



北 京 科 技 大 学

本科生毕业设计(论文)选题报告

题 目： 椭圆曲线密码算法

加速方法研究与实现

学 院： 计算机与通信工程学院

专 业： 信息安全

姓 名： 韦效伟

学 号： 42024032

指导教师签字： 姚宇浩

2024 年 03 月 18 日

目 录

本科生毕业设计(论文)选题报告	1
目 录	1
1 课题背景和研究意义	1
1.1 课题背景	1
1.2 研究意义	1
2 文献综述	3
2.1 基础数论知识	3
2.1.1 群	3
2.1.2 域	3
2.2 椭圆曲线	4
2.2.1 椭圆曲线的定义	4
2.2.2 素域中的椭圆曲线	4
2.2.3 椭圆曲线群运算规则	4
2.2.4 椭圆曲线点乘运算	5
2.3 SM2 简介及其原理	5
2.4 SM3 简介及其原理	7
2.5 几种椭圆曲线点乘计算加速方法	7
2.5.1 NAF 点乘	8
2.5.2 雅可比坐标系	8
2.5.3 Montgomery 模乘	10
2.6 FPGA	10
2.7 Git 简介和基本指令	10
3 研究内容、预期目标和研究方法	12
3.1 研究内容	12
3.2 预期目标	12
3.3 研究方法	12
4 研究进度安排	13
参考文献	15

1 课题背景和研究意义

1.1 课题背景

椭圆曲线密码算法（ECC）作为一种公钥密码算法，具有良好的安全强度，在密钥长度较短的情况下提供与 RSA 等传统算法相当的安全性。在现代密码学和信息安全领域，ECC 被广泛应用于数字签名、密钥交换、身份认证等场景。然而，随着物联网等资源受限环境的兴起，ECC 在这些场景中的应用也面临着挑战，主要体现在计算效率和硬件资源利用率上。

在物联网等资源有限环境中，ECC 的计算效率成为一个关键问题。特别是在对称加密算法中，频繁的倍点计算操作对资源有限的设备而言可能会成为性能瓶颈。因此，研究如何加速 ECC 中的倍点计算成为一项重要课题。通过对现有的倍点加速方法进行分析和比较，可以找到最适合物联网环境的加速方法，并进一步探索其在硬件上的实现，以提高计算效率和节约资源消耗。

本课题的背景针对 ECC 在物联网等资源受限环境中应用的需求和挑战，研究并实现一种高效的 ECC 倍点计算加速方法。通过深入分析现有的加速方法原理和优缺点，结合实验对比不同方法的性能表现，最终选择并深入研究一种最具潜力的加速方法。通过硬件实现，可以进一步提高计算效率，为物联网等资源受限环境下的安全通信提供更好的支持。

总之，本课题旨在探索如何在资源有限的环境中提高 ECC 的计算效率，从而促进其在物联网等领域的广泛应用，并为安全通信提供更加高效和可靠的解决方案。

1.2 研究意义

在资源受限的环境中提高椭圆曲线密码算法（ECC）的计算效率具有重要的理论和实际意义：

首先，ECC 作为一种高效的公钥密码算法，在信息安全领域有着重要的地位。通过提高 ECC 在资源受限环境下的计算效率，可以推动密码学领域的发展，为解决实际安全问题提供更加有效的技术支持。

其次，在物联网等资源受限的环境中，安全通信是一个关键挑战。通过优化 ECC 的计算效率，可以提高物联网设备的安全性，保护物联网系统免受恶意攻击和数据泄露的威胁，从而促进物联网的健康发展。

第三，资源受限环境下的设备通常具有有限的计算能力和存储空间。优化 ECC 的计算效率可以减少设备的计算负担，节约能源消耗，延长设备的使用寿命，并降低运行成本。

第四，有助于提高安全性。ECC 在数字签名、密钥交换等场景中广泛应用，其安全性直接影响到通信数据的保密性和完整性。通过提高 ECC 的计算效率，我们可以加强加密算法的安全性，防止密码学攻击，从而更好地保护通信数据的安全。

