椭圆曲线密码高效软件实现技术研究进展

董建阔[1),5)] 刘 哲[2)] 陆 盛[1)] 郑昉昱[3)] 林璟锵[4)] 肖 甫[1)] 葛春鹏[2)]

[1)](南京邮电大学计算机学院 南京 210023)

[2)](南京航空航天大学计算机学院 南京 211106)

[3)](中国科学院信息工程研究所 北京 100093)

[4)](中国科学技术大学网络空间安全学院 合肥 230026)

[5)](郑州信大先进技术研究院 郑州 450001)

摘 要 伴随云计算技术与物联网技术的快速发展,用户敏感数据呈现爆发式增长,为保障网络中用户隐私数据的安全,国家相继出台了以《密码法》为核心的一系列法律法规,进一步明确密码应用的规范要求.无论是以云计算为代表的服务侧,还是以物联网为代表的终端侧,结构复杂的公钥密码的计算能力都面临极大的挑战.椭圆曲线算法相比于传统的 RSA 密码算法,具有更短的密钥长度,在计算速度、资源存储、数据带宽等方面具有重要的优势,可用于实现密钥交换、数字签名、公钥加密等密码原语,是当前应用最为广泛的公钥密码技术之一.本文通过简要分析服务侧与终端侧两种不同的应用场景,明确端云两侧在软硬件、密码算法需求等方面所存在的巨大差异,归纳了各类椭圆曲线密码算法标准与硬件开发平台参数.基于上述内容,本文总结了椭圆曲线密码的高效软件实现技术研究进展,着重介绍了国产椭圆曲线密码的研究现状,并展望了椭圆曲线密码算法实现的未来发展趋势.

关键词 椭圆曲线密码;物联网;云计算;公钥密码;密码工程

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2023.00909

Research Progress on Efficient Software Implementation of Elliptic Curve Cryptography

DONG Jian-Kuo[1),5)] LIU Zhe[2)] LU Sheng[1)] ZHENG Fang-Yu[3)] LIN Jing-Qiang[4)]
XIAO Fu[1)] GE Chun-Peng[2)]

[1)](*School of Computer Science, Nanjing University of Posts and Telecommunications, Nanjing* 210023)

[2)](*School of Computer Science, Nanjing University of Aeronautics and Astronautics, Nanjing* 211106)

[3)](*Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093)

[4)](*School of Cyber Security, University of Science and Technology of China, Hefei* 230026)

[5)](*Zhengzhou Xinda Institute of Advanced Technology, Zhengzhou* 450001)

Abstract With the rapid development of cloud computing technology and Internet of Things technology, user-sensitive data shows explosive growth. In order to ensure the security of user privacy data in the network, the state has successively issued a series of laws and regulations with the "Cryptography Law" as the core, which defines the standard requirements of cryptography application. Whether it is the service side represented by cloud computing or the terminal side represented by the Internet of Things, complex public-key cryptography computing power is facing great challenges. Public key cryptography algorithm is one of the core algorithms of cryptography,

收稿日期:2021-12-27;在线发布日期:2022-10-09.本课题得到国家自然科学基金(62132008)、江苏省自然科学基金(BK20220388)、江苏省高等学校基础科学(自然科学)研究面上项目(22KJB520004)、中国博士后科学基金(2022M711689)、CCF-华为胡杨林基金、河南省网络空间态势感知重点实验室开放课题基金(HNTS2022026)和国家密码发展基金(MMJJ20180105)资助.**董建阔**,博士,讲师,中国计算机学会(CCF)会员,主要研究方向为高性能密码技术.E-mail:djiankuo@njupt.edu.cn.**刘 哲**(通信作者),博士,教授,主要研究领域为密码工程.E-mail:zhe.liu@nuaa.edu.cn.**陆 盛**,学士,主要研究领域为高性能密码实现技术.**郑昉昱**,博士,助理研究员,主要研究领域为应用密码学.**林璟锵**,博士,教授,博士生导师,主要研究领域为系统安全.**肖 甫**,博士,教授,博士生导师,主要研究领域为无线传感网与网络空间安全.**葛春鹏**,博士,副研究员,主要研究领域为密码学.

which is widely used in digital signature, key exchange, public key encryption and other cryptographic primitives. In 1987, ECC(Elliptic Curve Cryptography) algorithm based on elliptic curve discrete logarithm problem first came into the view of researchers. Compared with the traditional RSA algorithm, the elliptic curve algorithm has a shorter key length and has important advantages in computing speed, resource storage, data bandwidth, etc. It can be used to realize key exchange, digital signature, public-key encryption, and other cryptographic primitives. It is one of the most widely used public-key cryptography technologies. How to implement elliptic curve public key cryptosystem with high computational complexity safely and efficiently has always been a challenging topic for cryptographers. With the growing demand for application cryptography, the research on the implementation of high-performance elliptic curve public key cryptography algorithms presents a new development trend, but also brings about corresponding problems. In the cloud computing scenario, the server hardware itself has the characteristics of high performance computing capacity, large memory, etc. In the scenario of the Internet of Things, due to the massive deployment of terminal equipment, the hardware cost is limited, and the ROM space resources of a large number of low-end processor devices are seriously insufficient. However, a large number of application scenarios (such as intelligent transportation, intelligent medical treatment, etc.) have strict requirements on the delay of cryptographic algorithms. On the server side, the main requirements of cloud computing oriented elliptic curve cryptography algorithm computing technology are: high throughput, low latency, and high security; On the terminal side, the requirements of the elliptic curve cryptography algorithm for the Internet of Things are: lightweight, high-performance, and high security. Due to the differences between hardware platforms and requirements, they must be studied separately to ensure that both sides of the end cloud can meet the current development trend of the Internet of Everything. This article briefly analyzes two different application scenarios on the server side and the terminal side and clarifies the huge differences in software and hardware and cryptographic algorithm requirements on both sides of the terminal and cloud. Furthermore, the standards of various elliptic curve cryptography algorithms and the parameters of the hardware development platform are summarized. Based on the above content, this paper summarizes the efficient software implementation technology of elliptic curve cryptography, mainly introduces the research status of domestic elliptic curve cryptography, and looks forward to the future development trend of elliptic curve cryptography algorithm implementation.

Keywords elliptic curve cryptography; Internet of Things; cloud computing; public key cryptography; cryptographic engineering

1 引 言

伴随云计算技术与物联网技术的快速发展，人们正逐步迈向万物互联的智能世界. 互联网金融、移动支付等应用对于密码算法尤其是计算复杂度高的公钥密码算法的性能提出了严峻的挑战. 一方面，在以云计算技术为代表的服务侧，需要面对上亿级用户产生的海量数据，服务端需要在有限时间内完成这些用户的身份认证与数据保护，庞大的用户签名请求对云计算同样也是极具挑战的难题. 另一方面，在以物联网技术为代表的终端侧，物联网嵌入式芯片或移动设备受到成本限制，其计算资源往往严重不足，在 CPU 主频、内存资源甚至供电等方面存在一定的瓶颈，将结构复杂的公钥密码算法部署在资源受限的嵌入式设备上，在空间、性能、功耗等方面都面临挑战.

1976 年，公钥密码体制由美国斯坦福大学 Diffie 与 Hellman 两人在文献[1]中最早提出；紧接着，RSA 公钥密码算法在 1977 年被公布[2]，并基于三位数学

家的名字(Rivest、Shamir、Adleman)进行命名.根据美国国家标准与技术研究院(National Institute of Standards and Technology,NIST)相关标准[3],为了保障数据安全,推荐使用的算法安全强度不小于 112 比特长度,与之对应 RSA 公钥密码的密钥长度为 2048 比特;相比于对称密码算法,RSA 密码算法在存储空间、计算复杂度等方面存在较大的劣势.1987 年,基于椭圆曲线离散对数的 ECC(Elliptic Curve Cryptography)算法[4]首次进入科研人员的视野,相比于 RSA 公钥密码算法,椭圆曲线密码具有更短的密码长度与更快的计算速度,受到学术界与工业界的广泛关注.自 20 世纪 90 年代以来,我国许多学者对椭圆曲线密码体制展开研究[5-8],ECC 公钥密码相比 RSA 公钥密码算法,在密钥大小、生成效率与签名性能等方面有着显著的优势.

在我国,商用密码是密码的重要组成部分,用于保护不属于国家秘密的商业信息,被广泛应用于金融、通信、工业等多种商业领域.自 1999 年《商用密码管理条例》发布以来,经过二十余年的发展,国产密码算法从理论到技术等多个层面取得了重要的研究成果,经历了从无到有、从国家标准到国际标准的角色转变.当前,我国已具备完整的商用密码算法体系,包括对称密码、公钥密码、哈希函数等.2010 年,国家密码管理局发布了[9]椭圆曲线公钥密码 SM2.经过近些年来的不断发展,国产密码算法高性能实现技术已经取得了重要的进展,但相比国外通用密码算法,部分领域存在一定的差距.进一步,伴随密码应用需求快速增长,高性能国产公钥密码算法实现研究呈现了新的发展趋势,同时也带来了相应的问题.

面对公钥密码在计算性能方面存在的瓶颈,大量的密码学家对公钥密码的高性能实现技术展开了研究.其中,基于各类芯片平台的密码软件实现技术具有成本低、可移植性强、易部署的优势,是密码工程研究的热点.在以云计算为代表的服务器端,基于高性能 CPU、GPU(Graphics Processing Unit)芯片的 ECC 实现技术不断刷新吞吐峰值;同样的,在资源受限的嵌入式平台,包括各类物联网 CPU 以及嵌入式 GPU 等,椭圆曲线密码算法在空间和性能上也不断涌现新的优化技术.

2　椭圆曲线密码

经过近几十年的快速发展,椭圆曲线的理论研究趋于成熟,椭圆曲线密码算法现已成为公钥密码研究的新宠,其标准化工作受到了各方的重视,包括 NIST、国家密码局、RFC 等都已公布基于 ECC 的公钥密码算法标准;椭圆曲线相比于传统的公钥算法优势明显,在工业界也得到广泛的应用,文献[10]指出,椭圆曲线公钥密码算法可应用于移动互联、物联网、安全认证、RFID 等领域.

2.1　基本定义

相比于传统的 RSA 公钥密码算法,椭圆曲线密码学是一种较新的公钥密码技术,最远追溯到 1987 年[4],在 2005 年后,得到广泛推广应用.如表 1 所示,相同的安全强度下,椭圆曲线在密钥长度上具有显著的优势,且伴随安全强度的增大,优势愈发明显.在相关综述文献[10]中,作者指出世界上访问前 100 网站中,超过 70%的网站使用 HTTPS(Hypertext Transfer Protocol Secure)技术用以保障网络数据传输,其中有 69 个网站使用椭圆曲线密码技术用于密钥协商或用户的签名认证,可见椭圆曲线密码在工业界应用的广泛性.

表 1　ECC 与 RSA 安全强度对比

ECC 密钥长度	RSA 密钥长度	对比	安全强度
160	1024	1∶6	80
256	3072	1∶12	128
384	7680	1∶20	192
512	15360	1∶30	256

椭圆曲线一般表示为 Weierstrass 格式[11],本节我们主要介绍三种常见的椭圆曲线表示方式:Weierstrass 曲线、Montgomery 曲线与 Edwards 曲线.定义在有限域 GF(p)上的 Weierstrass 曲线的具体表示如下:

$$E: y^2+a_1xy+a_3y=x^3+a_2x^2+a_4x+a_6,$$

其中,参数需要满足$(a_1,a_2,a_3,a_4,a_6)\in \mathrm{GF}(p)$,上述方程一般被称作长 Weierstrass 方程,进一步可以表示为简化的 Weierstrass 方程,具体为

$$E: y^2=x^3+ax+b,\text{且 } 4a^3+27b^2\neq 0.$$

Montgomery 曲线是由密码学家 Montgomery 在 1987 年定义的一条曲线[12],同样也被广泛应用于密码学中,例如基于 Curve25519 的密钥交换算法,其具体的表示形式为

$$E: by^2=x^3+ax^2+x,\text{且 } b\times(a^2-4)\neq 0.$$

Edwards 曲线[13-14]由 Edward 在 2007 年定义,其表示的方程如下:

$$E: x^2+y^2=1+dx^2y^2,\text{且 } d\in \mathrm{GF}(p)\backslash\{0,1\}.$$

通过对比 Edwards 曲线与 Twisted Edwards 曲

线，Edwards 曲线是一种特殊的 Twisted Edwards 曲线，且满足($a=1$).

