**山东大学**

**博士学位论文开题报告**

**题 目：安全多方计算及其在分布式密码系统中的应用**

**培 养 单 位 网络空间安全学院（研究院)**

**学科（专业） 网络空间安全**

**导 师 陈宇**

**研 究 生 涂彬彬**

**学 号 202020881**

**开题报告日期 2021.11.**

**研究生院制**

**二〇一五年六月**

目录

[1. 研究背景和意义 3](#_Toc87027110)

[2. 国内外的研究现状及分析 5](#_Toc87027111)

[2.1通用MPC协议的研究现状 5](#_Toc87027112)

[2.2 门限密码的研究现状 7](#_Toc87027113)

[2.2.1 门限加密的研究现状 8](#_Toc87027114)

[2.2.2 门限签名的研究现状 9](#_Toc87027115)

[3. 前期的理论研究与试验论证工作的结果 12](#_Toc87027116)

[3.1 MPC前期理论研究 12](#_Toc87027117)

[3.1.1 安全性质 12](#_Toc87027118)

[3.1.2 相关模型 13](#_Toc87027119)

[3.2 门限密码前期理论研究 14](#_Toc87027120)

[4. 主要研究内容、实施方案及其可行性论证 16](#_Toc87027121)

[4.1主要研究内容 16](#_Toc87027122)

[4.2 实施方案及其可行性论证 17](#_Toc87027123)

[4.2.1基于ECDSA的两方签名算法 18](#_Toc87027124)

[4.2.2 ECDSA适配器签名及其两方扩展 21](#_Toc87027125)

[4.2.3 SM2适配器签名 27](#_Toc87027126)

[4.2.4 SM2签密方案及其两方扩展 31](#_Toc87027127)

[5. 论文进度安排和预期目标 34](#_Toc87027128)

[6. 论文预期创新点 34](#_Toc87027129)

[7. 课题研究的条件、经费支持及导师组人员配备情况 36](#_Toc87027130)

[8. 可能的困难与解决途径 36](#_Toc87027131)

[9. 参考文献 36](#_Toc87027132)

1. 研究背景和意义

安全多方计算（secure multi-party computation，MPC）旨在解决一组互不信任的参与方之间保护隐私的协同计算问题，并且在整个计算过程中，除了由输入输出可以推测出的信息外，没有更多额外的信息泄漏。安全多方计算的安全性可保证数据拥有方的隐私信息安全，同时安全多方计算的功能性可以为数据需求方提供数据之间的共享流通和计算，打破“数据孤岛”限制，发挥数据价值。因此，安全多方计算能在数据拥有方和需求方之间，可实现数据的可信互联互通，解决数据安全性、隐私性问题，大幅降低数据信息交易成本，提升数据使用价值，为数据拥有方和需求方提供有效的对接渠道，形成互惠互利的交互服务网络。

对于密码系统而言，密钥是实现密码功能操作的关键。传统公钥密码的密钥唯一，密钥的安全性完全依靠用户的秘密保存，一旦用户的密钥丢失，不仅难以恢复，还容易形成单点故障问题导致密码系统无法正常运行，甚至恶意敌手截获密钥后，可任意窃取用户隐私信息或者伪造用户身份。门限密码系统完美体现了“不要把所有的鸡蛋放在一个篮子里”的思想，将传统公钥密码的集中化权限进行分散，保证了多用户共同分享密码操作权限，并以一种多方安全计算的方式进行密码操作，使得多个用户分布式协同完成密码操作。门限密码系统可避免传统公钥密码由于密钥唯一而导致的单点故障问题，对于提升密码系统的健壮性和安全性等有着重要作用。在已有的研究中，理论上存在通用的方法解决一般性的安全多方计算问题，即对现有的密码系统采用通用的安全多方计算技术进行分布式扩展，构建安全的门限密码系统，但是Goldreich[1]指出直接应用现有的通用方法来解决具体的问题是低效且不切合实际的。传统密码系统的分布式研究需要兼顾密码系统的实现功能、应用效率、安全性等方面，需要定制的安全多方计算协议。

目前，安全多方计算研究在基础理论和实际应用上己经取得了较多的成果，其在门限密码系统的应用也日渐丰富。但是，随着网络技术的不断发展，特别是云计算、物联网、人工智能、区块链等分布式场景对于隐私保护和数据安全需求的不断提升。无论是在基础理论研究方面，还是在应用协议研究方面，安全多方计算都有待进一步的研究和探讨。例如，通用的安全多方计算协议的计算效率和通信效率需要继续优化；实用的安全模型和安全性分析方法还有待深入研究；应用问题的多样性，特别是对于具体的门限密码系统需要设计更多的特定安全协议来完成具体计算；网络技术的发展也使得安全多方计算不断地面临着新的应用需求。

综上所述，安全多方计算能够在数据不离开数据持有节点的前提下，完成数据的分析、处理和结果发布，能提供数据访问权限控制和数据交换的一致性保障，使得在确保参与各方利益前提下的协作计算成为可能，其研究对于打破“数据孤岛”，促进分布式网络环境下的信息交流与数据共享产生巨大的推动，并在诸多的领域拥有广泛的应用空间。此外，安全多方计算技术能对传统密码系统进行分布式扩展，细化密码操作的访问控制权限，防止单点故障问题，提升密码系统的健壮性和安全性，适用于当前分布式应用系统的应用需求。由于网络环境的复杂性和应用问题的多样性，安全多方计算研究尚不充分，较多问题有待进一步的研究和探讨。因此，对于基础安全多方计算协议展开研究，提升应用效率，加强安全性，同时针对特定的应用问题，特别是针对标准密码算法的进行分布式设计，满足门限密码系统的应用需求，不仅具有较重要的理论意义，而且对于我国经济和社会领域具有广泛的应用前景。

1. 国内外的研究现状及分析

1982年，Yao[2]用“百万富翁问题”的实例首次描述了两方安全计算问题，即两个百万富翁在不泄露自己财产数额的前提下，如何比较出谁更富有？该问题可以抽象为：拥有秘密输入的双方，如何在保证输入的隐秘性和结果正确性的同时完成数值大小比较的计算任务。随后，Goldreich，Micali和Wigderson[3]将两方安全计算概念推广到多方安全计算。

目前对于安全多方计算的研究主要可以分为两个方面：一方面是安全多方计算基础理论的研究，主要包括对安全多方计算基础协议的构建和完善，以及对设计通用安全多方计算协议的方法论的研究。另一方面的研究是将安全多方计算与应用问题相结合，为应用设计兼顾效率与安全需求的多方计算协议，特别是门限密码系统的研究，比如：门限解密和门限签名等。

## 2.1通用MPC协议的研究现状

安全多方计算的基础理论是安全多方计算应用协议研究的理论基石。具体包括协议的存在性、协议的计算模型，协议安全性、协议公平性以及协议构造方法等方面的内容。1987年Goldreich等[3]提出一种通用安全计算协议的构造方法，攻克了安全多方计算的存在性难题，为安全多方计算后期研究工作奠定了坚实的基础。1998年，Goldreich[1]对安全多方计算的概念、计算模型、半诚实模型下的安全性定义以及通用协议的描述进行全面概括。

目前，通用安全多方计算协议设计主要基于混淆电路、不经意传输、秘密分享、同态加密等技术构造。Yao[2]首次提出混淆电路技术并结合不经意传输，给出了第一个通用的两方安全计算协议——Yao协议。该方法将要计算的函数转换成一个布尔电路C，布尔电路可分解为AND、OR等基础的门电路。随后，依次实现基础门电路也就是获得了计算函数的最终结果。目前Yao混淆电路仍然是构造安全两方计算通用协议的一个非常有效的基础技术，后续有大量的工作给出了进一步的效率优化，如point-and-permute[4]，free XOR[5]，GRR3/2[6,7]，fleXOR[8]，half gates[9]，slicing and dicing[10]等。基于Yao混淆电路的通用安全两方计算协议的计算效率较高，通信轮数是常数，但是通信复杂度与电路的AND门数线性相关，而且扩展到多方场景较为复杂。