第五，随着物联网、云计算等新兴技术的快速发展，对安全通信的需求越来越迫切。优化 ECC 的计算效率可以为各种应用场景提供更加高效和可靠的安全通信解决方案，推动技术的广泛应用。

因此，研究如何提高 ECC 在资源受限环境下的计算效率具有重要的理论和实际意义，有助于推动密码学的发展、促进物联网安全、节约资源消耗、提高安全性，并促进技术的应用和发展。

2 文献综述

在本节中，我们将对文献综述中涉及各个主题进行详细阐述。首先，我们将介绍基础数论知识，包括群和域的概念。接着，我们将讨论椭圆曲线的基本定义以及在素域上的应用。然后，我们将深入探讨 SM2 密码算法及其原理，以及 SM3 密码杂凑算法及其原理。随后，我们将介绍几种椭圆曲线点乘计算加速方法，包括 NAF 点乘、雅可比坐标和 Montgomery 模乘。最后，我们将简要介绍 FPGA 的基本概念和应用，以及 Git 版本控制工具的简介和基本指令。

2.1 基础数论知识

2.1.1 群

将一种代数运算定义在一个非空集合 Q 内， Q 被称为群需要满足以下条件：

1. 封闭性。

$$x, y \in Q, \text{ 有 } x \cdot y \in Q.$$

2. 结合律。

$$\forall x, y, z \in Q, \text{ 有 } (x \cdot y) \cdot z = x \cdot (y \cdot z).$$

3. 有单位元。

$$\forall x \in Q, \text{ 有 } e \in Q, \text{ 使得 } x \cdot e = e \cdot x = x.$$

4. 有逆元。

$$\forall x \in Q, x^{-1} \in Q, \text{ 使得 } x \cdot x^{-1} = x^{-1} \cdot x = e, \text{ 称 } x^{-1}、x \text{ 互为逆元。}$$

Abel 群：

$$\text{若有一个群 } Q \text{ 中, 满足任意 } x, y \in Q, \text{ 均有 } x \cdot y = y \cdot x, \text{ 则称群 } Q \text{ 为 Abel 群。}$$

2.1.2 域

假设 F 为一个非空集合，集合 F 上定义了加法和乘法运算，且满足以下公理，则称 F 为域。其中， F 中元素的个数称为阶。

- 1) F 关于加法构成 Abel 群，且单位元为 0；
- 2) $F - \{0\}$ 关于乘法构成 Abel 群；

3) 乘法对加法满足分配律。

素数域：设有限域 F 的阶为 p^n ，当 $n=1$ 时，该有限域称为素数域。素数域被广泛应用于 ECC。

2.2 椭圆曲线

2.2.1 椭圆曲线的定义

椭圆曲线^[1]是域上亏格为 1 的光滑射影曲线。曲线方程如公式(2-1)所示。

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \quad (2-1)$$

上式被称为 Weierstrass 方程。其中， $a_1, a_2, a_3, a_4, a_6 \in K$ ，且 $\Delta \neq 0$ ， Δ 为曲线 E 的判别式，各个参数的具体定义如公式(2-2)所示。

$$\begin{cases} d_2 = a_1^2 + 4a_2 \\ d_4 = 2a_4 + a_1a_3 \\ d_6 = a_3^2 + 4a_6 \\ d_8 = a_1^2a_6 + 4a_2a_6 - a_1a_3a_4 + a_2a_3^2 - a_4^3 \\ \Delta = -d_2^2d_8 - 8d_4^3 - 27d_6^2 + 9d_2d_4d_6 \end{cases} \quad (2-2)$$

2.2.2 素域中的椭圆曲线

通过公式(2-3)可以把素数域上的椭圆曲线变换为 $y^2 = x^3 + ax + b$ ，其中， $a, b \in K$ ， $\Delta = -16(4a^3 + 27b^2)$ ，且 $4a^3 + 27b^2 \neq 0$ 。

$$(x, y) \rightarrow \left(\frac{x-3a_1^2-12a_2}{36}, \frac{y-3a_1^2x}{216} - \frac{a_1^3+4a_1a_2-12a_3}{24} \right) \quad (2-3)$$

