

2021 年全国大学生信息安全竞赛

作品报告

作品名称： 面向无人机网络的轻量级无证书安全通信系统

电子邮箱： bbcczhang@126.com

提交日期： 2021.06.05

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。
2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。
3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。（本页不删除）
4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。
5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

面向无人机网络的轻量级无证书安全通信系统

摘要

嵌入式技术迅速发展使得微型智能设备无处不在，无人机的体积小、灵活易操作的特点使它广泛应用于各个领域。如今，无人机安全已上升为国家安全。但是，在无人机通信过程中由于无人机载荷和计算资源有限，无人机通信面临着如何平衡性能与安全的问题。通过对无人机通信特征和无人机网络拓扑环境的分析，本作品总结了无人机网络安全通信面临的主要问题，并设计了基于无证书公钥密码学的无人机轻量级安全通信方案。

该方案主要包括：预部署、密钥协商、密钥更新三个阶段。首先，在预部署阶段，无人机与 CA 分别生成无人机公私钥的一部分，无人机将两部分合并生成完整的公私钥。若双方无人机首次建立通信，则进入密钥协商阶段。双方无人机根据轻量级安全通信协议协商一个仅双方可知的会话密钥。当无人机双方再次通信时，系统进入密钥更新阶段。系统根据密钥协商阶段存储的初步密钥值直接计算最终会话密钥。双方无人机利用最终会话密钥实现安全的信息传输。由于无人机存储着海量密钥信息，本作品采用 MySQL 搭建数据库环境存储密钥信息，为密钥更新阶段提供更加快捷的查询服务。

本作品在无人机中实现了该方案，利用 GTK 实现了图形界面的开发，构建了完备的无人机轻量级无证书安全通信系统。经过对系统密钥协商各部分时间的测试，系统能够实现轻量级的无人机通信，密钥协商时间只需 2.655ms。同时，该系统可以防止认证中心信息泄露引起的假冒攻击。因此，本作品可以实现无人机网络的轻量级安全通信。

关键词：无人机通信安全、轻量级安全通信、无证书公钥密码、Socket 通信

目录

第一章 作品概述	6
1.1 背景分析	6
1.2 研究现状	8
1.3 本文的主要工作	9
1.4 前景分析	10
第二章 无人机网络通信安全分析	11
2.1 无人机安全通信系统的架构	11
2.2 无人机网络安全通信流程	12
2.3 无人机安全通信系统隐患分析	13
第三章 作品的设计与实现	14
3.1 系统架构	14
3.2 符号说明	16
3.3 无证书通信方案设计	16
3.3.1 预部署阶段的设计	17
3.3.2 密钥协商阶段的设计	20
3.3.3 密钥更新阶段的设计	23
3.4 系统前端设计	24
3.4.1 预部署阶段	24
3.4.2 密钥协商与密钥更新阶段	25
3.4.3 前端构建关键代码	25
3.5 系统设计流程	26
3.6 安全性分析	27
第四章 密钥信息的安全存储方法	29
4.1 密钥信息的特征	29
4.2 密钥信息存储架构选择	30
4.3 MySQL 数据库的设计	30
第五章 系统测试与分析	32
5.1 测试环境	32
5.2 安全通信流程测试	32
5.3 系统效率测试	36

5.4 CA 信息泄露安全性对比测试.....	37
5.5 测试结果分析.....	39
第六章 创新性说明	40
第七章 总结	41
参考文献.....	42

第一章 作品概述

本章阐述了本文的研究背景和研究现状，重点介绍了当前无人机的通信安全问题和保证无人机通信安全的策略，最后总结了本文的主要工作。

1.1 背景分析

无人驾驶飞机简称“无人机”，是利用无线电遥控设备和自备程序控制装置操控的不载人飞机。在数字信息时代的背景下，无人机技术和产业得到了快速发展，已成为世界各国促进国民经济与社会发展的重要推动力。2020 年，Zipline 等公司使用非接触式无人驾驶飞机在五个非洲国家的偏远地区运输 COVID-19 测试样品^[1]。无人机与行业的深度融合，使无人机广泛应用于智慧农业、抢险救灾、快递运输、军事国防等各个领域。

随着 5G 技术的迅速发展，无人机与 5G 技术的结合加速了无人机在各个领域的应用。根据 Gartner 预测，2020 年全球物联网企业无人机售出量达到 52.6 万架，同比增长 50%，预计到 2023 年将达到 130 万架^[2]。无人机为各个领域节省了人力、物力，减少了时间成本，提供了更加全面的监管和记录。图 1.1 预计了未来两年内全球前 5 名 IoT 企业无人机的售出量。

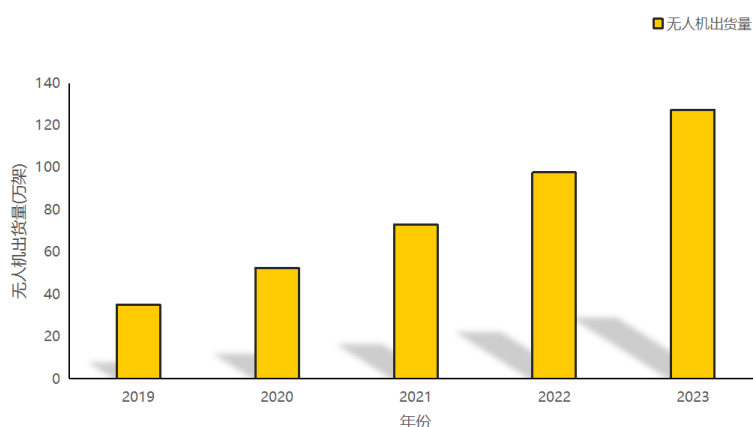


图 1.1 全球前 5 名 IoT 企业无人机售出量

目前，单个无人机因有限载荷、执行效率的限制，难以胜任复杂的任务。因此，网络化的无人机集群能极大程度上拓展无人机的应用范围与任务使命。在无人机网络中，无人机间主要以无线通信方式进行信息交互^[2]。受限于无人机载荷的影响，无人机安全通信主要面临着以下三个挑战：

首先，无人机网络的开放性使其易受攻击。在无人机通信过程中，无人机间发送的消息在开放的通信链路中传输。这些大量的信息很容易被攻击者利用 GPS 欺骗、信号拦截与篡改攻击等方式捕获。在 2016 年的“3.15”晚会上，黑客利用大疆无人机的无线通信安全漏洞，通过无线劫持技术完全取得了大疆无人机的控制权^[3]。图 1.2 显示了无人机通信过程主要遭受的攻击类型。

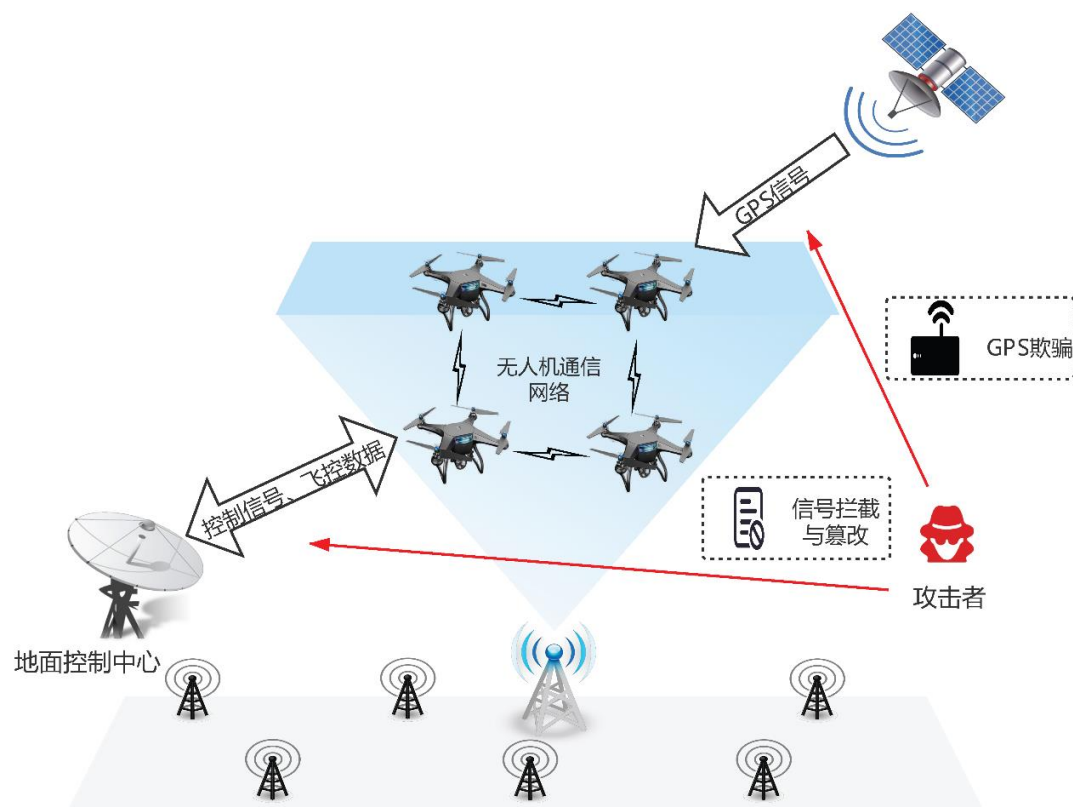


图 1.2 无人机网络常见攻击类型

其次，无人机的计算资源有限。无人机作为移动设备，主要由电池供电，其续航能力受到一定限制。部分方案为了保证无人机的通信安全，选用高负载的协议来认证无人机身份的合法性。这样不仅提高了无人机的能量损耗，而且影响无人机的正常工作。

另外，无人机无法抵抗认证中心（CA）信息泄露引起的假冒攻击。现有的许多方案中无人机密钥完全由一个可信的认证中心（CA）生成，这给密钥存储带来了极大隐患。2019 年，Trustico 公司将全部用户的私钥存储在自己的数据库中，这导致所有用户的证书被撤销^[5]。如果攻击者窃取了该公司的数据库，则可以假冒任意用户与其他用户进行通信，从而导致严重的安全问题。

现阶段针对无人机通信安全的研究主要结合身份认证技术，应用椭圆曲线数字签名算法、隐式证书等方案实现轻量级通信，但是对如何在保证轻量级通信的基础上抵御 CA 信息泄露引起的假冒攻击则研究的较少。

1.2 研究现状

现阶段已经有很多用于无人机安全通信的方案，常见的方案有以下三种：

- **基于 ECC 证书的密钥建立方案。**将证书作为可靠无人机的身份凭证，使用 ECDSA 技术对证书进行签名来验证无人机的身份，实现身份的可认证性^[6]。2019 年，Shuai 等人^[5]提出了一个使用 ECC 的 IOT 设备认证计划，然而，Fakroon 等人^[8]在 2020 年表明，该计划在计算和通信成本方面表现不佳，还遭受平行会话攻击、特权内部攻击。可以看到，基于 ECC 证书的密钥建立方案需要考虑计算负载提高、影响验证效率、存储空间增大、能耗增加等问题。
- **基于 ECQV 证书的密钥建立方案。**为了实现更加轻量级的身份认证，部分方案采用 ECQV 证书。Certicom Research 在 2013 年提出的基于证书的签密方案类型将依赖于 ECQV 隐式证书方案^[9]作为密钥管理协议，与基于 RSA 的公钥加密 (PKC) 系统相比^[10]，该方案使用椭圆曲线操作并产生更轻量级的 PKC 解决方案。在与方案一实现相同的安全等级下，这种技术可以降低一半的计算负载^[11]。但是上述所有方案的安全性完全依赖 CA，如果 CA 存储的私钥信息泄露，攻击者可以很轻易的与任意无人机建立通信。
- **无证书公钥密码方案 (CL-PKC)。**为了防止 CA 信息泄露引起的假冒攻击，很多研究者选用 CL-PKC 方案。2017 年，D.Q. Bala^[12]等人提出了使用 CL-PKC 对 IOT 智能设备进行相互认证的方案。然而，Malik M^[13]等人在 2020 年指出 CL-PKC 方案删除了传统 PKI 的计算开销，但加密公钥的成本依然很高。

