摘要

我们提出了一个新的多方协议,用于分布式生成双素数 RSA 模数,该协议假设有无意识传输并且在面对任意子集的恶意串谋参与者的情况下,具有安全性.我们的协议高度模块化,其最上层可以视为一个概括了先前工作结构并简化了安全证明的模板.我们引入了一种结合采样和筛选的技术,消除了 Fredriksen 等人(Crypto '18)方法中固有的泄露以及 Hazay 等人(JCryp '19)方法中对加法同态加密的依赖.我们将这一技术与一种高效的隐私检查相结合,用以在采样的候选不是双素数时追溯性地检测恶意行为,从而克服了隐蔽的拒绝采样攻击,并在前沿技术状态下实现了无论是渐进还是具体效率的改进.

关键词:阈值密码学,多方计算,RSA,双素数采样,具体效率.

目录

[1.引言 2](#_Toc166293948)

[1.1. 结果和贡献 3](#_Toc166293949)

[1.2 技术概览 4](#_Toc166293950)

[1.3. 其他工作 5](#_Toc166293951)

[1.4组织 5](#_Toc166293952)

[2. 预备知识 6](#_Toc166293953)

[3.假设和理想的功能 7](#_Toc166293954)

[3.1.保理假设 7](#_Toc166293955)

[3.2.分布式双质抽样功能 8](#_Toc166293956)

[4.分布式双质采样协议 11](#_Toc166293957)

[4.1.高级概述 11](#_Toc166293958)

[4.2.在协议中使用的理想功能 12](#_Toc166293959)

[4.3.协议本身 14](#_Toc166293960)

[4.4.安全示意图 15](#_Toc166293961)

[5.分布式双质数测试 16](#_Toc166293962)

[5.1.半诚实的环境 16](#_Toc166293963)

[5.2.恶意设置 17](#_Toc166293964)

[6.效率分析 19](#_Toc166293965)

[6.1.每个实例都有成功的概率 19](#_Toc166293966)

[6.2.实例化FBiprime和FAugMul的成本 20](#_Toc166293967)

[6.3综合上述内容 21](#_Toc166293968)

[6.4.严格恒定和预期恒定的回合 22](#_Toc166293969)

[6.5.与前期工作的比较 22](#_Toc166293970)

# 1.引言

双素数是形式为 N = p · q 的数,其中 p 和 q 都是素数.这样的数作为 RSA 加密系统中公钥的一部分使用(即模数),而因子分解则是秘钥的一部分.长期以来,研究人员已经研究了高效采样双素数的方法;在早期,这项任务需要专门的硬件并且通常不被认为是实际可行的.随着时间的推移,计算能力的提升使 RSA 进入了实用性和普遍性的领域.给定一个安全参数 k,采样 RSA 双素数的事实上的标准方法涉及选择随机的 k 位数并将它们提交给 MillerRabin 素性测试,直到找到两个素数;这些素数然后相乘形成一个 2k 位的模数.当单一方希望生成一个模数,并且被允许知道相关的因子分解时,这种方法是足够的.

Boneh和Franklin启动了分布式RSA模数生成的研究.这一问题涉及一组希望共同采样一个双素数的参与者,其方式是确保没有任何腐败和串谋的子集可以学习到这个双素数的因子.他们强调,将通用的多方计算(MPC)技术应用于标准采样算法会带来实际操作上的困难,特别是在执行MillerRabin素性测试时,需要对很多模数进行模幂运算.因此,Boneh和Franklin构建了一个避免使用秘密模数的模幂运算的新双素性测试协议.这一测试全部在公开的双素数N下进行,允许由各方执行幂运算.此外,他们为整体采样协议引入了一个三阶段结构:

1. 素数候选筛选p和q的候选值联合秘密采样,并通过使用简单但有效的试除法筛选,去除带有小因子的候选.

2. 模数重建:安全地计算并公开N = p·q.

3. 双素性测试:使用分布式协议检验N是否为双素数.如果不是,则重复此过程.

在其半诚实n方设置的核心工作中,Boneh和Franklin的方案采用了ElGamal和Paillier加密,并利用了零知识证明在密文上验证各种关系,从而提高了安全性,但这增加了具体的成本,因为使用了许多定制的零知识证明.

文档还提到,尽管先前的工作通常会考虑RSA密钥生成并包括生成e和d的步骤,这项工作仅关注采样RSA模数N的任务.在采样N之后,可以应用先前技术来采样e(公钥指数)和d(私钥指数),并且分布式生成的RSA模数有其独立的应用,例如为可验证延迟函数生成可信设置.

弗雷德里克森等人的后续协议主要依赖于无意识传输(OT),用于执行筛选和利用吉尔博亚经典乘法协议进行模数重建.他们通过使用"诚实证明"这一民间技术来实现恶意安全性,该技术在协议的最后阶段进行评估,并通过实施他们的协议证明了其实用性.然而,如何将他们的方法扩展到超过两方的情况还不清楚.此外,他们的筛选方法允许选择性失败攻击,他们通过在功能中加入一些泄漏来解决这一问题,这种泄漏也可能允许恶意对手有选择地、秘密地引发假阴性(即,在筛选阶段后强制排除真正的双素数),这一属性在他们的功能建模中再次体现.

这些属性在安全性和效率方面共同作用,导致安全性下降,因为对手可以基于额外泄露的信息拒绝样本,而排除恶意的假阴性则需要运行足够多的实例来使所有实例的统计失败概率微不足道.