椭圆曲线密码算法的标准化经过几十年的发展，已取得了重要的成果：(1) NIST 系列[15]. NIST 的椭圆曲线标准化工作启动较早，早在 1999 年，NIST 推荐了 15 条具有不同安全强度的标准椭圆曲线，其中包含 5 个不同长度的素数域(分别是 192、224、256、384 和 512 比特)和二进制域(分别是 163、233、283、409 和 571 比特)；(2) SM2 曲线[9]. SM2 椭圆曲线密码算法是我国具有自主知识产权的商用公钥密码算法，在 2010 年由国家密码管理局制定并发布. SM2 基于 256 比特长度的素数域(梅森素数)，在有限域层面可利用快速约减方案，性能上具有一定的优势. 可应用于密钥交换、数字签名、公钥加密(配合 SM4 国密算法). 2018 年，国密局宣布，我国 SM2 椭圆曲线成功入选 ISO 国际标准[16]，为我国的国密算法国际化迈出了重要一步；(3) 其他相关组织也定义了大量的椭圆曲线密码标准，包括美国国家标准化组织提出的椭圆曲线签名算法 ANSI1 X9.62[17]、德国的 Brainpool 曲线[18]、IEEE 的椭圆曲线密码标准 IEEE P1363[19]、韩国信息安全局的 KCDSA[26] 等.

早在 2006 年，著名的密码学家 Bernstein 基于梅森素数 $p=2^{255}-19$ 设计了 Curve25519 曲线[20]，该曲线是一条 Montgomery 曲线，安全强度为 128 比特，在密钥交换算法 X25519 设计中采用了单坐标计算方案，在 Montgomery Ladder 方案的加持下，其性能与安全性都得到了显著的提升. 2013 年，著名的斯诺登“棱镜门”事件爆发，美国 NIST 标准中椭圆曲线密钥生成依赖的随机数函数 Dual_EC_DRBG 可能存在后门，美国政府制定的椭圆曲线标准安全性受到广泛的质疑. 在此基础上，反观 Curve25519 算法参数来源明确，在性能与安全性相比 NIST-P256 同样具有明显的优势，该算法很快成为工业与学术界追捧的热点.

随后，不断有新的椭圆曲线方案被提出：2014 年，Bernstein 在文献中，设计了安全强度更高的 Curve41417 曲线；2015 年 Mike Hamburg 选取了有限域 $p=2^{448}-2^{224}-1$，基于 Twisted Edwards 曲线提出了 Ed448 算法[21]；同年，微软密码团队提出了 Four$\mathbb{Q}$ 曲线[22]，宣称其性能是 NIST 标准的 4～5 倍，是 Curve25519 性能的 2～3 倍；2016 年 12 月 IRTF 也宣布具有蒙哥马利曲线形式的 Curve25519 和 Curve448 成为 IETF 椭圆曲线标准(RFC7748[23])，同年，IRTF 公布了新的密码签名方案 RFC8032[24]，EdDSA 包含了基于 Ed25519/448 两条 Twisted Edwards 椭圆曲线的签名方案；最新版本 TLS 1.3[25] 也集成了这些特殊类椭圆曲线的实现，如表 2 所示，我们列出了几种常见的椭圆曲线密码标准与曲线的类型.

表 2 常见的椭圆曲线类型

算法类型	安全强度/bit	曲线类型
NIST P-256[15]	128	Weierstrass 曲线
SM2[16]	128	Weierstrass 曲线
Curve25519[23]	128	Montgomery 曲线
Curve448[23]	224	Montgomery 曲线
Ed25519[24]	128	Twisted Edwards 曲线
Ed448[24]	224	Twisted Edwards 曲线
Four$\mathbb{Q}$[27]	128	Twisted Edwards 曲线

2.2 椭圆曲线密码的需求现状

网络安全协议一般需要通信双方进行多次交互，例如 ECDH 协议，通信双方都要完成一次固定基点的标量乘法与一次非固定基点的标量乘法计算，服务侧与终端侧都要支持部署椭圆曲线密码算法协议栈. 如图 1 所示，由于服务侧与终端侧在硬件条件、算法需求等方面存在巨大的差异，我们将对两个应用场景进行分开分析. 研究者在进行密码算法研究过程中，需要根据不同的需求场景、硬件条件、算法需求，分别展开算法性能优化技术研究.

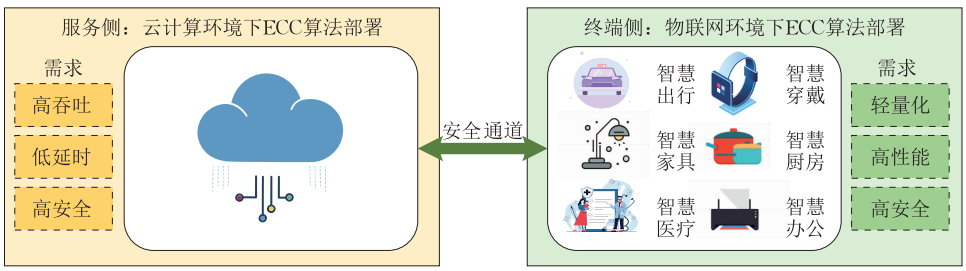

图 1 部署现状与需求特点

服务侧. 云计算场景下椭圆曲线密码的部署，面对海量的终端设备时，公钥密码计算复杂度高，经常会成为服务端处理的瓶颈. 例如，2020 年阿里“双十一”购物节[28]，交易峰值吞吐高达 58 万次每秒，若基于商用国密算法进行签名认证服务，需要进行一次验签操作完成用户认证、一次密钥交换操作用于数据保护、一次签名操作以确认支付完成，这只是理想交易模型，在实际实现中支付流程复杂度远远高过于此，在不考虑负载均衡的理想情况下，至少需要几十台甚至上百台商用签名服务器同时工作，才能满足支付宝峰值性能要求. 同样，为了保障用户体验，对服务端密码计算的延时性也有着较高的要求. 随着互联网规模的不断扩张，上述场景越来越普遍，例如：春运 12306 网站售票系统、微博热点事件等. 这类场景对于商用密码算法的主要需求集中在计算吞吐与安全实现上，兼顾延时用以保证用户体验.

终端侧. 物联网经过 10 多年的快速发展，已经渗透到各行各业：智慧出行、智能家居、智慧医疗等. 根据中国信通院 2019 年的统计与预测，全球的物联网链接数量每年都以 10 亿的数量级增长；放眼中国，物联网发展更为迅速，我国物联网设备数量每年增长超过 1 亿台. 物联网设备有着大规模部署、平台各异、资源不足等特点. 大量的物联网设备的存储资源严重不足，个别甚至在 1 MB 以内：例如，AVR 芯片仅有不到 10 K 字节的 RAM 与 100 KB 字节的 ROM. 有限的存储空间还需部署上层应用、操作系统等通用软件资源，留给密码算法的空间严重不足，面向物联网设备的椭圆曲线公钥密码算法实现研究，在保证安全实现的基础上，首先需要考虑算法轻量化问题. 由于物联网设备一般为少核或者单核处理器，算法吞吐效率与计算延时成反比，故我们将提高物联网设备密码计算效率统称为追求密码的高性能实现.

通过上述描述，在服务端，面向云计算的椭圆曲线密码算法计算技术的主要需求为：高吞吐、低延时、高安全；在终端侧，面向物联网的椭圆曲线密码算法的需求为：轻量化、高性能、高安全. 由于硬件平台与需求存在差异，两者必须分别展开研究，以确保在端云两侧都能满足当前万物互联的发展趋势.

2.3　实现平台

端云两侧在物理硬件方面存在较大的差异，服务侧计算资源会更加的充实，硬件一般具有多核心、高主频、大内存的特点，为了追求更高的性能，还会增加协处理器以补充计算能力. 在设备部署方面，一般具有专门的机房环境以及不间断电源支持，攻击者一般很难从物理层面与设备直接接触. 终端侧与此不同，一般为资源受限的物联网设备，受到成本限制，个别设备甚至仅有几十 KB 的存储资源；部署环境同样参差不齐，大量的设备使用电池供电，算法优化需首先考虑轻量化的需求，在此基础上，进一步追求高性能、高安全.

2.3.1　高性能计算平台

面对密码算法需求的爆发式增长，各类高性能计算硬件设备逐渐被应用于密码计算，具体包括 CPU、GPU、Intel Phi 处理器[29]等，本章我们主要介绍最为常用的两类计算平台：高性能 CPU 芯片与 GPU 芯片，其中 GPU 主要围绕 NVIDIA 公司发布的相关硬件产品.

高性能 CPU 芯片

本节主要介绍的是高性能 CPU 处理器芯片，该类芯片主要部署在台式机、服务器以及高性能移动终端设备上，一般具有高主频、高内存、多核等特点，相比于依赖电池供电的嵌入式设备，这类 CPU 芯片对于性能的要求更加严格. 例如：Intel Core 系列、志强系列以及基于 ARM V8 打造的骁龙 800 系列、华为麒麟系列等. 表 3 例举了一些当前主流的 CPU 芯片及其对应的核心数.

表 3　高性能 CPU 处理器举例

CPU 型号	指令集类别	平台	核心数
Intel Core i7-9800[32]	AVX 指令	桌面级	8
Intel Xeon 8360Y[33]	AVX 指令	服务器	36
AMD 霄龙 7642	AVX 指令	服务器	48
AMD 锐龙 7[34]	AVX 指令	桌面级	8
骁龙 888[35]	NEON	终端级	8
麒麟 990[36]	NEON	终端级	8

为了追求更好的密码计算性能，上述高性能计算 CPU 芯片为构造相对简单的对称密码算法（AES 等）提供底层的密码原语，例如在 Intel 部分平台支持的 AES-NI 指令集，能够大幅度提高该算法的整体性能，相比软件实现具有数量级的提升效果；ARM 同样具备类似的 AES 安全拓展指令集，在 ARM v8.2 后，还支持国密 SM4 算法的硬件底层加速.

但是，对于复杂的椭圆曲线密码算法，芯片暂时不支持包括标量乘法、大整数乘法等密码计算原语，开发者只能利用通用指令集进行加速，高性能 CPU 芯片会基于多核的特点提供高效的并行指令集，一般为单指令多数据（Single-Instruction-Multi-Data，

SIMD)并行指令架构.例如,Intel 平台的高级矢量拓展(AVX)指令[30]、ARM 平台的 NEON 指令集[31]等.Intel AVX 指令集已有多年的发展历史,最早源于 1996 年的 MMX 向量指令集,后衍变为 SSE 系列指令集,2008 年引入了 AVX-128 向量并行指令集,至今发展为 AVX-512 向量指令.值得一提的是,AVX 指令集作为一个通用并行指令集,不仅适用于 Intel 平台的 CPU,其同样适用于 AMD 平台的 CPU.并行指令集能够大幅度提升算法的执行效率,例如基于 AVX-128 指令,单条指令可完成 4 个定点数加法运算,普通软件实现需要执行 4 次,并行指令集的加速效果是毋庸置疑的.基于高性能 CPU 加速椭圆曲线密码的主要挑战在于:如何将复杂的椭圆曲线密码算法适配到 CPU 的并行指令中,并在保障算法安全的基础上,追求极致的计算性能.

高性能 GPU 芯片

早期的 GPU 主要用于图像加速,主要是为了弥补 CPU 在图像处理方面的不足.与 CPU 平台的架构不同,图形处理器将更多的硬件空间用于算术逻辑单元,而不是控制逻辑单元,GPU 具有大量的运算核心用于像素点的并行计算.NVIDIA 是 GPU 显卡制造的龙头,从 2010 年至今,GPU 从 Fermi 架构[37]到最新 Ampere 架构,其单精度浮点数计算性能增长超过了十倍.显卡具有高性能的计算能力逐渐引起研究人员的关注,2006 年,NVIDIA 推出通用并行计算架构[38](Compute Unified Device Architecture,CUDA),目前 CUDA 支持 C/C++、Fortran、Java 等多种高级计算机语言,这极大降低 GPU 开发者的学习门槛,方便跨学科领域快速掌握 GPU 开发技能.

在 NVIDIA 的硬件架构中,最上层的组织单元被称为流处理器(Streaming Multiprocessors,SMs),流处理器内部包含一定数量的计算核心(core)与缓存单元(Cache),当 CPU 调用 GPU 的 CUDA 程序时,计算任务会以线程块(Block)的方式分配到 SM 上,每个线程在 SM 被内部分配到各个计算核心上执行;当执行结束后 GPU 将计算结果拷贝到 CPU 内存中.在并行计算的规模上,GPU 可同时并行处理几千个线程,这种并行处理方式一般称为单指令多线程架构(Single Instruction Multiple Thread,SIMT).需要指出的是,GPU 执行过程中采用小端的方式表示数据,为了节省硬件空间,GPU 不存在分支预测与指令的预处理,线程运行的指令被分到每个计算核心,通过并行化的方案执行完成.