上述方案的确能够保护无人机的通信安全，但是依然要消耗无人机的大量资源。同时，现有的方案^[14]在密钥协商期间需要时刻与 CA 建立连接，无人机的密钥在工作期间无法得到更新。如何实现一个“轻量级安全通信”方案，在不增加无人机能耗的前提下保护无人机通信安全。目前还没有给出明确的方法。

1.3 本文的主要工作

为了实现无人机的轻量级安全通信，本文基于无人机密钥生成、密钥协商、密钥更新等流程，提出了基于 CL-PKC 的轻量级安全通信方案，该方案可以实现轻量级的无人机密钥协商，且适用于资源受限的无人机设备，主要实现了以下内容：

首先，在不增加无人机能耗的前提下解决无人机轻量级通信问题。主要包括以下两点：1、CA 的间歇性连接。在整个安全通信的过程中，CA 只需参与无人机密钥生成阶段和密钥协商的开始时刻，无需参与全部过程。2、轻量级密钥更新。当两无人机建立通信时，无人机首先接收远端无人机发送的加密元素，计算一个初步的密钥值并存储在数据库中。双方再次建立通信时，可以直接利用之前存储的初步密钥值计算最终会话密钥。上述特性极大程度上降低了整个方案的计算负载，从而实现了轻量级通信。

其次，解决 CA 信息泄露引起的假冒攻击问题。主要原理是：无人机私钥的第一部分由无人机自己生成，第二部分由 CA 生成并发送给无人机。整个过程除了无人机外任何一方都不知道它的全部私钥信息，从而有效避免了攻击者假冒任意无人机进行非法活动。

最后，解决无人机轻量级安全通信方案的实现问题。本文在树莓派中实现了无人机的轻量级安全通信，并实现了系统部署。其次，根据无人机存储的密钥信息数量多、需要快速存取的特点，本文采用 MySQL 搭建数据库环境存储会话密钥，使用 SOCKET 通信技术实现整个通信过程中密钥信息的传输。最后，为了实现良好的用户体验，本文采用 GTK 实现 Linux 下的图形界面开发，便于用户进行使用。

图 1.3 展示了本文的主要架构：

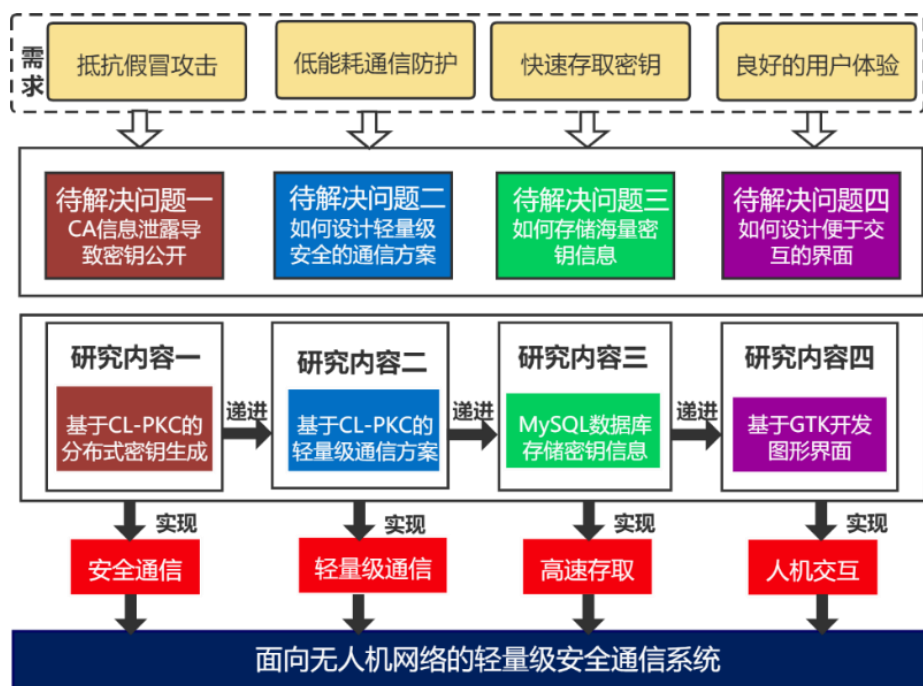


图 1.3 作品架构图

1.4 前景分析

无人机在各个领域都有着广泛的应用，具有体积小、造价低、使用方便、效率高等优点。在无人机网络的协同工作中，无人机之间的通信至关重要。但是在无人机通信工程中，无人机网络的开放性很容易使无人机遭受攻击。其次，无人机的计算资源有限，因此急需一种轻量级的方案保证无人机的通信安全。最后，某些公司将用户的私钥存储自己的数据库中，攻击者很容易通过数据库中存在的漏洞获取密钥。因此，保护无人机通信安全具有较强的实践价值。

本作品提供了一个无人机轻量级安全通信方案，可以在不增加无人机能耗的前提下实现无人机轻量级通信，同时能够防止 CA 信息泄露引起的假冒攻击。作品基于 CL-PKC 设计、MySQL 数据库存储数据，通过树莓派实现轻量级安全通信方案，利用 UDP 协议实现无人机间的信息传输，最后通过 GTK 实现图形化界面的开发。该系统适用于资源受限的无人机设备。

本作品可以应用于无人机、IoT 设备的轻量级安全通信，除了保证无人机网络的安全外，还可以减少通信对于无人机资源的消耗、有效抵抗 CA 信息泄露引起的假冒攻击，对于无人机安全领域有着很好的借鉴意义。

第二章 无人机网络通信安全分析

本章主要介绍了现阶段的无人机安全通信系统，包括无人机安全通信系统的架构和无人机安全通信的流程，并对无人机安全通信系统的隐患进行了分析。

2.1 无人机安全通信系统的架构

无人机安全通信系统广泛用于各个领域，但无人机安全通信系统的框架大体相同，本质都是期望通信的双方在身份认证安全的基础上协商会话密钥，并实现会话密钥的定期更新。

因此，一般的无人机安全通信系统主要包括如下模块：

（1）预部署管理：通过调用系统管理员部署的密码学元素，为 CA 生成公钥和私钥。同时在网无人机通过 CA 的公开参数生成唯一的公钥和私钥。

（2）认证管理：为保证入网无人机的安全性，CA 根据证书信息、签名信息等对无人机身份进行检查，避免假冒攻击。

（3）通信管理：为了实现安全的信息传输，无人机安全通信系统综合无人机负载状况、通信链路状况等因素选择合适的安全协议进行信息传输。

（4）存储管理：为实现无人机海量密钥信息的存储、提高密钥信息的查询效率，将无人机密钥信息存储在数据库中。

根据上面对无人机安全通信系统各模块功能的讨论，我们得出了一般的无人机安全通信系统所具有的功能图，如图 2.1 所示。

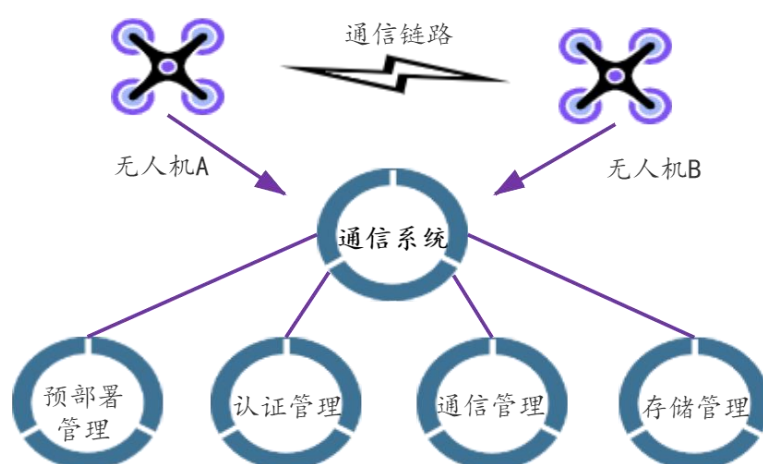


图 2.1 无人机安全通信系统功能图

2.2 无人机网络安全通信流程

无人机的安全通信在无人机网络中至关重要。一个安全的通信流程是指在无人机网络辐射范围内，根据无人机的密钥信息验证无人机身份，生成一个仅通信双方可知的会话密钥，并以最轻量级的方式实现会话密钥更新的过程。

现有无人机网络如果要想实现设备间的安全通信，主要包括以下两个步骤：

- (1) 生成会话密钥，即想要建立通信的无人机双方，按照系统部署的安全通信协议，生成会话密钥的过程。
- (2) 在双方期望再次进行通信时，如果当前会话密钥时间超出系统预设期限，依据密钥更新原则将重新生成新的会话密钥，完成信息传送。

在实际的无人机通信过程中，由于各无人机安全通信系统不同、通信协议不同、无人机功能多样，因而无人机安全通信的具体细节也不相同，但根据无人机安全通信的系统架构和主要步骤可以得出无人机安全通信一般流程，如图 2.2 所示。

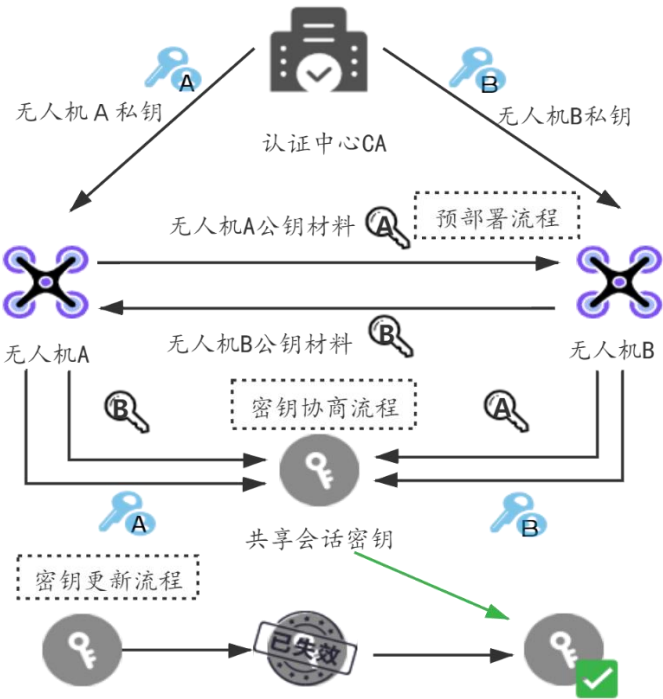


图 2.2 无人机安全通信的一般流程

根据无人机安全通信的主要步骤，可以将无人机安全通信大致分为三个主要阶段，即预部署阶段、密钥协商阶段和密钥更新阶段。

(1) 预部署：系统管理员首先为系统部署密码学材料。CA 与无人机分别生成自己的公私钥。无人机 A 首先向 CA 请求期望建立通信的无人机 B 的 IP 地址。无人机 A 在获取对方的 IP 地址后，通过 UDP 协议向目标无人机发送消息并建立通信。