因此,根据当前技术水平,尚不清楚是否可以在不泄露额外信息的情况下,在两方之间(其中一方可能是恶意的)采样RSA模数,或者在多方之间(除一方外均可能是恶意的)采样RSA模数,而不涉及重加密等重量级的密码学原语及其相关的性能损失.在本工作中,我们提出了一个高效实现这两项任务的协议.

## 1.1. 结果和贡献

清晰的功能性:我们定义了一个简单、自然的功能 F\_RSAgen,用于从先前工作中使用的众所周知的分布采样双素数,不包括采样失败时的敌对行为.

带有自然假设的模块化协议:我们提出了一个协议 TRSAgen 在 (F\_AugMul, F\_Biprime) 混合模型中,其中 F\_AugMul 是一个增强的乘法功能,F\_Biprime 是一个双素性测试功能,并证明它在恶意设置中实现了 UC功能性,假设因子分解的困难性.具体而言,我们证明了:

定理 1.1(主安全定理,非正式):在 PPT 恶意对手可能腐败任意子集的情况下,可以在 (F\_AugMul, F\_Biprime) 混合模型中安全计算 F\_RSAgen,假设因子分解的困难性.此外,因为我们的安全证明仅在对手作弊时依赖因子分解的困难性,我们惊讶地发现我们的协议在对抗半诚实对手时实现了完美安全性.

定理 1.2(半诚实安全定理,非正式):在计算能力无限的半诚实对手可能腐败任意子集的情况下,F\_RSAgen 可以在 (F\_AugMul, F\_Biprime) 混合模型中完美安全地计算.

支持功能和协议:我们定义 F\_Biprime,一个用于双素性测试的简单、自然功能,在半诚实设置中实现了 UC功能性,并表明这显著简化了 Frederiksen 等人 [23] 协议的组合.我们认为这种组合显著简化了这两个协议的构成,并因此改进了先前协议.

功能性 F\_AugMul:

我们定义了 F\_AugMul,一种用于采样和相乘秘密共享值的功能性,这些值是基于中国剩余定理特定形式派生的.在 TRSAgen 的背景下,这一功能性允许我们高效地采样特定范围内的数字,没有小因数,然后计算它们的乘积.这个功能可以完全通过使用无意识传输来实现,利用已知的乘法协议衍生品.

渐进效率:

我们对我们组合的协议进行了渐进分析,并发现我们的半诚实协议在带宽效率上比 Frederiksen 等人的协议高出 k / log k 倍,其中 k 是素数 p 和 q 的位长.我们的恶意协议在参与者遵守协议时比 Frederiksen 等人的高出 k / s 倍,s 是一个统计安全参数,当参与者偏离协议时效率更高.Frederiksen 等人的协议严格优于 Hazay 等人的协议,尤其在渐进带宽性能方面.

具体效率:

我们进行了一个闭式的具体通信成本分析,包括一些优化,如使用随机预言机.对比 Frederiksen 等人的协议,我们的协议在存在最坏情况下的恶意对手时性能提升约五倍,而在半诚实设置中提升约八十倍.随着采样的双素数位长的增加,我们协议的具体性能优势也随之增长.

## 1.2 技术概览

构建性采样与高效模数重建:

大多数先前的工作使用拒绝采样来生成一对候选素数,然后乘以 \(10^{20}\) 以确保它们有一定的间隔.他们均匀采样,然后反复运行试除法协议,如果试除法失败,就舍弃值和所做的测试工作.这意味着大约三分之二的工作被浪费,因为在筛选后候选数量扩大了两倍.我们提出一种不同的方法,即通过构建性采样来生成一对候选素数并高效地将它们相乘.这种筛选方法在精神上最初由Malkin等人以一种不同的设置形式化.

中国剩余定理(CRT):

CRT表明了一组值之间的同构关系,每个值都在一个由不同素数定义的字段中,以及一个环中相对于这些素数的乘积的单个值.我们将这组值称为一个值的CRT形式或CRT表示,它们彼此同构.我们基于这种同构制定了一种采样机制:对于每一个前 \(O(k / \log k)\) 个奇素数,参与者共同(且高效地)采样这些素数的份额,这些份额的和是一个保证不被集合中任何素数整除的非零 \(k\)位值.出于技术原因,我们同时采样两个这样的候选.

乘法与CRT形式的转换:

我们采用的方法是,不将候选素数从CRT形式转换为标准形式后再相乘,而是直接在CRT形式下对其逐元素进行乘法,然后将乘积转换为标准形式以完成协议.这有效地将单个"全宽度"乘法,其大小为 k,替换为 O(k / log k) 个各自大小为 O(log k) 的单独乘法.我们打算通过基于OT的协议执行乘法,这种协议的计算和通信复杂性至少与其输入的平方成正比.

复杂性与效率改进:

在半诚实情形中,我们的方法产生的总体复杂性为 O(k log k),与执行单个全宽度乘法的 O(k^2) 相比,提供了显著改进.在恶意情形中,结合已知最优乘法器,我们的方法产生的复杂性也随输入长度和一个统计参数的乘积增长,实现 O(k log k + k s) 的总体复杂性.这在具体实现中也有所体现,显示了我们的方法在统计上与其他方法相比如何提供显著改进.

安全性与效率的权衡:

我们的协议允许在素数采样成功时立即终止,而在采样失败时继续运行,这种直观的安全概念在我们的安全分析中起着关键作用.与以往的OT乘法器相比,我们的方法由于大多数候选项在双素性测试中失败,而避免了执行昂贵的隐私检查,从而节省了大量资源.这种"无隐私"的恶意安全策略(在验证诚实行为时牺牲隐私)在我们的案例中实现了显著的效率收益.

