对比了国内手机取证技术及工具的主要特点、优势与应用情况，立足于案件一线应用，给出手机数据取证工具目前需应对的挑战与未来的研究重点，并对其质量评价存在的难点提出思考[9]。

尽管取得了一定进展，但国内iOS设备隔空投送的取证溯源仍面临诸多挑战。iOS系统的更新迭代、设备安全性的提升以及数据加密技术的应用，都给取证工作带来了新的难题。北京网神洞鉴司法鉴定所的司法鉴定技术专家们通过深度解析 iPhone 设备日志，明确传输原理，找出了与 AirDrop 相关的记录，针对iOS系统中的取证需求，在iOS系统取证工具中增加了隔空投送取证相关的功能用于捕获、分析和提取隔空投送传输过程中的关键信息，以支持取证和溯源工作[12]。

综合来看，国内外对此课题的研究相对较少，但随着对移动设备安全性的不断增加，相关研究和工具的发展也可能会逐渐增多。未来，国内研究者需要进一步优化取证方法和工具，提升对iOS设备取证的深度和广度，以适应不断变化的技术和法律环境。综上所述，国内关于iOS系统中“隔空投送”取证溯源流程的研究已取得了一定的进展，但仍需要持续深入和创新，以应对日益复杂和多样化的智能终端取证需求，为维护社会安全和司法公正提供更加有力的技术支持和解决方案。

1.3 研究内容与研究方法

1.3.1 本文研究内容

iOS系统中的“隔空投送”功能，作为一种高效的文件传输机制，通过运用低耗蓝牙发现技术，能够迅速在邻近设备间建立连接。同时，借助Wi-Fi局域网进行数据传输，实现了快速且稳定的文件分享。为了深入理解其工作原理，本文将深入研究AirDrop的握手流程与传输机制，分析低耗蓝牙发现技术及Wi-Fi局域网的应用原理。

在对AirDrop日志的分析过程中，本文将聚焦于与取证紧密相关的内容。通过详细剖析日志中的传输记录、参与者身份信息以及传输文件类型等核心数据，我们将探讨如何有效提取证据，并分析嫌疑人可能泄露的信息。此外，本文还将探索发送方SIM卡信息、iMessage通信记录、FaceTime通话数据与接收方日志中记录的发送方手机号哈希值之间的潜在联系，并研究如何利用这些关联信息来追溯嫌疑人身份。

在设计溯源工具的功能模块时，本文将充分考虑如何高效地提取并分析日志信息，以确保工具的实用性和准确性。

通过对研究内容的深入探讨和实践，本文将得出结论，总结本研究的主要发现，讨论AirDrop取证溯源流程的研究与设计对于解决AirDrop滥用问题的重要意义，并展望未来的研究方向和应用前景。

1.3.2 研究方法

（1） 技术调研

在本研究中，将对隔空投送的工作原理、传输方式和安全机制进行深入的技术调研。通过查阅相关文献、官方文档和技术手册，收集和整理相关信息，建立对隔空投送技术的全面理解。这将为后续的研究和设计提供重要的基础和参考。

（2） 研究与设计

基于技术调研的结果，将设计并实现AirDrop取证溯源的功能模块与流程。通过分析隔空投送的工作流程和数据传输特点，确定关键的取证节点和信息提取方法。同时，设计合适的数据结构和算法，以确保取证溯源流程的有效性和可靠性。这一阶段将结合理论研究和实践经验，为后续的测试与验证奠定基础。

（3） 测试与验证

在设计实现取证溯源功能模块与流程后，将进行实验验证流程的可行性和有效性。通过模拟真实场景下的隔空投送传输过程，测试所设计的取证流程在不同情况下的表现和稳定性。同时，研究各种可能的结果，包括成功取证、取证失败以及数据不完整等情况，以进一步优化和完善取证溯源流程。这一阶段将通过实验数据和结果来支撑研究的结论和建议。

# 隔空投送传输原理

隔空投送的传输原理涵盖两大关键部分，即低耗蓝牙发现技术与WiFi局域网的应用原理。现针对这两大方面展开详细阐述。

2.1 低功耗蓝牙连续性协议

在AirDrop传输的初始阶段，设备会使用低功耗蓝牙（Bluetooth Low Energy，BLE）技术进行发现和建立连接。BLE技术具有低能耗、短距离、低延迟等特点，适合用于设备之间的快速发现和连接。通过BLE，设备可以在附近搜索其他支持AirDrop的设备，并建立起连接，准备进行后续的数据传输。

在BLE发出无线信号时，设备广播的短消息被称为广告数据包，用来向附近的设备宣布它们的存在和功能。广告数据包可以包括设备的名称、类型，也可以在“制造商特定”的字段中包括自定义数据。该字段通常由供应商用来为应用程序传输数据。苹果公司利用该字段来包含其设备中连续性协议的数据。

苹果公司开发了许多称为“连续性”的功能，旨在提高其产品的可用性。这些功能包括：活动传输、文件传输、Wi-Fi密码共享等。连续性服务所需的附近设备之间的通信就是通过BLE完成的，即将连续性数据嵌入BLE广告数据包中，并进行广播以供附近的设备接收。

在共享文件之前，设备通过BLE交换标识符来确定对方的身份，这些标识符通常是电子邮件地址或电话号码，也就是所谓的连续性数据。这些标识符不是明文发送的，而是使用加密哈希函数进行哈希处理。绕过这种混淆并恢复标识符即可得到用户的身份信息。当发送方使用AirDrop共享文件或图像时，他们将广播其电话号码哈希值的部分字节。虽然只有哈希的部分字节被广播，但在接收方的设备上提取出这些字节就足够来恢复完整的电话号码。攻击者可以创建一个数据库，其中包含其区域中每个电话号码的哈希值，通过暴力破解恢复明文。

2.2 Wi-Fi无线通信技术

一旦设备建立了连接并开始传输数据，AirDrop就会利用Wi-Fi技术进行实际的数据传输。在传输过程中，AirDrop使用了设备本身的Wi-Fi硬件，通过创建一个临时的无线局域网（Ad-hoc网络）来进行数据传输。这个临时的Wi-Fi网络允许设备之间直接通信，而无需经过路由器或者互联网连接。通过Wi-Fi，AirDrop能够实现高速、稳定的数据传输，适用于大文件和多媒体内容的传输需求。

遗憾的是，通过Wi-Fi传输的数据会被使用AES-256加密算法进行加密，这是一种高度安全的加密算法。这意味着，如果使用Wireshark抓包工具对无线传输过程进行监听，我们只能得到一堆难以解读的乱码数据。

以下是AirDrop工作原理图：

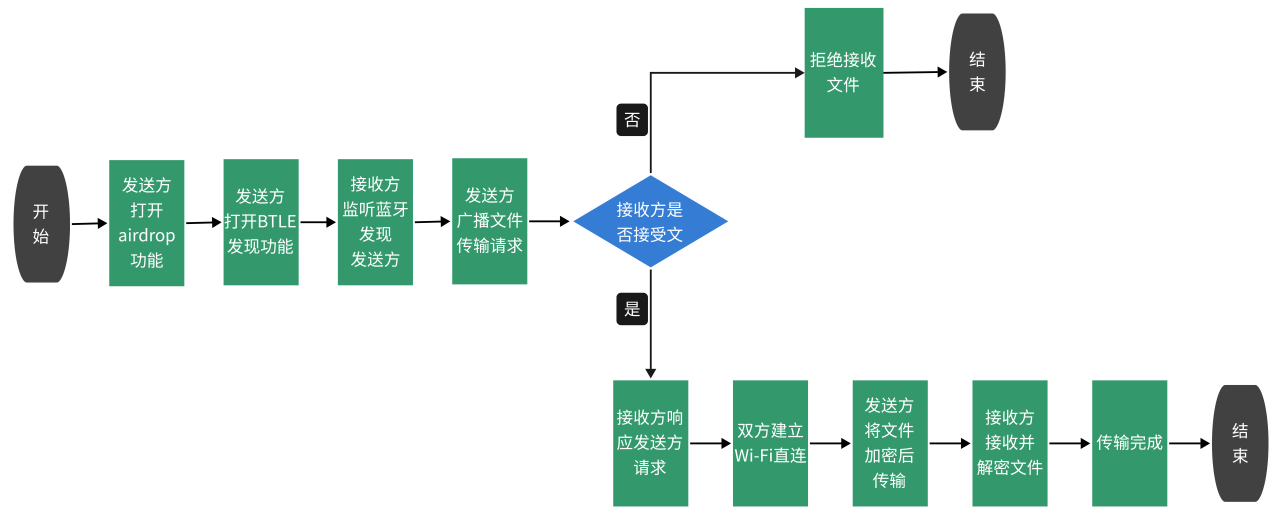


图1 AirDrop工作原理图

2.3 取证思路的提出

综合以上所述，经过对AirDrop传输原理的分析，我们不难得出结论：对AirDrop取证的最佳方式是通过设备的日志记录来进行，并且溯源的关键是利用蓝牙发现阶段的日志找到发送方的身份信息。在实际的侦查取证工作中，这一信息只能通过报案人即接收方的手机来获取。

另外，PacketLogger是macOS系统提供的一种工具，用于捕获和记录网络数据包。相比于Wireshark的优势是，它可以用于捕获AirDrop传输过程中的蓝牙发现阶段的数据，以帮助分析和调查相关问题。然而在研究过程中，我们发现PacketLogger并不能够掌握比统一日志更多信息，且在实战中，无法利用网络抓包工具进行实时取证，这里就不再赘述。因此，取证工作的核心是对接收方日志的收集与对AirDrop相关日志的提取，并且通过解析获取到的发送方用户信息实现溯源。下面将对于日志收集与提取模块与证据信息解析模块的进行设计。