Goldreich等[3]基于秘密分享和不经意传输技术给出了通用的两方安全计算协议——GMW协议。该协议通过对每个秘密输入进行异或分享，然后逐层逐门的对电路求值，最后通过恢复输出导线的值获得最终的函数结果。该技术较容易扩展到多方场景，但是要求各参与方之间进行多轮交互，通信轮数和电路的层数相关，通信复杂度较高。

Ben-Or等[11]基于Shamir的秘密分享[12]给出了通用的安全多方计算——BGW协议。该协议通过对每个秘密输入进行秘密分享，巧妙的利用Shamir秘密分享方案的同态性质，计算各门的输出分享，最后通过秘密分享的恢复算法组合最终的函数结果。该协议适用于对域上的包含加法、乘法、标量乘法门的算术电路求值，计算效率较高且容易扩展到多方场景，但是通信轮数和电路层数相关，而且在计算乘法门时会导致导致多项式次数扩张，要求更多的参与方进行降次。随后，Beaver[13]提出了“乘法三元组”的概念，将MPC协议划分为（参与方输入未知时）预处理阶段和（参与方选择好输入时）在线阶段，在预处理阶段各参与方生成一些关联性的随机量，在线阶段使用预存的随机量实现高效的计算。该方法极大的减少了BGW协议中的通信复杂度，避免了计算乘法门时的多项式扩张问题。

## 2.2 门限密码的研究现状

门限密码系统可看作关于密码功能操作(解密/签名)的特殊的多方安全计算，多个用户秘密地分享密钥信息，在进行解密/签名操作时，用户可使用自己的私钥，通过多方安全计算的模式多方协作解密密文/签名消息，对于解决单点故障问题，以及分布式环境下多用户的数据安全、隐私保护和身份认证有着重要的应用意义。虽然理论上可对现有的密码系统采用通用的安全多方计算技术进行分布式扩展，构建安全的门限密码系统，但是Goldreich[1]指出直接应用现有的通用方法来解决具体的问题是低效和不切实际的。因此，针对具体的门限密码系统的设计需要兼顾实现功能、应用效率、安全性等方面，构造特定的安全多方计算协议。1979年，Shamir和Blakley分别独立地提出了秘密分享方案，门限的思想便深入人心。随后，Desmedt和Frankel等[14,15]在研究面向群组的密码(group oriented cryptography)时，引入了门限密码的概念，彻底地打开了门限密码研究的大门。我们分别通过门限加密和门限签名两个方向介绍门限密码的研究现状。

### 2.2.1 门限加密的研究现状

1989年，Desmedt和Frankel[15]使用Shamir的门限秘密分享技术分割私钥，对ElGamal加密方案进行门限化，构造了高效的选择明文安全的门限加密方案。随后，De Santis等[16]提出函数分享的概念，并基于 RSA假设实例化了函数分享，构造了选择明文安全的门限加密方案。

对于构造选择密文安全的门限加密方案的探索，Lim和Lee[17]首先观察到设计选择密文安全的门限加密，需要密文可公开验证。随后，Shoup和Gennaro[18]给出了非交互的选择密文安全的门限加密的形式化定义，并基于Lim和Lee[17]的密文可公开验证思想，使用Chaum-Pedersen[19]的∑-协议，应用Fiat-Shamir启发式设计的非交互的零知识证明来保证密文的可公开验证性，并在随机预言机模型下设计了首个可实践的选择密文安全的门限加密方案。Canetti和Goldwasser[20]基于Cramer和Shoup[21]的选择密文安全的公钥加密方案，并在标准模型下将其门限化，构造了选择密文安全的门限加密方案。此外，Canetti和Goldwasser[20]还提出了基于多重加密思想构造门限加密方案，该思想后续也影响了Zhang等人[22]，Dodis和Katz[23]，以及Xie等人[24]构造门限加密的方法。在Eurocrypt 2011上，Wee[25]对该思想扩展，将可提取哈希证明系统门限化，引入新的密码组件——门限的可提取哈希证明系统，并基于门限的可提取哈希证明系统设计了门限加密的通用构造方法，然后分别基于整数分解假设和DDH假设给出具体的实例化方案。此后，Libert和Yung[26]对Wee提出的门限可提取哈希证明的思想进一步推广，引入了all-but-one perfectly sound threshold hash proof systems的概念，给出了自适应模型下选择密文安全的门限加密方案的构造，并分别基于素数阶的双线性群上的DLIN假设和 symmetric external Diffie-Hellman假设实例化了门限加密方案。涂彬彬等[27]提出了门限(有损)单向陷门函数的概念，给出了自适应模型下选择明文安全的门限加密方案的通用构造，并基于DDH假设和LWE假设给出具体的实例化方案。

### 2.2.2 门限签名的研究现状

与传统数字签名相比较，门限签名将签名私钥分散给多个用户，只有当不少于门限值个用户协同签名，才能完成正确的签名，而少于门限个用户无法通过合谋签名。因此，在构造门限签名时，不仅要考虑签名的不可伪造性，还需要考虑敌手对于目标用户的攻击。

1991年，Desmedt和Frankel[28]设计了可在阿贝尔群上运算的门限秘密分享方案，首次基于RSA假设设计了门限签名方案。随后，基于不同的密码假设的门限签名方案也相继被设计。Shoup[29]采用了消除分母的技术(clearing out the denominator)，在静态模型下基于RSA假设构造了非交互的门限签名方案。Katz和Yun[30]对修改版本的Rabin签名方案进行了门限化，获得静态模型下基于整数分解假设的非交互的门限签名方案。在Eurocrypt 2011上，Wee[25]基于门限可提取哈希证明系统设计了静态模型下门限签名的通用构造，并给出丰富的实例化。随后，Bendlin等人[31]将Gentry等人[32]的基于格的签名方案门限化，设计了基于格的门限签名方案。Boneh等人[33]提出了门限通用转化器的概念，并基于该组件可将格上的签名方案转化成门限版本，获得基于格的门限签名方案。

上世纪九十年代，随着数字签名标准(digital signature algorithm，DSA)的提出和广泛应用。特别是，近些年密码货币的爆发式发展，基于椭圆曲线的数字签名标准(ECDSA)在比特币等密码货币中承担重要应用。而在其中签名密钥的丢失也意味着具体经济的损失。由于比特币系统中使用了多重签名的解决方案，并非门限签名，限制了系统的灵活性，难以支持复杂的过程结构。因此，对DSA进行门限化具有重要的应用意义。

在门限DSA的研究探索中，由于签名算法是概率算法，对其门限化需要所有签名参与方通过交互运算共同协商随机数，导致门限化非常困难。1996年，Gennaro等人[34]给出了交互式门限签名的形式化定义，并基于联合随机秘密分享方案(joint random secret sharing，JRSS)和联合零秘密分享方案(joint zero secret sharing，JZSS)，并设计了计算乘积和求逆的多方安全计算方法，对DSA进行门限化，构造了全分布式的门限数字签名标准。具体而言，根据JRSS技术协商出DSA的签名验证公钥和签名私钥，并分别保留私钥分享。签名阶段，各参与方使用JRSS协商随机数k并各自保留随机数分享，使用多方安全求逆运算计k的逆，使用多方安全乘积运算计算签名私钥与随机数k的组合，各参与方可获得签名值的分享。最后各参与方公布签名分享，达到门限值即可组合出最终签名。但是该方案在组合签名阶段，各用户需要计算私钥与随机数的乘积与随机数的求逆，导致了多项式的次数扩张。具体而言，当门限值是t时，至少需要2t+1个用户才能组合出签名。因此，该方案并非门限最优化的，而且通信消耗非常大，难以应用。