(2) 密钥协商：双方无人机建立通信后，利用通信链路交换公钥信息。根据接收的对方无人机公钥及本无人机私钥等信息，双方计算一个相同的会话密钥，同时将该密钥保存在本地数据库中。

(3) 密钥更新：在双方再次期望通信时，系统判断当前会话密钥的有效性。如果密钥已失效，由系统的密钥更新机制对会话密钥进行更新，并自动替换当前数据库存储的密钥。

2.3 无人机安全通信系统隐患分析

无人机安全通信系统需要具有较高的安全性和较低的能量能耗，因此在研究无人机安全通信系统的隐患时，我们主要针对下面这三点进行分析。

首先，对于现有的部分安全通信系统，CA 存储设备的私钥信息。一旦 CA 被攻击者攻破，将造成大量的私钥信息泄露，攻击者利用私钥假冒网络中的任意无人机。

其次，在无人机通信过程中为了提高传输信息的安全性，选用高负载的协议生成会话密钥。对于无人机来说，这无疑加剧了无人机的能量损耗。

最后，为保证会话密钥的定期更新，需要在密钥更新阶段，完全重新计算会话密钥。由于缺少轻量级的更新机制，无人机的计算负载将会再次提高。

本作品的设计可以有效的解决上述情况下的无人机安全隐患和大量能源损耗问题，并且提升无人机的通信效率。

第三章 作品的设计与实现

本章针对无人机通信过程中的安全问题,设计了一个基于 CL-PKC 机制的无人机安全通信系统,该系统架构由无人机客户端和 CA 服务器端两部分组成,并介绍了系统的关键组成模块,最后对整个系统的流程进行了总结。

3.1 系统架构

本作品针对无人机通信的特征,提出了一种基于 CL-PKC 机制的安全通信系统 (Lightweight Secure Communication System based on Certificateless Public Key Cryptography, LSCS),来解决无人机的安全通信问题。

LSCS 主要包括预部署、密钥协商和密钥更新三个阶段,其中无人机与 CA 通过 UDP 协议进行信息交互。无人机的密钥信息存储在本地的 SQL 数据库中。方案的系统架构总体设计如图 3.1 所示:

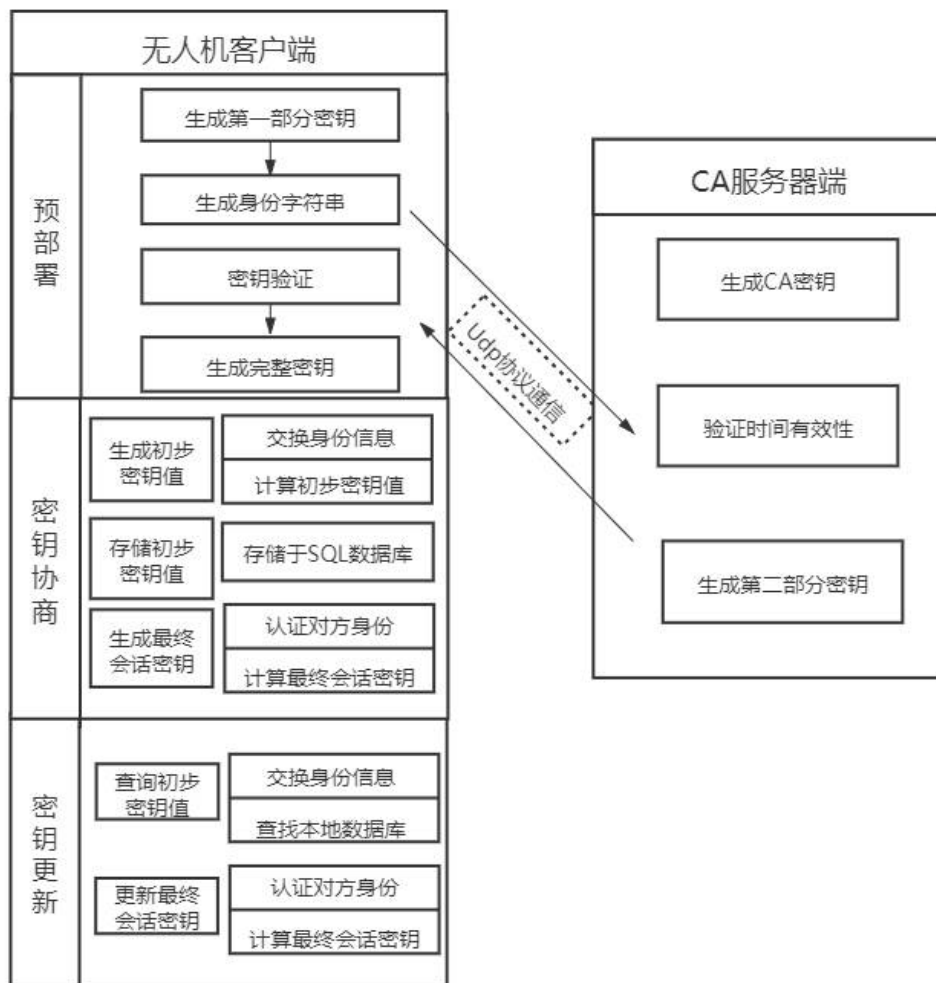


图 3.1 LSCS 的系统总体架构

系统主要包括以下两个部分：

➤CA 服务器端：

(1) 生成 CA 密钥：CA 根据系统管理员部属的密码学元素计算自己的公私钥。

(2) 验证时间有效性：CA 收到无人机身份字符串后，判断字符串是否有效，从而拒绝重放攻击。

(3) 生成无人机第二部分公私钥：CA 根据自身私钥生成无人机的第二部分公私钥，并通过 UDP 协议发送给无人机。

由于 CA 只保存无人机私钥的一部分，因此即使 CA 信息泄露给攻击者，攻击者仍然不能得到无人机的完整私钥。

➤无人机客户端：

(1) 预部署阶段：系统管理员首先部署必要的密码学材料。当一个无人机进入系统后，无人机与 CA 分别生成公私钥的一部分，最后将两部分拼接形成最终无人机的公私钥。

(2) 密钥协商阶段：如果两个无人机是首次建立通信，则启动密钥协商程序。该阶段分为生成初步密钥值、存储初步密钥值、生成最终会话密钥三部分。

(2-1) 生成初步密钥值：双方无人机交换身份信息，并根据对方的信息和本机私钥计算一个初步密钥值。

(2-2) 存储初步密钥值：无人机将(2-1)中计算的初步密钥值存储在本地 SQL 数据库中。

(2-3) 无人机根据(2-1)中接收的身份信息认证远端无人机的身份。认证成功后，双方计算一个唯一的最终会话密钥。

(3) 密钥更新阶段：如果两个无人机想要建立新的会话、密钥超过系统管理员设置的有效期或密钥泄露，则启动密钥更新程序。该程序主要包含查询初步密钥值和更新最终会话密钥两部分。

(3-1) 查询初步密钥值：在本地 SQL 数据库中查找与远端无人机初次建立通信时存储的初步密钥值。

(3-2) 更新最终会话密钥：双方无人机直接利用查找到的初步密钥值计算最终的会话密钥。

3.2 符号说明

符号表示	符号含义
ε	所选取的椭圆曲线组
G	椭圆曲线组 ε 的 n 阶生成元
F_q	素有限域
E/F_q	在素有限域定义的椭圆曲线
(c, C)	CA 的私钥和公钥
ID_i	无人机 i 的标识字符串
n_i	无人机 i 选取的随机数
t_i	无人机 i 密钥有效期
(x_i, X_i)	无人机 i 的第一部分公私钥
(p_i, P_i)	无人机 i 的第二部分公私钥
K_{ji}, K_{ij}	无人机 i 与无人机 j 的初步密钥值
H	Hash 函数
ψ	HMAC 函数
φ	KDF 函数
sk	初步会话密钥
σ_i	无人机 i 生成的消息认证标签

3.3 无证书通信方案设计

无证书公钥密码学（Certificateless Public Key Cryptography, CL-PKC）最早是由 Al-Riyami 和 Paterson^[15]最初构想的一类加密方法。它旨在克服现有的基于身份加密（IBC）方案的安全限制。在 IBC 方案中，系统中所有实体的私钥生成过程都完全由可信的机构即密钥生成中心（KGC）控制。因此，掌握 KGC 泄露的秘密信息的手对手可以完全冒充任何实体，而无法被发现。然而，CL-PKC 可以很好的解决这一问题，CL-PKC 方案中设备的私钥由设备和 CA 共同生成^[16]，主要分为以下几个阶段：

- ①初始阶段。此阶段由 CA 执行，CA 生成自己的公钥PK和私钥SK。

② 部分密钥生成阶段。CA 根据公钥PK、私钥SK 和请求实体的身份字符串 ID，为请求实体生成第一部分私钥d。

③生成密钥阶段。首先，请求实体根据 CA 公钥PK、身份字符串 ID 和随机数n，生成第二部分密钥x。其次，请求实体接收 CA 发送的第二部分私钥，根据PK、x，生成请求实体的全部私钥sk。最后，请求实体定义它的全部公钥，根据PK、x，生成实体的公钥pk。

3.3.1 预部署阶段的设计

预部署阶段负责的是无人机公私钥的生成。无人机公私钥的生成是不同无人机之间进行通信的基础，通信双方凭借本机私钥和远端无人机密钥信息建立会话密钥，进行消息传递。

在此阶段中，本作品采用 CL-PKC 的思想，实现了分布式无人机密钥生成方案，保证即使 CA 数据泄露，攻击者仍无法得到无人机的完整密钥，具体流程如图 3.2 所示：

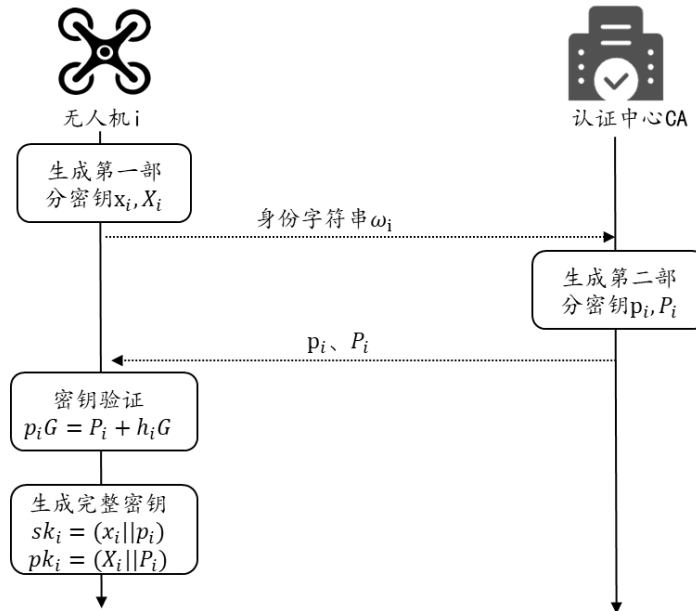


图 3.2 预部署阶段具体流程

预部署阶段主要有以下几个步骤：

Step1: 系统管理员部署如下公开参数：椭圆曲线群 \mathcal{E} 、椭圆曲线群的 n 阶生成元 G 、素数域 F_q 上定义的椭圆曲线 E 、SM3 哈希函数 H 、KDF 密钥派生函数 φ 。

Step2: CA 生成自身的公私钥。CA 随机选取一个 k 位素数 q ，利用哈希函数计算自己的私钥 $c = H(q)$ 。之后，CA 计算自己的公钥 $C = cG$ 。