# 隔空投送取证模块设计

3.1 日志收集与提取模块

3.1.1 统一日志分析与libimobiledevice工具

我们不妨选择使用macOS设备收集iOS设备上的日志记录，这是由于苹果引入了一种便于访问与分析日志的统一日志系统（Unified Logging System，ULS），即操作系统中的日志记录系统，用于记录系统的各种活动和事件方便开发者和安全人员进行安全审计。这种统一的、结构化的日志记录机制，允许以一致的方式访问和分析这些日志。此外，ULS还提供了命令行工具和图形界面工具（控制台），用于检索、查看和分析系统日志。

libimobiledevice是一个开源项目，提供了计算机与iOS设备通信的库和工具集合。它允许我们通过USB连接来访问iOS设备的文件系统、进行应用程序安装和调试等操作，也提供了一些用于访问设备日志和信息的工具，这些工具可以帮助我们取证过程中收集和分析iOS设备的统一日志。

3.1.2 手动日志收集过程

在安装好libimobiledevice工具包的macOS系统上可以在终端（Terminal）窗口使用命令“idevice\_id-l”来查看所连接的设备的[UDID（UniqueDeviceIdentifier），UDID是一个由40个字符组成的十六进制序列，用于唯一标识一台苹果设备。](https://zhuanlan.zhihu.com/p/668680307" \t "https://www.bing.com/_blank)然后使用日志收集命令将iOS设备的日志导入一个文件中。具体命令示例为：“sudo log collect --device -udid=<UDID> --last10m --output iPhone.logarchive”。其中根据接收方手机收到隔空投送信息的大致时间可以设置日志收集的时间范围，例如示例中的“--last10m”代表收集最近十分钟以内的日志记录。最后使用“open iPhone.logarchive”命令打开搜集到的日志，并利用搜索栏的过滤功能来找到我们所需要的蓝牙发现阶段的日志记录。结果如图2所示：

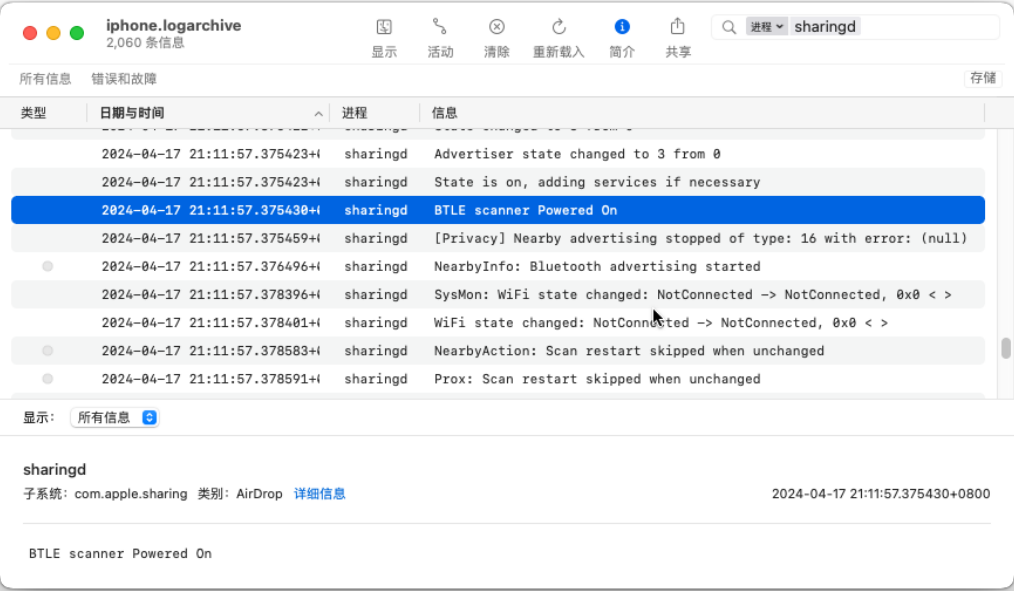


图2 iOS设备的日志搜集结果

3.1.2 python脚本实现

为了减少操作、提高效率，除了手动通过可视化的界面来进行日志收集，我们还可以通过编写python脚本来实现自动化的日志提取。即实现取证溯源工具中的日志收集与提取模块，经过分析，这一模块我们仅提取与蓝牙子系统中包含哈希值的信息即可。一种方法是使用libimobiledevice中的idevicesyslog工具，代码示例如下：

1. import subprocess
2. import re
3. def extract\_AirDrop\_btle\_logs():
4. #使用idevicesyslog工具获取iOS设备的系统日志
5. command='idevicesyslog'
6. result = subprocess.run(command,shell=True,capture\_output=True,text=True)
7. #解析日志
8. logs = result.stdout.split('\n')
9. btle\_logs=[]
10. for log\_entry in logs:
11. #使用正则表达式匹配
12. match = re.search('(.+)BTLE(.+)',log\_entry)
13. if match:
14. btle\_logs.append(match.group(1))
15. return btle\_logs
16. if\_\_name\_\_=="\_\_main\_\_":
17. AirDrop\_btle\_logs = extract\_AirDrop\_btle\_logs()
18. print("AirDrop BTLE Logs:")
19. for log in AirDrop\_btle\_logs:
20. print(log)

3.2 证据信息解析模块

通过手动分析日志，我们可以从接收方设备可以获取到：接收时间、接收状态、接收保存路径、发送方设备名、发送方相关邮箱Hash、发送方相关电话Hash等信息。

我们的以“王五的iPhone”和“张三的iPhone”两台手机作为实验对象。如图4所示，在接收端日志中过滤包含“AirDrop”的信息，可以得到对发送方相关信息的记录结果，其中“Email=[aa2b0...a94f0]”的方括号中就是发送方电子邮箱对应哈希值的开头和结尾部分，同理，“Phone=[e947c...a5ed0]”方括号之间就是发送方手机号的部分Hash值。

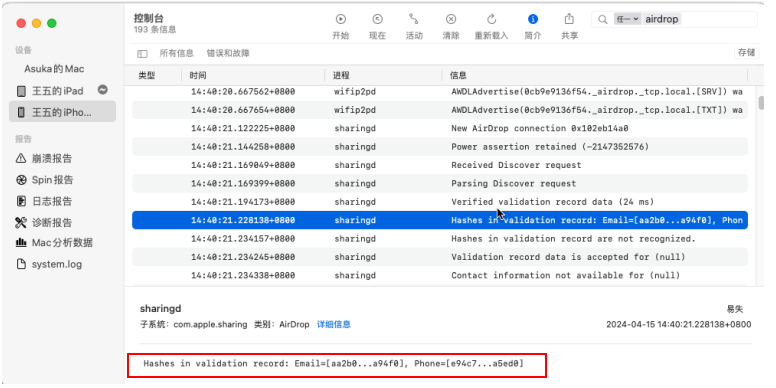


图3 在Mac控制台查看“王五的iPhone”的日志

当“王五的iPhone”作为接收方时，得到的是隔空投送发送方“张三的iPhone”信息的哈希值。我们再将“张三的iPhone”作为接收方，由“王五的iPhone”发送空投信息，得到的结果如图5所示，不同的是，这一次并没有发现发送方的电子邮箱或手机号的哈希值，这是由于发送方设备并没有插入SIM卡。关于发送方“Phone Hash”为空的各种情况我们将在下一章中讨论。

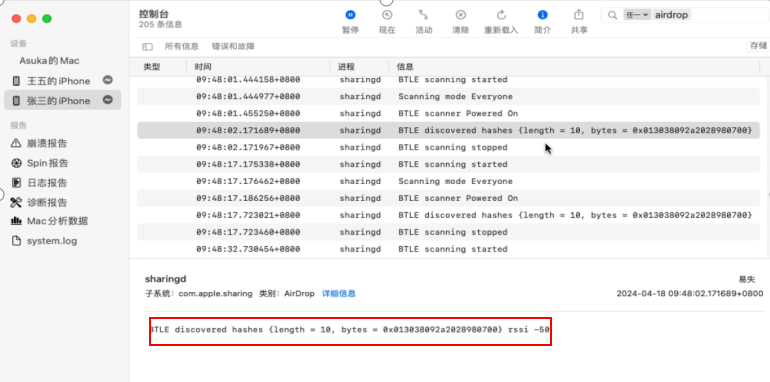


图4 在Mac控制台查看“张三的iPhone”的日志

结合图4、图5的结果，进一步对两部手机的用户信息进行哈希值计算发现，苹果使用SHA256（Secure Hash Algorithm 256bit）来计算电话、邮箱的哈希值，计算结果如表1、表2所示。

表1 用户Apple ID与对应哈希值

|  |  |  |
| --- | --- | --- |
| 设备名称 | Apple ID（电子邮箱） | SHA256-Hash（十六进制表示） |
| 张三的iPhone | huangzan695@gmail.com | aa2b0f4c5c00273f2a1676b56fdbdefa1ca3bab7b3732373d618432c2d1a94f0 |
| 王五的iPhone | 1282442146@qq.com | 79D337499BECE420DDDDAF860B901E67F8143AC36DC05549FD38588701866650 |