受到成本与技术迭代的影响,不同的 GPU 平台拥有不同的硬件计算单元,高性能的双精度浮点数计算仅在旗舰版本和科学计算的显卡上出现,例如 NVDIA Tesla 系列.从 Volta 架构开始,NVIDIA 提供一种专门应用于人工智能计算的硬件资源(Tensor Core),深度学习性能首次达到 100 万亿次(TFLOPS),是上一代 NVIVIDA Pascal 架构的 5 倍以上,表 4 总结了当前主流的 NVIDIA 计算显卡与其支持的硬件计算能力.

表 4 高性能 GPU 处理器举例

GPU 型号	Tensor Core	高性能 DPF	架构	核心数量
NVIDIA GTX 1080[39]	否	否	Pascal	2560
NVIDIA RTX 3080[40]	支持	否	Ampere	8704
NVIDIA Tesla P100[41]	否	支持	Pascal	3584
NVIDIA Tesla V100[42]	支持	支持	Volta	5120
NVIDIA Tesla A100[43]	支持	支持	Ampere	6912

图 2 展示了基于众核 GPU 平台的密码计算模型,首先,CPU 将密码计算需要输入的数据整合并分组处理,利用 GPU 拷贝引擎(Copy Engine)将数据从 CPU 内存拷贝到 GPU 内存;密码操作需要的密钥根据占用的空间大小与密钥性质,可存储在 GPU 的全局内存、纹理内存或者共享内存中;在数据部署完成后,GPU Kernel 启动大量线程,对数据流上的密码请求进行并行计算操作,然后再将并行计算结果拷贝到 CPU 内存中;最后,在 CPU 内存中对来自不同请求的输出数据重新分配,并返回计算结果.

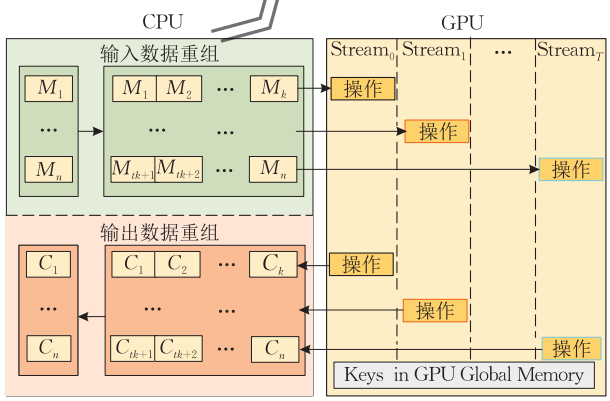

图 2 基于 GPU 平台的密码计算模型

由于硬件设计上的并行原则,流处理器中最小的执行单元被称为一个线程束(warp),单个线程束包含 32 个线程.GPU 对于线程的管理、调度以及执行都以线程束为执行单位.同一个线程束中的各个线程从相同的程序地址开始运行,但是每个线程都

有自己的指令计数器以及数据寄存器，可以单独运行与自由分支. 当一个或者多个线程块被分配到某个 SM 上执行时，线程块中的线程将被分配为多个线程束，这些线程束接受 warp 调度器的管理调度，分时段在流处理器中运行.

线程束中的所有线程每次都会执行同一个指令，当所有线程执行的指令保持一致时，GPU 能够达到最高的运行效率. 如果同一个线程束中线程间执行指令不同，出现条件分支的情况，线程束会分别执行每个线程的分支指令，分支执行过程中会确保无关线程处于“关闭”状态，以确保数据不会被无关指令“污染”. 上述同一个线程束的线程出现指令分支的情景被称为束分叉(warp divergency)，束分叉只会出现在同一个线程束中，不同的线程束指令不同，不会出现上述情况. 束分叉会严重影响 GPU 的执行效率，在 CUDA 算法编程过程中，应该尽量避免条件分支等指令的出现.

2.3.2　嵌入式设备

嵌入式设备相比于服务端情况更加的复杂. 由于生产嵌入式设备厂商的数量众多，芯片在物理形态、系统类型等方面存在较大差异，本节主要介绍两类最为常用的嵌入式芯片：嵌入式 CPU、嵌入式 GPU.

嵌入式 CPU 芯片

不同于传统的 PC 机，基于嵌入式 CPU 芯片的传感设备和移动设备大多是由电池进行供电，一旦被部署，将会持续运行几个月或者一年的时间. 这些嵌入式设备无论是在电池的电力、CPU 计算能力还是内存的容量都非常有限，例如，AVR 和 MSP 芯片的运算频率都是小于 20 MHz 的，ARM 的运算频率也只有 200 MHz 左右；存储空间上面，AVR 和 MSP 芯片都只有少于 10 K 字节 RAM 和 100 K 左右字节的 ROM. 同时还有其他常见的嵌入式 CPU 处理器，例如 AT mega16 的运算性能为 1 MIPS/MHz，ARM Cortex M0+的性能也仅有 0.9 MIPS/MHz，远低于正常 CPU 处理器的计算能力.

物联网或者无线传感网络的嵌入式传感设备经常被部署在无人监管的运行环境下，这些传感设备用来收集信息并且通过无线信号传输到基站，这就使得攻击者很容易接触到无线信号或者直接接触到传感器设备本身，从而使用相关侧信道攻击技术对嵌入式芯片运行的密码算法进行时间攻击、能耗攻击甚至于注入错误攻击，使得节点容易受到操控，网络的传输受到破坏.

软硬结合是算法优化的重要方案之一，基于通用的高级语言(C 语言等)开发的软件，一般在性能挖掘、空间优化方面存在较大的缺陷. 嵌入式平台由于平台种类繁多，各个平台之间参数各异，这为密码算法在各类平台上的优化提出了一定的挑战.

嵌入式 GPU 芯片

伴随着移动互联网与物联网的高速发展，嵌入式 GPU 也逐渐被引入到高性能计算领域中. 嵌入式 GPU 被引入并不是用于图像显示，而是在物联网设备中，缺乏一类具有高能效比的人工智能加速芯片. 针对物联网的人工智能应用需求，NVIDIA 提供了不同性能版本的硬件产品，如 Nano、TX2 等. 表 5 总结了一些当前主流的嵌入式 GPU 及其硬件计算能力.

表 5　嵌入式 GPU 处理器举例

型号	Tensor Core	架构	核心数量	性能 TFLOPS
TX2[44]	否	Pascal	256	1.3(FP16)
Nano[45]	否	Maxwell	128	0.5(FP16)
Xavier NX[46]	支持	Volta	384	21(INT8)
AGX Xavier[47]	支持	Volta	512	32(INT8)

嵌入式 GPU 同样支持 CUDA 编程，可以利用 SIMT 的并行思想进行编程，与传统的 GPU 的主要不同点如下：(1) 嵌入式 GPU 平台具有更少的寄存器资源，同为 Pascal 架构的 TX2，其单个 SM 拥有的寄存器数量是桌面级 GPU GTX 1080 的一半；(2) 嵌入式 GPU 在核心数量上远远少于普通的桌面级 GPU；(3) 嵌入式 GPU 与 CPU 共用同一块内存，但是桌面级 GPU 与 CPU 内存在物理上是隔离的.

嵌入式 GPU 芯片本身一般为信用卡大小，功耗从几瓦到几十瓦，但这类计算平台却有着强大的算术计算能力. 例如，基于图灵架构[48]的 Jetson TX2(Tegra X2、TX2)浮点数计算性能超过 1.3 GFLOPS，在机器视觉、机器人、智能车机等复杂的场景中有着广泛的应用. 相比于 GPU 平台，嵌入式 GPU 在功耗比方面具有巨大的优势，特别适合作为物联网设备或者边缘计算设备的节点.

3　面向服务侧的 ECC 实现技术

在本节中，我们主要介绍在高性能计算平台上，各类椭圆曲线密码的研究现状，为了方便研究，我们将研究对象分为两类：国际通用密码与国产椭圆曲线密码.

3.1 国际通用密码算法实现技术

国际通用公钥密码中,主流的椭圆曲线算法包括 NIST-P 系列、Brainpool 曲线等,受斯诺登事件影响,一些安全性更高的新型曲线受到重视,包括 Curve25519、Ed448 等. 当前,大量研究人员将 GPU、Intel 处理器作为实验平台,从不同层面上完成对椭圆曲线密码算法的实现.

高性能 CPU 平台部署广泛,一般是密码工程研究者首选的实现平台. 2010 年,滑铁卢大学[49]通过优化椭圆曲线的有限域、点算术以及标量乘法,在基于 x86 架构的 Intel 以及 AMD 芯片上完整实现了椭圆曲线点乘的优化操作,速度相比当时的最佳结果提升了 31%. 同时随着一些并行指令集的推出,开发者可以充分利用 AVX2 等 SIMD 并行指令集,从不同的维度并行加速椭圆曲线算法. 值得一提的是,AVX2 指令集是一种通用指令集,可以被运用在 Intel,AMD 等高性能 CPU 上来进行并行加速计算. 现有的密码算法库[50-51]都已经提供对应平台的向量指令(AVX2、NEON 等)加速方案. 但这并没有阻碍研究者进一步挖掘平台计算能力,实现新的性能突破. 2015 年,以色列海法大学团队[52]优化了椭圆曲线密码实现,基于 4 个 64 比特字表示椭圆曲线大整数,对椭圆曲线有限域进行并行加速,结合预计算表技术(窗口大小为 7),完成对 NIST P256 曲线的实现,在当时主流的 Intel Haswell CPU 中,实现性能是 OpenSSL 密码算法库的 2.33 倍. 2015 年,巴西坎皮纳斯大学团队在文献[53]中优化实现了 Curve25519 曲线 Montgomery Ladder 标量乘法,底层采用 AVX2 256 比特长度的向量 SIMD 指令. 例如向量乘法$[a_0b_3, a_0b_2, a_0b_1, a_0b_0]$的计算中,首先使用 BCAST 指令将 a_0 填充到 256 比特长度的寄存器 R_0,向量$[b_3, b_2, b_1, b_0]$保存在 R_1 中,然后利用乘法向量指令 MUL(R_0,R_1)即可完成计算. 2019 年,该团队在文献[54]中进一步优化椭圆曲线计算,并将并行方案一般描述为($n\times m$),其中有限域计算的并行数量为 n,单个向量的并发执行有限域计算的数量为 m(m-way),分别展开对 2-way 与 4-way 实现 Curve25519 与 Curve448 的实现,相比原有的工作性能提升超过 10%,在安全性能方面,利用比特操作(或操作与异或操作)替换了 Montgomery 乘法中的 CSWAP 与 CSMOV 中的 if-else 分支操作,降低了分支操作带来的安全风险,文献最后还探索了基于最新的 AVX-512 并行指令集的椭圆曲线密码实现.

GPU 将更多的芯片面积用于算术计算单元,相比于 CPU,GPU 的算术计算能力得到大幅度提升,密码学研究者利用 NVIDIA 公司提供的 CUDA 编程技术,可充分将 GPU 高性能计算的能力应用到椭圆曲线加速领域. 2012 年,葡萄牙里斯本理工大学团队在文献[55]使用 RNS 剩余数系统,在 GPU 平台上,完成针对 ECC 的大整数乘法实现. RNS 系统由于单个字运算之间相互独立以及乘法数量优势,方便在 GPU 平台中并行实现,但是在大整数高基表示与 RNS 表示之间转化需要耗费大量的运算,因此该文献存在一定的性能缺陷. 伊利诺伊大学芝加哥分校 Bernstein 团队发表文献[56-57],基于 GPU 平台,使用单个线程完成 Montgomery 高基乘法,Montgomery 乘法是一种广泛应用于 GPU 的模乘算法[58],ECC 计算过程中由于模数较小,在延时可接受的范围内,一般使用单个线程完成模乘计算. 大量的文献[59-62]在 GPU 平台上,对椭圆曲线梅森素数的快速约减算法进行优化. 最具代表性的是,2017 年,中科院信工所团队[60]在《TIFS》期刊上发表文章,基于 NVIDIA GTX 780Ti 平台实现每秒 92 万次 NIST P-256 验签操作,其底层的大整数运算采用了 PTX ISA 汇编指令编写,采用了高低进位乘法指令(MUL.LOW/HIGH)的思想完成大整数运算,利用 GPU 高内存的特点存储固定点乘法的预计算表,减少计算复杂度,最后在 CPU 平台上完成椭圆曲线坐标转换的大整数求逆运算,充分发挥 GPU-CPU 联合计算优势,为测试平台性能,该团队还搭建了网络平台,模拟测试了大规模吞吐情况下,椭圆曲线密码加速器的整体性能.