Step3: 首先，无人机 i 选取随机数作为私钥的第一部分 x_i ，通过椭圆曲线组生成第一部分公钥 $X_i = x_i G$ 。其次，该无人机 i 生成两个元素：字符串 ID_i 和 t_i ，其中，字符串 ID_i 唯一标识此无人机，与公钥 X_i 唯一绑定。 t_i 代表密钥的有效期。无人机将 ID_i 、 t_i 、 X_i 合并为字符串发送给 CA。

$$w_i = (ID_i || t_i || X_i) \quad (1)$$

Step4: CA 接收到无人机 i 发送的字符串 w_i 后，首先检查时间戳 t_i 的有效性。若该字符串 w_i 已过时，则将其丢弃，因此可以拒绝恶意攻击者的重放攻击。CA 保存无人机 i 的 ID_i 及通信的 IP 地址。

Step5: CA 选取随机数 r_i ，计算其在椭圆曲线上的投影 $P_i = r_i G$ ，将 P_i 作为无人机 i 的第二部分公钥。CA 计算无人机 i 的第二部分私钥 p_i ，并将 P_i 、 p_i 发送给无人机 i 。

$$\begin{cases} h_i = H(w_i || P_i) \\ p_i = r_i + h_i c \bmod n \end{cases} \quad (2)$$

用于生成无人机 i 第二部分公私钥的关键代码如下图所示。由于 SM3 为国密算法且安全性较高，因此 Hash 函数选用国密 SM3 哈希算法。

```

//CA选择随机值r并计算P=r*G
BN_rand(r, 160, -1, 1);
EC_POINT_mul(group, P, r, NULL, NULL, ctx);

//CA计算h=H(w||P)，使用SM3哈希算法
str = EC_POINT_point2hex(group, *P, 2, NULL);
EVP_DigestInit_ex(md_ctx, md, NULL);
EVP_DigestUpdate(md_ctx, w_P, strlen(w_P));
EVP_DigestFinal_ex(md_ctx, h, &h_size);

//CA计算p = r + h * c(mod n)
EC_GROUP_get_order(group, n, ctx);
BN_mod_mul(temp, *h, private_key_c, n, ctx);
BN_mod_add(temp, r, temp, n, ctx);

```

图3.3 生成无人机第二部分公私钥的关键代码

Step6: 无人机 i 收到 CA 发送的第二部分公私钥后，使用下列公式可以验证第二部分公私钥的真实性，避免公私钥被攻击者篡改：

$$p_i G = P_i + h_i C \quad (3)$$

Step7: 若密钥验证成功，则认为密钥是由 CA 生成且未被篡改。因此，无人机 i 将两部分密钥将其进行拼接即得到完整的公钥 pk_i 、私钥 sk_i ：

$$\begin{cases} sk_i = (x_i || p_i) \\ pk_i = (X_i || P_i) \end{cases} \quad (4)$$

上述过程中，无人机 i 的第一部分私钥存储在无人机本地且第一部分公钥 X_i 与其身份字符串 ID_i 唯一绑定，因此，即使 CA 生成的无人机第二部分私钥信息泄露，攻击者仍然不能假冒该无人机。

3.3.2 密钥协商阶段的设计

密钥协商阶段是 LSCS 系统的关键部分。LSCS 利用 elliptic curve Diffie Hellman (ECDH) 方法建立初始会话密钥。ECDH 是一种密钥交换方法，是 ECC 与 DH 的结合，它使得通信的双方能在公开的信道中安全的交换密钥，用于加密后续的消息。

在密钥协商过程中，通信双方通过两步生成最终会话密钥。初步密钥值是密钥协商阶段的中间值，存储在无人机的本地 SQL 数据库中，通过初步密钥值计算得到最终会话密钥。通信双方利用最终会话密钥对数据进行加解密实现安全通信。

在该阶段中，通信双方会交换四条不同的消息。前两条消息用于无人机间交换密钥材料，并利用 ECDH 算法生成初始会话密钥。两台无人机通过后两条消息验证彼此的身份，并计算最终会话密钥。

以无人机 A、B 为例，图 3.4 为密钥协商阶段具体流程

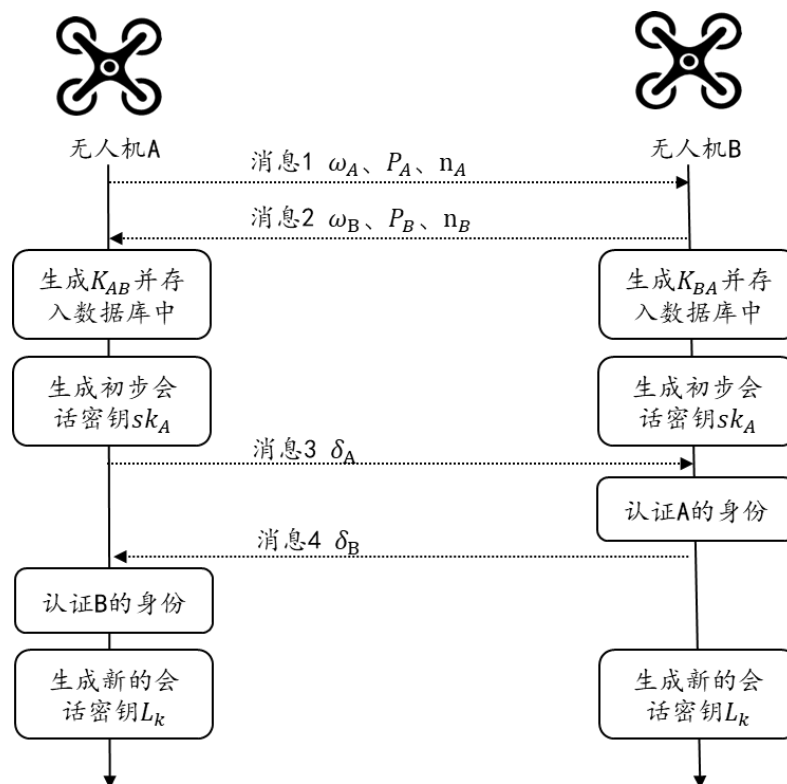


图 3.4 密钥协商阶段具体流程

下面介绍密钥协商阶段的具体实现方法：

Step1: 无人机 A 生成协议的第一条消息并发送给无人机 B, 该消息包括 w_A 、 P_A 、 n_A 。其中, w_A 唯一标识此设备, $w_A = (ID_A || t_A || X_A)$ 。 P_A 则供对方生成身份标签, n_A 是无人机 A 生成的随机数, 可以保证消息的新鲜性, 防止重放攻击

Step2: 无人机 B 接收到消息后, 检查 w_A 的时间有效性。若 w_A 有效, B 将回复该阶段的第二条消息, 即 $w_B || P_B || n_B$ 。其中, $w_B = (ID_B || t_B || X_B)$ 。

无人机 A 接收消息后, 同样判断时间有效性。若有效, 则无人机 A, B 的前两条消息交换完毕。

Step3: 无人机 A 计算初步密钥值 K_{AB} , K_{AB} 用于生成初始会话密钥 sk 。

$$\begin{cases} K_{AB,1} = p_A(P_B + H(ID_B || t_B || X_B)C) \\ \quad = p_A(r_B G + h_B cG) \\ \quad = p_A p_B G \\ K_{AB,2} = x_A X_B = x_A x_B G \\ K_{AB} = K_{AB,1} || K_{AB,2} \end{cases} \quad (5)$$

利用 KDF 函数 φ 对 K_{AB} 进行派生, 计算得到初始会话密钥 $sk = sk_{AB} = \varphi(K_{AB})$ 。 sk 用于对认证消息进行加密。其中。无人机 A 将生成的 K_{AB} 存储在本地数据库中, 供密钥更新阶段使用。

同样, B 计算初步密钥值 K_{BA} , 将其存储在本地数据库中:

$$\begin{cases} K_{BA,1} = p_B(P_A + H(ID_A || t_A || X_A)C) \\ \quad = p_B p_A G \\ K_{BA,2} = x_B X_A = x_B x_A G \\ K_{BA} = K_{BA,1} || K_{BA,2} = K_{AB} \end{cases} \quad (6)$$

由于 $K_{BA} = K_{AB}$, 故 B 也可以得到相同的初步会话密钥 sk_B 。

该过程的关键代码如下:

```

//生成初步密钥值 $k_{AB}$ 

EC_POINT_mul(group, temp_d, NULL, public_key_C, h, ctx);

EC_POINT_add(group, temp_d, P, temp_d, ctx);

EC_POINT_mul(group, temp_d, NULL, temp_d, p, ctx);

EC_POINT_mul(group, temp_d, NULL, temp_d, p, ctx);

K = EC_POINT_point2hex(group, temp_d, 2, NULL);

EC_POINT_mul(group, temp_d, NULL, X, x, ctx);

strcat(K, EC_POINT_point2hex(group, temp_d, 2, NULL));

```

图 3.5 生成初步密钥值 K_{AB} 关键代码

Step4： 无人机 A 准备认证消息。无人机 A 生成认证标签： $\sigma_A = \psi[sk, (w_A, P_A, w_B, P_B, n_A, n_B)]$ ，其中 ψ 是 HMAC。A 将认证标签使用 sk 进行加密，并发送给 B。

B 接收到 A 的认证消息后，通过重新计算认证标签 σ'_A ，验证对方的身份：

$$\sigma'_A = \psi[sk, (w_A, P_A, w_B, P_B, n_A, n_B)] = \sigma_A \quad (7)$$

若上式成立，无人机 B 相信确实与无人机 A 进行通信。若上式不成立，则中止会话。

无人机 B 也进行类似的操作。将准备好的认证标签 σ_B 加密后发送给 A，A 同样重新计算 σ'_B 来验证 B 身份的真实性。

该过程的关键代码如下

```

//生成A的认证消息： $\sigma_A = \psi[sk, (w_A, P_A, w_B, P_B, n_A, n_B)]$ 

strcpy(str, w_A); temp = EC_POINT_point2hex(group, P_A, 2, NULL);

strcat(str, temp); strcat(str, w_B);

temp = EC_POINT_point2hex(group, P_B, 2, NULL); strcat(str, temp);

temp = BN_bn2hex(n_A); strcat(str, temp);

HMAC(md, sk, strlen(sk), str, strlen(str), result, &size);

//A将自己的认证消息发给B

sendto(sockfd, result, size, 0, (struct sockaddr*)&addr_B, sizeof(addr_B));

```

```

//A对B进行认证
Auth_Message_generator(result, sk, w_B, P_B, w_A, P_A, n_B, n_A, group);

if(strcmp(result, Auth_Message_B) == 0)

    printf("A think B is credible\n");

else

    printf("A think B is not credible\n");

```

图 3.6 无人机验证对方身份关键代码

Step5: 双方验证身份后，无人机 A 和 B 生成一个仅双方可知的共享会话密钥 L_k ，利用 L_k 进行安全通信。

$$L_k = \varphi(K_{AB} || n_A || n_B) = \varphi(K_{BA} || n_A || n_B) \quad (8)$$

3.3.3 密钥更新阶段的设计

密钥更新阶段是系统的最后一个环节。当两个无人机首次建立通信时，由于之前从未交换过密钥信息，因此完全执行密钥协商阶段的所有流程。当两个无人机建立新的会话时，启动密钥更新阶段生成新的会话密钥。该过程包括同步密钥更新和异步密钥更新。

