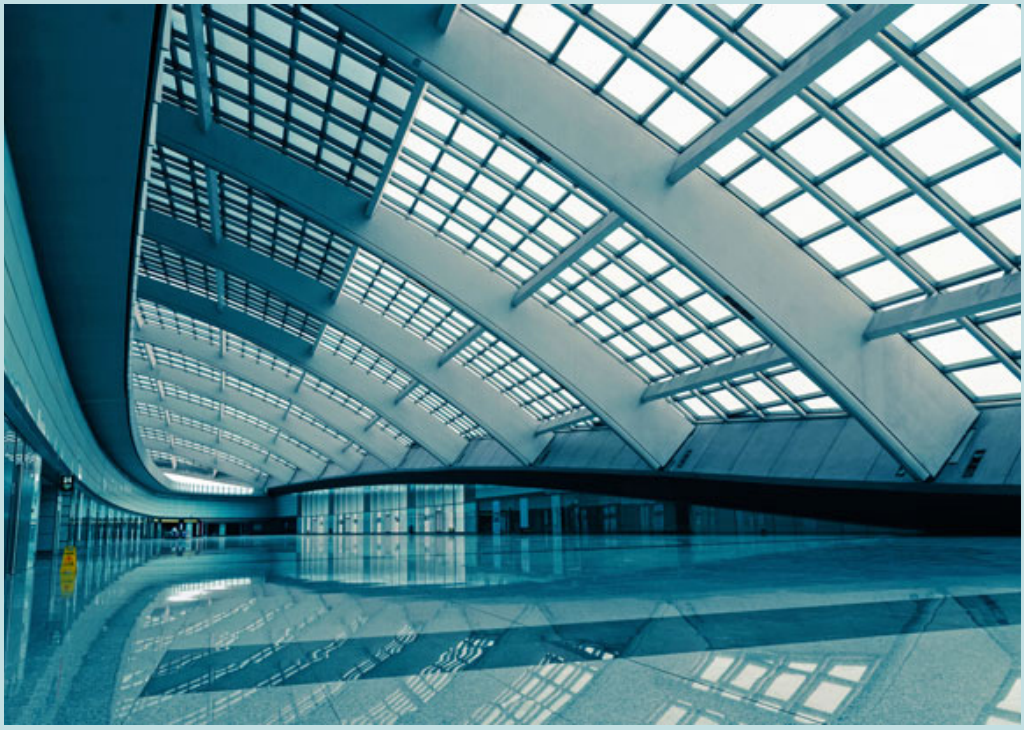
算法报告

2023.4.6
abel



基于门限制的密钥分割与恢复算法

1、简介

本报告主要介绍一种基于门限制的密钥分割与恢复算法，该算法通过分割和分发密钥来实现安全的密钥管理。此算法的核心原理是使用中国剩余定理（CRT）对密钥进行分割和恢复。

2、算法原理

2.1 相关数学理论部分

一、同余关系（Congruence）

同余关系是密码学和数论的基本概念。给定一个正整数 m，如果两个整数 a 和 b 满足 a - b 可以被 m 整除，那么我们说 a 和 b 对模 m 同余。数学表示为 a ≡ b (mod m)。同余关系具有以下性质：

反身性：a ≡ a (mod m)。

对称性：若 a ≡ b (mod m)，则 b ≡ a (mod m)。

传递性：若 a ≡ b (mod m) 且 b ≡ c (mod m)，则 a ≡ c (mod m)。

兼容性：若 a ≡ b (mod m) 且 c ≡ d (mod m)，则 a + c ≡ b + d (mod m) 且 ac ≡ bd (mod m)。

三、逆元素（Modular Inverse）

在模运算中，逆元素是一个重要概念。给定一个整数 a 和一个模数 m，若存在整数 b，使得 ab ≡ 1 (mod m)，则我们称 b 是 a 关于模 m 的逆元素。在密钥恢复的过程中，我们需要计算模逆来解决同余方程组。

四、欧几里得算法（Euclidean Algorithm）

欧几里得算法用于计算两个整数的最大公约数（GCD）。扩展欧几里得算法不仅能计算最大公约数，还能找到模逆。给定两个整数 a 和 m，扩展欧几里得算

法可以找到整数 x 和 y，使得 ax + my = gcd(a, m)。当 gcd(a, m) = 1 时，x 就是 a 关于模 m 的逆元素。

密钥分割：给定一个群组密钥，我们将其分割成多个部分，以便每个成员只能获得部分密钥。首先，我们为每个子服务器选择一个模数，然后根据门限值生成随机数。接着，通过计算模数与随机数的乘积和密钥的余数来得到密钥的分割部分。这样可以确保每个成员所获得的密钥部分都不同，从而增加密钥安全性；密钥恢复：当成员需要恢复群组密钥时，他们可以从子服务器中获取密钥部分。然后，使用中国剩余定理对这些部分进行组合，从而恢复群组密钥。具体而言，我们首先计算每个密钥部分与其他模数的乘积，然后求逆，最后将它们相加并取模，得到恢复的群组密钥。

在密钥分割和恢复的过程中使用了**同余关系**，我们使用了模运算来保证密钥的整数范围。同余关系的性质使我们能够方便地处理模运算，例如求和、求积等。**逆元素**：在密钥恢复阶段，我们需要计算模逆元素来解决同余方程组。通过求解逆元素，我们可以计算出密钥部分与其他模数的乘积，从而利用中国剩余定理恢复群组密钥。欧几里得算法：尽管程序中没有直接使用欧几里得算法，但它在计算模逆元素时库本身内部调用。而且我们可以利用扩展欧几里得算法找到整数 x 和 y，使得 ax + my = gcd(a, m)。当 gcd(a, m) = 1 时，x 就是 a 关于模 m 的逆元素。在程序中，我们使用了 Python 的 pow() 函数来计算逆元素，该函数在内部可能使用了类似的**欧几里得算法算法**。这些数学理论为我们的密钥分割与恢复算法提供了理论基础。它们在处理模运算、计算逆元素以及密钥恢复等方面发挥了关键作用，使得我们的算法既安全又高效。

3、技术细节

为了提高安全性，我们在生成密钥分割时引入了随机数，使得每次生成的密钥分割都不同。

我们对生成的随机数和加密分割密钥进行了检查，以确保它们的长度相等。

为了防止恶意攻击，我们增加了门限制的检查，以确保成员只能在满足一定条件下恢复密钥。

4、应用场景

基于门限制的密钥分割与恢复算法适用于需要对密钥进行安全管理的场景，例如金融、医疗、政府等领域。通过将密钥分割为多个部分，我们可以降低单个密钥泄露的风险，从而保护数据安全。

本报告介绍了一种基于门限制的密钥分割与恢复算法，该算法利用中国剩余定理对密钥进行分割和恢复。通过引入随机数、增加门限制检查等手段，我们确保了密钥的安全性和可靠性。该算法在需要对密钥进行安全管理的各种场景中都具有很高的实用价值。

在未来的研究中，我们可以进一步优化和改进该算法，以适应更多的应用场景和需求。例如，可以考虑使用更高效的数学方法或密码学技术来提高密钥分割和恢复的速度，或者引入新的安全机制以增强抗攻击能力

5、参考链接

中国剩余定理：https://en.wikipedia.org/wiki/Chinese_remainder_theorem

Sympy模块：https://docs.sympy.org/latest/index.html

随机数生成：https://docs.python.org/3/library/random.html

密钥分割与门限密码学：https://en.wikipedia.org/wiki/Secret_sharing