双素性测试作为黑盒:

我们定义了一个双素性测试功能,它可以通过 Boneh–Franklin 双素数测试的恶意安全版本实现,并且具有清晰的接口,例如不需要其输入在操作阶段确保它们是不变的.这种模块化的关键在于,如果对手试图通过提供错误的输入来欺骗双素性测试,这可以归结为因子分解问题:如果对手能够欺骗,我们展示如何利用这一点来分解双素数.我们确保只在 N 确实是双素数时依赖这种归约,从而防止对手影响候选分布.

模块化的好处:

我们声称模块化带来了简化协议描述和安全证明的优点.我们相信这种方法将引导更强安全属性的衍生工作.例如,我们证明了协议的半诚实版本只在它省略了反向一致性检查时有所不同,即完美安全.在恶意设置中,采用带安全保障的 F\_Biprime 和 F\_AugMul 可实现对适应性对手安全的 RSA 模数采样协议.

安全性调整和辨识中止:

主协议需要微小调整就能实现带辨识中止的安全性,假设底层功能 F\_AugMul 和 F\_Biprime 是一致性实例化的.这可以通过增强 F\_Biprime 来完成,允许全局请求(例如,当 N 不是双素数时),并在可识别中止时保证所有参与者都能看到此行为.通过这种方式,我们可以确保即使在多达 n1 个腐败者存在的情况下也能保证输出交付,从而通过标准技术来拒绝每个腐败方的一个双素数.

民间技术的应用:

民间技术涉及迭代地调用协议,每次迭代都会排除一个腐败方,直到成功为止.对于恒定比例的腐败,所隐含的线性回合复杂性开销可以被降低到超常数(例如,log n).

## 1.3. 其他工作

Frankel, MacKenzie, 和 Yung 调整了 Boneh 和 Franklin 的协议,以在诚实多数的设置中对抗恶意对手.他们的主要贡献是引入了一种强健的整数分布式乘法方法.Cocks 提出了一种在启发式假设下进行多方 RSA 密钥生成的方法,但后来 Coppersmith 和 Joye 与 Pinch 指出这种方法可能不安全.Poupard 和 Stern 提出了一个基于无意识传输的恶意安全两方协议.Gilboa 改进了这种方法,使之在半诚实两方模型中更为高效,并引入了一个新的多乘法方法.

Malkin, Wu, 和 Boneh 实施了 Boneh 和 Franklin 的协议,并通过引入优化的筛选机制来改进它,使其在整数字段中生成随机值的份额.这种协议通过一个半诚实的多数协议生成加法份额以进行双素性测试.Algesheimer, Camenisch, 和 Shoup 描述了一个依赖于密文之间的秘密共享转换技术来执行分布式 Miller–Rabin 测试的方法.尽管每次调用其 MillerRabin 测试的复杂性很高,但他们的整体协议在通信复杂性上也显著.

Damgård 和 Mikkelsen 对这个方法进行了改进,显著降低了通信和回合复杂性,并在诚实多数设置中实现了恶意安全性.Chen 等人在后续工作中利用基于 CRT 的双素数采样技术,结合了加法同态加密和零知识证明的最新进展,关注在一个强大的半诚实聚合器和许多弱小恶意客户的环境中的优化机会.

## 1.4组织

论文结构:

1.第二节:提供基本符号和背景信息.

2.第三节:定义理想的双素数采样功能.

3.第四节:展示实现第三节功能的协议.

4.第五节:介绍双素性测试协议.

5.第六节:进行效率分析.

6.附录:包含安全性的完整证明和乘法协议的详细信息.

表格1 先前工作的比较:

列出了各种不同协议的基本信息,包括参与方数、容忍的腐败数、安全性模型(如半诚实或恶意)、使用的通信渠道(私有、认证、广播)和所依赖的假设(如离散对数、无意识传输、强RSA、决策性复合剩余性、决策性DiffieHellman).

该表格对各个研究工作的核心参数和安全假设进行了归类和比较,有助于理解不同方法在安全性和实施复杂性上的权衡.

# 2. 预备知识

符号定义:

我们使用 "=" 表示等式,":=" 用于赋值,"<" 表示从分布中抽样,"≡" 用于同余,"≈c" 表示计算上不可区分,而 "≈s" 用于统计不可区分性.

通常,单个字母的变量用斜体表示,多字母的函数名和变量名使用无衬线字体,字符串字面量使用等宽字体.

我们使用 "mod" 表示模运算符,而 "(mod m)" 出现在一行的末尾表示该行的所有等价关系都应该在整数模 m 上考虑.

在讨论 RSA 双素数时,我们用 k 表示每个素数的位长度.我们使用 s 表示统计参数.

在具体效率方面,我们引入了第二个计算安全参数 λ,它表示与长度为 2k 3 的双素数等效强度的对称密钥的长度.k 和 λ 必须变化,NIST 已经给出了它们之间关系的建议.

向量和数组:

向量和数组用粗体表示,并通过下标索引;例如,x\_i 是向量 x 的第 i 个元素,与标量 x 不同.

当我们希望从二维数组中选择一行或一列时,我们在不选择的维度上放置一个星号;因此,y\_,j 是矩阵 y 的第 j 列,而 y\_j, 是第 j 行.

当只有两个参与方在场时,我们将它们称为 Alice 和 Bob.