同步密钥更新：系统管理员基于无人机的应用场景，设计一个密钥的更新周期，定期更换会话密钥。此外，当系统管理员发现某无人机会话密钥泄露时，立刻启动异步密钥更新程序。

异步密钥更新：双方无人机在本地数据库中查询之前存储的初步密钥值，若存在，则交换身份材料，计算最终的会话密钥。由于最终会话密钥的生成仍有新的随机数的参与，因此可以保证会话密钥的新鲜性，如图 3.7 所示：

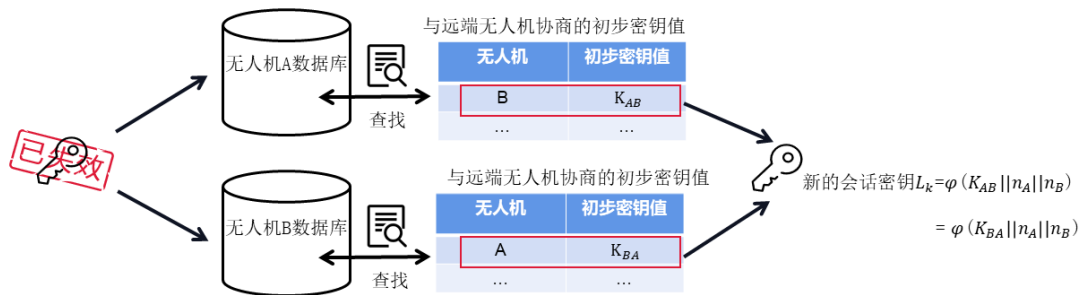


图 3.7 密钥更新阶段具体流程

从方案设计的角度，两无人机调用 `Get_data` 获取初步密钥值，通过 `socket` 通信实现认证消息的传输，该阶段关键代码如下：

```
//得到数据库中的数据
Get_data(mysql, res, query_str);

//发送与接收随机数
char *n_str = BN_bn2hex(n);

sendto(sockfd, n_str, strlen(n_str), 0, (struct sockaddr *)&addr, sizeof(addr));

recvfrom(sockfd, n_str, 256, 0, (struct sockaddr *)&addr, addrlen);
```

图 3.8 无人机查询数据库并交换身份材料关键代码

3.4 系统前端设计

前端主要负责将无人机轻量级无证书安全通信系统以图形界面的形式呈现，方便用户使用和管理人员进行系统维护，并且能够清晰地展现无人机安全通信系统的各个阶段。

GTK+（Gnome Toolkit）是一套跨多种平台的图形工具包，现已发展成为一个功能强大、设计灵活的通用图形库。GTK+被 GNOME 选中担当 GNOME 桌面的基础[18]，成为 Linux 下开发图形界面的应用程序的主流开发工具之一。GTK+具有设计良好、灵活、可扩展和简单易用等优良特性。

本作品的前端设计与后端轻量级安全通信方案一一对应，同样分为预部署，密钥协商，密钥更新三个阶段，且分别为 CA 和无人机设计了前端界面。下面将对三个阶段的前端设计进行详细说明。

3.4.1 预部署阶段

预部署阶段的前端界面主要包括 CA 地址绑定、无人机请求连接和密钥显示三部分。

无人机安全通信系统启动后，首先会出现 CA 的信息输入界面，系统管理员绑定 CA 的 IP 地址和端口号。在输入完成后，后端会同时生成系统所必要的密码材料。

在 CA 地址绑定完毕后，进入无人机信息输入界面，无人机输入 CA 的 IP 地址请求与 CA 建立连接。CA 在接收到连接请求后，检查该无人机设备是否已进

行过预部署。若没有进行预部署，则在后端 CA 与无人机协同生成无人机的公私钥。

在无人机的公私钥生成完毕后，前端页面显示无人机的公私钥信息。预部署阶段到此结束。

3.4.2 密钥协商与密钥更新阶段

在预部署阶段结束后，无人机前端界面显示当前已入网的无人机，用户选择想要进行通信的无人机设备的 ID。后端 CA 通过检索数据库向用户发送远端无人机的 IP 地址，并判断通信双方是否为首次通信。若为首次通信，则进入密钥协商阶段，否则进入密钥更新阶段。

在密钥协商阶段，系统后端启动密钥协商阶段计算通信双方的会话密钥，并显示在无人机前端界面。

在密钥更新阶段，系统后端检索数据库得到初步密钥值，进而生成最终会话密钥并显示在无人机前端界面。

通信双方的会话密钥生成完毕后，整个无人机安全通信系统操作执行完毕。前端逻辑结构图如图 3.9 所示：

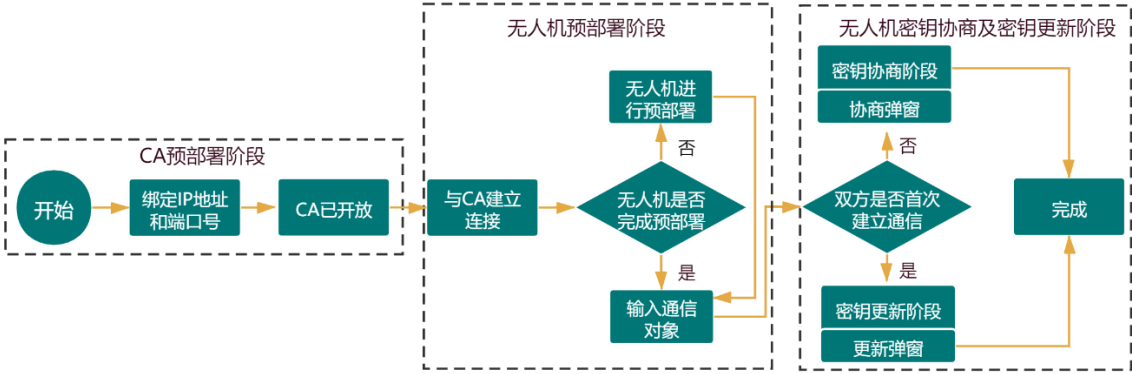


图 3.9 前端逻辑结构图

3.4.3 前端构建关键代码

无人机和 CA 的前端界面构建主要包括：页面切换、页面内部布局、页面信息显示。

前端使用 GTK_NOTEBOOK 模块模拟页面的切换。首先构建数据结构体，利用 gtk_notebook_append_page()生成新页面，gtk_notebook_next_page()用于切

换页面。之后，构建 button 按钮，当 button 被按下时通过 g_signal_connect()携带数据结构体进入 window2 窗口。

页面信息显示的关键代码如下所示：

```
/*构建窗口间所要传输的数据*/
struct data{
    GtkWidget *window;
    GtkWidget *page;
    const BIGNUM *XXX;
};
struct data *DATA = (struct data *)malloc(15000);
DATA->window = ((struct data *)entry)->window;
DATA->XXX = XXX;

/*新建下一窗口*/
GtkWidget *window2 = gtk_box_new(GTK_ORIENTATION_VERTICAL, 0);//建立垂直布局
gtk_notebook_append_page (GTK_NOTEBOOK (((struct data *)entry)->page), window2, label);
gtk_widget_show_all(((struct data *)entry)->window);
gtk_notebook_next_page(GTK_NOTEBOOK (((struct data *)entry)->page));

/*当 button 被“click”时，携带 DATA 进入 window2*/
g_signal_connect(button, "clicked", G_CALLBACK(window2), (void *)DATA);
```

图 3.10 页面信息显示关键代码

3.5 系统设计流程

设计完三个阶段后，我们将系统架构图 3.1 补充完整，就可以得到 LSCS 系统的设计流程图，如图 3.11 所示。从流程图中，我们可以清晰的看到在各个阶段，无人机与 CA，无人机之间详细的交互过程，无人机只需要在预部署阶段和密钥更新的开始阶段与 CA 进行交互，在之后的阶段中无需 CA 的连接。在密钥协商阶段中，分两步生成会话密钥，并将初步密钥值存储在数据库中。在密钥更新阶段，无人机只需查询本地数据库即可得到初步密钥值，进而计算最终会话密钥。整个过程在没有增加无人机能耗的情况下实现了轻量级安全通信。

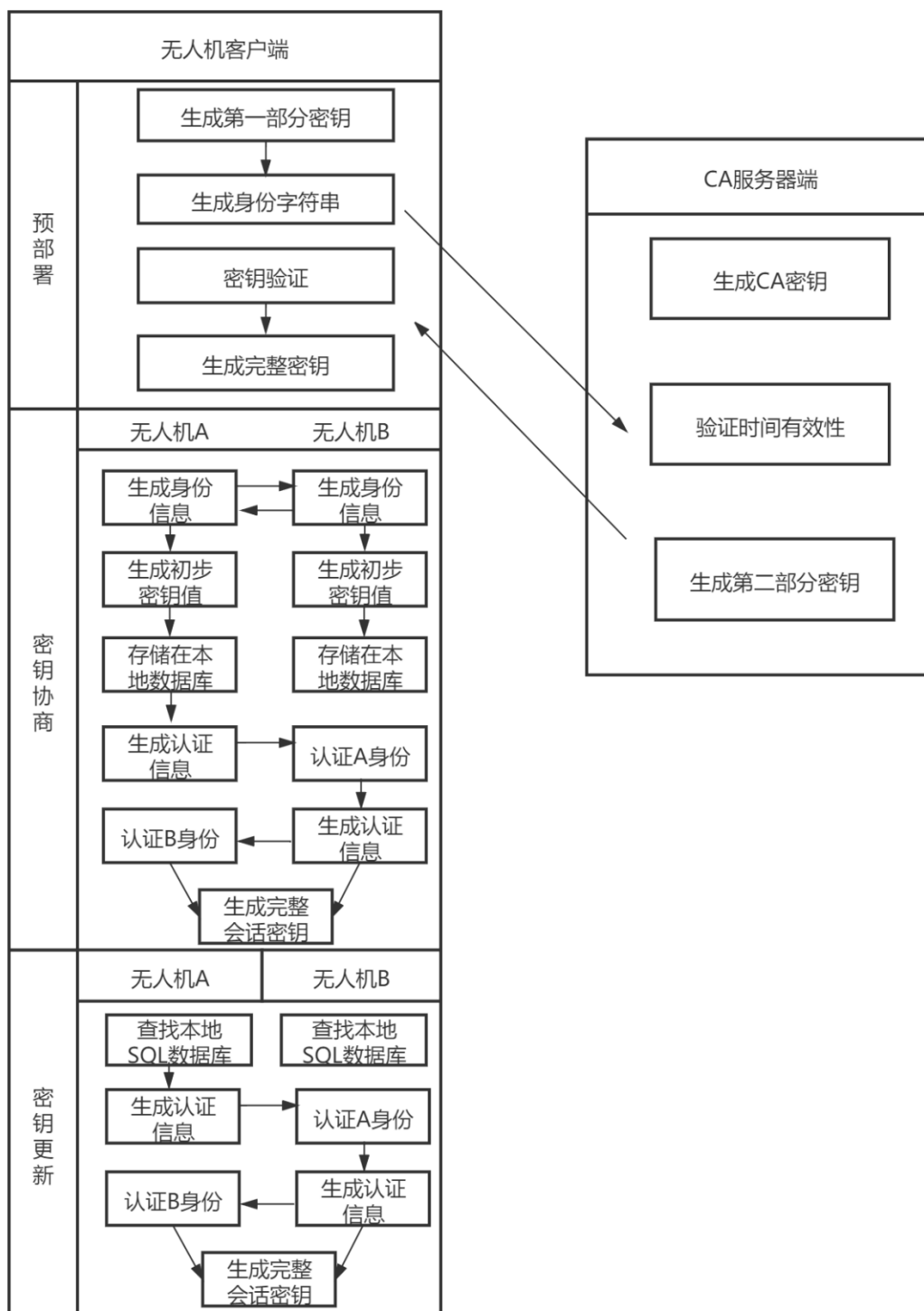


图 3.11 流程图

3.6 安全性分析

(1) **抵抗 CA 信息泄露攻击。**在本作品中，我们假设无人机的密钥信息储存在 CA 中，并且泄露给攻击者。同时我们假设攻击者只能访问 CA 上的密钥信

息而无法完全获得 CA 的控制权。在上述的场景中，传统基于证书的方案（例如，使用 X.509 证书和 ECQV 证书）无法保证无人机之间通信的安全性。然而，在采用本作品方案时，无人机私钥由两部分生成，即使 CA 信息泄露，攻击者仍然无法获取全部的私钥信息。