表2 用户手机号与对应哈希值

|  |  |  |
| --- | --- | --- |
| 设备名称 | 手机号（中国大陆） | SHA256-Hash（十六进制表示） |
| 张三的iPhone | 8616563288777 | e94c70acbc1afe4a02bc82c0b739452430ccac042bb4896d5e8154da2faa5ed0 |
| 王五的iPhone | 8617863291968 | 43a8835e9c668c81ba6ca57babcf0177576ba3b57b2c4eb2ffe6388e04097a82 |

下面介绍SHA256算法计算哈希值的特点，并介绍将哈希值破解，恢复成明文的途径。

3.2.1 SHA256算法

SHA256算法是一种常用的哈希算法，用于生成256位长度的哈希值。SHA256算法通过对输入数据进行一系列复杂的计算和处理，产生一个唯一的256位二进制数字，通常表示为64个十六进制字符。SHA256算法具有三个特性：唯一性，不同的输入数据通常会生成不同的SHA256哈希值；固定长度，输出的SHA256哈希值是固定长度的256位二进制数，通常表示为64个十六进制字符；抗碰撞性，SHA256算法在目前已知的攻击下，具有很高的碰撞阻力，即不同的输入数据很难生成相同的哈希值。

SHA256算法的具体实现可分为如下五个步骤：

首先，对数据进行填充，使其长度模512余448，即填充至长度为64位的倍数减去64位。这里使用1后跟n个0进行填充。其次，在填充之后的消息末尾添加一个64位的块，这个块包含了原始消息的长度值。接着，算法初始化8个32位的哈希缓冲区（即所谓的a,b,c,d,e,f,g,h），这些值固定且为硬编码的初值。然后，处理每一个512bit的消息块：将其分为16个32bit的字，并使用64个定义好的常量进行64次迭代操作，这些操作基于非线性函数对这些字进行混合与替换。最后，将64次迭代操作得到的输出与初始哈希值累加。在所有消息块处理完成后，这8个32位的缓冲区最终的值就形成了256位的SHA256哈希值。

3.2.2 暴力破解与彩虹表攻击

理论上，哈希函数被设计成单向函数，即从哈希值反推明文应该是不可能的。此外，SHA256具有高度的抗冲突性，即找到两个不同的输入但输出相同哈希值（称为碰撞）的概率极低。因此，从数学上破解SHA256几乎是不可能的。

相对于破解随机大型数据集的SHA256哈希值，破解特定区域特定格式的手机号码成功的可能性有所提高，在iOS系统中“隔空投送”取证溯源的实际应用中，可以通过暴力破解与彩虹表攻击方式尝试获取原始数据。

暴力破解是指尝试所有可能的输入组合以找到匹配的哈希值。彩虹表是一种预先计算的哈希值数据库，可以用于加速破解过程。对于特定格式的数据（如手机号码），可以预先生成一个针对该格式数据的彩虹表。如果破解者拥有一个目标SHA256哈希值，并且知道这是一个特定区域的手机号码生成的，那么使用彩虹表可能会有所帮助。

3.2.3 python脚本实现

通过暴力破解的方法得到发送方手机号需要用到python标准库itertools中的permutations函数，用于生成可迭代对象的所有可能的排列组合，同时使用hashlib库中的SHA256函数来执行SHA-256加密算法。脚本示例如下：

1. from hashlib import SHA256
2. from itertools import permutations
3. #目标哈希值
4. target\_hash = "目标SHA-256哈希值"
5. #定义一个生成所有手机号排列的函数
6. def generate\_phone\_number\_permutations():
7. #以"86"开头，后面跟着11位数字
8. prefix = "86"
9. digits = "0123456789"
10. for p in permutations(digits,11):
11. yield prefix+"".join(p)
12. #遍历所有手机号排列
13. for phone\_number in generate\_phone\_number\_permutations():
14. #计算当前手机号的哈希值
15. hashed\_phone\_number = SHA256(phone\_number.encode()).hexdigest()
16. #比较哈希值是否与目标哈希值相同
17. if hashed\_phone\_number == target\_hash:
18. #如果匹配，则输出手机号和哈希值，并结束程序
19. print("匹配的手机号:",phone\_number)
20. print("对应的SHA-256哈希值:",hashed\_phone\_number)
21. break

即使知道特定区域的手机号码格式，在这种情况下，暴力破解尝试的数量仍然是非常庞大的。所以采用预生成的哈希值与原始数据对照表，尝试寻找匹配的哈希值来恢复原始数据是更好的方法。简单的彩虹表攻击脚本示例如下：

1. from hashlib import SHA256
2. #已知的哈希值的开头五位和末尾五位
3. known\_prefix = "已知的开头五位"
4. known\_suffix = "已知的末尾五位"
5. #定义一个简单的彩虹表生成函数，用于生成指定范围内的手机号及其哈希值的开头五位和末尾五位
6. def generate\_rainbow\_table(start,end):
7. rainbow\_table = {}
8. for i in range(start,end+1):
9. phone\_number = "86165{:081d}".format(i)#生成以86165开头的11位手机号
10. #计算手机号的哈希值
11. hashed\_phone\_number = SHA256(phone\_number.encode()).hexdigest()
12. #仅保留哈希值的开头五位和末尾五位，构成彩虹表的键
13. hash\_key = hashed\_phone\_number[:5]+hashed\_phone\_number[-5:]
14. rainbow\_table[hash\_key] = (phone\_number,hashed\_phone\_number)
15. return rainbow\_table
16. #在指定范围内生成彩虹表
17. start\_number = 0#起始手机号的后8位
18. end\_number = 99999999#终止手机号的后8位
19. rainbow\_table = generate\_rainbow\_table(start\_number,end\_number)
20. #根据已知的哈希值的开头五位和末尾五位查找匹配的手机号
21. target\_key = known\_prefix + known\_suffix
22. if target\_key in rainbow\_table:
23. phone\_number,hashed\_phone\_number = rainbow\_table[target\_key]
24. print("匹配的手机号:",phone\_number)
25. print("对应的SHA-256哈希值:",hashed\_phone\_number)
26. else:
27. print("未找到匹配的手机号")

将代码中哈希值“已知的开头五位”和“已知的末尾五位”替换为从日志中发现的“e947c”和“a5ed0”即可顺利破解出“165”号段的手机号码明文，本质上是碰撞攻击，程序运行结果如图5所示。

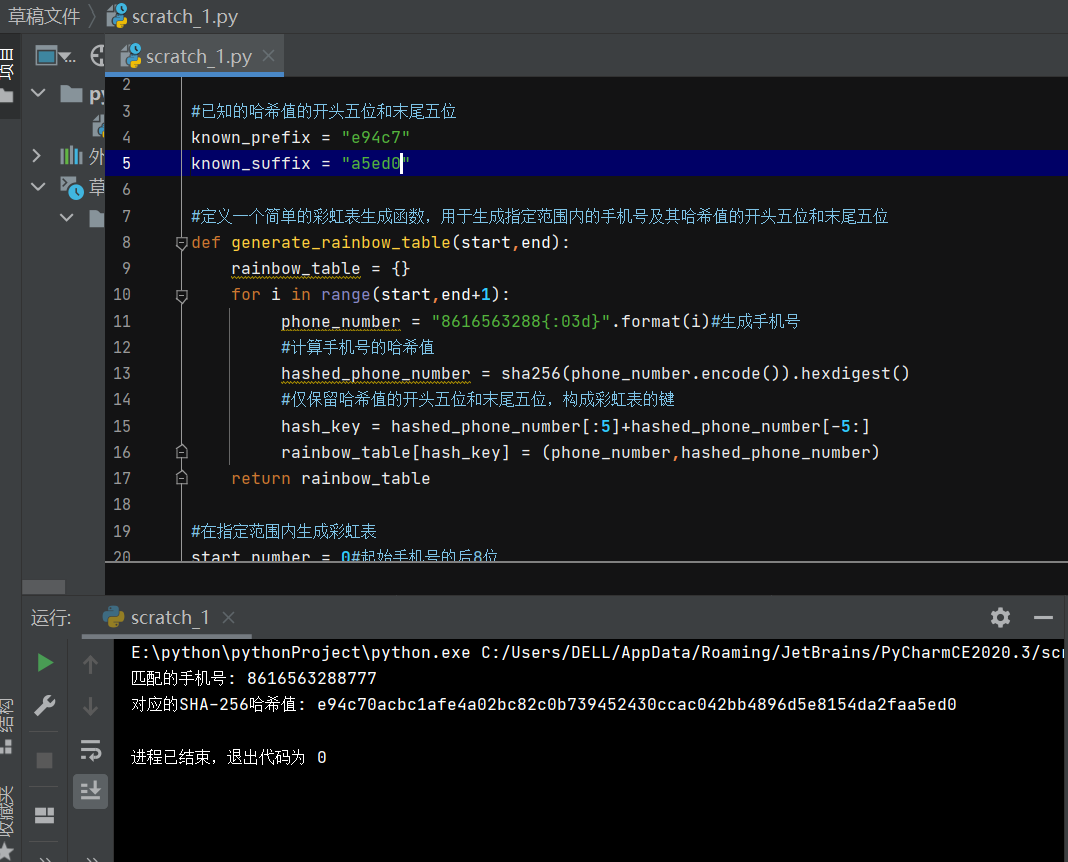


图5 彩虹表碰撞结果

创建彩虹表需要大量的计算和存储资源，对于SHA256，提高彩虹表破解哈希值成功率的核心在于优化彩虹表的生成过程和使用策略。对于破解手机号的哈希值，具体可以从以下几个方面进行优化。