2016年，Gennaro等人[35]针对门限数字签名标准的多项式扩张问题，基于Paillier[36]的同态加密方案的门限版本配合陷门承诺技术，构造了最优化的门限数字签名标准。具体而言，该方案采用了门限同态加密技术，各参与方可以使用同态加密的性质对于明文进行密态操作，保证了各方隐私信息的安全性，并可完成复杂的签名运算。但是该方案依旧要求各用户之间进行大量的交互运算通信消耗较高，而且签名算法需要复杂的零知识证明。随后在CCS 2018上，Gennaro和Goldfeder[37]对上述方案进行优化，使用了特殊的多方安全计算技术，避免了使用复杂的零知识证明方法，设计了简单的门限ECDSA方案。但是该方法依旧使用了Paillier的同态加密方案，而目前并没有针对Paillier加密方案的高效分布式密钥生成算法。针对该问题，Lindell等人[38]使用指数ElGamal的加法同态算法代Paillier的加法同态加密算法设计了首个高效的门限 ECDSA分布式密钥生成算法和签名算法。

在2010年底，我国国家密码管理局发布了SM2椭圆曲线公钥密码算法[39]，对我国商用密码产品和信息安全体系建设具有重大意义。尚铭等人[40]填补了SM2算法在多方合作应用环境下的空缺，基于秘密分享的思想，对SM2签名算法进行门限化，分别针对存在可信中心和不存在可信中心的情况下，设计门限的SM2签名算法。在门限值为t，总用户数量为n>2t时，任何2t+1个参与者集合可以生成一个有效的签名。2020年, Yudi Zhang等[41]提出了基于同态加密的 SM2 两方协同签名算法，并给出了可证明安全分析。随后，涂彬彬等[42]归纳了三种安全多方计算方法, 分别是基于门限加法同态加密算法、基于多个加法同态加密算法和基于单一加法同态加密算法的安全多方计算协议，并设计了门限化的SM2签名算法和SM9密钥生成算法。

1. 前期的理论研究与试验论证工作的结果

## 3.1 MPC前期理论研究

### 3.1.1 安全性质

安全多方计算是指在分布式网络中多个互不信任而又相互独立的参与方利用自己的隐私信息作为输入，协同计算某个预先约定的功能函数，并在计算结束后得到各自相应的函数输出的过程。在此过程中，各参与方均不能获得其它参与的任何私有信息。

一般来说，安全多方计算中的安全性包括隐私性、正确性、输入独立性、确保获得输出正确性和公平性五个方面[43]：

隐私性(Privacy)：各参与方只能获取自身输入、输出以及可由其推导出的信息，而不能获得其他额外信息；

正确性(Correctness)：保证协议的正确执行，各参与方获得正确的输出结果；

输入独立性(Independence of inputs)：保证不诚实的参与方选择的输入与其他诚实参与者的输入独立；

确保获得输出(Guaranteed out delivery)：不诚实的参与方不能阻止诚实参与方获得自己的输出；

公平性(Fairness)：不诚实的参与方能够获得输出，当且仅当诚实的参与方获得各自输出。

### 3.1.2 相关模型

(1)参与方模型

根据协议执行中的参与者行为的规范性，可以将参与方分为诚实参与方、半诚实参与方和恶意参与方三种类型。诚实参与方完全按照协议执行，不存在提供虚假数据、泄漏信息、窃听消息和中止协议的行为。

半诚实参与方能够按照协议要求执行，也不存在数据作假、恶意中止协议等行为，但是他们会保留所有收集到的信息，试图从中推断出更多的额外消息。

恶意参与方可无视协议执行要求，会恶意地破坏协议。

(2)攻击能力模型

根据对受控方的控制方式不同，攻击者可以分为静态攻击和动态攻击。静态攻击只能够在协议开始前选择受控参与者；协议执行过程中，攻击者只能控制已选择的参与者集合。动态攻击者可以根据自己的意愿，在协议开始之前或协议执行过程中随意选择受控参与者，但在执行过程中受控方不再更改。

根据攻击者的计算能力不同，攻击者可以分为无穷计算能力的攻击者和有限能力的攻击者。具有无穷计算能力的攻击者，可以计算任何数学问题，其计算能力不受任何限制。此类攻击者的安全协议称为信息论安全。有限能力的攻击者，即具有概率多项式时间计算能力的攻击者。此类的安全协议称为密码学安全。

根据对控制方进行控制的能力不同，攻击者可以分为被动攻击者和主动攻击者。被动攻击者只能通过被腐败的参与者收集各类信息，包括输入、输出以及相关中间结果，但他们不能控制协议，而且完全遵守协议的执行步骤。主动攻击者则能够完全控制通信信道，具有中间人攻击能力，不但能够对信道中的消息实施删除、注入、修改、重放、阻止等操作，而且能够调度协议的执行。

(3)通信模型

安全多方计算协议的实现离不开分布式环境。通常情况下，网络通信模型包括同步通信模型和异步通信模型两种。同步通信模型是指所有协议的参与方共享一个时钟服务器，消息的收发都在瞬间完成。这种模型能够保证协议运行和消息通信的有序进行。异步通信模型的各参与方则按照不同的时钟周期接收或者发送消息。因此，消息在传输时可能存在很大延时或乱序的情况。

信道模型可分为安全信道和不安全信道。在安全信道中，每一个成员身份标识的以及其发送的消息都能够被其他成员收到，而且在该信道上传输的消息不会被第三方获取，即便是攻击者也不能够窃听或者修改通信内容。因此，参与者在安全信道中的交互被看作是安全的。在不安全信道中，存在攻击者窃听或者篡改参与方之间的通信内容的情况。

## 3.2 门限密码前期理论研究

门限密码系统主要包括门限加密和门限签名，具体形式类似公钥系统中的加密和签名。门限加密包括密钥生成、加密和解密过程，以及一个消息组合器 (combiner)。在解密过程中，用户使用自己的私钥解密密文，并将解密分享发送给消息组合器，消息组合器在接收到门限值个解密分享时可组合出原消息。同样，门限签名包括密钥生成、签名和验签过程，以及一个签名组合器。在签名过程中，用户使用自己的私钥签名消息，并将签名分享发送给签名组合器，签名组合器在接收到门限值个签名分享时可组合出该消息的签名。如果门限密码的密钥生成阶段不需要密钥生成中心，系统的公钥和各用户的私钥是由用户自己交互完成，则可称该门限密码是全分布式的 (full distributed)[44,45]。如果门限密码的解密/签名阶段，不需要各用户之间或者与组合器进行交互，则称该门限加密/签名是非交互的(non-interactive)[46]。

与传统公钥密码相比，门限密码有着丰富的功能性、更强的安全模型和特殊性质。在密码功能方面，门限密码将传统公钥密码操作从以往的单一模式扩展到多用户合作模式。在安全性模型上，门限密码除了考 虑类似公钥密码方案的安全性外，还要考虑敌手对于用户的攻击，比如: (1)弱的攻击模型——静态攻击模型 (static corruption model)。 敌手在系统参数发布前先选定攻击的目标用户，可获得目标用户的私钥信息；(2)强的攻击模型——自适应攻击模型 (adaptive corruption model)。敌手可在系统运行的任何时刻，根据具体的攻击情况自适应地选择攻击的目标用户，获得目标用户的私钥信息。除了丰富的功能性和更强的安全性，门限密码还具有一些其它特性：(1)紧致性 (compactness)：公钥尺寸和密文/签名尺寸与参与用户的数量无关。(2)鲁棒性(robustness)：各用户的解密/签名分享的正确性可被公开验证。(3)全分布式：系统的公钥和用户的私钥可由用户自己通过交互生成，避免了密钥生成中心的权限过大或者被攻击者控制等带来的安全风险。(4)抗合谋：门限密码系统要求参与的用户数量达到门限值，才能正确完成密码操作，防止了单个用户失败导致整个系统瘫痪，同时防止任意少于门限值个用户合谋。

1. 主要研究内容、实施方案及其可行性论证

4.1主要研究内容

（1）探索高效且安全性更强的通用MPC协议

调研现有MPC的安全性模型，比如半诚实模型、恶意模型、静态模型、自适应模型等，研究各种安全模型下MPC的存在性问题，以及具体的MPC协议构造技术。同时，结合实际应用和具体攻击模式，探索更贴近现实的安全性模型，并给出具体的MPC协议设计。

