



信息加密系统需求性分析报告

学 院： 数学与计算机学院

姓 名： 谢焕豪 许洛熙 彭锦锋

关超华 邓佳诚 李佳伦

展浩铭

科任老师： 蔡浩

课 程： 软件工程

报告时间： 2025年1月1日

**需求性分析**

**1.功能性需求**

**（1）.加密算法支持**

**Caesar密码**

* 算法特点：Caesar密码是一种简单的替换加密算法，通过将字母表中的每个字母移动固定数目的位置来进行加密。
* 实现要求：系统应允许用户指定偏移量，并能够对输入的文本进行相应的偏移操作。
* 性能考量：由于Caesar密码较为简单，系统应能够快速处理大量文本数据的加密和解密。

**Transposition密码**

* 算法特点：Transposition密码通过重新排列明文中的字母顺序来实现加密，保持字母的集合不变。
* 实现要求：系统应支持用户定义的排列规则，并能够按照这些规则对文本进行加密和解密。
* 性能考量：对于小文件和中等复杂度的加密，系统应保持高效的处理速度。

**Vigenere密码**

* 算法特点：Vigenere密码是一种多表代换密码，使用关键字对明文进行加密，提高了密码的安全性。
* 实现要求：系统应允许用户输入关键字，并根据关键字生成Vigenere方阵进行加密。
* 性能考量：系统应优化算法以处理较长的文本数据，同时保持合理的加密和解密速度。

**RSA加密**

* 算法特点：RSA是一种非对称加密算法，使用一对公钥和私钥进行加密和解密。
* 实现要求：系统应支持密钥的生成、管理和使用，以及使用公钥加密和私钥解密的过程。
* 性能考量：考虑到RSA加密计算量较大，系统应优化算法以减少对计算资源的消耗，尤其是在处理小数据文件时。

**（2） 数据一致性**

* 加密解密验证：系统必须确保加密后的数据在解密后能够完全恢复到原始状态，没有数据丢失或损坏。
* 错误检测：系统应具备错误检测机制，如校验和或哈希函数，以验证数据在加密和解密过程中的完整性。
* 异常处理：系统在遇到错误或异常时，应能够提供清晰的错误信息，并保持原始数据的完整性。

**（3） 多平台支持**

* 跨平台兼容性：系统应设计为跨平台兼容，能够在Windows、macOS和Linux等主流操作系统上运行，无需额外的修改。
* 用户界面一致性：不同平台上的用户界面应保持一致，提供相同的用户体验。
* 性能优化：系统应针对不同平台的硬件特性进行优化，确保在各种环境下都能达到最佳性能。

**2.安全性需求**

**（1）加密算法的安全性**

* 算法强度：系统所采用的加密算法必须经过充分验证，能够抵抗当前已知的密码分析技术。特别是对于RSA加密，密钥长度至少应为2048位，以确保足够的安全性。
* 算法更新：系统应能够支持算法的更新和替换，以应对新出现的安全威胁和密码学进展。

**（2）密钥管理**

* 密钥生成：系统必须能够生成强随机密钥，且密钥生成过程必须安全，不受预测或复现。
* 密钥存储：所有密钥必须安全存储，使用加密技术保护，防止未授权访问。对于RSA密钥，应使用硬件安全模块(HSM)或其他安全存储解决方案。
* 密钥分发：非对称加密中的公钥和私钥的分发过程必须安全，防止密钥在传输过程中被截获或篡改。

**（3）数据传输安全**

* 数据加密：在数据传输过程中，所有敏感数据必须加密，使用安全的传输协议（如TLS）来保护数据不被窃听或篡改。
* 身份验证：系统应实施强身份验证机制，确保只有授权用户才能访问加密和解密功能。

**（4）抗攻击能力**

* 抵抗暴力破解：系统设计的加密算法应具有足够大的密钥空间，使得暴力破解在实际中不可行。
* 抵抗侧信道攻击：系统应设计以抵抗侧信道攻击，如通过功耗分析、电磁泄漏等非传统途径获取加密密钥的攻击。
* 抵抗重放攻击：系统应能够识别并处理重放攻击，确保旧的数据包不能被重新发送以欺骗系统。

**（5）系统完整性**

* 代码和数据分离：系统应确保代码和数据存储在不同的环境中，防止潜在的数据篡改。
* 系统监控：系统应具备监控机制，能够检测和记录异常行为，如未授权访问尝试或系统配置更改。

**（6）用户数据隐私**

* 数据访问控制：系统应实施严格的数据访问控制，确保用户数据只能被授权用户访问。
* 数据匿名化和去标识化：在可能的情况下，系统应提供数据匿名化或去标识化的功能，以保护用户隐私。

**3. 性能需求**

**（1）加密和解密速度**

* 实时性能：系统应能够快速响应用户的加密和解密请求，特别是在处理实时数据时，如通信过程中的数据加密。
* 大数据量处理：系统必须能够高效处理大量数据的加密和解密，对于Vigenere密码和RSA加密，应优化算法以减少对计算资源的消耗，尤其是在处理大文件时。

**（2）系统资源消耗**

* 资源优化：系统在执行加密和解密操作时，应优化CPU和内存的使用，确保在高负载下不会过度消耗系统资源，影响其他程序的运行。
* 存储效率：加密数据的存储格式应设计为空间效率最大化，避免不必要的存储开销。

**（3）并发处理能力**

* 多用户支持：系统应能够支持多用户同时进行加密和解密操作，保持稳定的性能，特别是在多用户环境中。
* 负载均衡：系统应具备负载均衡机制，能够在高并发请求下分配计算资源，以防止系统过载。

**系统稳定性**

* 错误恢复：系统在遇到错误时，应能够快速恢复，保证操作的连续性和数据的完整性。
* 高可用性：系统应设计为高可用性，减少系统故障时间，确保关键业务的连续运行。

**4. 用户体验需求**

* 操作简便性：系统界面应友好，操作简便，用户可以轻松进行加解密操作。
* 错误信息反馈：系统在用户操作错误时能够提供明确且有用的错误提示，帮助用户正确使用系统。
* 文档和帮助系统：提供清晰的文档和帮助系统，确保用户能够理解并顺利完成加密过程。

**5. 发展方向需求**

* 新加密方式的添加：系统应设计为易于添加新的加密方式，以适应未来加密技术的发展。
* 版权问题规避：使用Godot开发可以避免一些版权问题，系统应继续利用开源资源，确保合法合规。
* 服务器和管理员功能：由于缺少服务端和管理员方面的功能，系统应开发相应的服务器端功能，以支持更广泛的应用场景。
* 通信功能：系统可以考虑增加通信功能，以支持加密数据的传输。

**6. 测试和维护需求**

* 测试覆盖：系统应进行全面的测试，包括功能性测试、安全性测试、性能测试和用户体验测试，确保系统的稳定性和可靠性。
* 持续维护：系统需要定期更新和维护，以修复已知问题并提高系统性能。