2.2.3 椭圆曲线群运算规则

椭圆曲线群的运算主要有两类，点加和点乘。

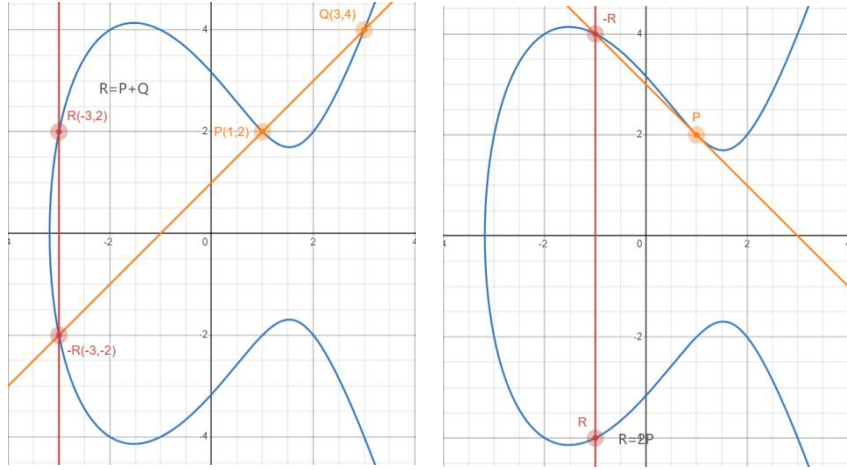
1) 点加

椭圆曲线上两个点相加称为点加。如图 2.1(a)所示，对于椭圆曲线 E 上任意两点 $P = (x_1, y_1)$ 和 $Q = (x_2, y_2)$ ，过这两点的直线与 E 相交于另一点，这一点与 x 轴对称得到点 R ，定义 $R = P + Q$ 称为点加运算。

2) 倍点

在点加的基础上，如果 P 和 Q 是同一点，则做切线与 E 相交于另一点，这一点与 x 轴对称得到点 R ，定义 $R = 2P$ 为倍点运算。

下图为 $y^2=x^3-7x+10$ 上点加和倍加的示意图



椭圆曲线 $E(F_n)$ 的运算法则有：

(1) 单位元，对于所有 $P \in E(F_n)$ ， $P + O = O + P = P$

(2) 负元素，若 $P = (x, y) \in E(F_p)$ ，则 $(x, y) + (x, -y) = O$ ，记点 $(x, -y)$ 为 $-P$ ，并称其为 P 的负，此外， $-O = O$ 。

(3) 点加，设 $P_1 = (x_1, y_1) \in E(F_n) \setminus \{O\}$ ， $P_2 = (x_2, y_2) \in E(F_n) \setminus \{O\}$ ，且 $x_1 \neq x_2$ ，设 $P_3 = (x_3, y_3) = P_1 + P_2$ ，则

$$\begin{cases} x_3 = \lambda^2 - x_1 - x_2 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

，其中， $\lambda = \frac{y_2 - y_1}{x_2 - x_1}$ 。

(4) 倍加，设 $P_1 = (x_1, y_1) \in E(F_n) \setminus \{O\}$ ，且 $y_1 \neq 0$ ，设 $P_3 = (x_3, y_3) = 2P_1$ ，则

$$\begin{cases} x_3 = \lambda^2 - 2x_1 \\ y_3 = \lambda(x_1 - x_3) - y_1 \end{cases}$$