棱镜门事件爆发后,新型椭圆曲线成为密码工程领域的研究热点. 2014 年,法国团队[63]首次在 GPU 平台上实现 Curve25519 点乘算法,由于其使用 OpenGL(Open Graphics Library)作为开发工具,代码通用型较强,可以部署在 NVIDIA、AMD 等多种平台,但是相比于 NVIDIA 的 CUDA 开发,性能存在较大缺陷,其椭圆曲线点乘性能约为每秒 52 万次运算. 2018 年,中科院信工所团队利用 GPU 平台定点数[64]对有限域大整数算法加速,并提出了单轮进位约减方案,在性能上有了新的突破,在同一 GPU 平台上,将 Curve25519 性能提升到每秒 139 万次点乘运算. 2020 年,该团队[65]利用计算类显卡的双精度浮点数计算能力,相比于传统的定点数表示方案,双精度浮点数表示方案存在溢出与比

特位操作指令缓慢等问题，密码算法设计更加复杂，该团队针对 Curve25519 提出了单字 51-bit 的表示方案，并结合定点数比特位操作完成了快速约减方案，最后在NVIDIA Tesla P100显卡上，将 Curve25519 标量乘法的性能提升到每秒 619 万次操作，创造了新的性能记录.

3.2　国产椭圆曲线密码实现

在云计算的场景下，服务端设备硬件本身具有高性能计算、大容量内存等特点. 大多数参考文献面向密码高速计算的需求，利用 GPU、Intel 等通用高性能计算器件，对涉及性能瓶颈的国产公钥密码算法进行了广泛的研究.

国产 SM2 公钥算法标准化时间较早，且应用最为广泛，是学术界与产业界的研究热点. 2014 年，清华大学团队[66]在 ASIC 芯片上实现了高吞吐的 SM2 公钥密码算法，在有限域层面，通过多轮测试与性能对比，最终采用单轮运算的 256-bit 的乘法器以及 radix-4 方案优化实现模逆运算；在保障正确的前提下，重构点加运算与倍点运算的执行流程；利用 NAF 标量乘法实现方案，有效降低点加算术的数量，并采用两级流水线技术加速算法性能，该实现创造了当时最高的性能记录. 2014 年，中国科学院信工所团队[62]在 GPU 平台上，实现了一种面向 GPU SM2 实现的大整数模乘算法，旨在最小化算法的加法指令数量，所需的加法指令数量从传统算法的 $O(n^2)$ 降低到 $O(n)$，在 GTX Titan 上实现的模乘和模平方计算的吞吐率分别达到每秒 33 亿次和每秒 59 亿次，性能提升了近 43%，但是该文献在完成有限域层的优化后，并未完整实现点乘运算，仅提供了理论性能值.

2019 年，上海交通大学团队[67]在 Intel 处理器上，利用现代处理器拓展指令集（$madx$，$adcx$，$adox$）与大容量 Cache 的优势，从底层到上层（大整数运算、模运算、点运算、标量乘法等）对 SM2 算法进行优化实现，该工作还优化了随机数发生器，进一步提升了 SM2 算法的整体性能，相比于 OpenSSL 算法实现具有较大的优势. 2020 年，南京航空航天大学团队[68]在 Intel 处理器上，利用 SIMD（单指令多线程技术）AVX2 并行技术，优化了 SM2 公钥密码 Weierstrass 曲线的 Co-Z 坐标计算方案[69]，结合预计算技术利用空间换时间，尽可能将椭圆曲线层算术的时间复杂度降低，将标量乘法的性能提升了 31%. 同年，中科院信工所团队[70]基于 sjcl 密码算法库，实现了 JavaScript 版本的 SM2 算法，该工作采用了固定 comb 的方案对 ECC 中的上层点乘算法进行加速，算法的性能提高了一倍以上；该工作通过优化代码使国密 SM2 的算法空间降低到 50KB 以下，保障了浏览器加载时延.

北京大学关志团队[71]长期致力于商用密码算法的计算技术研究，其开源密码算法库（GmSSL）广泛用于工业界，对标 OpenSSL[50]开源密码算法库，支持包括公钥密码 SM2/9 在内的各类商用密码算法实现. 为保障算法通用性，该算法库主要基于高级计算机语言（C 语言等）开发，其团队也在不断探索汇编层面的技术优化. 在表 6 中，本文展示了面向服务侧 ECC 软件的代表性成果，吞吐性能（kops/s）表示（千次标量乘法计算操作/秒）.

表 6　面向服务侧的 ECC 实现技术举例

实现工作	椭圆曲线类型	实现平台	吞吐性能/(kops/s)	计算延时/ms
Huang 等人[68]	SM2	Intel Cascade Lake	7.20	0.14
Gueron 等人[52]	NIST P-256	Intel Haswell	10.70	0.09
Cheng 等人[72]	Curve25519	Intel i5-6360U	32.00	—
Faz-Hernández 等人[54]	Curve25519	Intel i7-7820X	44.00	0.02
FourQlib[27]	FourQ	Intel Xeon E5-2699v3	58.97	—
Cui 等人[73]	224 bit Edwards	GTX285	115.00	19.20
Mahe 等人[63]	Curve25519	GTX TITAN	524.00	—
Zheng 等人[62]	SM2	GTX 780Ti	391.00	—
Pan 等人[60]	NIST P-256	GTX 780Ti	929.00	25.82
Dong 等人[64]	Curve25519	GTX TITAN	1394.00	—
Gao 等人[65]	Edwards25519	TITAN V	7216.00	1.51
	Curve25519		13558.00	2.84

4　面向终端侧的 ECC 实现技术

本节主要介绍在物联网设备上，尤其是资源受限的轻量化芯片，面向椭圆曲线算法的研究现状. 我们着重介绍了在嵌入式平台上国际通用的椭圆曲线密码优化实现技术，并引出国密算法相关技术研究.

4.1　国际通用密码算法实现技术

近些年来，在嵌入式平台上针对国际通用密码算法的高效实现工作层出不穷. 嵌入式平台受到成

本限制,计算、内存资源极度匮乏.在相同的安全强度下,RSA 算法密钥长度远大于 ECC 算法,因此在资源受限的嵌入式平台上,ECC 公钥密码算法是学者研究的热点.

2004 年,Oracle 实验室 Gura 团队发表文献[74],在嵌入式芯片上提出了能够减少计算指令的混合乘法技术,在有限域算法的基础上分别实现 ECC 算法与 RSA 算法,其 NIST P-160 的性能超过 RSA-1024 算法 10 倍.2008 年开源的 TinyECC 是基于 TinyOS 的椭圆曲线密码库,是比较早在 16 位 MSP430 嵌入式平台上支持各种不同安全级别椭圆曲线加密,密钥交换和数字签名协议的密码库[75].2013 年,格拉茨技术大学团队发表文献[76],在 MSP 嵌入式芯片上,实现椭圆曲线密码;MSP430 平台仅包含 27 种运算指令,但是缺少乘法指令,该方案首先利用有限的汇编指令(XOR 等操作)组合(64×1)-bit 的多项式乘法,进一步完成(64×32)乘法器,最后实现了 192-bit 的大整数乘法实现,该乘法性能是已有工作的 2.7 倍;此外,该团队在 MSP430 嵌入式芯片的 CPU 和内存之间设计了一个新的硬件加速器,同样性能出众.

在嵌入式平台,新型 ECC 算法实现研究工作同样层出不穷.2015 年,德国波赫鲁尔大学 Düll 等人[77]分别在 8 比特、16 比特与 32 比特微处理器上实现 Curve25519 密钥协商算法,有限域层面仍然选用 Karatsuba 乘法,而且单独实现了大整数平方预算,基于三种不同类型的平台(AVR、MSP430、ARM Cortex-M0)运算指令优化实现,该文献是 Curve25519 算法在嵌入式平台的代表性工作;2017 年,坎皮纳斯大学团队[78]在 Cortex-M4 平台上,通过分解 256 比特的大整数乘法方案,提出了二次迭代 Karatsuba 乘法技术,并采用了操作数缓存技术减少内存的访问次数,在有限域层对 ECC 算法进行实现优化,该工作不仅实现了基于 Montgomery Ladder 的 Curve25519 算法,而且评估了 Ed25519 算法的性能.2018 年,滑铁卢大学团队[79]在三种嵌入式平台上(AVR、MSP430、ARM Cortex M4),针对微软提出的 Four$\mathbb{Q}$ 椭圆曲线标量乘法进行优化实现,相比 Curve25519 算法的实现工作[77],性能有进一步的提升;该团队还评估过 MSP430 系列芯片的多种硬件乘法器性能[80-81],提出了之字形多精度乘法实现(Zigzag)算法,实现了抗侧信道攻击的蒙哥马利曲线标量乘算法,并在此基础上完成高效的密钥协商,数字签名协议的实现.2020 年,中科院信工所团队[82]将嵌入式 GPU 应用在公钥密码算法高性能实现上,完成了椭圆曲线 X25519/448 密钥交换算法与 EdDSA 签名算法,分别从有限域、点算术、点乘层面对上述算法展开性能优化实现,在功耗仅有 10W 的嵌入式平台上,吞吐性能甚至超过服务器级别的 CPU,为服务器端或边缘计算设备提供了新的解决方案.

为了更好地适配嵌入式设备,ARM 公司提供了 MbedTLS 开源算法库[83],支持大部分种类的椭圆密码算法,其主要特点是轻量化,对于资源短缺的设备能够较好地适配.相比于 OpenSSL 算法库,MbedTLS 开源算法库的社区活跃度不高,暂不支持较新的椭圆曲线密码(例如 EdDSA 签名算法).

4.2 国产椭圆曲线密码实现

《密码法》二十一条明确指出"国家鼓励商用密码技术的研究开发、学术交流、成果转化和推广应用",国产公钥密码得到国内学者的广泛关注.在物联网的场景下,终端设备由于海量部署,硬件成本受限,大量的低端处理器设备的 ROM 空间资源严重不足.然而,大量应用场景(例如智慧交通、智慧医疗等)对密码算法的时延有较为苛刻的要求.大多数参考文献在物联网场景实现公钥密码算法时,首先考虑密码算法的轻量化问题,在解决可部署的问题后再进一步追求密码算法的高性能与高安全实现.

2019 年,日本会津大学团队[84]在 8-bit AVR 嵌入式平台实现 SM2 国密算法与 NIST P-256 曲线,该工作优化实现了大整数加减操作,并使用内存效率更高的 Karatsuba 方案实现大整数乘法与平方,最后利用 Montgomery 阶梯技术实现标量乘法,对比其实验结果,国密算法 SM2 相比于 NIST P-256 拥有更好的性能与轻量化优势.2019 年,在计算机顶级期刊《TCAD》上,广东工业大学团队[85]在 110-nm 的 SMIC 平台上,完整地实现 SM2/3/4 三种国密算法,在软硬结合的设计框架下,对计算负载核心运算进行硬件实现,例如,SM2 密码算法的有限域运算、点算术运算使用硬件实现,上层签名与验签部分使用软件完成;并利用 TRNG 模块加速随机数产生效率;该工作部分软硬件实现还通过并行手段进行加速,进一步提高了算法的整体性能;其算法设计过程中也考虑了安全因素,声明能够抵御能量侧信道攻击,性能上的提升超过 10%.

2018 年,中电集团相关团队[86]提出了一种针对国密 SM9 标识密码算法的加速方案的专利,在高

复杂度的双线性对运算中，基于算法的 12 次扩展域结构以及数学性质，大幅度地提高了 12 次扩展域的运算效率，该方案通过研究双线性对的 Miller 算法特征与塔式扩张的结构，运算效率提升接近 50%. 2019 年，国家电网研究团队发表文章[87]，该工作指出 SM9 的安全性基于椭圆曲线的双线性映射性质，由于双线性对计算复杂度较高，在有限资源的设备上，性能是 SM9 密码算法所面临的重要挑战. 该工作分析了 SM9 国密算法双线性对的计算流程，并通过实验对比了基于两种坐标（Projective 坐标与 Jacobian 坐标）Miller Loop 的实现性能，后者相比前者性能提升 5%，在一定程度上提升了 SM9 算法的整体性能. 表 7 展示了面向终端侧椭圆曲线密码实现的部分研究成果.