（2）抵抗假冒攻击。本作品中每个无人机通过字符串 w_i 将公私钥第一部分（无人机自生成的公私钥部分）与无人机的 ID 唯一绑定。攻击者如果假冒任意无人机 ID 与其他无人机进行通信，会导致通信双方计算得出不同的初步会话密钥。因此，远端无人机可通过消息认证码判断攻击者的虚假身份。

（3）防止重放攻击。在本方案中，尽管在不同的会话中两无人机的初步会话密钥保持不变，但每次会话都需要两个新的随机数参与。这两个随机数保证了无人机有效验证远端无人机的身份，且为通信双方更新最终会话密钥。因此，任何重放的旧消息都会导致身份认证失败。

第四章 密钥信息的安全存储方法

无人机网络中的无人机集群协同工作时，需要频繁的对密钥进行存储和读取，以进行通信过程中消息的加解密。因此，在设计无人机网络安全通信保护方法时，无人机客户端如何选择密钥信息存储架构，既能不增加无人机的能量损耗，又能实现快速存取密钥，成为一个重要问题。面对大量的密钥信息带来的数据安全存储、低能耗等需求，本作品主要解决了以下问题：

（1）密钥信息存储问题。密钥信息具有大量、高速、频繁存取、机密性等特征，对于数据存储需要更高的要求。因此，本作品根据密钥信息的存储要求，提出采用 MySQL 搭建数据库环境存储无人机密钥信息。

（2）数据库密钥信息安全问题。对于当前无人机安全通信系统容易遭受黑客的攻击问题，本作品只在 CA 数据库中存储无人机的部分密钥，因此即使数据库信息泄露，攻击者仍然无法得到全部的密钥信息

4.1 密钥信息的特征

无人机应用场景的日趋广泛使得现在无人机安全通信系统高度依赖对于大量密钥信息的存取。为实现提高效率、降低负载、安全存储的目的，无人机安全通信系统向着网络信息化、数据标准化等趋势进行发展。因此，在研究密钥信息的存储问题时，需要先对密钥信息相关特征进行分析。密钥信息主要具有以下特征：

（1）庞大的信息量：多无人机协同工作时，每两个相互通信的无人机均需计算一个会话密钥。密钥信息具有信息量大、源点多的特点。

（2）轻量级存取：无人机的协同工作需要实时的进行信息交互。因此，对于密钥信息的存取需要快速的进行。

（3）结构化：在无人机数据库中，本作品将密钥信息分条存储，力求数据的结构性，以满足数据提取与可视化要求。

（4）快速更新：在无人机安全通信系统中，无人机会话密钥需要定期进行更新。因此如果直接存储最终会话密钥，会造成大量密钥信息的修改与删除，系统的能耗会急剧增加。

因此，密钥信息具有大量、轻量级存取、快速更新等大数据特征，对于密钥信息的存储提出了更高的要求。

4.2 密钥信息存储架构选择

本作品选择 MySQL 作为数据存储架构，之所以选择该存储架构，其主要原因与密钥信息的特征相关，主要原因如下：

(1) 密钥信息的存储需求。密钥信息的主要特征是海量化、结构化、快速存储等，因此适合使用关系型数据库存储数据。

(2) 低成本。MySQL 是一个开源的、轻型的、关系数据库管理系统。由于其轻型、快速、存储使用成本低等特点，使得 MySQL 在因特网上被广泛的应用。

在无人机网络中，CA 与无人机安装 MySQL 用于数据存储与查询。CA 首先在预部署阶段存储所有的无人机身份字符串与通信地址。在密钥协商过程中，无人机与期待建立通信的远端无人机分别在本地数据库中存储一个初步的密钥值。

4.3 MySQL 数据库的设计

1、MySQL 数据库的建立

首先使用 `mysql_init()` 初始化 MySQL，接着使用 `mysql_real_connect()` 输入数据库参数，连接设备上的 MySQL。MySQL 数据库建立的关键代码如下：

```
//初始化mysql  
  
mysql_init(&mysql);  
  
mysql_real_connect(&mysql, NULL, "water dog", "xcl12345", NULL, 0, NULL, 0);  
  
query_str = "use Client";  
  
mysql_real_query(&mysql, query_str, strlen(query_str));  
  
res = mysql_store_result(&mysql);
```

图 4.1 数据库建立关键代码

2、无人机与 MySQL 的交互

在整个系统密钥协商的执行过程中，首次建立通信的无人机双方将计算得到的初步密钥值存储在数据库中。当通信双方再次建立新的会话时，首先检查本地数据库，若已存储初步密钥等信息，则会跳过系统的初步密钥生成阶段。双方交换认证消息，计算最终会话密钥。由于最终会话密钥的生成依赖于双方交换的认证信息中的随机数，因此可以实现轻量级安全的密钥更新。无人机与 MySQL 交互的关键代码如下：

```

//无人机间第一次通信

xX_generator(&x, &X, group, ctx);//无人机获取私钥x与公钥X

w_generator(ID, X, group, w);//无人机获取身份ID与字符串w

UDP_PrePhase(sockfd, addr, &p, &P, &C, w, group);//调用socket与CA进行通信

addr=UDP_Formal(mysql, &LK, x_A, w_A, p_A, P_A, sockfd, C);//无人机之间认证并协商密
钥

Save_Repeat(mysql, w, x, X, p, P, C, LK);

//双方再次建立通信

query ="select *from Client_A_key;";

if(mysql_real_query(&mysql, query, strlen(query)) ==0)

{

Get_Repeat(mysql, w_A, w_B, &P_A, &P_B, K, sk, &C);//获得第一次启动时的各种参数

if(UDP_Formal_Repeat(sockfd, K, w_A, P_A, P_B, sk) == 0) //无人机之间进行第二次通信

{

printf("Repeat failed\n");

return 0;

}

}

}

```

图 4.2 无人机与数据库交互关键代码

3、CA 与 MySQL 的交互

预部署阶段中，无人机向 CA 请求生成第二部分的公私要。CA 将网络内所有无人机的 IP 地址存储在数据库中。在密钥协商阶段，无人机向 CA 请求期待建立通信的远端无人机的 IP 地址，CA 查询数据库利用 UDP 套接字发送给无人机。CA 与 MySQL 交互的关键代码如下：

```

//预部署阶段时 CA 保存无人机的通信地址
Save_Repeat(mysql, w_A, x_A, X_A, p_A, P_A, public_key_C, LK_A, group1, ctx);

//密钥协商阶段时CA提供通信地址

Get_Repeat(mysql, w_A, w_B, &P_A, &P_B, K, sk, &C);//获取CA数据库的信息

sendto(sockfd, w_A, strlen(w_A), 0, (struct sockaddr*)&addr, sizeof(addr));

```

图 4.3 CA 与 MySQL 交互的关键代码

第五章 系统测试与分析

本章对第三章提出的安全通信方案和第四章中的密钥信息安全存储架构进行原型系统实现，并对系统进行性能效率与安全性测试。

5.1 测试环境

系统主要包括无人机客户端和 CA 服务器端两部分，其中 CA 服务器端主要参与预部署阶段生成无人机客户端第二部分密钥。下面说明无人机环境和 CA 服务器环境。

1、无人机客户端

通过将程序烧录到树莓派中，并将树莓派轻量级搭载在无人机上，以此作为客户端设备环境，主要参数如下表所示：

表 5.1 无人机环境参数

名称	参数
系统	Raspbian GNU
CPU	BroadcomBCM2711B0(CortexA-72) 1.5GHz*4
内存	1GB
存储	树莓派 4B 16GB 专用 TF 内存卡

2、CA 服务器环境

在 Ubuntu 系统下搭建 CA 服务器端，其参数如下表所示：

表 5.2 CA 服务器环境参数

名称	参数
系统	Ubuntu 20.04
CPU	Intel(R) Core(TM) i5-8300H CPU 2.3GHz*1
内存	2GB
存储	30GB

5.2 安全通信流程测试

安全通信流程测试主要测试本作品能否实现无人机间的轻量级安全通信，并通过 GTK+ 设计的前端界面进行展示。主要流程如下：

(1) 系统管理员开启 CA 服务器端，并绑定 CA 的 IP 与端口。绑定完成后，无人机可根据 CA 的 IP 地址与端口地址与 CA 建立通信。若无人机首次加入网络，则 CA 与无人机之间进行预部署阶段。CA 为无人机生成部分公私钥。若已进行预部署，则 CA 可向无人机提供网络内其他设备的 IP 地址。

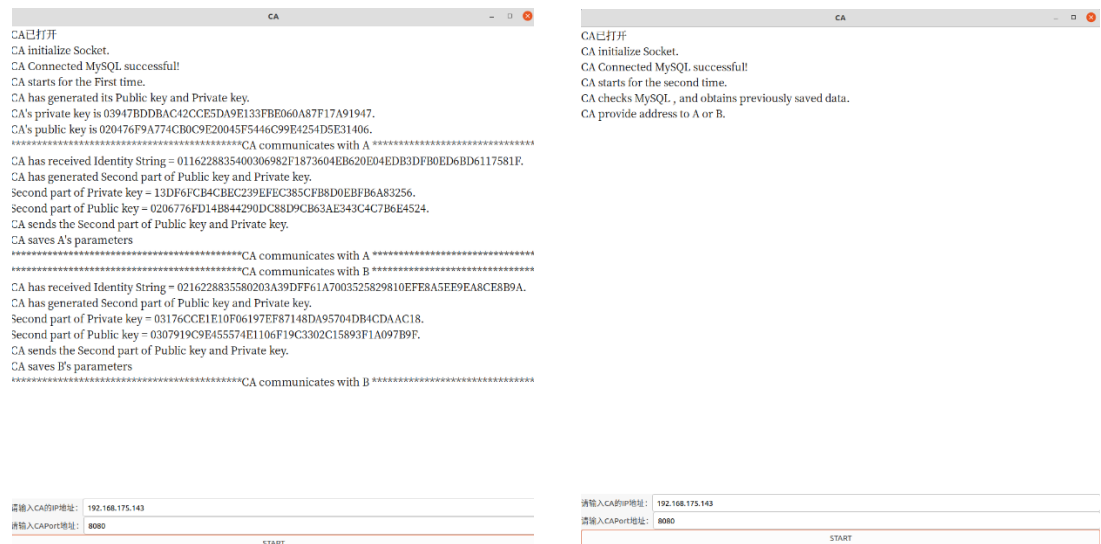


图5.3(a)入网功能 (b)信息提供功能

(2) 打开无人机 A 的图形化程序，如图 5.4 所示。首先需要输入 CA 的 IP 地址与端口号，检查是否已经进行了预部署。若无人机进行了预部署，则直接进入下一阶段；否则，无人机进行预部署，与 CA 进行交互生成无人机公私钥。