，其中， $\lambda = \frac{3x_1^2 + a}{2y_1}$ 。

2.2.4 椭圆曲线点乘运算

设 $P \in E(F_p)$ ，记 kP 为将点 P 进行 k 次自相加，如公式(2-4)所示。 kP 运算即为椭圆曲线点乘运算，也称为标量乘。

$$kP = \underbrace{P + P + \cdots + P}_{k \uparrow} \quad (2-4)$$

2.3 SM2 简介及其原理

SM2 是我国自主知识产权的商用密码算法，基于椭圆曲线公钥密码算法设计，加密强度为 256 位。SM2 算法的安全性基于椭圆曲线离散对数问题，安全性高。

SM2 算法的加密过程如下：

第一步：通过真随机数发生器生成一个大于 1 小于 n (n 为所选的椭圆曲线阶次) 的随机数 k 。

第二步：根据如下步骤计算椭圆曲线点，其中 G 和 C_1 在椭圆曲线上， G 为在椭圆群中挑选的元点，并将 C_1 数据类型转为比特串。

$$C_1 = [k]G = (x_1, y_1) \quad (2-5)$$

第三步：计算椭圆曲线上的非无穷点 S ， h 大于 1 小于 $(n-1)$ ， P_B 为接收方公钥。

$$S = [h]P_B \quad (2-6)$$

第四步：根据以下公式计算曲线点 x_2 、 y_2 ，并转化为比特串。

$$[k]P_B = (x_2, y_2) \quad (2-7)$$

第五步：计算 t 的值，KDF 为密钥派生函数，以 SM3 计算，若 t 为全 0 比特串则返回第一步。

$$t = \text{KDF}(x_2 || y_2, \text{len}) \quad (2-8)$$

第六步：计算 C_2 的值为 M 异或 t 。

第七步：计算 C_3 的值，Hash 为 SM3 杂凑算法。

$$C_3 = \text{Hash}(x_2 || M || y_2) \quad (2-9)$$

第八步：根据以下步骤计算密文 C ，并发送给用户 B 。

$$C = C_1 || C_2 || C_3 \quad (2-10)$$

SM2 算法解密过程如下：

第一步：从 C 中取出比特串 C_1 ，将 C_1 的数据类型转换为椭圆曲线上的点，验证 C_1 是否满足椭圆曲线方程。

第二步：计算椭圆曲线点 S ，若 S 为无穷远点直接报错退出。

$$S = [h]G \quad (2-11)$$

第三步：根据私钥及 C_1 计算 x_2 、 y_2 ，将坐标的数据类型转换为比特串。

$$[d_B]C_1 = (x_2, y_2) \quad (2-12)$$

第四步：根据公式 2-13 计算 t 的值，若 t 为零比特串，报错退出。

$$t = \text{KDF}(x_2 || y_2, \text{len}) \quad (2-13)$$

第五步：从密文 C 中取出 C_2 ，计算 M' 的值为 C_2 异或 t 。

第六步：计算 u 的值，取出比特串 C_3 ，若 u 与 C_3 不相等，报错退出。

$$u = \text{Hash}(x_2 || M' || y_2) \quad (2-14)$$

第七步：用户 B 解密出明文 M' 。

2.4 SM3 简介及其原理

SM3 算法适用于商用密码应用中的数字签名和验证、消息认证码的生成等多方面。此算法对输入长度小于 2 的 64 次方的比特消息，生成长度为 256 比特的杂凑值。其主要流程由填充，迭代压缩，消息拓展和压缩函数构成，这里不对此算法进行过多探究，只使用其对输出的消息进行编码以便得到 256 比特的数据。

2.5 几种椭圆曲线点乘计算加速方法

ECC 中最核心的运算是点乘运算，也称标量乘运算。针对标量乘的计算，朴素的算法是 Kaliski^[2]提出的加减法。目前，标量乘计算的优化已经存在多种：Morain F^[3]提出的加减链算法，通过对标量 k 进行编码减少其汉明重量；Solinas J^[4]提出一种 width- w NAF 算法，使用预计算表减少点加运算次数；Blakely G^[5]提出加法型模乘算法，将乘法转换为加法实现，达到加速效果 Montgomery P^[6]提出一种 Montgomery 模乘法，通过位移操作避免大数取模时的试除操作；Bernstein D^[7]通过将坐标系转换到雅可比坐标，避免点加和倍点操作时最耗时的模逆运算；Awaludin A^[8]设计的 Karatsuba 乘法器，将大数模乘转换为位宽更小的操作。

表 2-1 标量乘的优化方法

学者	方法	简述
Kaliski	朴素加减法	
Morain F	加减链算法	通过对标量 k 进行编码来减少其汉明重量
Solinas J	width- w NAF 算法	利用预计算表的方法来减少点加运算的次数
Blakely G	加法型模乘算法	将乘法转换为加法实现，减少乘法操作的复杂度
Montgomery P	Montgomery 模乘法	通过位移操作来避免大数取模时的试除操作
Bernstein D	雅可比坐标	避免点加和倍点操作中最耗时的模逆运算
Awaludin A	Karatsuba 乘法器	将大数模乘转换为位宽更小的操作，分治算法，可以减少乘法操作的次数