使用更多的彩虹表:制作多个彩虹表，每个表使用不同的简化函数或初始密码集。

优化彩虹表的结构:对彩虹表进行结构优化，比如使用更高效的查找算法和存储结构，减少在查找过程中的时间开销和提高查找成功率。

精细构造简化函数:简化函数负责将哈希值转换成彩虹表中的下一个潜在密码，设计高效且能均匀覆盖所有可能密码的简化函数，是提高破解效率和成功率的关键。

3.3 总结

本章节基于隔空投送传输过程和通信原理，按照的二章中提出的iOS系统中隔空投送取证思路，完成了两大取证分析模块的设计。同时也遇到了一些问题，即在接收方提取出的日志不具有发送方用户Phone Hash的情况。接下来，我们将在第四章中进行具体的实验，模拟非法应用隔空投送功能扰乱公共秩序案件中的侦查取证工作，研究AirDrop日志中Phone Hash为空的各种情况并分析实验结果。

# 实验过程与结果

为了全面验证取证流程的合理性、实际取证工作中的可行性，本实验设计共分为三个部分，分别是对受害者手机的日志分析实验、对嫌疑人身份信息的哈希解密实验、以及关于取证过程中手机号码哈希值为空情况的研究。本次实验的软硬件环境如表3所示。

表3 实验设备信息

|  |  |  |
| --- | --- | --- |
| 类别 | 名称 | 信息 |
| iOS设备 | 张三的iPhone | 型号名称：iPhone6sPlus  操作系统：iOS13.6.1 |
| 王五的iPhone | 型号名称：iPhone6s  操作系统：iOS15.7 |
| Mac虚拟机 | Asuka的Mac | 操作系统：macOS Sonoma版本14.4.1  开发环境：python3.9.6、PyCharm Community Edition 2024.1 |
| Windows主机 | Dell | 操作系统：Windows 10家庭中文版  开发环境：VMware Workstation 17Pro、python3.9.13、PyCharm Community Edition 2020.3.5x64 |

4.1 日志分析实验

在本次实验中，为更贴近实际侦查工作的需要，我们将着重分析来自接收方iOS设备的日志记录。首先，我们要确定所要溯源的目标发送过程的AirDrop ID，在设备的生命周期内AirDrop ID是不一致的，会一直发生变化。最近一段时间的的AirDrop ID可以在iOS设备上的/private/var/mobile/Library/Preferences/com.apple.sharingd.plist中找到，但由于一些设备的AirDrop功能不常用，导致这个plist中没有AirDrop ID。

图6所示的查询可以提供统一日志中仍保存的AirDrop ID。

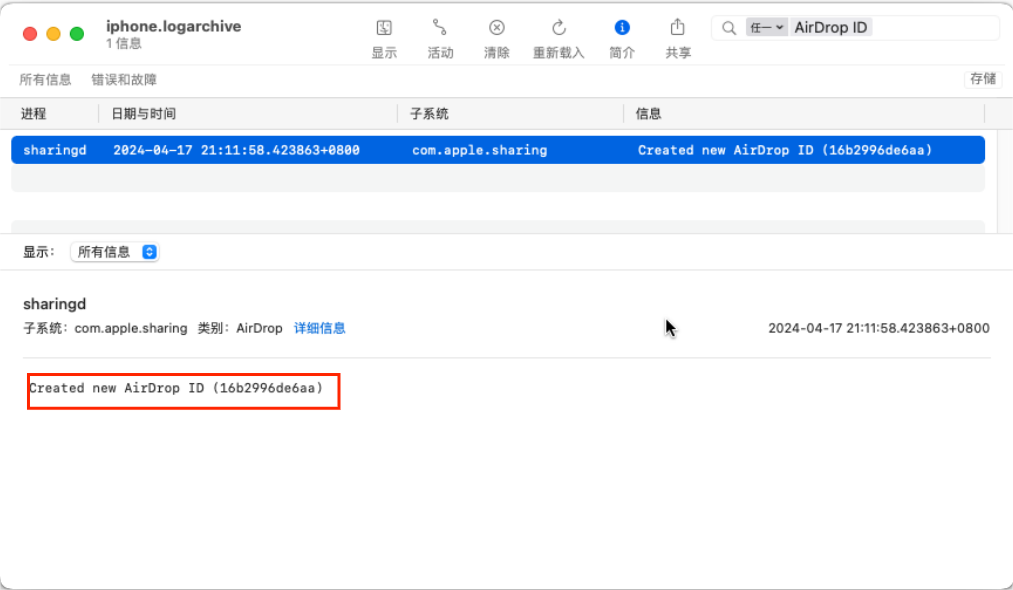


图6 查询日志中AirDrop ID信息

对于当时使用的是什么可发现模式，快速的查询方法是寻找"scanning mode"。如图7所示在接收方收到骚扰的案件中，接收方往往将其可发现模式设置为"Everyone"。

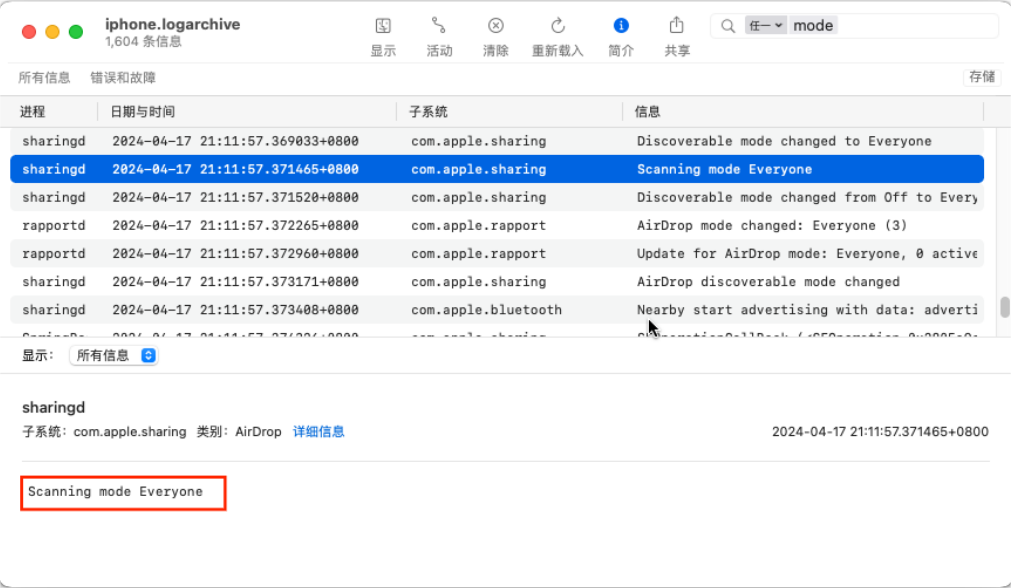


图7 查询日志中AirDrop可发现模式

对于一个传入的连接，AirDrop连接由一个标识符建立，如图8所示我们可以发现本次AirDrop连接的标识符是0x10dedc900。

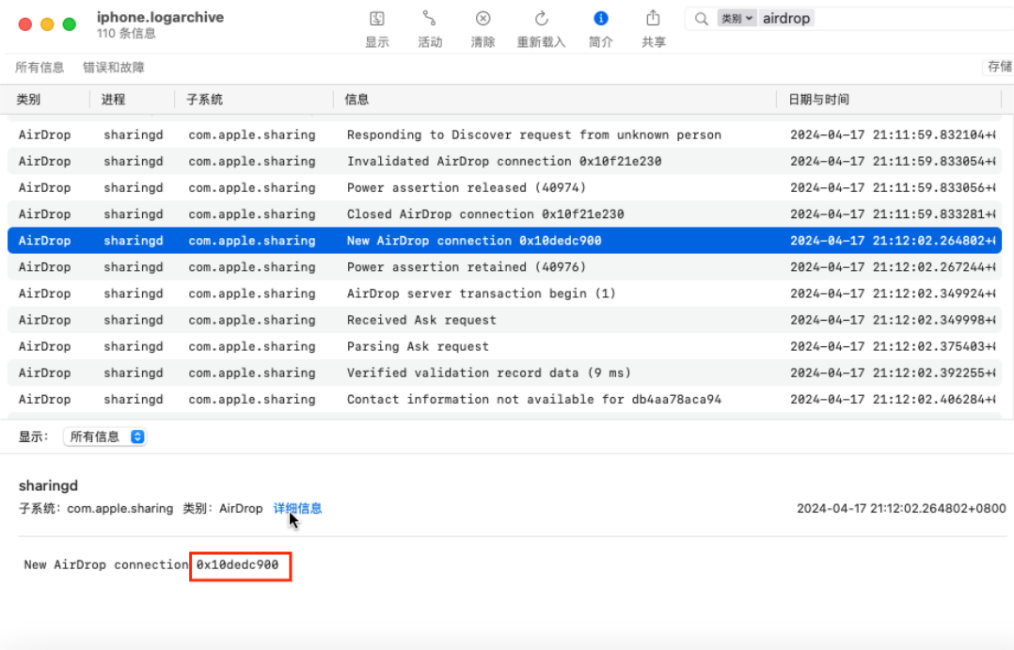


图8 AirDrop建立连接的标识符

在这个记录之后，有很多信息详细说明了被传输的是什么类型的文件以及它来自哪里。从图9中能够发现发送方设备名称是“王五的iPhone”，被传输的是JPEG格式的图像文件。