变量通常可能带有一个索引以表明它们属于特定的参与方.

当数组被一个特定的参与方所拥有时,该方的索引总是放在前面.

我们使用 |y| 来表示向量 y 中的元素数量.

通用可组合性:

文章证明了协议在通用可组合性(UC)框架下的安全性,并使用了标准的UC符号.

为了方便处理功能中的隐式元素,例如假设功能在会话标识符被不当重用时中止,提供了一个函数GenSID,该函数能从任意数量的参数中确定性地派生出一个独特的会话ID.

中国剩余定理(CRT):

CRT定义了一组模数互质的整数值集与它们的乘积的模数之间的同构关系.这为采样过程提供了基础.

定理2.1说明了如何从一组模两两互质的正整数中,根据它们的模产生一个唯一的整数值y,使得对于每个j,y在模m\_j下的值等于x\_j.

为了完整性,提供了CRTRecon算法,用于根据m和x计算唯一的y值.

CRTRecon算法:

1. 计算 M = ∏ m\_j,其中 |m| = ℓ 是向量 m 的长度.

2. 对每个 j,计算 a\_j = M / m\_j 并找到 b\_j,使得 a\_j b\_j ≡ 1 (mod m\_j),这可以通过扩展的欧几里得算法实现.

3. 输出 y = Σ (a\_j b\_j x\_j) mod M,这是将 y 的 CRT 表示转换回其标准形式的结果.

# 3.假设和理想的功能

标准的因子分解实验(实验3.1),该实验由对手 \(A\) 和一个生成双素数(biprime)的算法 GenModulus 参数化.这里提到的对手和算法是基于 Katz 和 Lindell 的形式化定义的.在实验中,输入一个1后跟 \(k\) 个0的二进制串(记为 \(1^k\)),GenModulus 算法输出三个值 \(N\)、\(p\) 和 \(q\),其中 \(N = p \cdot q\),且 \(p\) 和 \(q\) 都是 \(k\) 位的质数.

## 3.1.保理假设

实验3.1的具体步骤包括:

1. 运行 GenModulus(1\(^k\)),得到 \(N\)、\(p\)、\(q\).

2. 将 \(N\) 发送给对手 \(A\),对手 \(A\) 返回 \(p'\) 和 \(q'\),这两个值的乘积应该大于1.

3. 如果 \(p' \cdot q' = N\),则输出1;否则输出0或不输出.

这个实验的目的是验证 \(p'\) 和 \(q'\) 是否确实是 \(N\) 的因子,从而测试 GenModulus 算法的有效性以及对手 \(A\) 的破解能力.文中还提到,Katz 和 Lindell 规定了在忽略概率极小的抽样失败的情况下,要求 GenModulus 必须在严格的多项式时间内运行.

在密码学应用中使用 GenModulus 函数生成 \(p\) 和 \(q\) 的方法.该函数旨在从范围 \([2^{k1}, 2^k)\) 中均匀地选择质数 \(p\) 和 \(q\).然而,文献指出,按照这种均匀双素数分布有效地进行抽样是困难的,尤其是在多方场景中.因此,大多数先前的工作选择按照不同的分布进行抽样,并因此使用的模数需要略有不同的因子分解假设.

文中特别提到了 Boneh 和 Franklin 提出的分布,该分布由抽样算法 BFGM 定义,适用于多方抽样.BFGM 算法考虑多方数量 \(n\) 为额外参数,它抽样 \(n\) 个整数份额,每个在范围 \([0, 2^{k\log n}]\) 内,然后将这些份额求和以生成候选质数.这种方法并不产生 \(k\)位质数的均匀分布.此外,BFGM 只抽样那些 \(p \equiv q \equiv 3 \pmod{4}\) 的质数 \(p\) 和 \(q\),以便于有效地进行素性测试,并过滤出那些有效的模数 \(N = p \cdot q\),满足 \(p \equiv 1 \pmod{q}\) 或 \(q \equiv 1 \pmod{p}\) 的情况.

此外,文中指出 Boneh 和 Franklin 的方法实际上提出了两种变体,选择了其中一种因为它提供了更有效的抽样协议,且没有假阴性结果.文献还提到了对 \(p\) 和 \(q\) 的下界的解释不一,但大多数先前的工作采用的是 \(0\) 的下界,作者认为这种差别并不重要.

算法3.2(BFGM)是为了生成两个大质数 \( p \) 和 \( q \) ,这两个质数的乘积 \( N \) 用于密码学中的模数.该算法详细步骤如下:

1. 对于每个 \( i \) 从 1 到 \( n \),随机选择 \( pi \) 和 \( qi \),范围都是 \([0, 2^{k\log n})\).其中,第一个 \( p1 \) 和 \( q1 \) 满足 \( p1 \equiv q1 \equiv 3 \pmod{4} \),而其他 \( pj \) 和 \( qj \) 满足 \( pj \equiv qj \equiv 0 \pmod{4} \).

2. 计算所有 \( pi \) 的和得到 \( p \),计算所有 \( qi \) 的和得到 \( q \),然后计算 \( N = p \cdot q \).

3. 检查 \( N \) 和 \( p + q 1 \) 的最大公因数(gcd)是否为1,同时确认 \( p \) 和 \( q \) 是否都是质数.如果都符合条件,则输出 \( N \) 和 \( \{(pi, qi)\}\_{i \in [n]} \);否则,从步骤1重复整个过程.