2.5.1 NAF 点乘

NAF 点乘是利用 NAF 表示法进行点乘运算的一种方法，其中 NAF 表示法是一种整数的二进制表示法，使用的数字只包含 1 和 -1，并且没有连续的 1。NAF 点乘可以提高点乘运算的效率，在椭圆曲线密码学中常被使用。

NAF 点乘可以通过一下六个步骤来实现：

第一步，将点乘的标量（通常是一个整数）用 NAF 表示法表示。例如，13 的 NAF 表示为 1101。

第二步，选取点 P 作为椭圆曲线上的一个基点。

第三步，初始化结果点 Q 为无穷远点（椭圆曲线上的零点）。

第四步，从 NAF 表示法的最高位开始，按照如下三种情况逐位处理

- （1）如果遇到 1，则将结果点 Q 与基点 P 相加。
- （2）如果遇到 -1，则将结果点 Q 与基点 P 相减。
- （3）如果遇到 0，则不进行操作。

第五步，继续处理下一位，直到处理完所有位。

第六步，最终得到的结果点 Q 即为点乘的结果。

例如，假设我们要计算 $5P$ 的点乘，其中 P 是椭圆曲线上的一个点，NAF 表示为 101。具体步骤如下：

第一步，初始化结果点 Q 为无穷远点。

第二步，从 NAF 表示的最高位开始：

- （1）遇到 1，将 Q 与 P 相加，得到 $Q=Q+P$ 。
- （2）遇到 0，不进行操作。
- （3）遇到 1，将 Q 与 P 相加，得到 $Q=Q+P$ 。

最终得到 $5P$ 的结果。

NAF 点乘可以减少点乘运算中的乘法操作次数，提高计算效率，特别适用于椭圆曲线密码学等领域。

2.5.2 雅可比坐标系

在仿射坐标系中进行点加和倍点时，都会涉及有限域上的除法，即域运算层的模逆运算，而域运算层最复杂的运算就是模逆运算。为了加速运算，研究人员对不同坐标系下的点加和倍点运算的计算量进行了对比，如表 2-2

所示。

表 2-2 不同坐标系计算量对比

	仿射坐标	射影坐标	雅可比坐标
点加	$1I+2M+1S$	$12M+2S$	$12M+4S$
倍加	$1I+2M+2S$	$7M+5S$	$4M+6S$

表 2.2 中, I 为模逆, M 为模乘, S 为模平方。可见射影坐标和雅可比坐标都避免了求逆运算, 综合看来雅可比坐标相对射影坐标计算量更小。

由于雅可比坐标系的点加和倍点都不涉及模逆运算, 所以整个点乘过程只需进行雅可比坐标系到仿射坐标系转换过程时的一次模逆, 所以坐标系转换对模逆计算量的减少非常显著。

1) 雅各比坐标系的定义在椭圆曲线密码学中, 一般使用雅各比坐标系表示椭圆曲线上的点。雅各比坐标系中的一个点由三个坐标表示: (X,Y,Z) , 其中 X, Y, Z 是有理数。仿射坐标系中椭圆曲线上的点 (x, y) 可以通过雅各比坐标系表示为 (X, Y, Z) , 其中 $x=X/Z^3, y=Y/Z^3$ 。

2) 雅各比坐标系的特点

(1) 简化点加法运算: 在雅各比坐标系中, 点的加法运算可以通过一系列的加法、减法、乘法和平方运算来实现, 从而避免了椭圆曲线上的除法运算, 提高了计算效率。

(2) 无穷远点的表示: 雅各比坐标系中的无穷远点通常表示为 $(0, 1, 0)$, 在点加法运算中起到类似于零元素的作用。

(3) 点的标量乘法: 雅各比坐标系也适用于点的标量乘法运算, 可以通过一系列的点加法和点倍乘来实现。这在椭圆曲线密码学中经常被使用。

(4) 转换到仿射坐标系: 雅各比坐标系中的点可以通过一定的变换转换为仿射坐标系中的点, 进行其他操作, 如验证、加密等。

假设有一个椭圆曲线上的点 P , 其在雅各比坐标系中表示为 (X_P, Y_P, Z_P) , 另一个点 Q 在雅各比坐标系中表示为 (X_Q, Y_Q, Z_Q) 。那么可以通过雅各比坐标系中的点加法规则计算 $P+Q$ 的结果。