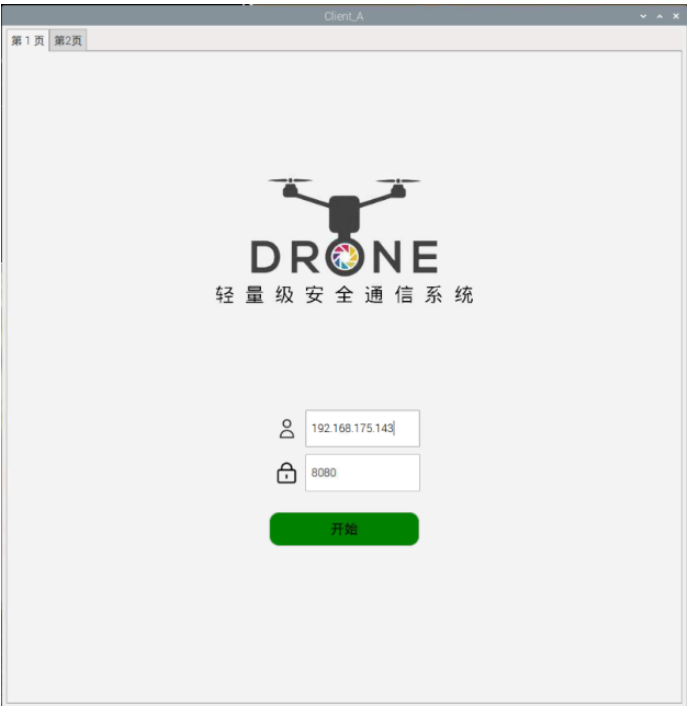


图 5.4 程序首页

(3) 预部署完成后，无人机 A 界面会显示已生成的无人机公私钥，并列出当前可通信的无人机列表，如图 5.5 左图所示。此时，无人机 A 可以通过输入通信对象的标识，来与通信对象建立连接。可以看到，当前已部署的无人机有 A 和 B。选择与无人机 B 建立连接，此时系统弹出窗口显示“正在与 Client_B 协商密钥”，点击确定按钮，系统自动进入下一阶段。



图 5.5 信息显示与对象选择界面

(4) 若与通信对象间为首次通信，则后端程序进行密钥协商阶段，否则进行密钥更新阶段。图 5.6 表示与无人机 B 的密钥协商阶段，图中可以看到密钥协商阶段的流程：首先交换双方的密钥参数，计算并保存初步密钥值，然后双方进行身份认证，最后生成无人机 A 与 B 的最终会话密钥，并显示在界面上。



图 5.6 无人机 A 的密钥协商阶段

（5）若无人机 A 再次与无人机 B 建立连接，系统会进入密钥更新阶段，图 5.7 表示无人机 A 的密钥更新阶段，图中可以看到密钥更新阶段的流程：首先交换密钥参数，然后直接从 Mysql 数据库（图 5.8）中提取初步密钥值，相互认证后生成无人机 A 与 B 的新的最终会话密钥，并显示在界面上。



图 5.7 与无人机 B 的密钥更新阶段

name	val
w_B	0216228835580203A390FF61A7003525829810EFE8A5EE9EA8CE8B9A
P_B	0307919C9E455574E1106F19C3302C15893F1A097B9F
sk_AB	f6cce0a155be781537d6f19564f0564ccfc9a1
K_AB	030690FD05410D340F7A24A0750D2B84AF47E8AE440F03019697E9C321FFA8671E9A829D8A4617007B564141
w_A	0116228835400306982F1873604EB620E04EDB3DFB0ED6BD6117581F
x_A	2A7173398BC101665B4591B3A3AD29ED42D76B65
X_A	0306982F1873604EB620E04EDB3DFB0ED6BD6117581F
p_A	130F6FCB4C8EC239EFC385CFB8D0EBFB6A83256
P_A	0206776FD148844290DC88D9CB63AE343C4C7B6E4524
public_key_C	020476F9A774CB0C9E20045F5446C99E4254D5E31406
LK_A	032FA44479B4128E9AF201E577861AEFC2F51F

11 rows in set (0.00 sec)

图 5.8 无人机 A 的 Mysql 数据库

(6) 上述流程结束后, 无人机 A 与 B 之间可利用最终会话密钥进行安全通信, 到此, 安全通信测试流程结束。

5.3 系统效率测试

图 5.9 提供了 Raspbian GNU 系统平台上 LSCS 方案的密钥协商阶段简化时间图, 其中显示了各个子流程的具体时间。LSCS 密钥协商总时间为 2.655ms, 子流程中初步密钥值 K_{AB} 所需时间较长, 耗时为 1.34ms, 计算 sk 所需时间为 0.433ms。由于无人机间距离较近, 因此发送延迟相对较短。

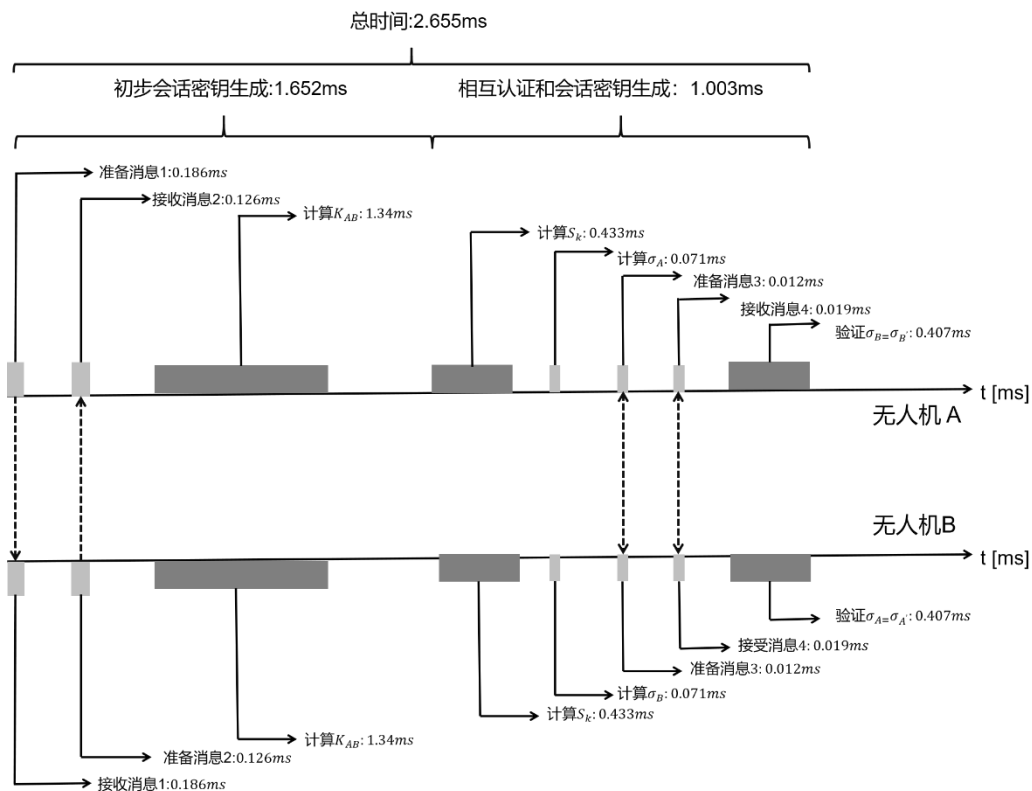


图 5.9 LSCS 密钥协商阶段各流程时间图

图 5.10 提供了不同方案与 LSCS 运行时间的对比图，在 Raspbian GNU 系统平台上，对[18][19][20]每一方案单独运行 30 次，并计算其所用时间。可以看出，LSCS 方案运行时间始终维持在 2.3ms 左右，且耗时最短。方案[20]运行时间较长，且波动幅度较大。

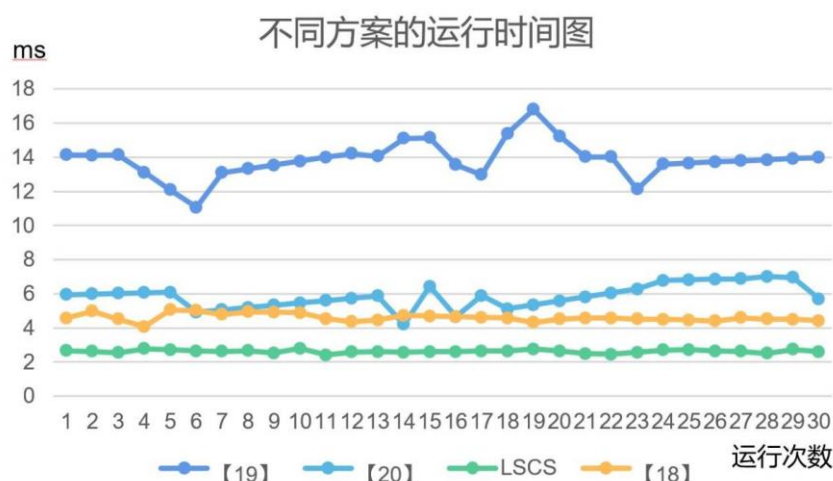


图 5.10 不同方案的运行时间图

5.4 CA 信息泄露安全性对比测试

无人机网络需要确保在 CA 信息泄露后依然能够正常运行，从而保证系统的安全性。本节通过模拟攻击者对传统方案（例如，使用 ECDH 密钥交换算法）的攻击过程来说明本方案的优越性。

攻击过程如下：

(1) 无人机 A 与 B 的 IP 地址分别为 192.168.175.38 和 192.168.175.143，通过 UDP 协议进行信息交互。无人机 A 和 B 通过 ECDH 协议来生成会话密钥，并利用该会话密钥进行加密通信。