表 7　面向终端侧的 ECC 实现技术举例

实现工作	椭圆曲线类型	实现平台	吞吐性能/(kops/s)	计算延时/ms
Zhou 等人[84]	SM2	AVR ATmega128	0.0006	1562.000
Wenger 等人[88]	NIST P-256	AVR ATmega128	0.0004	2174.000
Hinterwälder 等人[89]	Curve25519	MSP430X 8 MHz	0.0010	833.000
Düll 等人[77]	Curve25519	AVR ATmega128	0.0010	909.000
		MSP430X 16 MHz	0.0030	370.000
Gouvêa 等人[90]	NIST P-256	MSP430X 8 MHz	0.0020	667.000
Fujii 等人[78]	Curve25519	Cortex-M4 84 MHz	0.0900	11.000
Liu 等人[79]	FourQ	MSP430X 8 MHz	0.0020	526.000
		Cortex-M4 84 MHz	0.2000	5.590
		AVR ATxmega256A3	0.0050	204.000
Dong 等人[82]	Curve448	NVIDIA Tegra X2	17.9000	0.050
	Curve25519		155.5000	0.006

5　椭圆曲线安全实现技术研究

1996 年，美国旧金山密码研究所[91]提出了侧信道的攻击思想，其核心是通过监控特定资源从而获取密钥等敏感信息. 与之对应的为侧信道防御技术，即分析已有的侧信道攻击方案与密码算法实现流程，设计并实现各类防护策略，用以提高密码算法的安全性.

此外，白盒攻击也是一种面向椭圆曲线密码算法攻击的常见手段. 针对密码的白盒攻击思想最早在 2002 年由 Chow 等人[92]提出，其核心是指，攻击者可以很轻易得到携带密码算法设备的控制权并对其进行包括不限于二进制追踪、动态分析、读取密钥等一系列操作，从而能够得到密码算法运行时的内部数据. 这对密码算法的安全提出了严峻的挑战，而能抵御白盒攻击的密码实现一般被称为白盒密码实现.

5.1　服务侧安全实现

面向服务侧，高性能计算平台侧信道攻击防护技术研究. 随着硬件性能不断提升，各种类型的加解密算法在高性能平台实现上都取得了巨大的突破. 2021 年，上海交通大学团队[93]分析了基于 C 语言有理算术库 MIRACL 的 SM2 软件实现，针对 wNAF 滑动窗口的标量乘法，采用了 Flush＋Reload 的攻击方案，恢复了 SM2 的密钥. 该实验结果表明，固定时长的标量乘法与 MIRACL 提供的标量乘法相比，具有更高的防护效果.

目前针对于椭圆曲线密码算法的白盒实现技术也有相关研究. 2015 年，上海交通大学来学嘉教授团队对于白盒密码研究发表综述[94]，详细介绍了白盒密码的相关理论基础，实现技术以及研究现状. 2018 年，上海交通大学谷大武团队[95]基于国密 SM2 算法，设计实现了一种白盒密码实现方法，在传统 SM2 数字签名的基础上通过增设验证参数与验证参数表来确保数据完整性、不可抵赖性等安全需求，防止数字签名私钥被破解. 经过实验，此方法签名时间是标准 SM2 的 2～3 倍，验证时间约为其 2 倍，符合实际的应用需求. 2020 年，成都卫士通发表文献[96]，将椭圆曲线倍点运算中的倍数用一种特殊的形式即构造分量的方式进行表达，与此同时还构建了一个查找表来保护每个分量的安全性防止攻击者获取分量从而恢复出倍数，最后通过消除掩码的方式来实现正确的结果. 此外，在此方案中，作者还使用余数系统嵌套分解大数计算，有效降低了查找表的大小，确保了倍点运算中的倍数不会被泄露. 该方案可运用在国密 SM2/SM9 中，以提高 SM2/SM9 私钥操作在第三方不可信平台上的安全性.

高性能计算平台一般作为高性能密码机，部署在服务器后端，由于基于能量的侧信道攻击方式实现难度较大，因此针对服务器端的主要的侧信道攻击方式为时序攻击. 例如针对 GPU 高性能计算，2016 年美国东北大学发表文献[97]，利用时序侧信道

的攻击方法，在 30 min 之内能够破译出基于 GPU 平台实现的对称密码 AES-128 算法的密钥. 目前，虽然还未出现针对基于 GPU 平台实现的公钥密码算法攻击方法，但是传统 CPU 平台的时序攻击方式(例如文献[98])完全具有延伸到 GPU 上的可能性. 当前，大部分基于 GPU 实现的公钥密码算法忽略了侧信道攻击防御或者研究不足.

进一步地，在实现密码算法过程中，高性能计算平台具有多线程以及存储结构多样化等特性，这些特性会产生针对 GPU 特有的攻击漏洞. 例如，GPU 处理同一个 warp 分叉操作时[38]，产生更大的延时差别，安全风险更大，尤其是在同一线程束的不同请求，其计算延时一致的特性，使之很容易成为计时攻击的切入点. GPU 多线程配合时也存在特有安全风险，文献[99]在 Montgomery 乘法实现中，使用 while、any 等指令构造的 lazy-carry 方法，用以完成约减过程中线程间进位的问题，但是 while 循环对于不同的源数据，循环次数不同，进而导致延时不同；同时 CUDA 中特有的 any 指令的返回内容为整个线程束的信息，容易泄露同一个线程束中其他解密操作的信息，存在一定的安全威胁；同样共享内存的使用，会带来 bank conflict 的问题[38]，易泄露同一个线程束中其他解密操作的信息. 上述都是 GPU 实现密码算法中特有的安全威胁，但是并没有受到足够的重视.

5.2 终端侧安全实现

面向终端侧，基于嵌入式平台的侧信道防护技术与服务器侧部署位置不同. 针对终端侧的侧信道攻击方案更加的丰富，尤其是在成本受控的嵌入式芯片上，硬件防护手段受到严重限制，攻击者可以采用功耗、电磁、时序等攻击手段.

卢森堡大学团队发表文献[100]，在 8-bit AVR 嵌入式平台上，针对椭圆曲线点乘算法，采用安全的 Montgomery Ladder 技术保证算法实现的安全性. 该团队的另一个工作[101]在有限域层面，针对椭圆曲线大整数乘法的乘积扫描算法进行优化，该方案从实现角度上能够达到固定延时(constant-time)的效果，具有侧信道攻击防护能力. 2015 年，中科院周永彬团队[102]在 8 比特的 AVR 嵌入式平台上，针对 Simon 算法提出了具有侧信道攻击防护的高阶掩码方案，兼顾了性能与安全. 2021 年，中科院信工所团队在计算型显卡 GPU 上[103]，利用双精度浮点数熔加指令高效地完成大整数乘法，进一步在实现过程中，利用比特位操作避免分支操作，保证密码计算延时固定，具有时序侧信道攻击防护能力.

在上述两类部署环境中，大量的防侧信道技术方案具有通用性. 例如，针对椭圆曲线算法的点乘层面，采用固定延时的固定窗口计算方案而摒弃具有更高计算性能的非固定窗口计算方案. 同样，采用掩码防护方案[91]也具有一定的普适性，不过随着攻击技术的不断提高，掩码方案的安全阶数不断提高，在嵌入式设备上，掩码方案也面临着轻量化的设计需求.

6 分层优化技术总结

如图 3 所示，本节由底层到上层，从三个维度对上述的优化方案进行总结：有限域层面、点算术层面、标量乘法层面.

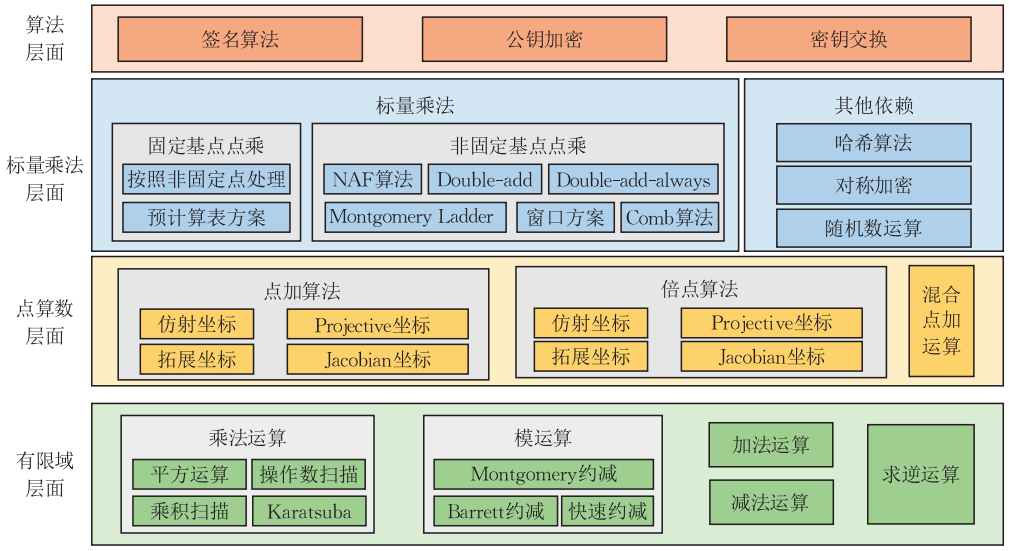

图 3 椭圆曲线密码算法架构

有限域层面. 有限域是椭圆曲线的最底层运算，其中，多精度模乘运算($C=A\times B \bmod p$)是决定椭圆曲线最底层有限域计算效率的关键操作，是密码工程研究人员优化创新的重点. 在算法实现过程中，由于平方运算存在一定数量的重复计算单元，可单独处理用以降低乘法指令的数量. 根据计算流程，可以简单将多精度模乘运算分为两类：(1) 首先计算多精度乘法，然后对大整数乘积进行求模运算. 例如：C(512-bit)$=A$(256-bit)$\times B$(256-bit). 常用的多精度乘法计算方案有操作数扫描(Operand Scanning Method)、乘积扫描(Product Scanning Method)和 Karatsuba 算法，得到 C(512-bit)乘积后，利用约减算法计算模乘的最终结果，常用的约减算法有快速约减(梅森素数)、Montgomery 约减与 Barrett 约减等；(2) Montgomery 乘法[104]是另一类典型的求解模乘的计算方案，可以将乘法与约减操作交替进行，避免复杂的试除运算. 俄勒冈州立大学发表文献[105]详细讨论了五种 Montgomery 乘法的优缺点，并总结出粗粒度整合操作数扫描方式(Coarsely Integrated Operand Scanning)计算效果最好.

Gura 等人[74]在 CHES 2004 年的经典论文提出了一种混合多精度乘法计算方案，主要是结合了操作数扫描与乘积扫描的优点，减少乘法指令数量；CHES 2011 年 Wenger 团队发表论文[106]提出了一种优化操作数字节执行顺序的计算方案，降低了内存指令的数量，并获得当年的最佳论文奖；2016 年，文献[80]提出了一种之字形多精度乘法实现(Zigzag)方案；同年，德国慕尼黑工业大学团队发表文献[107]提出了 2 层迭代的 Karatsuba 乘法方案(2-level subtractive Karatsuba)，主要是利用 32-bit 长度字表示，优化多精度大整数乘法；2020 年，文献[65]提出了一种 51-bit 字长度的双精度浮点数高低进位计算方案，主要是利用浮点数的熔加指令(Fused Multiply-Add)，降低乘法指令数量，大幅度提高多精度乘法的性能.

在有限域层面，模逆运算同样是密码工程优化的重点内容，最常用方案包括拓展的 Euclid 算法[108]与 Fermat 方法[109]. 二进制拓展的 Euclid 算法(Binary Extended Euclidean Algorithm，BEEA)使用右移操作替换了多精度除法操作，性能提升显著，但该类方案不满足固定延时的要求，具有侧信道攻击风险，针对 Euclid 算法安全实现同样是研究的热点，基于模幂的 Fermat 方法在涉及私钥运算的算法常作为优选方案. 2019 年，Bernstein 团队[110]提出了一种高效安全的 Euclid 方案，方案的核心操作被称为 division steps，该方法具有固定时延的特性，最终通过基于 Intel 平台上的两种计算场景证实(Curve25519 的模逆运算、ntruhrss701 与 sntrup4591761 格密码的密钥生成操作)，该方法在安全性与性能方面具有显著优势. 2021 年，耶鲁大学团队在 FPGA 平台上对 Bernstein 的 Euclid 方案进行性能评估[111]，并结合高效的底层实现，其模逆性能提升了 90%.