雅各比坐标系在椭圆曲线密码学中具有重要的作用, 能够简化点的运算, 提高计算效率, 同时也便于实现各种密码算法。

2.5.3 Montgomery 模乘

由表 2.2 可以得出,点加和倍点运算中最常见的就是模乘运算。模乘运算是 SM2 算法的域运算层使用率最高的运算,在 SM2 算法实现中,模乘运算的乘数是两个 256 位的整数,朴素算法会在计算过程中得出 512(256 + 256) 位的乘积,等待进行大数取模,大数取模在硬件实现中采用试除法,效率极低。所以模乘运算的速度是影响签名、验签算法的瓶颈。

Montgomery 算法的思想是将除法运算转化为位运算来简化模乘。算法实现分为乘法和约减两个阶段。

Montgomery 模乘算法的步骤如下:

1) 选择参数:

选择模数 N 和参数 R , 其中 N 是模数, $R=2^k$, 使得 N 和 R 互质。

2) 计算参数 R^{-1} , 既 $R \cdot R^{-1} \equiv 1(\text{mod } N)$

3) 给定两个整数 A 和 B , 可按公式 (2-15) 计算它们的 Montgomery 乘积。

$$T \equiv A \cdot B \cdot R^{-1}(\text{mod } N) \quad (2-15)$$

4) Montgomery 约减: 如果 $T \geq N$, 则 $T=T-N$ 。

5) 返回结果:

返回 T 作为模乘的结果。

2.6 FPGA

FPGA 是一种由查找表(Look up Table)、触发器(Flip-flop)、Block RAM (Random Access Memory)和寄存器(Register)等器件集成的可编程逻辑芯片。其可编程特性是指可以根据需要实现的功能使用硬件描述语言 Verilog 将 FPGA 配置成相应的电路。FPGA 的开发一般按照 Verilog 代码编写、功能仿真(Simulation)、逻辑综合(Synthesis)、实现(Implementation)和生成比特流(Generate Bitstream)五个流程进行^[9]。

2.7 Git 简介和基本指令

Git^[10]是一个分布式版本控制系统,最初由 Linus Torvalds 为管理 Linux 内核开发而设计。它具有高效的分支管理、版本控制和协作功能,被广泛应

用于软件开发项目中。Git 的核心理念是分布式版本控制，每个开发者都可以拥有完整的代码仓库，并可以在本地进行版本控制和提交。

Git 的基本使用方法：

初始化仓库：使用 `git init` 命令在当前目录初始化一个 Git 仓库。

添加文件：使用 `git add <file>` 命令将文件添加到暂存区。

提交更改：使用 `git commit -m "commit message"` 命令将暂存区的更改提交到本地仓库。

查看状态：使用 `git status` 命令查看工作区、暂存区和本地仓库的状态。

查看历史：使用 `git log` 命令查看提交历史记录。

创建分支：使用 `git branch <branch_name>` 命令创建新的分支。

切换分支：使用 `git checkout <branch_name>` 命令切换到指定分支。

合并分支：使用 `git merge <branch_name>` 命令将指定分支合并到当前分支。

远程操作：使用 `git remote add origin <remote_URL>` 命令添加远程仓库，`git push origin <branch_name>` 命令推送本地分支到远程仓库。

3 研究内容、预期目标和研究方法

3.1 研究内容

椭圆曲线密码算法虽然相对 RSA 等其他公钥算法，在密钥长度，安全性和计算效率方面都具有明显的优势，但在物联网等资源有限的环境中，其原始的标量乘（倍点）计算方法仍然不够有效，因此，本课题研究 ECC 中标量乘计算的加速方法及 C/C++实现。