梳理现有的通用MPC协议设计技术，比如基于Yao混淆电路的MPC、基于不经意传输的MPC，以及基于秘密分享的MPC等。探索是否存在其他通用MPC协议设计技术，丰富现有的通用MPC协议构造模式。同时，深入了解不同通用MPC协议的优化技术，比如对于Yao-协议的优化技术：point-and-permute[4]，free XOR[5]，GRR3/2[6,7]，fleXOR[8]，half gates[9]，slicing and dicing[10]等，进一步探索通用MPC效率优化技术，实现更加高效的通用MPC协议。

（2）MPC在门限密码系统的丰富应用

在物联网、云计算、区块链等分布式应用场景下，针对传统公钥密码应用的局限性，比如：功能性上难满足具体的应用需求，实际应用效率应用系统难以承受，安全性上难以达到安全需求等。通过深挖安全多方计算技术的应用模式，结合分布式应用需求，针对具体的密码算法，探索安全高效的门限密码构造。

目前对于ECDSA，SM2，SM9等标准密码算法的分布式应用，依旧存在门限密码构造需求[42,47]。基于ECDSA，SM2，SM9的门限签名需要的MPC技术较为复杂，想要实现恶意模型下的安全性也需要昂贵的零知识证明技术。SM9的主私钥的托管机制在应用中存在限制，比如签名私钥要求用户独自拥有，通过门限密钥管理方式，探索SM9主私钥的分布式管理机制，实现用户私钥由多个密钥管理中心协同生成。

探索传统密码算法，特别是标准密码算法的功能扩展，跟踪研究新型应用和具体功能，基于安全多方计算技术进行分布式设计，解决分布式应用场景下的密码算法应用需求。比如：适配器签名 (adaptor signature, AS)[48,49,50]，也称为无脚本脚本 (scriptless script) 可看作是数字签名对困难关系的扩展，联合了签名授权和证据提取两种功能，是目前解决区块链应用 (如密码货币) 中扩展性差、吞吐量低等问题的重要密码技术。我们针对现有适配器签名存在的问题，一方面探索门限适配器签名的形式化定义，为后续方案设计提供安全性分析基础，另一方面，我们对现有ECDSA的适配器签名进行效率优化，并采用安全多方计算技术进行分布式设计，同时 探索基于SM2构造适配器签名算法的构造方法和门限化的设计技术。

4.2 实施方案及其可行性论证

结合安全多方计算技术，对现有标准密码算法进行分布式门限化的设计，扩展的现有密码算法的应用功能，满足了分布式应用场景的应用需求。目前，我们构造了基于ECDSA的两方签名算法、高效的ECDSA适配器签名、两方ECDSA适配器签名算法、高效的SM2适配器签名算法，以及SM2签密算法及其两方扩展。

在可行性论证方面，我们对已构造的方案进行代码实现，实验结果表明具体方案可完美实现相关功能且应用效率较高。

### 4.2.1基于ECDSA的两方签名算法

目前基于ECDSA构造的两方签名算法需要复杂的安全多方计算技术，比如同态加密、不经意传输、秘密分享等[47,51,52]，同时需要昂贵的零知识证明技术抵抗恶意的敌手。我们发现ECDSA的签名结构具有可组合性质，即签名相同消息的两个ECDSA签名值可以组合成新的签名。基于这个观察，我们首次提出组合ECDSA算法，并基于ECDSA安全性给出了组合ECDSA签名的安全性。同时，对组合ECDSA签名算法进行两方分布式设计，构造安全高效的两方签名协议。该协议依旧可基于ECDSA可证明安全。

（1）组合ECDSA算法

组合ECDSA算法包含三个算法：密钥生成算法Gen，签名算法Sign和签名验证算法Vrfy，具体如下：

* : 密钥生成算法输入公开参数，均匀随机选择,计算。其中，签名私钥是,签名验证公钥是;
* : 签名算法输入签名私钥和消息，计算消息的哈希值，选择随机数,计算，如果r=0则重新选择随机数k，最后计算,输出签名值;
* : 签名验证算法输入签名验证公钥，消息，签名值，计算消息的哈希值，计算，如果，则输出1，否则输出0。

**引理1.**如果ECDSA是EUF-CMA安全的，则组合ECDSA也是EUF-CMA安全的.

（2）基于组合ECDSA的两方签名协议

基于组合ECDSA的两方签名协议包含分布式密钥生成协议DGen，分布式签名协议DSign和签名验证算法Vrfy。其中，签名验证算法和组合ECDSA的验证算法相同。令两方参与方分别是A和B。分布式密钥生成协议DGen和分布式签名协议DSign具体如下：

* 分布式密钥生成协议输入公共参数DGen(pp)：

1. 参与方A首先运行，并发送给;
2. 参与方B验证的消息并接受，否则拒绝，然后运行，并发送给;
3. A验证的消息并接受，否则拒绝；

输出：

1. A计算，存储；
2. B计算，存储；

* 分布式签名协议DSign(sid,m)：

1. A选择随机数，并计算，然后A发送给；
2. B接收的消息，选择随机数，并计算，发送给；
3. A验证的消息并接受，否则拒绝，然后发送给；
4. B验证的消息并接受，否则拒绝，然后计算；在接收到待签名的消息m后，B计算，并将发送给A；
5. A计算，计算，，A验证签名，如果正确，则输出该签名，否则拒绝。

**引理2.** 如果ECDSA是EUF-CMA安全的，以及两方签名协议中使用的零知识证明和承诺方案也是安全的，则上述构造的两方签名协议也是安全的。

（3）可行性分析

我们基于OpenSSL库，使用C++语言，实现了ECDSA，组合ECDSA，以及基于ECDSA的两方签名协议。实验结果表明我们的两方签名协议的签名算法和ECDSA的算法耗时相近，各项操作的耗时都是微秒级的，非常高效。实验结果如下所示：

表1. ECDSA，组合ECDSA以及两方签名协议的效率对比（微秒）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 方案 | 密钥生成 | 离线签名 | 在线签名 | 验证 |
| ECDSA | 11.07 | 37.03 | 2.05 | 87.56 |
| 组合ECDSA | 22.91 | 34.90 | 3.91 | 140.62 |
| 两方签名-参与方A | 61.42 | 100.2 | 2.06 | —— |
| 两方签名-参与方B | 65.41 | 84.29 | 1.98 | —— |

其中，—— 表示不需要该操作

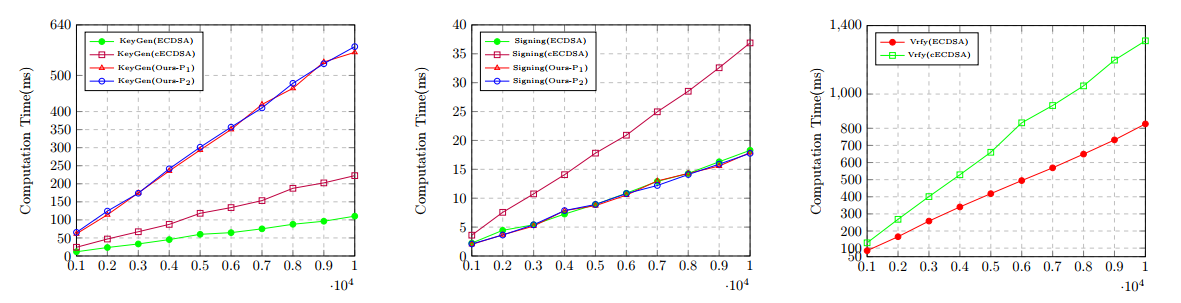


图1. 效率对比（左）密钥生成算法（中）签名算法（右）验证算法

### 4.2.2 ECDSA适配器签名及其两方扩展

现有的ECDSA适配器签名计算预签名值时的零知识证明和签名使用的随机数相关，难以离线执行，导致预签名过程效率较低。现有的两方适配器签名的形式化定义满足公钥可聚合的性质，该性质与较多的高效两方ECDSA签名算法不符合。因此，我们设计支持离线证明的ECDSA适配器签名，且给出满足交互式密钥生成的两方适配器签名的形式化定义，然后给出两方ECDSA适配器签名，并给出可证明安全。