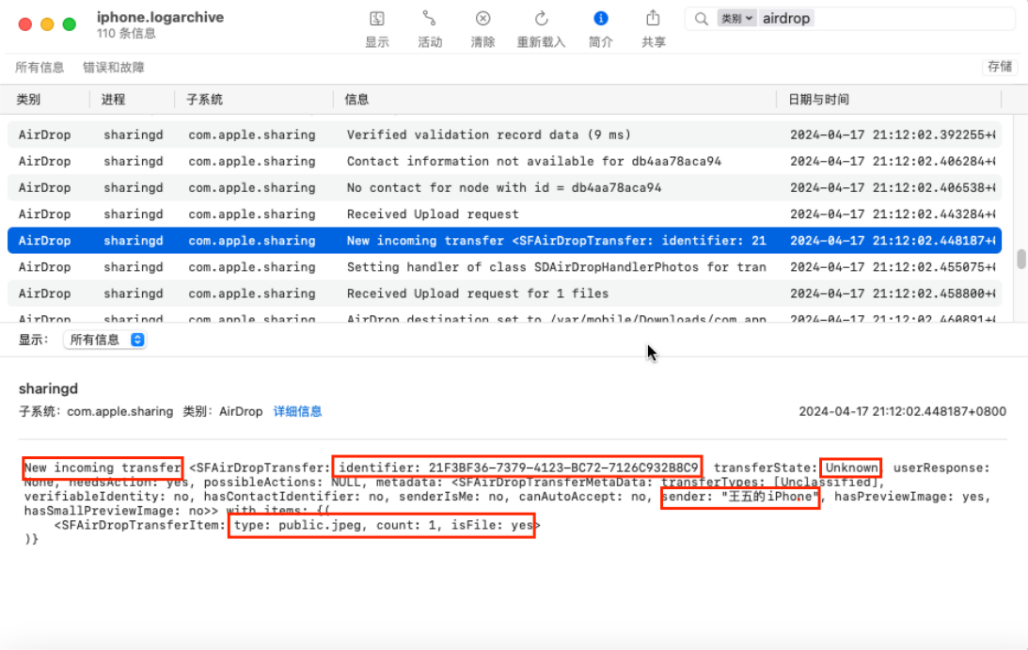


图9 AirDrop过程详细信息

接收方收到文件时可以选择接受或拒绝。这些选择都记录在统一日志，图10显示的条目是向用户展示的弹出式警报，以进行这一操作。要知道用户是否接受或拒绝了传输，我们需要关注“userResponse”部分。

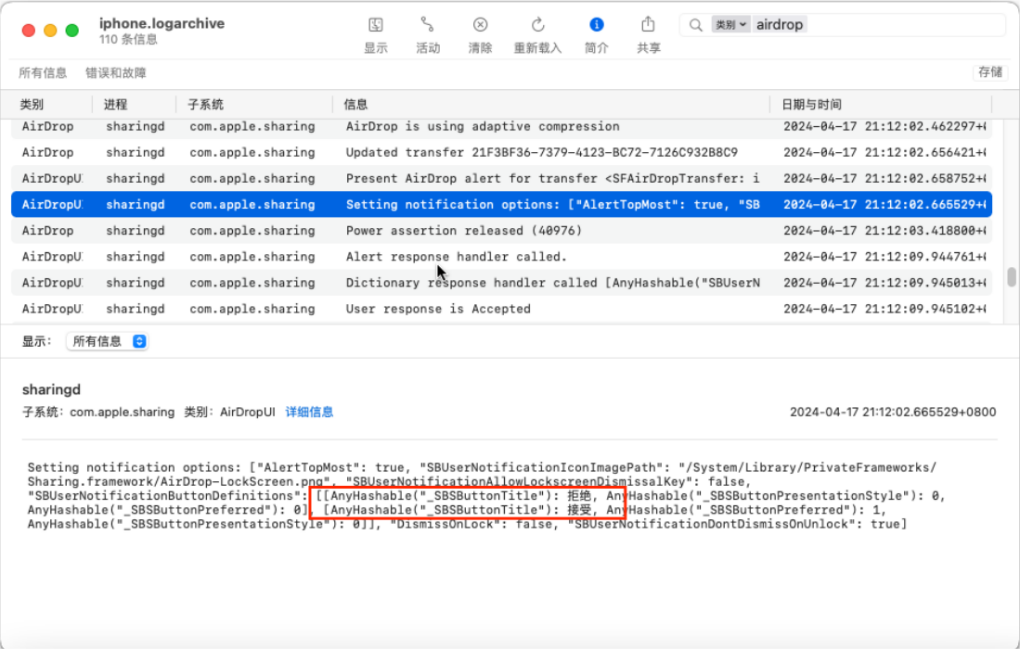


图10 AirDrop弹出式警报日志



图11 接收方响应情况

一旦用户选择了接受，传输就会继续，并在该文件的默认应用程序中打开。如图12中显示照片被导入并在照片（com.apple.mobileslideshow）中打开。一旦完成，隔空投送连接就终止了。文件路径为/var/mobile/Downloads/com.apple.AirDrop/21F3BF36-7379-4123-BC72-7126C932B8C9。如果用户拒绝AirDrop传输，同样的信息会显示拒绝或取消，AirDrop连接也会终止。

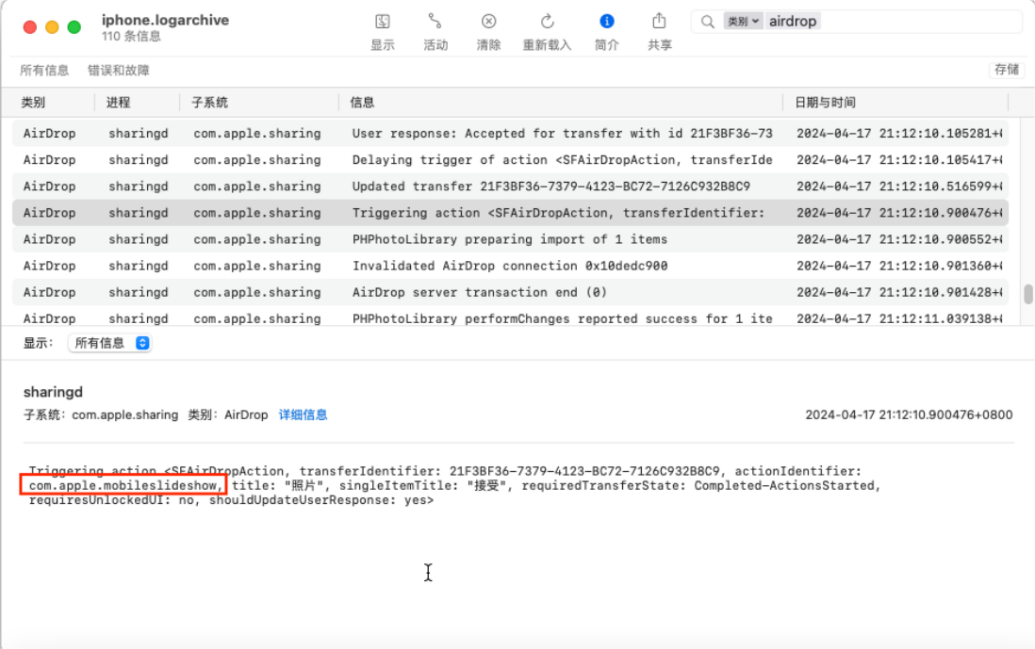


图12 文件在com.apple.mobileslideshow子系统中打开

4.2 哈希解密实验

通过上述手动分析日志，我们从接收方设备获取到：接收时间、接收状态、接收保存路径、发送方设备名等信息，完整的呈现了接收方的视角下收到隔空投送文件后的情况。根据第二章中对隔空投送原理的分析，我们可以得到发送方相关邮箱Hash、发送方相关电话Hash等信息，如图13是利用盘古石手机取证分析系统中的苹果日志提取工具从接收方日志中提取出的AirDrop哈希列表。接下来进行哈希解密实验即证据信息解析模块的测试。