(2) 攻击者可以通过抓取数据包来分析无人机 A 与 B 的通信消息，下面是通过 Wireshark 抓取到的 CA 与 A 通信的密文包：

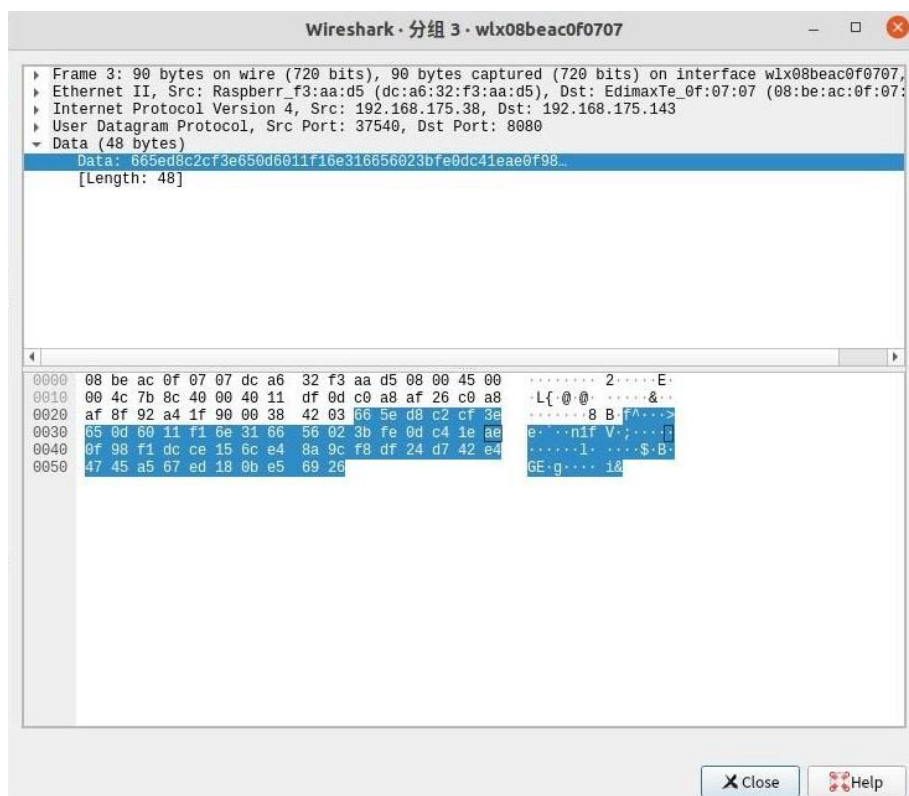


图 5.10 攻击者截获的密文包信息图

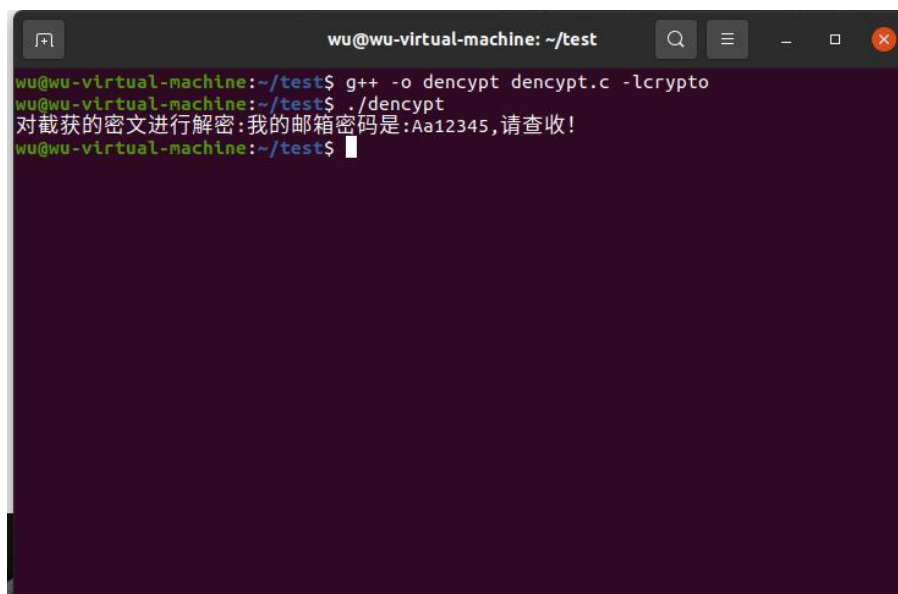
(3) 分析该数据包，可以得到 CA 与无人机通信的密文：

665ed8c2cf3e650d6011f16e316656023bfe0dc41eae0f98f1dcce156ce48a9cf8df24
d742e44745a567ed180be56926

(4) CA 信息泄露后，攻击者可以得到无人机 A 的私钥，从而计算通信的会话密钥为：

0401C0210ACA155517A795C22339A1CA705675CC13C7050A585FE7EEA5C
CCF723AAC7A2BA1CBC259CA5489

(5) 此时攻击者即可对截取的密文进行解密，得出明文：



```
wu@wu-virtual-machine: ~/test
wu@wu-virtual-machine:~/test$ g++ -o dencrypt dencrypt.c -lcrypto
wu@wu-virtual-machine:~/test$ ./dencrypt
对截获的密文进行解密:我的邮箱密码是:Aa12345,请查收!
wu@wu-virtual-machine:~/test$
```

图 5.11 密文解密所得结果图

上述实验证明,若安全方案过度依赖 CA,即 CA 存储了每个无人机的私钥,那么在 CA 的信息泄露后,攻击者可以利用无人机的私钥获取整个系统的所有通信消息,这样的系统存在巨大的安全隐患。对于 LSCS 方案,即使 CA 信息泄露,攻击者也不能获取到无人机的完整私钥,因此不会对系统的安全性造成影响。

5.5 测试结果分析

本作品从安全通信流程、性能效率与系统安全性方面对系统进行分析。

(1) LSCS 具有良好的通信性能。LSCS 系统使用 GTK+进行前端设计、数据库进行密钥信息存储、UDP 实现信息交互后,能够顺利完成整个安全通信流程,并提供给使用者良好的体验。

(2) LSCS 具有更好的性能效率。LSCS 的密钥协商时间约为 2.655ms,同设备下 ECC 协议运行时间约为 2.985ms,相比而言, LSCS 具有更好的性能。

(3) LSCS 具有极高的安全性。CA 信息泄露后,基于 ECDH 的传统无人机通信方案会存在巨大的安全隐患,攻击者可以掌握无人机的私钥,从而截取并破解加密消息。而 LSCS 被证明安全,说明 LSCS 方案有更好的安全性。

第六章 创新性说明

本作品提出了一种基于 CL-PKC 机制的轻量级安全通信方案，能够解决无人机网络中无人机安全通信问题。具体创新点如下：

（1） 无人机集群轻量级通信

本作品中创新性的在实现了无人机网络的轻量级通信，主要包括以下两个方面：1、支持与 CA 的间歇性连接。通信双方只在预部署阶段和密钥协商的开始时刻与 CA 进行信息交互，CA 无需持续连接；2、轻量级密钥更新。再次建立通信的无人机利用数据库存储的初步密钥值直接计算最终的会话密钥，无需重新执行整个密钥协商过程。

（2） 基于无证书公钥密码学的安全通信

本作品的安全通信主要体现在以下两个方面：1、密钥具有时效性。本作品为每个无人机密钥设置一个有效期，任何针对有效期的改动都可以被无人机检测。2、解决 CA 信息泄露引起的假冒攻击问题。无人机和 CA 分别生成一部分公私钥，即使 CA 信息泄露，攻击者仍然无法获取无人机的全部私钥。同时作品还解决了密钥托管问题。

（3） 基于 MySQL 数据库实现轻量级存取密钥

无人机网络中的无人机集群协同工作时，需要频繁的存取密钥进行通信过程中消息的加解密。本作品将无人机密钥信息存储在 MySQL 数据库中，既能不增加无人机的能量损耗，又能快速的存取密钥。

（4） 构建了完备的无人机轻量级无证书安全通信系统

本作品在系统实现与分析部分表明了设计的无人机轻量级无证书安全通信系统的各个模块是正确可行的，主要包括了预部署、密钥协商、密钥更新和无人机密钥分布式存储模块等。本作品使用 SOCKET 通信技术实现整个通信过程中信息的传输，同时为了具有良好的用户体验，本文采用 GTK 实现 Linux 下的图形界面开发。最终，本作品在树莓派中实现了该系统，并轻量级的搭载在无人机上，具有良好的应用前景。

第七章 总结

本作品通过对无人机网络安全通信系统的分析，介绍了现有的无人机网络的安全通信方案与通信流程。本作品提出了面向无人机网络的轻量级无证书安全通信系统。同时，针对无人机密钥海量、需要轻量级存储的特点，本文利用 MySQL 数据库搭建分布式环境存储无人机密钥信息。

在系统实现和分析阶段，通过对安全通信流程、性能效率和耗能开销的测试。可以证明，系统在实现无人机完全通信的同时，能够在低开销下完成无人机任务。

参考文献

- [1] 无人机运输 COVID-19 测试样品。 <https://www.thedronegirl.com/2020/06/09/gartner-delivery-drone-forecast/>
- [2] 未来两年内全球前 5 名 IoT 企业无人机的售出量。 <https://www.gartner.com/en/newsroom/press-releases/2019-12-04-gartner-forecasts-global-iot-enterprise-drone-shipmen>
- [3] Zeng, Yong, and Rui Zhang. "Energy-efficient UAV communication with trajectory optimization." *IEEE Transactions on Wireless Communications* 16.6 (2017): 3747-3760.
- [4] 何道敬,杜晓,乔银荣,朱耀康,樊强,罗旺.无人机信息安全研究综述[J].计算机学报,2019,42(05):1076-1094.
- [5] R. Wright. (2018). 23,000 Symantec Certificates Revoked Following Leak of Private Keys. Accessed: Oct. 17, 2019. [Online]. Available: <https://searchsecurity.techtarget.com/news/252436120/23000-Symantec-certificates-revoked-following-leak-of-private-keys>
- [6] Ammar, Mahmoud, Giovanni Russello, and Bruno Crispo. "Internet of Things: A survey on the security of IoT frameworks." *Journal of Information Security and Applications* 38 (2018): 8-27.
- [7] Shuai, Mengxia, et al. "Anonymous authentication scheme for smart home environment with provable security." *Computers & Security* 86 (2019): 132-146.
- [8] Fakroon M, Alshahrani M, Gebali F, Traore I (2020) Secure remote anonymous user authentication scheme for smart home environment. *Internet of Things*, p 100158
- [9] Certicom Research 2013, SEC4: Elliptic Curve Qu-Vanstone Implicit Certificate Scheme, Standards for Efficient Cryptography Group, Version 1.0 (Jan 2013).
- [10] D. Hankerson, A. J. Menezes, and S. Vanstone, "Guide to Elliptic Curve Cryptography", ISBN: 038795273X, Springer-Verlag New York, Inc., 2003.

- [11] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Syst. Lett.*, vol. 9, no. 1, pp. 1–4, Mar. 2017
- [12] D.Q. Bala, S. Maity, S.K. Jena, Mutual authentication for iot smart environment using certificate-less public key cryptography, in 2017 Third International Conference on Sensing, Signal Processing and Security (ICSSS) (IEEE, 2017), pp. 29–34
- [13] Malik M., Kamaldeep, Dutta M. (2020) On the Applicability of Certificateless Public Key Cryptography (CL-PKC) for Securing the Internet of Things (IoT). In: Dutta M., Krishna C., Kumar R., Kalra M. (eds) *Proceedings of International Conference on IoT Inclusive Life (ICIIL 2019)*, NITTTR Chandigarh, India. *Lecture Notes in Networks and Systems*, vol 116. Springer, Singapore. https://doi.org/10.1007/978-981-15-3020-3_5
- [14] J. Shen, Z. Gui, S. Ji, J. Shen, H. Tan, and Y. Tang, "Cloudaided lightweight certificateless authentication protocol with anonymity for wireless body area networks," *J. Netw. Comput. Appl.*, vol. 106, pp. 117–123, Mar. 2018.
- [15] Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Adv. Cryptol. ASIACRYPT*, 2003, pp. 452–473.
- [16] J. Baek, R. Safavi-Naini, and W. Susilo, "Certificateless public key encryption without pairing," in *Information Security*. Berlin, Germany: Springer, 2005, pp. 134–148.
- [17] <https://zh.wikipedia.org/wiki/GNOME>
- [18] D. He, S. Padhye, and J. Chen, "An efficient certificateless two-party authenticated key agreement protocol," *Comput. Math. Appl.*, vol. 64, no. 6, pp. 1914–1926, 2012.
- [19] S.-B. Wang, Z.-F. Cao, and H.-Y. Bao, "Efficient certificateless authentication and key agreement (CL-AK) for grid computing," *Int. J. Netw. Security*, vol. 7, no. 3, pp. 342–347, 2008.

- [20] M. E. S. Saeed, Q.-Y. . Liu, G. Tian, B. Gao, and F. Li, “AKAIoTs:Authenticated key agreement for Internet of Things,” *Wireless Netw.*, vol. 25, no. 6, pp. 3081–3101, Aug. 2019.