（1）ECDSA适配器签名

关于困难关系R的ECDSA适配器签名包含预签名算法pSign，预签名验证算法pVrfy，适配算法Adapt和提取算法Ext。令困难关系和，令P和V分别表示证明和验证算法。

* : 预签名算法输入公钥Q和签名私钥x，消息，以及困难实例，计算，运行证明算法，并选择随机数，计算，，，输出预签名值；
* : 预签名验证算法输入公钥Q，消息m和困难实例和预签名值，首先验证Z结构的证明，如果，输出0，否则计算，如果 则输出1，否则输出0;
* : 适配器算法输入证据y和预签名值，计算，并输出签名值；
* : 提取算法输入签名值，预签名值和困难实例，计算，如果，则输出y，否则输出；

**引理3.**如果ECDSA是安全的且R是困难关系，则ECDSA适配器签名是安全的。

1. 支持离线证明的ECDSA适配器签名

令P和V分别表示证明和验证算法，令困难关系，表示存在证据使得成立，表示存在证据使得且成立。

* : 预签名算法输入公钥Q和签名私钥x，消息，以及困难实例，如果，输出0，否则选择随机数，计算，，，输出预签名值；
* : 预签名验证算法输入公钥Q，消息m和困难实例和预签名值，首先验证Z结构的证明，如果，输出0，否则计算，如果 则输出1，否则输出0;
* : 适配器算法输入证据y和预签名值，计算，并输出签名值；
* : 提取算法输入签名值，预签名值和困难实例，计算，如果，则输出y，否则输出；

1. 支持分布式密钥生成的适配器签名

关于困难关系R和两方签名算法的两方适配器签名包含交互式预签名协议IpSign，预签名验证算法pVrfy，适配算法Adapt和提取算法Ext。其中，两方签名算法包含交互式密钥生成算法IGen和交互式签名算法ISign，以及签名验证算法Vrfy。

* ：交互式预签名协议输入签名消息和困难实例Y，以及参与方A的私钥和参与方B的私钥，输出预签名值；
* pVrfy，Adapt 和Ext是非交互的和适配器签名中的对应算法相同。

两方适配器签名的正确性和可适配性类似适配器签名，除了使用IGen和IpSign代替Gen和pSign算法

不可伪造性的定义类似两方签名，除了敌手在签名伪造实验中可以访问额外的预签名预言机。具体而言，在签名伪造实验中，敌手可以访问交互式密钥生成预言机，交互式签名预言机和交互式预签名预言机。上述预言机模拟诚实参与方操作。其中，交互式密钥生成预言机和交互式签名预言机的定义和两方签名的不可伪造实验中的定义相同。令交互式密钥生成预言机，交互式签名预言机和交互式预签名预言机初始化机器，，，其中，拥有的状态。

定义交互式预签名预言机如下：

1. 接收到，检测是首次接收到会话符号，则初始化新的机器。该机器拥有在交互式密钥生成结束时存储的签名私钥分享和其他状态，输入消息m，以及困难实例Y，并模拟诚实的参与方执行交互式签名协议；
2. 接收到，检测不是首次接收到会话符号时，预言机操作机器进行交互，如果机器结束，则输出机器的输出。

**定义1.**支持交互式密钥生成的两方适配器签名满足不可伪造性，如果对任意的PPT敌手，存在可忽略函数negl，，满足

其中签名伪造实验定义如下：

|  |  |
| --- | --- |
|  |  |
| 返回 |  |
|  |
|  |
|  |
|  |

**定义2.**支持交互式密钥生成的两方适配器签名满足证据可提取性，如果对任意的PPT敌手，存在可忽略函数negl，，满足

其中证据提取实验定义如下：

|  |  |
| --- | --- |
|  |  |
| 返回 |  |
|  |
|  |
|  |
|  |

**定理1.**支持分布式密钥生成的两方适配器签名算法是安全的，如果该算法满足两方预签名值的可适配性，两方适配器签名的选择消息不可伪造性，以及两方适配器签名的证据可提取性。

（4）两方ECDSA适配器签名

两方ECDSA适配器签名协议主要包含交互式密钥生成算法IGen和交互式预签名算法IpSign，其中交互式密钥生成算法IGen和两方ECDSA签名的密钥生成算法相同，预签名验证算法、适配算法和提取算法和ECDSA适配签名相同。协议描述如下：

* :

1. 参与方A选择私钥，计算，然后A发送给；
2. 参与方B接收的消息，选择私钥，计算，然后A发送给；
3. A接收的消息并验证，如果接受，则计算，并运行Paillier加密的密钥生成算法，其中，，加密获得，发送给，发送给，发送给B；本地存储;
4. B接收的消息，接收的消息，接收A的消息，B验证接收的消息，如果通过则计算，并本地存储;

* :

1. 参与方A选择随机数，计算，然后A发送给；
2. 参与方B接收的消息，然后选择随机数，计算，并发送消息给;
3. A接收的消息，如果验证通过，则计算，，发送给；
4. B接收的消息，验证通过则计算，随机选择，计算,计算,并发送给A；
5. A接收到消息，解密，，，即预签名值，验证预签名的正确性，如果通过则输出预签名值。

（5）可行性分析

我们基于OpenSSL库，使用C++语言，实现了ECDSA适配器签名， 我们的ECDSA适配器签名，以及支持离线证明的ECDSA适配器签名方案。实验结果表明，支持离线证明的ECDSA适配器签名的在线预签名效率比现有的ECDSA适配器签名快了近2.5倍，运行在线预签名操作1000次，三个方案的预签名操作的平均每次耗时分别为174.75 µs, 198.91 µs 和74.74 µs。实验结果如下所示：

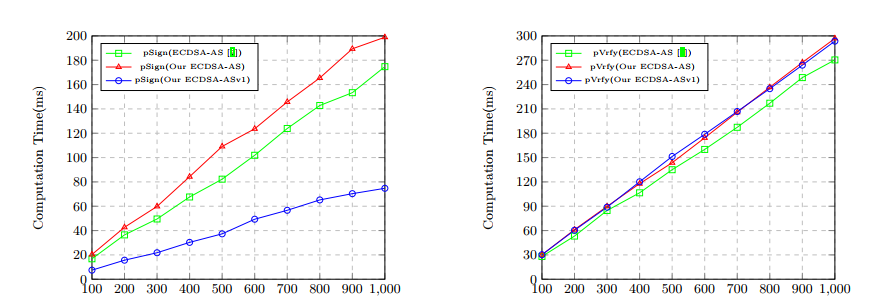


图2. 预签名算法以及预签名验证算法的效率对比

### 4.2.3 SM2适配器签名

(1)SM2适配器签名

SM2 适配器签名是 SM2 签名和困难关系的扩展, 主要包含了预签名算法pSign、预签名验证算法pVrfy、适配算法Adapt、证据提取算法Ext, 以及 SM2 的密钥生成算法Gen、签名算法Sign和签名验证算法Vrfy. 令和， P 和 V 分别表示零知识证明系统中的证明和验证算法. 基于 SM2 的适配器签名算法SM2-ASv1构造如下:

* : 预签名算法输入公私钥对，消息值m和困难实例，计算 ，运行, 计算, 选择随机数，并计算，，，最后输出预签名值；
* : 预签名验证算法输入公钥X, 消息m, 困难实例, 以及预签名值, 运行, 则输出 0, 否则计算e = H(m), ，如果，则输出1，否则输出0；
* : 适配算法输入预签名值和证据y，计算, 并输出签名值;
* . 证据提取算法输入签名值, 预签名值和困难实例 , 计算，如果，输出 y，否则输出;

**引理4.**如果SM2签名算法是安全的且R是困难关系，则SM2适配器签名是安全的。

（2）支持离线证明的SM2适配器签名

我们对 SM2 适配器签名SM2-ASv1进一步优化, 基于离线证明技术设计更加高效的 SM2适配器签名. 根据 , 可分别以 x 和 y 为证据证明 Z 的结构, 其中, 以 x 为证据, 该证明只能由预签名方生成; 以 y 为证据, 该证明可由困难关系生成方生成.根据 SM2-ASv1, 该方案主要包含了两对困难关系 和的零知识证明, 其中的证明可由困难关系生成方离线生成并证明, 后者在预签名算法阶段以预签名私钥 x 为证据, 证明离散对数相等的困难关系. 根据 SM2签名算法特点, 的证明过程与预签名过程中的随机数选择以及签名消息是独立的, 该部分可由预签名方在离线阶段执行. 离线证明构造较为直观, 即预签名方在接收 (或生成) 困难关系 后, 先离线生成 Z 并进行离线证明, 后续对消息 m 进行预签名时, 再选择随机数并生成预签名值. 因此, 该方案可根据 SM2-ASv1 直接得到. 同时, 该过程依旧保持困难关系的自证明结构, 并不影响 SM2 适配器签名算法的安全性.