具体的研究内容如下：

- 1) 椭圆曲线密码算法简介：研究椭圆曲线密码算法的基本原理和应用领域。
- 2) 椭圆曲线标量乘计算和加速方法：深入了解椭圆曲线上的标量乘运算方法，包括传统的倍点算法和 Montgomery 算法等，并分析当前标量乘计算方法在资源有限环境下的局限性和不足之处。
- 3) C/C++实现：基于 SM2 和 SM3 算法设计并实现椭圆曲线标量乘计算的加速算法，使用 C/C++语言编写。
- 4) 性能评估与比较：进行实验评估，比较不同加速方法再计算速度等性能表现。
- 5) 硬件实现：挑选一种加速方法使用 verilog 语言进行实现。

3.2 预期目标

- (1) 分析现有倍点加速方法的基本原理和优缺点；
- (2) 通过实验对比主流 ECC 加速方法；
- (3) 选择一种加速方法进行硬件实现。

3.3 研究方法

使用 SM2 算法作为基本框架，分别实现常见的多种 ECC 标量乘加速方法，进行实验对比，并挑选一种加速方法使用 verilog 语言进行实现。

4 研究进度安排

1 周，确定题目和内容，了解相关知识；

2-4 周，学习椭圆曲线密码算法，学习实验设计方法，完成选题报告；

5-8 周，配置 Git，分析现有倍点加速方法的基本原理和优缺点。

9-12 周，过实验对比主流 ECC 加速方法，选择一种加速方法探索其硬件实现，对系统性能评估测试；

13-15 周，总结前期工作，完成毕设论文撰写，准备毕业答辩。

学生本人签字：

年 月 日

参考文献

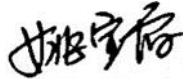
- [1] Hankerson D , Menezes A J , Vanstone S A . Guide to Elliptic Curve Cryptography[J]. Springer-Verlag New York, Inc. 2004.
- [2] Kaliski B S. IEEE p1363: A standard for rsa, diffie-hellman, and elliptic-curve cryptography[C]//Security Protocols: International Workshop Cambridge, United
- [3] Kingdom, April 10–12, 1996 Proceedings 4. Springer Berlin Heidelberg, 1997: 117-118. Morain F, Olivos J. Speeding up the computations on an elliptic curve using addition-subtraction chains[J]. RAIRO-Theoretical Informatics and Applications, 1990, 24(6): 531-543.
- [4] Solinas J. Efficient arithmetic on Koblitz curve[J]. Designs, Codes and Cryptography, 2000:195-249.
- [5] Blakely G R. A computer algorithm for calculating the product AB modulo M [J]. IEEE Transactions on Computers, 1983, 100(5): 497-500.
- [6] Montgomery P L . Modular multiplication without trial division[J]. Math of Computation, 1985, 44(170):519-521.
- [7] Bernstein D J, Lange T. Faster addition and doubling on elliptic curves[C]//Advances in Cryptology–ASIACRYPT 2007: 13th International Conference on the Theory and Application of Cryptology and Information Security, Kuching, Malaysia, December 2-6, 2007. Proceedings 13. Springer Berlin Heidelberg, 2007: 29-50.
- [8] Awaludin A M, Larasati H T, Kim H. High-speed and unified ECC processor for generic Weierstrass curves over $GF(p)$ on FPGA[J]. Sensors, 2021, 21(4): 1451.
- [9] Koc C , Acar T , Jr B . Analyzing and Comparing Montgomery Multiplication Algorithms[J]. Micro, IEEE, 1996, 16(3):26-33.
- [10] Loeliger, Jon, and Matthew McCullough. "Version Control with Git: Powerful tools and techniques for collaborative software development." O'Reilly Media, 2012.

指导教师意见

韦效伟同学针对椭圆曲线密码算法中倍点运算的加速方法查阅了相关文献，并对主要的加速运算方法进行了分析和对比。研究内容略显笼统，研究目标明确，研究方法合理。选题与专业方向一致，具有一定理论意义和应用价值，满足信息安全专业本科毕业设计选题要求。

选题报告内容完整，结构合理，书写基本规范。

请对研究内容进一步细化，并按照研究目标尽快开展各项研究工作。

指导教师签字: 

2024 年 3 月 21 日