点算术层面. 主要包含两类点算术运算：点加算法与倍点算法. 由于传统的仿射坐标(x,y)在计算过程中需要执行复杂的求逆运算，故一般首先完成坐标转换再执行对应的算法操作. Weierstrass 曲线在实现的时候一般采用雅克比坐标(X,Y,Z)来表示. Edwards 曲线一般采用拓展的爱德华坐标(X,Y,Z,T)进行点运算操作. 早在 2007 年，中船重工联合武汉大学团队发表文献[112]，对比了仿射坐标、投影坐标与雅克比坐标在点乘与倍点算法中的性能，最后采用了雅克比坐标与仿射坐标混合点加与雅克比坐标倍点的实现方案. 坐标转换本质上是一种利用空间换取时间的操作，能够在一定程度上减少复杂有限域操作的数量，达到提高计算效率的结果.

标量乘法层面. 根据基点的不同，标量乘法的运算可以分为两类：

(1) 可变基点标量乘法运算. 最传统的计算方案为 Double-add 方案，这种方案由于计算延时与输入有关，易受攻击且效率较低. 研究人员一般通过减少标量的海明重量或表示长度来减少点加或倍点运算的执行次数，例如 Comb 方案、固定窗口方案以及滑动窗口方案等[113]. 从安全角度层面考虑，Double-add 方案与滑动窗口方案都存在时序攻击的风险，Montgomery Ladder 方案、固定窗口方案由于计算具有固定延时的特性，可以很好地抵御侧信道攻击. 早在 2012 年，西南交通大学团队针对滑动窗口的快速标量乘法进行研究[114]，提出了混合表示下直接计算(2^kQ+P)的策略，改进了基于滑动窗口技术的标量乘法.

(2) 固定基点标量乘法. 该类算法一般应用在密钥生成、签名生成等流程，由于基点固定，研究人员一般会产生一个预计算表，并利用多个(混合)点加运算完成标量乘法. 进一步，还可以通过选取特殊形式的椭圆曲线，比如 GLV 曲线[115]、GLS 曲线[116]

等,可进一步来减少点算术运算的执行数量;2018年,中山大学团队从理论层面优化了二元域上椭圆曲线 Weierstrass 形式到 Edwards 形式的转换算法[117],该工作主要基于 Shallue-Woestijne 与半分有理点算法,在很小计算量的前提下,将椭圆曲线 Weierstrass 形式转换为 Edwards 形式,为椭圆曲线密码计算提供了更加高效与安全的解决方案.利用扭曲爱德华兹曲线(Twisted Edwards)和蒙哥马利(Montgomery)曲线的有理等价来进行加速运算[113],由于其标量乘法算法的复用,还可以在一定程度上降低算法空间,有利于在资源受限设备上部署完整的安全认证协议栈.2020 年,西安电子科技大学团队发表文献[118],研究面向 Edwards 曲线的快速标量乘法,引入了加速倍点运算的 CDA 算法,并提出了 4-NNAF 形式的标量 k 表示方案,结合并行计算结构,效率提升 36%以上.

加法链[119]与双基链[120]同样被广泛应用于椭圆曲线标量乘法加速.2019 年,西安邮电大学刘双根团队发表文献[121],提出了一种基于广义的斐波那契数列的青铜比例加法链(Bronze Ratio Addition Chain,BRAC),结合投影坐标,相比黄金比例加法链性能提升超过 30%.2021 年,该团队发表文献[122],改进了$(5P+Q)$的计算方法,使得该算法复杂度降低为 8M+16S,在此基础上,提出一种改进的斐波那契数列加法链(Improved Fibonacci Type Addition Chain),该方案不仅在性能上具有显著优势还具有抵抗 SPA 攻击的能力.2020 年,中科院信工于伟团队在欧密会议(Eurocrypt2020)发表文献[123],提出了一种基于动态规划算法来生成最优双基链的方案,该方案的性能相比现有算法提升了 6 倍,最优双基链汉明重量相比于 NAFs 方案降低了 60%,整体标量乘法性能提高了 13%,并具有抵御侧信道攻击的能力.

7 发展趋势

随着计算技术的不断发展以及用户隐私意识的不断增强,对于公钥密码性能要求不断提高,椭圆曲线密码实现技术也呈现出新的发展趋势.

7.1 追求极致性能

在万物互联的背景下,各类应用对公钥密码算法的性能与实时性要求越来越高.一方面,服务端在处理海量数据的密码运算的请求时,主要的解决方案是:增加密码计算设备的部署量来提高密码计算的性能,包括提高吞吐性能(ops/s)、计算延时(ms)等,与此同时还需要配套一定数量的负载均衡设备,导致成本进一步增加.提高单机设备处理椭圆曲线密码算法的性能,可大幅度降低企业运营成本.另一方面,大量的物联网应用对计算延时有着较高的要求,例如智能驾驶、智慧医疗等,计算复杂度高的公钥密码算法会带来计算延时,在实际应用中可能会引起不可估计的损失,必须在有限的计算资源下,追求密码计算的极致性能.

7.2 不断压缩空间

由于单个设备造价的要求极高,一般在芯片设计之初,海量部署的物联网设备便已完成空间划分.部分平台在内存划分之初,一般首先考虑应用的功能模块,安全内容重视不足或滞后部署;上述做法往往给攻击者带来便利,随着各项法律的不断完善,以及用户对于安全重视程度不断提高,安全模块逐渐作为产品升级的重要组成部分.滞后部署的密码算法往往面临内存不足的困境,压缩密码算法库的空间占用为上层应用以及安全方案的部署节省内存空间,具有重要的实际应用价值.

7.3 平衡安全效率

在保障上层应用安全的基础上,密码实现自身的安全同样是重要的研究课题.物联网设备部署环境多样,常被部署于无人监守的地方,受到成本限制,这些设备物理防护能力有限,容易被敌手所控制.但对于密码算法的加固一般会设计专门的防护策略,从而导致成本较高.根据鲁汶大学文献[124]的实现结果,针对 AES 算法的一阶防护实现代价在代码量、数据量和时钟周期等方面上都至少超过原有的 1.8 倍以上,其中数据量更是达到 16 倍之多,而公钥密码的防护策略与实现更加艰难,同时代价也更高.一方面,研究者不断降低防护策略对密码计算性能带来的负面影响;另一方面,在密码算法实现过程中,平衡安全策略与实现性能之间的关系,对于密码实现有着重要的意义.

7.4 重视国产研究

近些年来,针对云计算领域高性能国产密码算法实现已取得快速的进步,但是相比于国际主流密码算法仍有一定差距.如表 8 所示,中科院团队[62]在高性能 GPU 平台上,针对国产椭圆曲线公钥 SM2 算法,完成了对椭圆曲线有限域层面大整数算法优化实现,并未探究标量点乘算法以及上层完整的签

名验签算法，该文献按照比例对椭圆曲线标量点乘算法进行预估，其性能与当前最新工作 Curve25519 曲线性能已有数量级差别. 为了获得更加高效的计算性能，研究人员会针对特定平台采用专门的优化方案，例如针对椭圆曲线 Curve25519 算法，已有文献[77-78，89，107]分别在 ARM Cortex-M0、ARM Cortex-M4、AVR 8/16/32-bit 等多种平台上优化实现，不断取得新的实验突破. 反观同属于椭圆曲线类型的国密 SM2 算法，相比于国际通用的同类型密码算法，针对嵌入式平台的研究较少，已知文献[84]在 AVR 8-bit 嵌入式芯片上完成该算法的实现. 伴随云计算、物联网技术的快速发展，以及《密码法》等法律法规的不断完善，基于国密算法的网络安全方案面临重大的历史机遇.

表 8　椭圆曲线类型性能优化对比（国密 SM2 vs. Curve25519）

针对平台特定优化方案	高性能计算平台		嵌入式计算平台		
	GPU	Intel CPU	AVR 芯片	ARM M4	嵌入式 GPU
Curve25519	13 558 000 ops/s[65]	44 000 ops/s[54]	1.15 ops/s[77]	93.981 ops/s[78]	155 459 ops/s[82]
国密 SM2	391 000 kops/s[62]	7200 ops/s[68]	0.64 ops/s[84]	不支持#	不支持
性能对比*	34.67∶1	4.44∶1	1.79∶1	无	无

注：*性能对比：在同类别平台上，Curve25519 vs. SM2 的性能对比数值，SM2 性能作为 1；性能单位为每秒执行标量乘法数量(ops/s).
#不支持：具体是指针对特定平台的优化方案，不包含通用 C 等高级语言实现的密码算法库.

8　小　结

椭圆曲线公钥密码算法可用于密钥交换、数字签名、公钥加密等技术，是当前主流安全认证协议(TLS、SPEKE 等)的重要组成部分. 由于计算复杂度较高，一直是安全模块部署的重要瓶颈点. 面向椭圆曲线的软件高性能实现研究是密码工程学科研究的热点技术. 本文主要针对端云两侧不同的应用场景，对椭圆曲线密码算法高性能软件计算技术研究进行梳理与总结.

本文首先介绍了椭圆曲线基本知识与标准化现状，并阐述了后斯诺登时代新型椭圆曲线密码算法的发展趋势. 之后围绕服务侧与终端侧不同的应用场景，讨论了端云两侧对椭圆曲线密码不同的性能需求，紧接着列举了当前主流的密码实现平台的硬件特性. 由于椭圆曲线密码在端云两侧截然不同的应用现状，本文分别面向服务侧与终端侧详细介绍了椭圆曲线密码算法的研究现状，重点突出了国产密码算法在各类平台上实现成果，进一步安全实现技术是密码工程重要的组成部分，本文还补充了业界针对椭圆曲线密码安全实现的相关成果. 紧接着，我们分别从有限域、点算术与标量乘法总结了椭圆曲线密码优化的常用技术手段. 最后我们列举了椭圆曲线密码未来发展趋势，希望为相关领域的研究者提供一定的帮助.

致　谢　在此，衷心地感谢编辑部老师和审稿人给予本文的宝贵意见！

参　考　文　献

[1] Diffie W, Hellman M. New directions in cryptography. IEEE Transactions on Information Theory, 1976, 22(6): 644-654

[2] Rivest R L, Shamir A, Adleman L M. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 1978, 21(2): 120-126

[3] Barker E, Burr W, et al. Recommendation for key management: Part 1: General[M]. MD, USA: National Institute of Standards and Technology, Technology Administration, 2006

[4] Koblitz N. Elliptic curve cryptosystems. Mathematics of Computation, 1987, 48(177): 203-209

[5] Zhang Yin, Wang Chao. Research on the implementation of elliptic curve cryptosystem//1994 Proceedings of Advances in Theoretical Computer Science. Changsha: National University of Defense Technology Press, 1994: 104-108(in Chinese)
(张引，王潮. 椭圆曲线密码体制实现研究//1994 理论计算机科学进展论文集. 长沙：国防科技大学出版社，1994：104-108)

[6] Xu Qiu-Liang, Li Da-Xing. Elliptic curve cryptosystem. Journal of Computer Research and Development, 1999, 36(11): 1281-1288(in Chinese)
(徐秋亮，李大兴. 椭圆曲线密码体制. 计算机研究与发展，1999，36(11)：1281-1288)

[7] Chen Xiao-Feng, Wang Yu-Min. Research and development of public key cryptosystem. Journal on Communications, 2004, 25(8): 109-118(in Chinese)
(陈晓峰，王育民. 公钥密码体制研究与进展. 通信学报，2004，25(8)：109-118)

[8] Liu Duo, Dai Yi-Qi, Wang Dao-Shun. Stability and balance—Strategies and methods for anti-side-channel attacks in elliptic curve cryptosystems. Journal of Computer Research and Development, 2005, 42(10): 1667-1672(in Chinese)

(刘铎，戴一奇，王道顺. 平稳与平衡——椭圆曲线密码体制抗旁信道攻击的策略与手段. 计算机研究与发展，2005，42(10)：1667-1672)

[9] 商用密码检测中心. 最新标准公布. 2021. http://www.scctc.org.cn/templates/General/index.aspx?nodeid=16&pagesize=1&pagenum=10

[10] Harkanson R, Kim Y. Applications of elliptic curve cryptography: A light introduction to elliptic curves and a survey of their applications//Proceedings of the 12th Annual Conference on Cyber and Information Security Research. Oak Ridge Tennessee, USA, 2017: 1-7

[11] Blake I, Seroussi G, Seroussi G, et al. Elliptic Curves in Cryptography: Volume 265. UK: Cambridge University Press, 1999

[12] Montgomery P L. Speeding the pollard and elliptic curve methods of factorization. Mathematics of Computation, 1987, 48(177): 243-264