构造由困难关系生成方以y为证据离线证明Z结构的SM2适配器签名方案SM2-ASv2.令困难关系, 其中 P 和 V 分别表示零知识证明中的证明和验证算法. 基于 SM2 的适配器签名算法构造如下:

* : 预签名算法输入公私钥对，消息值m和困难实例，如果，则输出0，否则计算, 选择随机数，并计算，，，最后输出预签名值；
* : 预签名验证算法输入公钥X, 消息m, 困难实例, 以及预签名值, 运行, 则输出 0, 否则计算e = H(m), ，如果，则输出1，否则输出0；
* : 适配算法输入预签名值和证据y，计算, 并输出签名值;
* . 证据提取算法输入签名值, 预签名值和困难实例 I, 计算，如果，输出 y，否则输出;

困难实例I包含Y = yG和 Z = y(X + G)，皆以y为证据，具体的零知识证明可由困难关系生成方生成。因此，预签名方接收到该困难实例I可直接离线验证其正确性，随后需要对消息进行预签名时，根据离线验证的， 并选择随机数对消息进行预签名。该方法避免了在线预签名阶段中计算Z的过程，并可通过零知识证明其结构，与在线SM2-ASv1和现有ECDSA-AS相比较，SM2-ASv2在线预签名的计算效率更高。由于Z的零知识证明加入到离线证明的困难关系中，预签名中不需要过多的包含Z的信息。因此，SM2-ASv2签名值更小。同样，根据对SM2-ASv2的分析，离线证明过程依旧保证自证明结构，不影响适配器签名的安全性。

（3）可行性分析

我们基于OpenSSL库，使用C++语言，实现了别实现了 ECDSA适配器签名(ECDSA-AS)，SM2适配器签名(SM2-ASv1)和支持离线证明的SM2适配器签名(SM2-ASv2)。我们对比了三种方案在线预签名操作的计算效率以及预签名验证算法的计算效率。我们分别运行各方案的程序 1000 次, 发现 ECDSA-AS, SM2-ASv1 和 SM2-ASv2 的在线预签名操作单次平均耗时分别为 166.54µs，176.27µs，37.14µs；预签名验证操作单次平均耗时分别为 252.86µs，238.46µs，234.80µs。实验结果符合理论分析，SM2-ASv2 的在线预签名耗时仅为 ECDSA-AS 的 1/4。

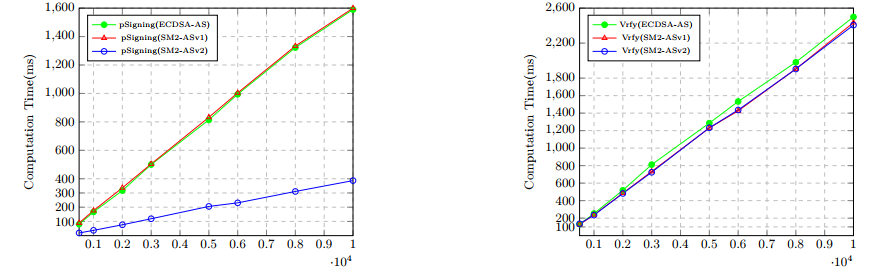


图3. 预签名算法和预签名验证算法的效率对比

### 4.2.4 SM2签密方案及其两方扩展

根据签密的模块化构造，我们分别采用椭圆曲线上的 Pedersen 承诺、 SM2 签名算法和 SM2 加密算法实例化上述通用构造，获得了具有消息同态计算功能的 SM2 签密方案，主要包含参数建立、密钥生成算法、承诺算法、签密算法、验证算法、解密算法和打开算法。具体构造如下：

参数建立阶段：椭圆曲线系统参数包括有限域的规模q，定义椭圆曲线的方程的两个元素，上的基点，其中和是中的两个元素，G的阶为n。

* ：密钥生成算法输入安全参数，SM2 椭圆曲线系统参数，运行 SM2 签名的密钥生成算法，输出验证公钥和签名私钥 运行 SM2 加密的密钥生成算法，输出加密公钥和解密私钥 选择随机数 ，计算承诺方案参数 Q = v · G;然后输出算法公开参数：承诺方案参数 、签名验证公钥 和加密公钥 ；秘密参数：签名私钥 和解密私钥 ；
* ：承诺算法输入承诺方案参数 Q 和消息 m；选择随机数，计算承诺值；然后输出：承诺值 c 和承诺打开参数；
* ：签密算法输入：加密公钥 pk、签名私钥 sk、承诺值 c 和承诺打开参数 d；运行 SM2 签名算法和 SM2 加密算法，计算 和；输出签密值 sc=；
* ：验证算法输入：验证公钥 vk、承诺值 c 和签名值 ；运行 SM2 签名算法，计算 ，输出1 表示验证正确，0 表示验证错误；
* ：解密算法输入：解密私钥dk和密文ct，运行 SM2 解密算法，计算；输出：承诺打开参数d；
* ：打开算法输入：承诺打开参数 ，承诺值c，以及承诺方案参数 ；判断 c = m · G + k · Q 是否成立，成立则输出消息 m，否则输出；

**正确性：**对于任意消息 m 的承诺，进行签密获得 和，使用 SM2 的签名验证算法可正确验证承诺值，，通过 SM2 解密算法解密出打开参数，可正确打开消息值

**安全性分析：**基于 SM2 的签密方案是对签密通用构造框架的实例化。根据 An J.H. 等的分析，采用安全的底层组件构造，可保证整体的签密方案的安全性。本文构造的 SM2 签密基于具有信息论安全的隐藏性和计算安全的绑定性的承诺方案、安全的 SM2 标准签名算法和 SM2 标准加密算法作为模块化构造组件。因此可实现整体 SM2 签密方案的安全性：SM2 签密的机密性由承诺方案的信息论安全的隐藏性和 SM2 加密算法的机密性保证；SM2 签密的完整性、 认证和不可否认性由承诺方案的计算安全的绑定性和 SM2 签名算法的完整性、认证和不可否认性保证。

**消息同态运算：**本方案中，我们采用了椭圆曲线上的 Pedersen 承诺方案实例化底层承诺模块，使得SM2 签密方案具有消息同态运算功能。

1)消息的承诺值 ；

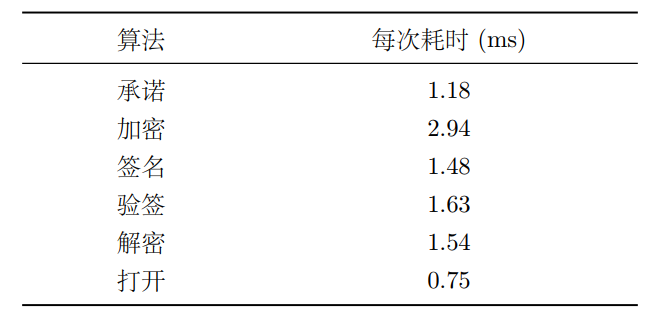
2)消息的承诺值 ；

**签密消息的加法同态运算：**该过程输入，计算 ，输出同态运算结果c。

签密的消息同态性质只能实现承诺消息的同态运算功能，不能实现加密的打开参数的同态运算。但是对于打开参数的同态运算的功能缺少不会影响消息同态运算的功能，只是在签密方案通信效率上有所影响，即：对于多个消息同态运算时，需要同时传输多个承诺打开参数的密文。