这个算法的安全性假定基于对手 \( A \) 解决 Factor\_A, BFGM(\( k, n \)) 问题的概率非常小,即小于一个可忽略的函数 negl(\( k \)).

此算法特别适合于需要分布式生成的场景,例如在多方安全计算中,各方可以共同参与 \( p \) 和 \( q \) 的生成,而无需单独一方掌握所有信息.这样不仅增强了安全性,还通过多方验证确保了 \( p \) 和 \( q \) 的质量.

## 3.2.分布式双质抽样功能

一种新的抽样算法,名为 CRTSample,它是作为 BFGM 算法的替代提出的.BFGM 算法的运行时间仅为期望的多项式时间,而 CRTSample 算法在成功时输出的分布在统计上与 BFGM 相同,但在严格的多项式时间内运行.该算法专门定义了失败的分布,这与协议设计和故障模式有关.

这里还引入了一些定义:

Primorial Number(原始数):第 \(i\) 个原始数定义为前 \(i\) 个质数的乘积.

((k, n)NearPrimorial Vector(接近原始向量):定义一个向量 \(m\),它的长度 \( \ell \) 是使得 \( \ell \) 个原始数小于 \( 2^{k\log n1} \) 的最大数,向量 \( m \) 的元素 \( m\_1, m\_2, ..., m\_{\ell} \) 是第 \( \ell \) 个原始数的奇数因子,按升序排列.

mCoprime(m互素):一个整数 \( x \) 是 \( m \)互素的,当且仅当它不被向量 \( m \) 中的任何元素整除.

这些定义为 CRTSample 算法提供了理论基础,使得算法能够有效地分配 \( p \) 和 \( q \) 的份额,并保证它们的生成是符合要求的.通过这种方式,CRTSample 算法能够在一定程度上控制失败的概率,并提高整个加密系统的安全性和效率.

算法3.6(CRTSample)的流程如下,这是一个为生成两个大质数 \(p\) 和 \(q\),其乘积用作密码学中的模数的过程:

1. 初始化:

\(m\) 是 (\(k\), \(n\))近原始向量,长度为 \(\ell\).

\(M\) 是 \(m\) 的乘积.

2. 抽样:

对于 \(i\) 在 \([n] \setminus P^\)(即除了一些特殊处理的索引以外的所有索引)中,随机选择 \(p\_i\) 和 \(q\_i\),范围是 \([0, M)\).

对于 \(i = 1\),\(p\_1\) 和 \(q\_1\) 被设为满足 \(3 \mod 4\).

对于 \(i > 1\),\(p\_i\) 和 \(q\_i\) 被设为满足 \(0 \mod 4\).

确保 \(p\) 和 \(q\) 为 \(m\)互素,即它们不被 \(m\) 中的任何元素整除.

3. 计算和验证:

计算 \(p = \sum\_{i \in [n]} p\_i\) 和 \(q = \sum\_{i \in [n]} q\_i\).

检查 \(gcd(p \cdot q, p + q 1) = 1\),确保 \(p\) 和 \(q\) 都是质数,且 \(p \equiv q \equiv 3 \mod 4\).

4. 输出:

如果上述条件都满足,则输出"成功",以及 \(p\) 和 \(q\).

如果不满足,则输出"失败",以及 \(p\) 和 \(q\).

此算法特别关注在多方计算环境中,如何有效且安全地分布式生成这两个质数,以便所有参与者都不需要完全信任其他任何一方.通过 \(m\)互素和特定的模余条件,算法旨在减少生成非质数或不满足安全要求的 \(p\) 和 \(q\) 的概率.

这一段文献中提出了一个关于多方功能性的理论基础,特别是围绕一个理想的多方功能 \(F\_{SAGen}\),它是算法 CRTSample 的自然嵌入.下面是关键内容的概括:

Lemma 3.7

情境: 当算法 \(A\_2\) 成功输出 \(N = p \cdot q\) 的因子 \(p\) 和 \(q\) 时,\(A\_1\) 接收到状态和 \(N\),并假设 \(N\) 是由 CRTSample 生成的双素数.

结论: 如果 \(A\_2\) 在给定的约束下(即 \(p\) 和 \(q\) 是由 CRTSample 生成的)能够输出 \(N\) 的因子,那么存在一个预期多项式时间算法 \(M\),该算法在实验 \(Factor\_{B,BFGM}(k, n)\) 中至少以 \(1/2^{n^3k^d}\) 的概率成功,减去一个可忽略的函数 negl(\(k\)).

Multipart Functionality (\(F\_{SAGen}\))

功能定义: \(F\_{SAGen}\) 接收来自参与者的输入 \((p\_i, q\_i)\),并在被调用时执行一次 CRTSample 算法.它输出有效的模数 \(N = p \cdot q\),或者指示采样失败(可能是因为未能生成质数或者算法被敌对方中断).

实现细节: \(F\_{SAGen}\) 使得在其使用中可以灵活地调节资源使用和执行时间之间的权衡.这种功能不仅简化了成功率的分析,还因为采样过程相同,简化了实现 \(F\_{SAGen}\) 的协议 \(T\_{SAGen}\) 的分析.

安全性考: 如果 \(F\_{SAGen}\) 未能向诚实方提供 \(N\),可能是因为 CRTSample 未能采样出质数或敌对方干扰了计算导致必须中止.在这种情况下,敌对方无法区分这两种情况,这基本上是将终止的标准安全概念应用于多方计算的结果.

这一理论表明,使用 \(F\_{SAGen}\) 可以在保持较高安全性的同时,实现算法的高效执行,适用于需要高安全性和可调节性能需求的环境.