[13] Bernstein D J, Lange T. Faster addition and doubling on elliptic curves//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Kuching, Malaysia, 2007, 2007: 29-50

[14] Edwards H M. A normal form for elliptic curves. Bulletin of the American Mathematical Society, 2007, 44(3): 393-422

[15] Kerry C F, Gallagher P D. Digital signature standard (DSS). FIPS PUB, 2013: 186-4

[16] Standardization I O. Preview SM2 digital signature mechanism. https://www.iso.org/standard/70631.html

[17] Ansi X. Public key cryptography for the financial services industry: The elliptic curve digital signature algorithm (ECDSA). ANSI X9 Catalog, 1999

[18] Lochter M, Merkle J. Elliptic curve cryptography (ECC) Brainpool standard curves and curve generation. RFC 5639, March, 2010

[19] IEEE P1363 Working Group. Standard specifications for public key cryptography. IEEE P1363/D20 (Draft Version 20), 2005

[20] Bernstein D J. Curve25519: New Diffie-Hellman speed records//Proceedings of the International Workshop on Public Key Cryptography. New York, USA, 2006: 207-228

[21] Hamburg M. Ed448-Goldilocks, a new elliptic curve. IACR Cryptology ePrint Archive, 2015, 2015: 625

[22] Järvinen K, Miele A, Azarderakhsh R, et al. FourQ on FPGA: New hardware speed records for elliptic curve cryptography over large prime characteristic fields//Proceedings of the International Conference on Cryptographic Hardware and Embedded Systems. Santa Barbara, USA, 2016: 517-537

[23] Langley A, Hamburg M, Turner S. Request for comments: Number 7748 Elliptic Curves for Security. RFC Editor, 2016. https://rfc-editor.org/rfc/rfc7748.txt. DOI: 10.17487/RFC7748

[24] Josefsson S, Liusvaara I. Request for comments: Number 8032 Edwards-Curve Digital Signature Algorithm (EdDSA). RFC Editor, 2017. https://rfc-editor.org/rfc/rfc8032.txt. DOI: 10.17487/RFC8032

[25] Rescorla E. The transport layer security (TLS) protocol Version 1.3. RFC, 2018, 8446: 1-160. https://doi.org/10.17487/RFC8446

[26] LIM C. The KCDSA (Korean Certificate-based Digital Signature Algorithm) with appendix[C]//Proceedings of the Conference on Information Security and Cryptology (CISC'97). Korean, 1997: 251-264

[27] Costello C, Longa P. FourQ: Four-dimensional decompositions on a Q-curve over the Mersenne prime//Proceedings of the International Cryptology Conference. Santa Barbara, USA, 2015: 214-235

[28] Briefing C. China's Double 11 Shopping Festival Tests Consumption Strength after COVID-19. 2020-11-13. https://www.china-briefing.com/news/chinas-double-11-shopping-festival-tests-consumption-strength-after-covid-19/

[29] Yao S, Yu D. PhiOpenSSL: Using the Xeon Phi coprocessor for efficient cryptographic calculations//Proceedings of the International Parallel and Distributed Processing Symposium. Orlando, USA, 2017: 565-574

[30] Intel. AVX Instruction Set. 2021. https://software.intel.com/content/www/cn/zh/develop/articles/introduction-to-intel-advanced-vector-extensions.html

[31] ARM. ARM Technologies ARM NEON. 2021. https://www.arm.com/why-arm/technologies/neon

[32] Intel. Intel Core i7-9800X X-series Processor. 2021. https://ark.intel.com/content/www/us/en/ark/products/189122/intel-core-i7-9800x-x-series-processor-16-5m-cache-up-to-4-50-ghz.html

[33] Intel. Intel Xeon Platinum 8360Y Processor. 2021. https://www.intel.cn/content/www/cn/zh/products/sku/212459/intel-xeon-platinum-8360y-processor-54m-cache-2-40-ghz/specifications.html

[34] AMD. AMD锐龙 7 3800X. 2020. https://www.amd.com/zh-hans/products/cpu/amd-ryzen-7-3800x

[35] Qualocomm. Qualocomm 888 processor. 2021. https://www.qualcomm.cn/snapdragon/888-5g-mobile-platform

[36] Huawei. Kirin 990 processor. 2021. https://consumer.huawei.com/cn/campaign/kirin-990-series/

[37] NVIDIA. NVIDIA's Next-Generation CUDA Compute and Graphics Architecture, Code-Named Fermi, Adds Muscle for Parallel Processing. https://www.nvidia.in/content/PDF/fermi_white_papers/T.Halfhill_Looking_Beyond_Graphics.pdf

[38] NVIDIA. CUDA C programming guide 9.0. 2017. https://docs.nvidia.com/cuda/cuda-c-programming-guide/

[39] NVIDIA. NVIDIA GEFORCE GTX 1080. 2021. https://www.nvidia.cn/geforce/products/10series/geforce-gtx-1080/

[40] NVIDIA. NVIDIA GEFORCE GTX 3080. 2021. https://www.nvidia.cn/geforce/graphics-cards/30-series/rtx-3080/

[41] NVIDIA. NVIDIA TESLA P100. 2021. https://www.nvidia.cn/data-center/tesla-p100/

[42] NVIDIA. NVIDIA TESLA V100. 2021. https://www.nvidia.com/en-gb/data-center/tesla-v100/

[43] NVIDIA. NVIDIA TESLA A100. 2021. https://www.nvidia.cn/data-center/a100/

[44] NVIDIA. NVIDIA JETSON TX2. 2021. https://www.nvidia.cn/autonomous-machines/embedded-systems/jetson-tx2/

[45] NVIDIA. NVIDIA JETSON NANO. 2021. https://www.nvidia.cn/autonomous-machines/embedded-systems/jetson-nano/

[46] NVIDIA. NVIDIA JETSON XAVIER NX. 2021. https://www.nvidia.cn/autonomous-machines/embedded-systems/jetson-xavier-nx/

[47] NVIDIA. NVIDIA JETSON XAVIER. 2021. https://www.nvidia.cn/autonomous-machines/embedded-systems/jetson-agx-xavier/

[48] NVIDIA. NVIDIA TURING GPU ARCHITECTURE. https://developer.nvidia.com/blog/nvidia-turing-architecture-in-depth/

[49] Longa P, Gebotys C. Efficient techniques for high-speed elliptic curve cryptography//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Santa Barbara, USA, 2010: 80-94

[50] Foundation OS. OpenSSL Cryptography and SSL/TLS Toolkit. 2016. http://www.openssl.org/

[51] Google. BoringSSL Cryptography and SSL/TLS Toolkit. 2021. https://github.com/google/boringssl

[52] Gueron S, Krasnov V. Fast prime field elliptic-curve cryptography with 256-bit primes. Journal of Cryptographic Engineering, 2015, 5(2): 141-151

[53] Faz-Hernández A, López J. Fast implementation of Curve-25519 using AVX2//Proceedings of the International Conference on Cryptology and Information Security in Latin America. Guadalajara, Mexico, 2015: 329-345

[54] Faz-Hernández A, López J, Dahab R. High-performance implementation of elliptic curve cryptography using vector instructions. ACM Transactions on Mathematical Software (TOMS), 2019, 45(3): 1-35

[55] Antão S, Bajard J C, Sousa L. RNS-Based elliptic curve point multiplication for massive parallel architectures. The Computer Journal, 2012, 55(5): 629-647

[56] Bernstein D J, Chen H C, Chen M S, et al. The billion-mulmod-per-second PC//Workshop Record of SHARCS: Volume 9. Lausanne, Switzerland, 2009: 131-144

[57] Bernstein D J, Chen T R, Cheng C M, et al. ECM on graphics cards//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cologne, Germany, 2009: 483-501

[58] Jeffrey A. Robinson B D. Fast GPU based modular multiplication. 2014. http://on-demand.gputechconf.com/gtc/2014/poster/pdf/P4156_montgomery_multiplication_CUDA_concurrent.pdf

[59] Bos J W. Low-latency elliptic curve scalar multiplication. International Journal of Parallel Programming, 2012, 40(5): 532-550

[60] Pan W, Zheng F, Zhao Y, et al. An efficient elliptic curve cryptography signature server with GPU acceleration. IEEE Transactions on Information Forensics and Security, 2017, 12(1): 111-122

[61] Yang Y, Guan Z, Sun H, et al. Accelerating RSA with fine-grained parallelism using GPU//Proceedings of the International Conference on Information Security Practice and Experience. Beijing, China, 2015: 454-468

[62] Zheng F, Pan W, Lin J, et al. Exploiting the potential of GPUs for modular multiplication in ECC//Proceedings of the 15th International Workshop on Information Security Applications (WISA 2014). Jeju Island, Korea, 2014: 295-306

[63] Mahe E, Chauvet J M. Fast GPGPU-based elliptic curve scalar multiplication. IACR Cryptology ePrint Archive, 2014, 2014: 198

[64] Dong J, Zheng F, Cheng J, et al. Towards high-performance X25519/448 key agreement in general purpose GPUs//Proceedings of the IEEE Conference on Communications and Network Security. Beijing, China, 2018: 1-9

[65] Gao L, Zheng F, Emmart N, et al. DPF-ECC: Accelerating elliptic curve cryptography with floating-point computing power of GPUs//Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). New Orleans, USA, 2020: 494-504

[66] Zhao Z, Bai G. Ultra high-speed SM2 ASIC implementation//Proceedings of the 2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications. Beijing, China, 2014: 182-188

[67] Mai L, Yan Y, Jia S, et al. Accelerating SM2 digital signature algorithm using modern processor features//Proceedings of the International Conference on Information and Communications Security. Beijing, China, 2019: 430-446

[68] Huang J, Liu Z, Hu Z, et al. Parallel implementation of SM2 elliptic curve cryptography on Intel processors with AVX2//Proceedings of the Australasian Conference on Information Security and Privacy. Perth, Australia, 2020: 204-224

[69] Rivain M. Fast and regular algorithms for scalar multiplication over elliptic curves. IACR Cryptology ePrint Archive, 2011, 2011: 338

[70] Wei Rong, Zheng Fang-Yu, Lin Jing-Qiang. Implementation of JavaScript general cryptographic library supporting national secret algorithms. Journal of Cryptography, 2020, 7(5): 595-604(in Chinese)
(魏荣，郑昉昱，林璟锵. 支持国密算法的 JavaScript 通用密码库的实现. 密码学报, 2020, 7(5): 595-604)

[71] Guan Z. The GmSSL Project. 2021. http://www.gmssl.org/

[72] Cheng H, Großschädl J, Tian J, et al. High-throughput elliptic curve cryptography using AVX2 vector instructions//Proceedings of the International Conference on Selected Areas in Cryptography. Halifax, Canada, 2020

[73] Cui S, Großschädl J, Liu Z, et al. High-speed elliptic curve cryptography on the NVIDIA GT200 graphics processing unit//Proceedings of the International Conference on Information Security Practice and Experience. Fuzhou, China, 2014: 202-216

[74] Gura N, Patel A, Wander A, et al. Comparing elliptic curve cryptography and RSA on 8-bit CPUs//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Cambridge, USA, 2004: 119-132

[75] Liu A, Ning P. TinyECC: A configurable library for elliptic curve cryptography in wireless sensor networks//Proceedings of the 2008 International Conference on Information Processing in Sensor Networks (IPSN 2008). Louis, USA, 2008: 245-256

[76] Wenger E. Hardware architectures for MSP430-based wireless sensor nodes performing elliptic curve cryptography//Proceedings of the International Conference on Applied Cryptography and Network Security. Banff, Canada, 2013: 290-306

[77] Düll M, Haase B, Hinterwälder G, et al. High-speed Curve25519 on 8-bit, 16-bit, and 32-bit microcontrollers. Designs, Codes and Cryptography, 2015, 77(2-3): 493-514

[78] Fujii H, Aranha D F. Curve25519 for the Cortex-M4 and beyond//Proceedings of the International Conference on Cryptology and Information Security in Latin America. Havana, Cuba, 2017: 109-127

[79] Liu Z, Longa P, Pereira G, et al. FourQ on embedded devices with strong countermeasures against side-channel attacks. IEEE Transactions on Dependable and Secure Computing, 2018, 17(3): 536-549

[80] Liu Z, Huang X, Hu Z, et al. On emerging family of elliptic curves to secure Internet of Things: ECC comes of age. IEEE Transactions on Dependable and Secure Computing, 2016, 14(3): 237-248

[81] Liu Z, Seo H, Hu Z, et al. Efficient implementation of ECDH key exchange for MSP430-based wireless sensor networks//Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. New York, USA, 2015: 145-153