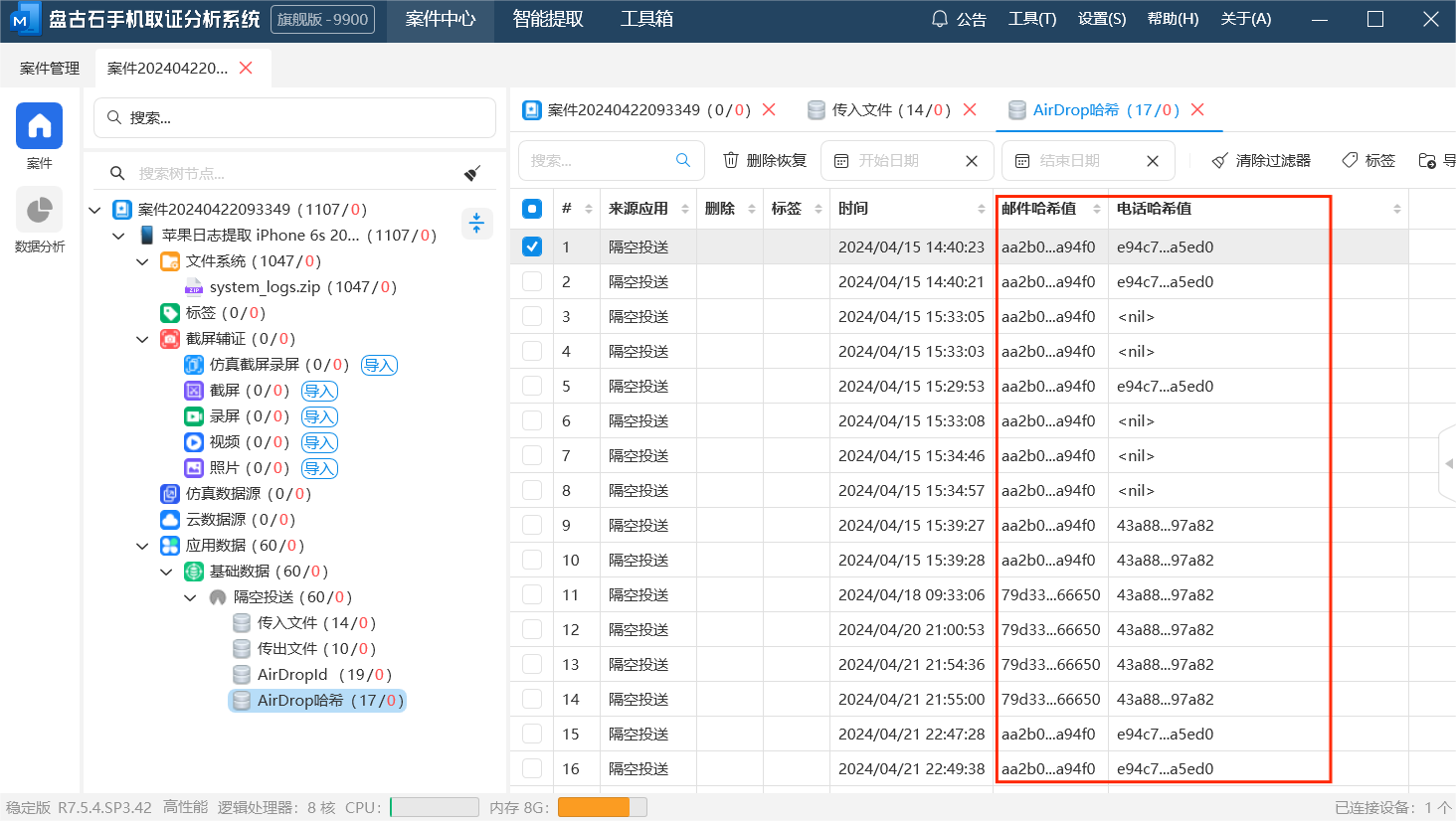


图13 接收方日志中的AirDrop哈希列表

将提取出的手机号哈希导入盘古石AirDrop哈希转换工具，如图14所示通过在线转换即可得出发送方的手机号明文。这里我们在第三章讨论过，对于特定格式的手机号码，可以预先生成一个针对该格式数据的彩虹表。当我们拥有目标SHA256哈希值，并且已知哈希值是特定区域的手机号码生成的，实验证明通过选择号段来缩小明文范围可以快速破解哈希值。



图14 将手机号码哈希转换为手机号明文

同理，当我们预先构造一个电子邮件地址列表，只要列表中包含了发送方嫌疑人的电子邮箱，就可以通过计算哈希值的方式破解出目标邮箱哈希值对应的邮箱明文。



图15 将邮箱哈希转换为邮箱明文

4.3 关于手机号码哈希值为空的研究

我们在第二章的研究中曾遇到这样一个问题，在接收方的日志中，并不总能发现发送方手机号的哈希值。图13所展现的AirDrop哈希列表中也出现了只有邮件哈希值而电话哈希值为空的情况。

首先，我们猜测电话哈希值是否为空，与手机是否装载SIM卡有关，只有当手机装载SIM卡时电话哈希值不为空。我们分别在发送方装载SIM卡与不装载SIM卡的状态下，以“张三的iPhone”为发送方，向接收方“王五的iPhone”隔空投送文件，并在接收方提取蓝牙发现阶段收到的广播的哈希值。如图15所示实验结果发现，当未插入SIM卡时，即使发送方“张三的iPhone”在Apple ID中手动设置了自己的手机号码，日志中也无电话哈希值记录。

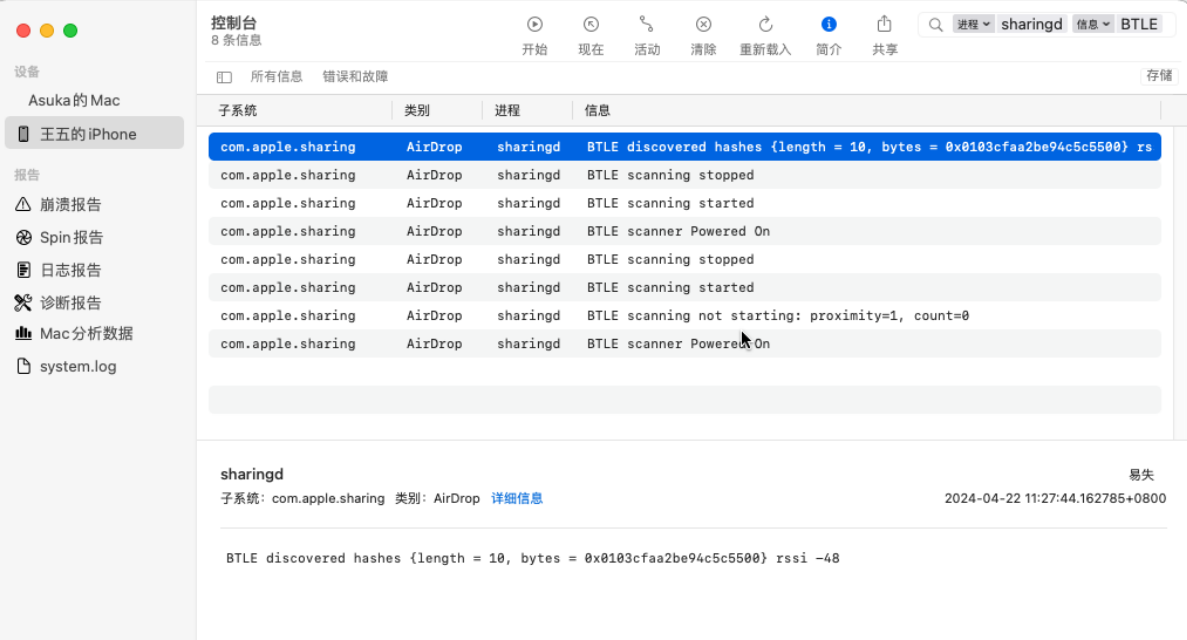
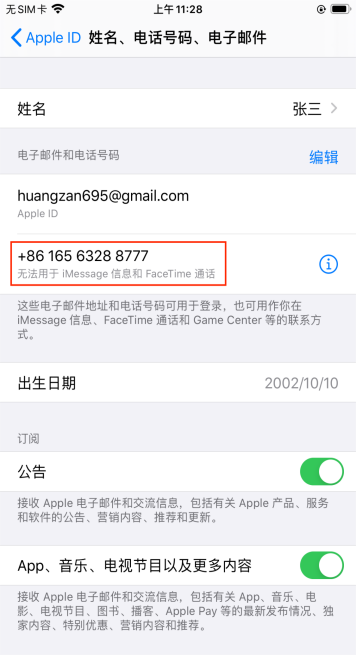


图16 发送方未插入SIM卡时的情况

将发送方“张三的iPhone”设备装载SIM卡后，再次测试，依然无法在接收方的日志中发现电话哈希值。这与我们的猜测不符。

经过进一步观察与思考，我们发现图16中，张三的Apple ID中手动添加的手机号下方有一段灰色字体——“无法用于iMessage信息和FaceTime通话”。这次我们猜测，电话哈希值是否为空与Apple ID中“姓名、电话、电子邮件”中“联络方式”的Facetime或iMessage的手机号码关联情况有关。

我们将手动编辑的手机号码删除，插入SIM卡，并打开iMessage与FaceTime。再次进行隔空投送，如图17所示，可以看到Apple ID中“姓名、电话、电子邮件”中“联络方式”自动关联了手机号码，灰色字体显示——“这些电子邮件地址和电话号码可用于登录，也可用作你在iMessage信息、FaceTime通话和GameCenter等的联系方式”。与此同时在Mac的控制台上从接收方日志中发现了发送方的电子邮件地址与电话哈希值。

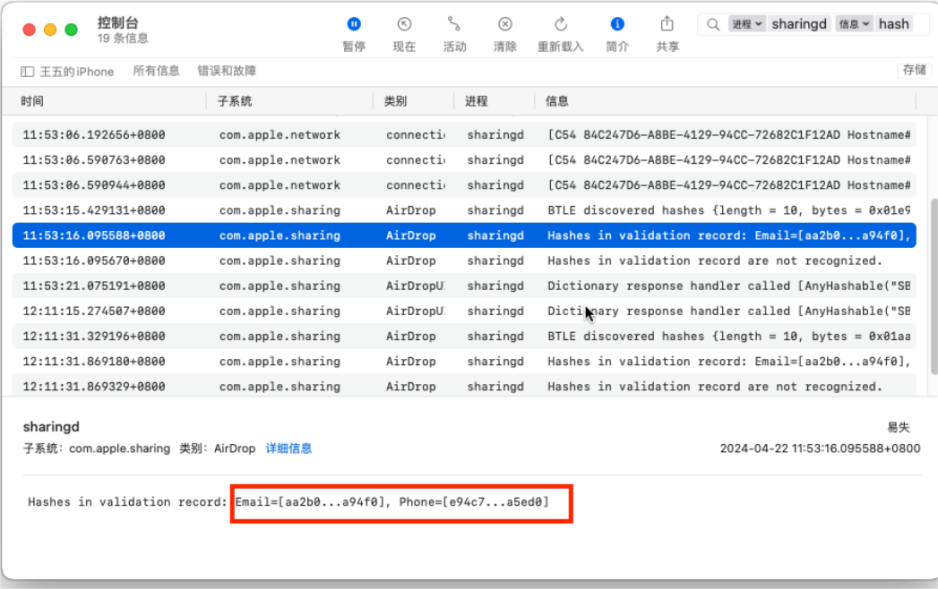


图17 发送方插入SIM卡且手机号与iMessage、FaceTime关联的情况

经过上述实验结果的深入分析，我们得出了一个重要结论：日志中电话哈希值的存在与否与手机上Facetime和iMessage功能的开关状态以及是否使用Apple ID登录具有直接关联性。在Apple ID的设置中，这种关联性具体表现为“联系方式”是否能够从Facetime和iMessage中自动提取手机号码并与之建立联系。若用户在Facetime和iMessage中使用Apple ID登录，那么日志中将会发现相应的哈希值。然而，值得注意的是，如果发送方用户在此之前手动编辑了“联络方式”，那么即使后续使用Apple ID登录Facetime和iMessage并实现了关联，日志中也不会发现哈希值。对于电话哈希值为空的情况，在实际取证中只能通过电子邮件哈希爆破，这种情况下的侦查工作具有一定的挑战性。