功能性 3.8 \(F\_{RSAGen}(k, n)\) 描述了一个分布式双素数采样过程,用于生成具有素数长度 \(k\) 的 RSA 模数.该功能性允许 \(n\) 个参与方参与,能够与可能破坏特定参与方的理想对手 \(S\) 直接交互.以下是这个功能性的核心流程和细节:

输入与采样流程

最大数 \(M\) 的确定:\(M\) 是小于 \(2^{k\log n}\) 的最大的原始数的两倍.

采样请求:每个诚实的参与方 \(P\_i\),对于 \(i \in [n] \setminus P^\),从对手 \(S\) 接收采样请求,每个 \(P\_i\) 会在 \([0, M)\) 范围内为 \(p\_i\) 和 \(q\_i\) 采样.

执行 CRTSample 算法:收到所有 \(p\_i\) 和 \(q\_i\) 后,执行 CRTSample 算法,接收结果为成功或失败,连同 \(p\) 和 \(q\).

结果处理

对模 \(4\) 的额外检查:

如果 \(p \not\equiv 3 \mod 4\) 或 \(q \not\equiv 3 \mod 4\),则终止并通知所有参与方失败信息.

如果 \(p \equiv q \equiv 3 \mod 4\) 但采样失败,保存失败结果,并通知所有参与方.

如果 \(p \equiv q \equiv 3 \mod 4\) 并且采样成功,计算 \(N = p \cdot q\) 并保存结果.

输出

成功或继续进行的情况:如果收到 \(proceed\) 指令,根据内存中保存的信息,将结果作为延迟输出发送给所有参与方.

欺诈检测:如果收到 \(cheat\) 指令,终止操作,通知所有参与方中止,并将因子分解结果直接发送给 \(S\).

安全性与实用性

这种功能性设计允许在多方设置中安全地生成 RSA 模数,即使在某些参与方可能受到对手干扰的情况下.通过要求 \(p\) 和 \(q\) 满足特定的模 4 条件,加强了对生成的模数的控制,使其更适合用于某些特定类型的密码协议.同时,这种设计通过集中处理欺诈和失败情况,保证了系统的整体可靠性和安全性.

# 4.分布式双质采样协议

这段文字提供了一个关于分布式双素数采样协议 \(T\_{RSAGen}\) 的简介,该协议实现了理想功能 \(F\_{RSAGen}\).下面是该部分内容的概要:

1. 高层概述:开始时提供了 \(T\_{RSAGen}\) 协议的高层概述,解释其目标和基本思路.

2. 理想功能定义(Sect. 4.2):然后,详细定义了协议依赖的两个理想功能.这些功能性是协议设计的基础,提供了操作的理论模型和安全基准.

3. 协议本身(Sect. 4.3):接下来,介绍具体的协议步骤和操作.这部分是实际实施的核心,详细说明了各参与方如何互动,以及如何集体生成 RSA 模数.

4. 安全性证明(Sect. 4.4):最后,提供了关于半诚实和恶意安全的证明草图.这部分展示了协议在不同威胁模型下的安全性,证明了它能抵御各种潜在攻击.

整体来看,这段介绍突出了 \(T\_{RSAGen}\) 协议的系统结构和其在保障密钥生成过程中的安全性方面的重要性.通过这样的结构化介绍,读者可以更清楚地理解协议的设计理念及其在实际应用中的效用和安全保障.

## 4.1.高级概述

分布式密钥生成协议的三个主要阶段,这种协议主要用于生成不含小因子的两个大质数 \(p\) 和 \(q\),并通过它们的乘积 \(N\) 学习其结果.该协议特别使用了中国剩余定理(CRT)来整合生成过程.下面是协议的三个主要阶段的详细说明:

1. 候选筛选(Candidate Sieving)

在第一阶段,参与方共同采样两个共享向量 \(p\) 和 \(q\),这些向量对应候选的质数 \(p\) 和 \(q\).

每个参与方 \(P\_i\) 采样 \(p\_i\) 和 \(q\_i\) 的份额,使得它们满足一定的同余条件以确保 \(p\) 和 \(q\) 符合质数的要求.

通过运用中国剩余定理重建方法(CRTRecon),每个参与方可以重建出整个 \(p\) 和 \(q\) 的值,如果每个 \(x\_{i,j}\) 的总和不为零(模 \(m\_j\)),那么 \(x\) 不会被 \(m\) 中的任何质数整除.

2. 双素性测试(Biprimality Test)

在这个阶段,参与方共同执行一个双素性测试来验证 \(N = p \cdot q\) 是否是双素数.

如果 \(N\) 通过了测试,那么它被认为是有效的RSA模数;如果没有通过,就表示至少一个数不是质数.

3. 一致性检查(Consistency Check)

这一阶段是为了确保在恶意模型下的安全性,参与方需要验证其他参与方在之前阶段的行为是否诚实.

如果双素性测试失败,参与方将检查共享数据以确认数据一致性;如果双素性测试成功,则进一步确保 \(p\) 和 \(q\) 的扩展是诚实进行的.

对于半诚实安全模型,此阶段可以省略,协议可以在双素性测试后结束.

这种协议设计考虑了加密系统在面临不同威胁模型时的安全性需求,通过多个阶段的验证确保了生成的密钥的安全性和一致性.