**可行性分析：**我们使用 C++ 语言实现了SM2 签密方案的承诺、加密、签名、解密、验证、以及打开算法，并循环测试了1000次，获得各个操作的平均效率均是毫秒级，分别是承诺：1.18ms/次；加密：2.94ms/次；签名：1.48ms/次；验签：1.63ms/次；解密：1.54ms/次；承诺打开：0.75ms/次。

表2. SM2 签密方案各算法的效率



1. 论文进度安排和预期目标

2020.9——2021.11 跟踪调研安全多方计算的研究现状，以及安全多方计算技术在门限密码领域的应用情况。探索安全多方计算和门限密码存在的问题，积极思考并给出初步的解决方案。

2021.12——2022.08 深入学习现有安全多方计算协议的各类安全模型和存在性情况，以及各类通用安全多方计算协议的构造技术。熟练掌握安全多方计算协议的构造方法，以及相关的优化技术，并结合安全多方计算在门限密码中的应用，在安全模型、功能扩展，以及效率优化等方面给出更加安全实用的解决方案，发表论文。

2022.09——2023.01 跟进现有通用安全多方计算协议的效率优化技术，并积极探索新型的优化技术，在现有协议基础上给出更加高效的通用安全多方计算协议。聚焦安全多方计算的应用场景，构造更贴合具体应用的门限密码方案，发表论文。

2023.02——2023.12 积极探索新型的通用安全多方计算构造方法，同时研究不同应用场景下的敌手攻击形式，结合实际应用，探索更强的安全性模型，并在该模型下给出具体的安全多方计算协议的构造，发表论文。

2024.01——2024.06 撰写毕业论文，通过毕业答辩。

1. 论文预期创新点

（1）基于ECDSA的两方签名算法避免使用复杂的安全多方计算技术和昂贵的零知识证明技术，相较于现有基于ECDSA的两方签名更加高效。本文实现的两方签名算法在签名过程中，各参与方的签名操作和ECDSA的签名操作类似，具体实现方案对于ECDSA签名变动较小，可重用较多的原始ECDSA算法的软硬件实现，对于存在的ECDSA应用进行两方升级扩展比较友好。

（2）本文首次提出了支持交互式密钥生成的适配器签名的形式化定义，可为交互式密钥生成的适配器签名的安全性分析提供研究基础。本文实现的ECDSA适配器签名算法与现有ECDSA适配器签名相比，预签名阶段的零知识证明可以离线执行，保证在线预签名阶段更加高效。此外，我们对ECDSA适配器签名进行分布式设计，构造了高效的两方ECDSA适配器签名方案，并给出了可证明安全性分析。

（3）本文首次基于国密 SM2 签名构造了适配器签名方案, 并基于SM2 的安全性给出其在随机预言机模型下的安全性证明. 随后, 根据 SM2 签名的结构特点, 采用离线证明技术进行优化, 获得了可支持离线证明的SM2适配器签名方案，并基于适配器签名的现有应用, 分别给出 SM2 适配器签名在区块链上的两个具体应用：原子交换协议和支付通道网络, 为后续 SM2 签名算法的应用推广提供参考。

（4）本文通过采用成熟的签密通用构造框架，基于SM2 标准算法构造了安全高效的SM2签密方案。该方案通过模块化构造，可实现丰富的功能性扩展，比如：消息的同态运算性质，以及无证书、交互式、批量验签等丰富的应用功能，同时该方案可实现SM2签名和加密算法的并行执行，极大的提升了签密的应用效率（max{cost{签名}， cost{加密}}<cost{签名}+cost{加密}）。

1. 课题研究的条件、经费支持及导师组人员配备情况

山东大学网络空间安全学院（研究院），老师们拥有雄厚的知识基础，优秀的科研攻关能力，实验室配备了先进的计算机设备。项目课题：实验室承担了“973”项目、国家自然科学基金重点项目、国家杰出青年基金项目课题，提供了良好的学习科研环境与财力支持。实验室为科研的顺利进行提供了强有力的保障。

1. 可能的困难与解决途径

（1）研究前期很有可能遇到的困难是对晦涩难懂的学术论文的理解，因为知识储备的不足，思维方式的偏差，对论文的理解很有可能有偏差和误解。解决途径：加强交流，与老师和同学们多沟通交流，共同探讨对同一个问题、同一个方法的理解和想法；

（2）阅读相关文献，通常研究人员会对一个问题有多方面的研究，相似或者互补，而论文中的参考文献正为我们提供了了解相关领域最直接的指导，在费解论文要点时，对相关文献的阅读往往能为我们提供新的视角，从而更全面地掌握相关技术与方法。

（3）后面很有可能遇到的是创新工作的困难，相信厚积薄发，认真钻研前人经验很重要，但也要养成独立思考问题的习惯。具体问题具体分析，方法不太可能天然普适，这就需要我们依据具体算法先做其特性的研究。

1. 参考文献

[1] O. Goldreich. The Foundations of Cryptography，Volume 2，Basic A pplications，Cambridge: Cambridge University Press 2004.

[2] YAO A C. Protocols for secure computation[C]. In: Proceedings of 2013 IEEE 54th Annual Symposium on Foundations of Computer Science. IEEE, 1982: 160–164. [DOI: 10.1109/SFCS.1982.88]

[3] Goldreich, O., S. Micali, and A. Wigderson. 1987. “How to Play any Mental Game”. In:Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC 1987). ACM. 218–229.

[4] Donald Beaver, Silvio Micali, and Phillip Rogaway. The round complexity of secure protocols. In Proceedings of the twenty-second annual ACM symposium on Theory of computing, pages 503–513. ACM, 1990.

[5] Vladimir Kolesnikov and Thomas Schneider. Improved garbled circuit: Free-xor gates and applications. In Automata, Languages and Programming, pages 486–498. Springer, 2008.

[6] Moni Naor, Benny Pinkas, and Reuban Sumner. Privacy preserving auctions and mechanism design. In Proceedings of the 1st ACM conference on Electronic commerce, pages 129–139. ACM, 1999.

[7] Benny Pinkas,Thomas Schneider,Nigel PSmart,and Stephen CWilliams. Secure two-party computation is practical. In Advances in Cryptology–ASIACRYPT 2009, pages 250–267. Springer, 2009.

[8] Vladimir Kolesnikov, Payman Mohassel, and Mike Rosulek. Flexor: Flexible garbling for xor gates that beats free-xor. In Advances in Cryptology–CRYPTO 2014, pages 440–457. Springer, 2014

[9] Samee Zahur, Mike Rosulek, and David Evans. Two halves make a whole. In Advances in Cryptology-EUROCRYPT 2015, pages 220–250. Springer, 2015.

[10] Rosulek M., Roy L. Three Halves Make a Whole? Beating the Half-Gates Lower Bound for Garbled Circuits. In: Malkin T., Peikert C. (eds) Advances in Cryptology – CRYPTO 2021. CRYPTO 2021.

[11] Ben-Or, M., S. Goldwasser, and A. Wigderson. 1988. “Completeness Theorems forNon-Cryptographic Fault-Tolerant Distributed Computation”. STOC 1988. ACM. 1–10.

[12] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612–613.

[13] Beaver, D. 1991. “Efficient Multiparty Protocols Using Circuit Randomization”. In:Proceedings of the 11th Annual International Cryptology Conference (CRYPTO 1991).Vol. 576.

[14]DESMEDT Y. Society and group oriented cryptography: A new concept[C]. In: Advances in CRYPTO ’87. Springer Berlin Heidelberg, 1987: 120–127.

[15] DESMEDT Y, FRANKEL Y. Threshold cryptosystems[C]. In: Advances in Cryptology—CRYPTO ’89. Springer Berlin Heidelberg, 1989: 307–315.

[16] DE SANTIS A, DESMEDT Y, FRANKEL Y, et al. How to share a function securely[C]. In: Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing. ACM, 1994.

[17] LIM C H, LEE P J. Another method for attaining security against adaptively chosen ciphertext attacks[C]. In:CRYPTO ’93. Springer Berlin Heidelberg, 1993: 420–434.