4.4 总结

本章主要围绕取证流程的合理性和实际可行性展开了一系列实验，以验证取证方法和工作流程的有效性。以下是对这些实验内容的总结。

首先，第一部分实验旨在分析受害者手机的日志信息，通过对手机日志的深入分析，模拟了侦查人员可以识别并提取的有价值信息，例如文件名、发送时间、接收方状态等，从而重建可能发生的事件或活动。在第二部分实验中，通过使用预先制作的彩虹表或其他解密工具，对哈希值进行破解，以获取嫌疑人的真实身份信息。这项实验旨在验证哈希解密技术在取证工作中的应用效果和可靠性。最后一部分实验着重研究了在取证过程中遇到手机号码哈希值为空的情况。通过以上实验，我们全面验证了取证流程的合理性和实际操作的可行性。这些实验不仅有助于理解和优化取证方法，还为实际取证工作提供了宝贵的经验和指导。

# 取证思源流程与应对建议的提出

5.1 取证溯源流程的提出

经过前三个章节的深入研究与实验，结合侦查机关的实际工作情况，我们提出了一套针对iOS系统中隔空投送骚扰案件的完整取证溯源流程。该流程旨在为侦查机关提供一套官方且高效的解决方案。

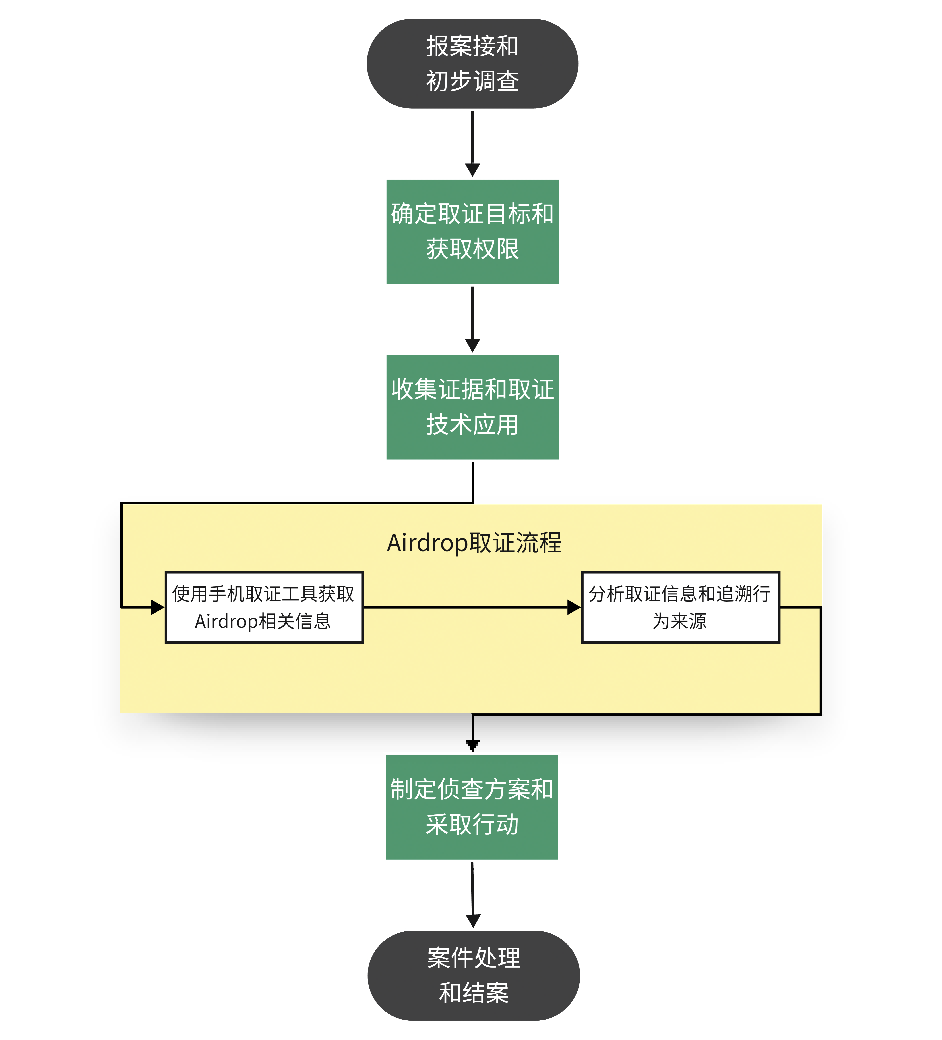


图18 iOS中“隔空投送”取证溯源流程图

如图18所示，首先，报案接收和初步调查。侦查机关在接收到关于iOS隔空投送骚扰案件的报案后，应立即记录报案信息，并指派专人对案件进行详细调查。通过初步调查，了解报案人所受骚扰的具体情况，包括骚扰内容、发生时间、地点等信息，并确保获得报案人的授权，以便进一步开展调查工作。

其次，确定取证目标和获取取证权限。根据初步调查的结果，侦查机关应明确取证目标，包括需要获取的设备信息、发送方身份信息等。为确保取证行为的合法性和有效性，侦查机关应申请相应的法律授权，可能涉及法院批准或其他法定程序。

接下来，收集证据和应用取证技术。侦查机关应获取报案人使用的iOS设备的相关信息，如设备型号、操作系统版本等。同时，如果可能的话，侦查机关应记录和保留骚扰内容的相关证据，如截图、消息记录等。此外，利用专业的手机取证工具或软件，连接目标iOS设备，提取与AirDrop相关的日志、系统记录和网络传输信息。

然后，分析取证信息和追溯行为来源。侦查机关应对收集到的取证信息进行深入分析，包括AirDrop的日志、发送方身份信息等。根据这些信息，侦查机关应努力追溯隔空投送的具体来源，包括发送方的设备信息、网络地址等。

在此基础上，制定侦查方案和采取行动。根据取证分析的结果，侦查机关应制定具体的侦查方案和行动计划，可能涉及调取相关证人、追踪设备位置等。在行动过程中，侦查机关应严格按照法定程序进行，如传唤相关当事人、调取网络日志或行踪等。

最后，案件处理和结案。根据调查结果，侦查机关应对骚扰案件进行妥善处理，采取适当的法律措施，如警告、罚款或起诉行为等。在结束调查后，侦查机关应编制完整的侦查报告，详细记录取证过程和结果，并提交给相关法律机构或法庭。

通过这一严谨、稳重且理性的取证溯源流程，我们期望为侦查机关在处理iOS系统中隔空投送骚扰案件时提供有力的支持和指导。

5.2 关于公民个人隐私的思考

在探讨公安机关获取蓝牙发送方信息是否侵犯个人隐私的问题时，必须明确两个基本点。首先，个人隐私权在现代法治社会中是受到法律严格保护的基本权利。然而，这并不意味着个人隐私权是绝对的，当公共利益或司法调查需要时，法律允许在特定条件下对个人隐私进行适当的侵入。但这样的侵入必须严格遵循法定程序，并限制在合理范围内。

在多数国家的法律体系中，对于电子数据的调取都设定了严格的法律程序。通常，警方需要依据相关法律，如犯罪嫌疑人的搜查和扣押程序，并经过司法部门的授权，才能进行电子数据的调查和收集工作。

在本文所讨论的情境中，如果蓝牙发送方通过隔空投送方式发送非法信息给接收方，这本身已经构成了对接收方合法权益的侵犯。这种行为显然违背了个人隐私和个人数据保护的原则。因此，为了维护更多公民的合法权益和社会秩序，公安机关有必要介入调查，以查明并惩处违法行为人。

在处理这类问题时，我们需要在个人隐私权与公共利益之间寻求平衡。如果发送者利用蓝牙隔空投送方式散播非法信息，损害他人权益，特别是在公共场所造成广泛影响，那么公安机关的介入调查是维护公共安全、公共秩序和他人权益的必要手段。此时，通过合法程序收集电子数据证据并不构成对个人隐私权的无理侵犯。

在收集电子数据证据的过程中，必须遵循最小侵害原则。这意味着在不影响调查工作的前提下，应尽可能减少对被调查者隐私权的侵扰。这就要求调查单位在调查过程中，必须严格保护与被调查案件无关的个人信息，防止其泄露或被不当使用。

因此，当公安机关依法采取措施，按照法定程序获取发送方的手机号码、电子邮箱等电子数据证据，以追查非法信息发送者时，只要这些行为遵循了适当的法律程序，并且在调查过程中尊重并保护了个人数据，那么这些行为既是必要的，也是合法的，不构成对个人隐私权的侵犯。