## 4.2.在协议中使用的理想功能

该文档描述的"增强多方乘法器"(Augmented Multiparty Multiplier, \(F\_{AugMul}\))是一个反应式功能,它在多个阶段运作并存储内部状态,允许在操作中存储和正确重用一组份额.它包括五个主要接口,使其适用于恶意版本的协议,而前三个接口适用于半诚实版本.以下是各接口的功能:

主要接口功能

1. sample接口:

允许参与方采样非零乘积的三元组份额,保证这些份额不为零模某个小质数.

在\(T\_{RSAGen}\)的上下文中,此接口用于采样 \(p\) 和 \(q\) 的CRT份额.

2. multiply和input接口:

使得参与方能够加载份额(相对于某个小质数模数),并在稍后执行模数乘法.

在\(T\_{RSAGen}\)的上下文中,这些接口用于对CRT份额的 \(p\) 和 \(q\) 进行长度扩展.

3. check接口:

允许参与方安全地计算存储值上的谓词,以验证操作的正确性.

用于检查 \(p\) 和 \(q\) 的CRT份额扩展是否正确.

4. open接口:

在双素性测试失败时,允许参与方回溯公开他们的输入.

这是验证采样过程和双素性测试时的透明度和诚实性的关键接口.

5. 透明接口(未详细描述):

供对手使用,允许对手请求泄露存储值并确定对sample或multiply接口调用的输出.

安全调整

文档提及,为了应对恶意模型,功能性记录对手使用透明接口的行为,并在使用check或open接口时通知诚实方终止操作,以防止数据泄露或不当操作.

这些功能性的设计意在通过细粒度控制和丰富的接口选择,提高协议的安全性和灵活性.通过这种方式,系统能够应对各种威胁模型,并在需要时提供必要的数据透明度和验证途径.

增强的多方乘法功能 \(F\_{AugMul}(n)\),该功能支持复杂的多步骤操作和内部状态管理,为密钥生成协议中的多方计算提供安全支持.下面是这一功能性的核心组件和操作流程:

功能概览

1. Cheater Action Recording:

功能性可以记录作弊者的行为,包括存储和发送特定会话ID和相关值的记录,帮助识别和管理潜在的不诚实参与方.

2. Sampling Interface:

在接收到来自每个参与方的采样请求后,功能性根据设定的条件(如模数和会话ID未被标记为作弊)采样新的三元组 \(x\_i, y\_i, z\_i\).

3. Multiplication Interface:

允许参与方提交他们的份额进行乘法操作,功能性在确保所有输入有效的前提下执行乘法,并将结果分配给各方.

4. Predicate Cheater Check:

提供一个接口用于检查基于存储值的谓词,这有助于验证操作的正确性,并在发现作弊行为时通知所有参与方.

5. Input Revelation:

在特定条件下,如会话结束或需要验证数据真实性时,允许参与方打开并公开他们的输入值.

操作细节

在采样和乘法操作中,如果存在关联的作弊记录,功能性将修改其正常操作,向作弊者发送特定信号,并可能调整输出以保护协议的完整性.

在检查谓词时,如果谓词或会话ID与作弊相关,功能性将终止操作,并可能拒绝所有未来的消息.

在输入公开阶段,功能性会检查是否有作弊行为,如有,将终止操作并通知所有参与方.

这些接口和操作机制为处理复杂的安全需求提供了强大的工具,尤其是在需要处理可能的内部作弊或数据泄露风险的场景中.通过这种方式,\(F\_{AugMul}(n)\) 功能性为多方计算提供了一个坚实的安全基础,确保即使在参与方之间存在潜在的不信任时,协议的执行也能保持安全和公正.

功能性 \(F\_{Biprime}(M, n)\) 是一个分布式双素性测试,它抽象化了 Boneh 和 Franklin 提出的双素性测试行为.此功能接收来自各方的候选双素数 \(N\) 以及其因子 \(p\) 和 \(q\) 的份额,检查 \(p\) 和 \(q\) 是否为质数以及 \(N = p \cdot q\) 是否成立.此外,对手 \(S\) 被提供了一个额外的接口,允许请求泄露诚实方的输入,但使用此接口会导致功能性向诚实方报告 \(N\) 不是双素数,即使它是.

双素性测试过程:

1. 收集输入:功能性等待每个参与方 \(P\_i\) 提交 \(N\)、\(p\_i\) 和 \(q\_i\).

2. 计算:对整数进行计算,以便将 \(p\_i\) 和 \(q\_i\) 的总和分别汇总为 \(p\) 和 \(q\),并计算 \(N' = p \cdot q\).

3. 验证条件:

确认所有参与方就 \(N\) 的值达成一致.

验证 \(N = N'\),\(p\) 和 \(q\) 均为质数,且满足 \(p \neq 1 \mod q\) 和 \(q \neq 1 \mod p\).

检查 \(p\) 和 \(q\) 是否在预定的范围内.

4. 输出结果:

如果满足所有条件,发送消息 \((biprime, sid)\) 表示 \(N\) 是有效的双素数.

如果对手 \(S\) 使用作弊接口或任何条件不成立,输出 \((leakedshares, sid)\) 和 \((notbiprime, sid)\).

安全性注意:

这个功能性旨在在密钥生成过程中提供诚实性和透明度,通过允许对手接口来增加额外的安全测试和数据验证步骤.

功能性具备对恶意行为的高度敏感性,可以在对手试图通过不诚实方式影响过程时,保护诚实方免受损害.

通过这种设计,\(F\_{Biprime}(M, n)\) 在确保双素数生成过程中的公正和安全性方面发挥了关键作用,特别是在涉及多方计算和潜在对手存在的情况下.