[82] Dong Jian-Kuo, Zheng Fang-Yu, Lin Jing-Qiang. High-speed Implementation of X25519/448 key agreement algorithm based on GPU. Journal of Cyber Security, 2020, 5(6): 60-74(in Chinese)
(董建阔, 郑昉昱, 林璟锵. 基于 GPU 的 X25519/448 密钥协商算法的高速实现. 信息安全学报, 2020, 5(6): 60-74)

[83] ARM. Mbed TLS Cryptography and SSL/TLS Toolkit. 2021. https://tls.mbed.org/

[84] Zhou L, Su C, Hu Z, et al. Lightweight implementations of NIST P-256 and SM2 ECC on 8-bit resource-constraint embedded device. ACM Transactions on Embedded Computing Systems (TECS), 2019, 18(3): 1-13

[85] Zheng X, Xu C, Hu X, et al. The software/hardware co-design and implementation of SM2/3/4 encryption/decryption and digital signature system. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2019, 39(10): 2055-2066

[86] Li Dan, Xu Mao-Zhi. An acceleration method of SM9 identification cipher algorithm. China: CN108650078A, 2018(in Chinese)
(李丹, 徐茂智. 一种 SM9 标识密码算法的加速方法. 中国: CN108650078A, 2018)

[87] Zhen P, Hu X, Yu Y, et al. Research on the optimization computation of SM9 bilinear pairings//Proceedings of the 2017 2nd International Conference on Communication and Information Systems. New York, USA, 2017: 256-261

[88] Wenger E, Unterluggauer T, Werner M. 8/16/32 shades of elliptic curve cryptography on embedded processors//Proceedings of the International Conference on Cryptology in India. Mumbai, India, 2013: 244-261

[89] Hinterwälder G, Moradi A, Hutter M, et al. Full-size high-security ECC implementation on MSP430 microcontrollers//Proceedings of the International Conference on Cryptology and Information Security in Latin America. Florianópolis, Brazil, 2014: 31-47

[90] Gouvêa C P, Oliveira L B, López J. Efficient software implementation of public-key cryptography on sensor networks using the MSP430X microcontroller. Journal of Cryptographic Engineering, 2012, 2(1): 19-29

[91] Kocher P, Jaffe J, Jun B. Differential power analysis//Proceedings of the Annual International Cryptology Conference. Santa Barbara, USA, 1999: 388-397

[92] Chow S, Eisen P, Johnson H, et al. White-box cryptography and an AES implementation//Proceedings of the International Workshop on Selected Areas in Cryptography. Newfoundland, Canada, 2002: 250-270

[93] Wang Teng-Fei, Zhang Hai-Feng, Xu Sen. Security analysis and protection of SM2 algorithm software implementation. Application Research of Computers, 2021, 38(9): 2811-2815(in Chinese)
(王腾飞, 张海峰, 许森. SM2 算法软件实现的安全性分析与防护. 计算机应用研究, 2021, 38(9): 2811-2815)

[94] Lin Ting-Ting, Lai Xue-Jia. Research on white box cryptography. Journal of Cryptography, 2015, 2(3): 258-267(in Chinese)
(林婷婷, 来学嘉. 白盒密码研究. 密码学报, 2015, 2(3): 258-267)

[95] Gu Da-Wu, Wang Lei, Ding Ning, Lu Hai-Ning. SM2 white box password implementation method. China: CN10825-9506, 2018(in Chinese)
(谷大武, 王磊, 丁宁等. SM2 白盒密码实现方法. 中国: CN108259506, 2018)

[96] Pan Wen-Lun, Zhang Li-Ting. White-box realization and application of double point operation. Journal of Cryptography, 2020, 7(3): 311-325(in Chinese)
(潘文伦, 张立廷. 倍点运算的白盒化实现及应用. 密码学报, 2020, 7(3): 311-325)

[97] Jiang Z H, Fei Y, Kaeli D. A novel side-channel timing attack on GPUs//Proceedings of the on Great Lakes Symposium on VLSI 2017. New York, USA, 2017: 167-172

[98] Schindler W. A timing attack against RSA with the Chinese remainder theorem[C]//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Heidelberg, Berlin: Springer, 2000: 109-124. Springer Berlin Heidelberg, 2000: 109-124

[99] Emmart N, Weems C. Pushing the performance envelope of modular exponentiation across multiple generations of GPUs //Proceedings of the 2015 IEEE International Parallel and Distributed Processing Symposium (IPDPS 2015). Hyderabad, India, 2015: 166-176

[100] Liu Z, Großschädl J, Wong D S. Low-weight primes for lightweight elliptic curve cryptography on 8-bit AVR processors//Proceedings of the International Conference on

Information Security and Cryptology. Guangzhou, China, 2013: 217-235

[101] Liu Z, Seo H, Großschädl J, et al. Reverse product-scanning multiplication and squaring on 8-bit AVR processors//Proceedings of the International Conference on Information and Communications Security. Hong Kong, China, 2014: 158-175

[102] Tang J, Zhou Y, Zhang H, et al. Higher-order masking schemes for simon//Proceedings of the International Conference on Information and Communications Security. Beijing, China, 2015: 379-392

[103] Gao L, Zheng F, Wei R, et al. DPF-ECC: A framework for efficient ECC with double precision floating-point computing power. IEEE Transactions on Information Forensics and Security, 2021, 16: 3988-4002

[104] Montgomery P L. Modular multiplication without trial division. Mathematics of Computation, 1985, 44(170): 519-521

[105] Koç C K, Acar T, Kaliski B S. Analyzing and comparing Montgomery multiplication algorithms. IEEE Micro, 1996, 16(3): 26-33

[106] Hutter M, Wenger E. Fast multi-precision multiplication for public-key cryptography on embedded microprocessors//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Nara, Japan, 2011: 459-474

[107] DeSantis F, Sigl G. Towards side-channel protected X25519 on ARM Cortex-M4 processors//Proceedings of the Software Performance Enhancement for Encryption and Decryption, and Benchmarking (SPEED-B). Utrecht, the Nederland, 2016: 1-12

[108] Shoup V. A Computational Introduction to Number Theory and Algebra. UK: Cambridge University Press, 2009

[109] Bizony M. Fermat's last theorem, simon singh (4th estate, 1997): book review. Learning and Teaching Mathematics, 2004, 2004(1): 48-49

[110] Bernstein D J, Yang B Y. Fast constant-time GCD computation and modular inversion. IACR Transactions on Cryptographic Hardware and Embedded Systems, 2019: 340-398

[111] Deshpande S, del Pozo S M, Mateu V, et al. Modular inverse for integers using fast constant time GCD algorithm and its applications//Proceedings of the 2021 31st International Conference on Field-Programmable Logic and Applications (FPL). Dresden, Germany, 2021: 122-129

[112] Zhang Jia-Hong, Chen Jian-Hua, Zhang Li-Na. Implementation of elliptic curve cryptographic algorithm with parallel structure. Computer Engineering and Design, 2007, 28(23): 5598-5600(in Chinese)
(张家宏，陈建华，张丽娜. 并行结构的椭圆曲线密码算法实现. 计算机工程与设计，2007，28(23)：5598-5600)

[113] Hankerson D, Menezes A J, Vanstone S. Guide to Elliptic Curve Cryptography. Germany: Springer Science & Business Media, 2006

[114] Li Zhong, Peng Dai-Yuan. Fast scalar multiplication based on sliding window technique. Computer Science, 2012, 39(B06): 54-56(in Chinese)
(李忠，彭代渊. 基于滑动窗口技术的快速标量乘法. 计算机科学，2012，39(B06)：54-56)

[115] Gallant R P, Lambert R J, Vanstone S A. Faster point multiplication on elliptic curves with efficient endomorphisms//Proceedings of the Annual International Cryptology Conference. Santa Barbara, USA, 2001: 190-200

[116] Galbraith S D, Lin X, Scott M. Endomorphisms for faster elliptic curve cryptography on a large class of curves//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Cologne, Germany, 2009: 518-535

[117] Zhang Jing-Wei, Zhao Chang-An. Conversion algorithm from Weierstrass form to edwards form of elliptic curve on binary domain. Journal of Cryptography, 2018, 5(3): 315-323(in Chinese)
(张婧炜，赵昌安. 二元域上椭圆曲线的 Weierstrass 形式到 Edwards 形式的转换算法. 密码学报，2018，5(3)：315-323)

[118] Ming Jiao-Jiao, Gao Xian-Wei, Dong Xiu-Ze et al. Research on fast scalar multiplication algorithm of edwards curve. Application Research of Computers, 2020, 37(9): 2776-2780(in Chinese)
(明娇娇，高献伟，董秀则等. Edwards 曲线快速标量乘算法研究. 计算机应用研究，2020，37(9)：2776-2780)

[119] Goundar R R, Shiota K, Toyonaga M. SPA resistant scalar multiplication using golden ratio addition chain method. International Journal of Applied Mathematics, 2008, 38(2): 83-88

[120] Dimitrov V, Imbert L, Mishra PK. Efficient and secure elliptic curve point multiplication using double-base chains//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Chennai, India, 2005: 59-78

[121] Liu Shuang-Gen, Li Dan-Dan, Li Xiao. Elliptic curve scalar multiplication algorithm based on bronze proportional addition chain. Journal of Shandong University: Natural Science, 2019, 54(11): 12-19(in Chinese)
(刘双根，李丹丹，李潇. 基于青铜比例加法链的椭圆曲线标量乘算法. 山东大学学报：理学版，2019，54(11)：12-19)

[122] Liu S G, Wang X, Liu Y W, et al. Fast scalar multiplication algorithms based on $5P+Q$ of elliptic curve over GF(3^m). International Journal of Network Security, 2021, 23(4): 604-611

[123] Yu W, Musa S A, Li B. Double-base chains for scalar multiplications on elliptic curves//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 538-565

[124] Roy S S, Reparaz O, Vercauteren F, et al. Compact and side channel secure discrete Gaussian sampling. IACR Cryptology ePrint Archive, 2014, 2014: 591

[125] Bernstein D J, Chuengsatiansup C, Lange T. Curve41417: Karatsuba revisited[C]//Proceedings of the International Workshop on Cryptographic Hardware and Embedded Systems. Springer, Berlin, Heidelberg, 2014: 316-334

DONG Jian-Kuo, Ph. D., lecturer. His main research interest is high performance cryptography.

LIU Zhe, Ph. D., professor, Ph. D. supervisor. His main research interest is cryptographic engineering.

LU Sheng, B. S. His main research interest is high performance cryptography implementation.

ZHENG Fang-Yu, Ph. D., assistant professor. His main research interest is applied cryptography.

LIN Jing-Qiang, Ph. D. professor, Ph. D. supervisor. His main research interest is system security.

XIAO Fu, Ph. D., professor, Ph. D. supervisor. His main research interests include wireless sensor network and cyberspace security.

GE Chun-Peng, Ph. D., associate professor. His main research interest is cryptography.

Background

With the rapid development of cloud computing technology and Internet of Things technology, people are moving towards the intelligent world of interconnected things. The number of applications, including internet finance and mobile payment, show an explosive growth trend. These applications have strong requirements for data privacy protection. This poses a severe challenge to the performance of cryptographic algorithms, especially public-key cryptographic algorithms with high computational complexity.

On the one hand, the service side represented by cloud computing technology needs to face the massive data generated by hundreds of millions of users. The server needs to complete the identity authentication and data protection of these users in a limited time. The huge number of user signatures request is also a challenging problem for cloud computing. On the other hand, on the terminal side represented by the Internet of Things technology, the embedded chips or mobile devices of the Internet of Things are limited because of the cost of production, and their computing resources are often seriously insufficient. There are certain bottlenecks in CPU frequency, memory resources and even power supply. Compared with the traditional RSA algorithm, the elliptic curve algorithm has a shorter key length and has important advantages in computing speed, resource storage, data bandwidth, etc. It can be used to realize key exchange, digital signature, public-key encryption, and other cryptographic primitives. It is one of the most widely used public-key cryptography technologies.

By analyzing the two different application scenarios on the service side and the terminal side, this paper further analyzes the huge differences between the two sides in terms of software, hardware, and cryptographic algorithm requirements. Then we summarize various elliptic curve cryptographic algorithm standards and hardware development platform parameters. Based on the above contents, this paper summarizes the efficient software implementation technology of elliptic curve cryptography. We also focus on the research of domestic elliptic curve cryptography and look forward to the future development trend of the elliptic curve cryptography algorithm. This paper has certain a guiding significance and reference value for the implementation of elliptic curve cryptography engineering.