[18] SHOUP V, GENNARO R. Securing threshold cryptosystems against chosen ciphertext attack[J]. Journal of Cryptology, 2002, 15(2): 75–96.

[19] CHAUM D, PEDERSEN T P. Wallet databases with observers[C]. In: CRYPTO ’92.Springer Berlin Heidelberg, 1992: 89–105.

[20] CANETTI R, GOLDWASSER S. An efficient: Threshold public key cryptosystem secure against adaptive chosen ciphertext attack[C]. In EUROCRYPT ’99. Springer Berlin Heidelberg, 1999: 90–106.

[21] CRAMER R, SHOUP V. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack[C]. In CRYPTO ’98. Springer Berlin Heidelberg, 1998: 13–25.

[22] ZHANG R, HANAOKA G, SHIKATA J, et al. On the security of multiple encryption or CCA-security+CCAsecurity=CCA-security[C]? In PKC 2004. Springer Berlin Heidelberg, 2004: 360–374.

[23] DODIS Y, KATZ J. Chosen-ciphertext security of multiple encryption[C]. In TCC 2005. Springer Berlin Heidelberg, 2005: 188–209.

[24] XIE X, XUE R, ZHANG R. Efficient threshold encryption from lossy trapdoor functions[C]. In PQCRYPTO 2011. Springer Berlin Heidelberg, 2011: 163–178.

[25] WEE H. Efficient chosen-ciphertext security via extractable hash proofs[C]. In CRYPTO 2010. Springer Berlin Heidelberg, 2010: 314–332.

[26] LIBERT B, YUNG M. Non-interactive CCA-secure threshold cryptosystems with adaptive security: New framework and constructions[C]. In TCC 2012. Springer Berlin Heidelberg, 2012: 75–93.

[27] Binbin Tu, Yu Chen, Xueli Wang. Threshold Trapdoor Functions and Their Applications.IET Information Security, Vol.14(2), 2020, pp.220-231.

[28] DESMEDT Y, FRANKEL Y. Shared generation of authenticators and signatures (extended abstract)[C]. In CRYPTO ’91. Springer Berlin Heidelberg, 1991: 457–469.

[29] SHOUP V. Practical threshold signatures[C]. In EUROCRYPT 2000. Springer Berlin Heidelberg, 2000: 207–220.

[30] KATZ J, YUNG M. Threshold cryptosystems based on factoring[C]. In ASIACRYPT 2002. Springer Berlin Heidelberg, 2002: 192–205.

[31] BENDLIN R, KREHBIEL S, PEIKERT C. How to share a lattice trapdoor: Threshold protocols for signatures and (H)IBE[C]. In ACNS 2013. Springer Berlin Heidelberg, 2013:218–236.

[32] GENTRY C, PEIKERT C, VAIKUNTANATHAN V. Trapdoors for hard lattices and new cryptographic constructions[C]. In: Proceedings of the 40th Annual ACM Symposium on Theory of Computing. ACM, 2008: 197–206.

[33] BONEH D, GENNARO R, GOLDFEDER S, et al. A lattice-based universal thresholdizer for cryptographic systems[J]. IACR Cryptology ePrint Archive, 2017: 2017/251.

[34] GENNARO R, JARECKI S, KRAWCZYK H, et al. Robust threshold DSS signatures[C]. In EUROCRYPT ’96. Springer Berlin Heidelberg, 1996: 354–371.

[35] GENNARO R, GOLDFEDER S, NARAYANAN A. Threshold-optimal DSA/ECDSA signatures and an application to bitcoin wallet security[C]. In ACNS 2016. Springer Berlin Heidelberg, 2016: 156–174.

[36] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[C]. In EUROCRYPT ’99. Springer Berlin Heidelberg, 1999: 223–238.

[37] GENNARO R, GOLDFEDER S. Fast multiparty threshold ECDSA with fast trustless setup[C]. In CCS 2018. ACM, 2018: 1179–1194.

[38] LINDELL Y, NOF A, RANELLUCCI S. Fast secure multiparty ECDSA with practical distributed key generation and applications to cryptocurrency custody[C]. In CCS 2018. ACM, 2018: 1837–1854.

[39] State Cryptography Administration. SM2 elliptic curve cryptographic algorithm[OL]. 2010. http://www.gmbz.org.cn/main/viewfile/20180108015515787986.html

[40] SHANG M, MA Y, LIN J Q, et al. A threshold scheme for SM2 elliptic curve cryptographic algorithm[J]. Journal of Cryptologic Research, 2014, 1(2): 155–166.

[41] ZHANG Y D, HE D B, ZHANG M W, et al. A provable-secure and practical two-party distributed signing protocol for SM2 signature algorithm[J]. Frontiers of Computer Science, 2020, 14: 143803

[42] TU B B, WANG X F, ZHANG L T. Two distributed applications of SM2 and SM9[J]. Journal of Cryptologic Research, 2020, 7(6): 826–838.

[43] Yehuda Lindell. Secure Multiparty Computation (MPC). IACR Cryptol. ePrint Arch. 2020: 300.

[44]GENNARO R, JARECKI S, KRAWCZYK H, et al. Robust threshold DSS signatures[C]. In: EUROCRYPT’96. 354–371.

[45] CANETTI R, GENNARO R, JARECKI S, et al. Adaptive security for threshold cryptosystems[C]. In: CRYPTO ’99. 98–115.

[46] BONEH D, BOYEN X, HALEVI S. Chosen ciphertext secure public key threshold encryption without random oracles[C]. In CT-RSA 2006. Springer Berlin Heidelberg, 2006: 226–243.

[47] Yehuda Lindell. Fast Secure Two-Party ECDSA Signing. J. Cryptol. 34(4): 44 (2021)

[48] A. Poelstra. Lightning in Scriptless Scripts. mimblewimble team mailing list. https://lists.launchpad.net/mimblewimble/msg00086.html.

[49] Aumayr, L., Ersoy, O., Erwig, A., Faust, S., Hostáková, K., Maffei, M., Moreno-Sanchez, P., Riahi, S.: Generalized bitcoin-compatible channels. IACR Cryptol. ePrint Arch. 2020, 476 (2020)

[50] Esgin, M.F., Ersoy, O., Erkin, Z.: Post-quantum adaptor signatures and payment channel networks. In ESORICS 2020, Lecture Notes in Computer Science, vol. 12309, pp. 378–397.

[51] Castagnos, G., Catalano, D., Laguillaumie, F., Savasta, F., Tucker, I.: Two-party ECDSA from hash proof systems and efficient instantiations. In CRYPTO 2019.

[52] Doerner, J., Kondi, Y., Lee, E., Shelat, A.: Threshold ECDSA from ECDSA assumptions: The multiparty case. In: 2019 IEEE Symposium on Security and Privacy, SP 2019.

山东大学博士学位论文开题报告评议表

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 学 号 | | 202020881 | | 姓名 | 涂彬彬 | | 指导教师 | | 陈宇 | 入学年月 | | 2020。09 |
| 培养单位 | | 网络空间安全学院（研究院） | | | | | 学科专业 | | 网络空间安全 | | | |
| 论文题目 | |  | | | | | 开题时间  开题地点 | |  | | | |
| 评审组对开题报告的评议（对论文选题的合理性、研究思路的可行性及对文献综述、博士生工作能力等方面的评价；存在的主要问题和改进措施）  开题报告评议结果：□优秀 □通过 □建议修改，重新开题 □不通过  评审组长签字： 年 月 日 | | | | | | | | | | | | |
| 评审  组  名  单 | 姓 名 | | 专业技术职务、是 否 博 导 | | | 学科（专业） | | 所在单位 | | | 签 名 | |
|  | |  | | |  | |  | | |  | |
|  | |  | | |  | |  | | |  | |
|  | |  | | |  | |  | | |  | |
|  | |  | | |  | |  | | |  | |
|  | |  | | |  | |  | | |  | |

|  |
| --- |
| 开题报告答辩过程的原始记录（主要内容为答辩专家提出的问题，学生回答情况，包括专家提出的修改意见等。） |