5.3 防止隔空投送功能被非法应用的应对建议

为确保信息安全并防止非法应用利用AirDrop进行传播，我们提出以下预防性措施建议：

首先，与苹果公司建立紧密的沟通协作关系至关重要。当用户接收到匿名发送的文件时，我们提议系统能够自动记录发送设备的相关信息，以便后续进行溯源追查。这将有助于我们更好地了解非法活动的来源并采取相应措施。

其次，针对用户个人行为，我们强烈建议苹果系统用户在不使用AirDrop功能时将其设置为关闭状态，或仅允许接收来自联系人的文件。此外，当不使用蓝牙、WI-FI等无线功能时，我们鼓励用户在系统设置中关闭这些功能，以降低潜在的安全风险。

对于非法文件的接收处理，我们要求用户一旦接收到疑似非法文件，应立即采取果断措施，包括但不限于删除文件、断开与发送者的联系，并及时向当地公安机关报告。这将有助于我们及时查处非法活动，维护网络环境的清朗。

此外，为了加强公共区域的管理，我们建议在关键聚集性和人员密集的场所，如商场、超市、地铁、公交等，安装具备AirDrop投送行为发现能力的探测器。这些探测器可以独立部署，也可以与现有的无线Wi-Fi等基础设施集成。通过这些探测器，我们能够及时发现AirDrop的非正常扫描及文件投送等行为，并及时部署相应的AirDrop应用监管平台。这将使我们能够实现对地区内非法AirDrop应用行为的实时和动态监测与管理。

综上所述，通过技术与管理双重手段的实施，我们将努力确保信息安全，降低非法应用通过AirDrop进行传播的风险。我们期望这些措施能够为用户提供一个更加安全、可靠的网络环境。

# 总结与展望

6.1 总结

随着移动互联网技术的迅猛发展和智能手机的广泛普及，iOS系统中的隔空投送功能作为一种高效且便捷的文件传输工具，已逐渐在广大用户中得到了普遍的应用与认可。此类面向陌生人的交流方式，既蕴含潜在的机会，亦伴随着一定的风险。部分怀有恶意意图的个体，可能会利用此平台传播非法图片、视频、音频等文件，甚至随意发泄个人情绪。本研究通过深入探索iOS系统中隔空投送的工作原理和传输方式，成功设计了用于隔空投送取证溯源的全流程示例。实验验证了取证方法和工作流程的有效性，为侦查取证工作提供了重要的技术支持和应对策略。本文进一步强调了公民对隔空投送行为的识别和防范重要性，提出了防止隔空投送功能被滥用的建议，旨在促进相关领域的研究和实践发展，为应对移动互联网时代面临的挑战提供了理论和实践支持。

6.2 展望

在隔空投送取证溯源方面的研究为未来的工作和发展提供了许多启示和方向。以下是对未来研究和实践的展望：

（1） 技术优化和工具开发：随着移动设备和iOS系统的不断更新，隔空投送功能可能会出现新的变化和特性。iOS17已经修复了低耗蓝牙发现协议中暴露用户信息哈希值的漏洞，未来的研究可以致力于优化取证过程中的技术方法和工具，确保能够有效应对新版本和新特性的隔空投送功能。

（2） 深入研究隐私保护机制：隔空投送涉及到用户个人信息和文件的传输，因此隐私保护是至关重要的。未来的研究可以进一步探讨隔空投送中的隐私保护机制，包括加密技术、身份验证和访问控制等，以确保用户数据的安全性和隐私性。

（3） 扩展取证范围和应用场景：隔空投送取证溯源不仅局限于传输文件的分析，还可以扩展到其他应用场景，例如社交互动、位置共享等。未来的研究可以探索更多的取证范围和应用场景，拓展隔空投送取证的深度和广度。

（4） 加强公众教育和意识提升：隔空投送作为一种便捷的文件传输方式，公众对其使用和风险认知不足。未来的研究可以通过加强公众教育和意识提升，帮助用户更好地识别隔空投送滥用行为，从而提高个人信息安全意识和防范能力。

（5） 跨学科合作与应用：隔空投送取证涉及到计算机科学、信息安全、法律等多个学科领域。未来的研究可以促进跨学科合作，结合不同领域的专业知识和技术手段，推动隔空投送取证研究的应用和实践。

隔空投送取证溯源的研究具有重要的理论和实践意义。未来的工作可以进一步完善取证技术和方法，加强隐私保护和公众教育，推动隔空投送取证在移动互联网时代的应用和发展，为保障个人信息安全和社会公共安全作出更大的贡献。

参考文献：

1. [Ben Lovjoy](https://9to5mac.com/author/benlovejoy/" \o "Posts by Ben Lovejoy). AirDrop security flaw can allow bad actors to see your phone number[EB/OL].(2019-8-1)[2024-3-16].https://9to5mac.com/2019/08/01/AirDrop-security-flaw/.

1. [Kinga Kieczkowska](https://kieczkowska.wordpress.com/).AirDrop Forensics[EB/OL].(2022-6-15)[2024-3-16].https://kieczkowska.wordpress.com/2020/06/15/AirDrop-forensics/.
2. Anonymous. [Analysis of Apple Unified Logs: Quarantine Edition[EB/OL].(2020-6-5)[2024-3-16].https://www.mac4n6.com/blog/2020/6/5/analysis-of-apple-unified-logs-quarantine-edition-entry-11-AirDropping-some-knowledge](https://www.mac4n6.com/blog/2020/6/5/analysis-of-apple-unified-logs-quarantine-edition-entry-11-airdropping-some-knowledge)
3. Mathieu Cunche. Apple devices are leaking sensitive data over BLE[EB/OL].(2019-12-5)[2024-3-16].https://team.inria.fr/privatics/apple-devices-are-leaking-sensitive-data-over-ble/.
4. 王晨语,薛亮.iOS平台的数字取证技术研究[J].数字技术与应用,2015,(02):49-51.
5. 刘浩阳.iOS设备取证技术[J].信息网络安全,2014,(09):167-170.
6. 许锘.iPhone手机在线取证研究[D].北京化工大学,2013.
7. 彭建新,周元建.iOS设备取证技术研究[J].中国人民公安大学学报(自然科学版),2012,18(04):37-41.
8. 仲利静,田雪梅,龙源,等.手机取证工具调研及质量评价[J].智能计算机与应用,2023,13(10):35-39.
9. 许正达.iOS平台数字取证系统的设计与实现[D].厦门大学,2019.
10. 陈佳霖,王轶骏,薛质.iOS系统数据安全研究[J].信息安全与通信保密,2012(08):100-102.
11. 宋波.[透视AirDrop：如何防范“隔空投送”骚扰? [EB/OL].(2023-8-24)[2024-3-16].https://](https://mp.weixin.qq.com/s/mpTbSughJZr6XN6opbS8Ow)mp.weixin.qq.com/s/mpTbSughJZr6XN6opbS8Ow
12. 赵飞,迟健.手机取证——关于iPhone手机数据提取方式的探讨[A].2019年全国公共安全通信学术研讨会优秀论文集[C].中国通信学会:2019:163-168.
13. 苏庆超. 基于iOS平台的手机取证系统的设计与实现[D].厦门大学, 2017.
14. Brandon E, Benjamin K, Derek F. Analysis of sysdiagnose in iOS 15 to identify the sending phone number of AirDrop data.[J].Journal of forensic sciences, 2022,67(4):1704-1707.
15. Wu H M, Chang C T, Min L Y. Digital Forensics Security Analysis on iOS Devices[J].Journal of Web Engineering,2021,20(3):775-794.
16. Bullock D, Aliyu A, Maglaras L, et al. Security and privacy challenges in the field of iOS device forensics[J].AIMS Electronics and Electrical Engineering,2020,4(3):249-258.
17. Koganti H, Rao N S G. Forensic Acquisition of iOS Devices[J].International Journal of Recent Technology and Engineering (IJRTE),2019,8(4):10847-10855.

附录：

附录A:图目录

[图1 AirDrop工作原理图](#_Toc2346) 6

[图2 iOS设备的日志搜集结果](#_Toc8775) 8

[图3 在Mac控制台查看“王五的iPhone”的日志](#_Toc15178) 9

[图4 在Mac控制台查看“张三的iPhone”的日志](#_Toc7654) 10

[图5 彩虹表碰撞结果](#_Toc7654) 13

[图6 查询日志中AirDropID信息 1](#_Toc18748)5

[图7 查询日志中AirDrop可发现模式 1](#_Toc31874)5

[图8 AirDrop建立连接的标识符 1](#_Toc4036)6

[图9 AirDrop过程详细信息 1](#_Toc21413)6

[图10 AirDrop弹出式警报日志 1](#_Toc14425)7

[图11 接收方响应情况](#_Toc15653) 17

[图12 文件在com.apple.mobileslideshow子系统中打开](#_Toc29995) 18

[图13 接收方日志中的AirDrop哈希列表](#_Toc4940) 18

[图14 将手机号码哈希转换为手机号明文 1](#_Toc15972)9

[图15 将邮箱哈希转换为邮箱明文](#_Toc14389) 19

图16 发送方未插入SIM卡时的情况 20

图17 发送方插入SIM卡且手机号与iMessage、FaceTime关联的情况 21

图18 iOS中“隔空投送”取证溯源流程图 23

附录B:表目录

[表1 用户AppleID与对应哈希值](#_Toc20073) 10

[表2 用户手机号与对应哈希值](#_Toc25041) 10

[表3 实验设备信息 1](#_Toc20320)4