## 4.3.协议本身

分布式双素数采样协议 Protocol 4.4. TRSAGen(k, n, B),该协议旨在生成具有特定属性的RSA模数 \(N = p \cdot q\),其中 \(p\) 和 \(q\) 是大质数.这个过程依赖于中国剩余定理(CRT)重构和增强的多方乘法器功能(\(F\_{AugMul}\))以及双素性测试功能(\(F\_{Biprime}\)).下面是协议的关键步骤:

协议步骤

1. 候选筛选:

每个参与方 \(P\_i\) 计算三个会话ID向量并发送给 \(F\_{AugMul}(n)\),然后接收三个向量 \(p\_i, q\_i, N\_i\) 的乘积结果.

初始设定 \(p\_1, q\_1 = 3\) 和 \(N\_1 = 9\),其他 \(p\_i, q\_i\) 初始为0.

2. 中国剩余定理重构:

每个参与方使用收到的份额和CRT重构出 \(p, q\) 和 \(N\).

3. 长度扩展和乘法:

对 \(p\_i\) 和 \(q\_i\) 进行模 \(m\) 运算以构建完整的 \(p\) 和 \(q\).

向 \(F\_{AugMul}(n)\) 发送序列化的输入和乘法请求,以计算 \(N\) 的扩展份额.

4. 最终重构:

从所有参与方收集 \(N\_i\),使用CRT重构最终的 \(N\).

5. 局部试除:

每个参与方对 \(N\) 进行试除,检查是否有小于某个界限 \(B\) 的质数可以整除 \(N\).

6. 双素性测试:

所有参与方将 \(N, p, q\) 发送给 \(F\_{Biprime}(M, n)\),等待双素数的确认.

7. 一致性检查:

如果 \(N\) 通过双素性测试,参与方确认 \(N\) 的一致性;如果不通过或存在作弊,进行安全检查并可能终止协议.

安全性和实用性

安全性:协议设计包括对作弊行为的检查,确保即使在参与方不完全诚实的情况下也能正确执行.

效率:使用中国剩余定理和有效的多方计算减少了计算和通信的复杂性,同时确保了高安全性.

这个协议体现了在分布式环境中生成RSA模数的复杂性和对安全机制的需求,特别是在涉及多个不完全可信方的场景中,如何保持高安全性和操作的有效性.

## 4.4.安全示意图

TRSAGen 协议如何实现理想功能 FRSAGen 在半诚实和恶意模型下的详细理论基础.通过混合模型使用 \(F\_{AugMul}\) 和 \(F\_{Biprime}\),协议能够抵御最多 \(n1\) 个恶意参与方的攻击.

主要论点和证明概述:

完美实现:

半诚实模型:模拟器 \(S\) 能够在没有被告知作弊的情况下完美模拟 \(TRSAGen\) 协议的行为.\(S\) 简单地转发环境和对手 \(A\) 之间的消息,使得理想世界中的行为与真实世界无法区分.

恶意模型:提供了一个在附录C中详细证明的全面证明.如果对手遵循协议规范而不作弊,模拟器可以像在半诚实情况下一样操作.如果对手试图作弊,任何偏离协议规范的行为将被诚实的参与方识别并导致操作中止.

安全论证的关键元素:

对手在输入给 \(F\_{AugMul}\) 或 \(F\_{Biprime}\) 时作弊的行为会被检测出来,因为所有的输入和计算都必须符合预设的规则和条件,否则会被系统识别并拒绝.

特别是,\(F\_{Biprime}\) 的设计确保只有当 \(N = p \cdot q\) 是有效的双素数时,它才会被确认.任何试图通过错误的 \(p\)、\(q\) 或 \(N\) 欺骗系统的行为都会在一致性检查阶段被揭示.

理论框架:

理论分析涵盖了对手可能采取的所有潜在策略,包括试图误导系统关于 \(N\) 的双素性状态的尝试.证明展示了在每种情况下系统如何响应,以保护协议的完整性和正确性.

每种情况的处理都基于假设:因子分解 \(N\) 是困难的.只有当 \(N\) 是双素数且被正确重建时,系统才会接受 \(N\);任何其他情况都会导致操作失败或被中止,从而防止了对手通过欺诈手段成功影响系统.

这些理论框架和证明不仅显示了 \(TRSAGen\) 协议的强大安全性,还突出了它在保护密钥生成过程中的可靠性,即使面对几乎全部的参与方都可能是恶意的.

# 5.分布式双质数测试

在本节中,我们将介绍实现FBiprime的协议.

## 5.1.半诚实的环境

文档中描述了 \(F\_{Biprime}\) 的实现细节,特别强调了它是如何在半诚实设置中通过仿照 Boneh 和 Franklin 提出的双素性测试协议实现的.该协议是基于 MillerRabin 测试的变种,用于确定一个数 \(N\) 是否为双素数.

测试过程的关键要点:

1. 测试原理:

对于随机选择的 \(y \in \mathbb{Z}\_N^\)(即 \(y\) 与 \(N\) 互素),使用 Jacobi 符号来测试 \(y^{(Np+1)/4} \equiv \pm 1 (\mod N)\).

该测试主要用于检测 \(N\) 是否为合数,因为对于大多数非双素数 \(N\),该等式通常不成立.

2. 测试执行:

测试重复执行 \(s\) 次以减少错误接受合数的概率,其中 \(s\) 是统计安全参数,错误概率为 \(2^{s}\).

此过程可有效筛除非双素数,但可能对特定形式的非双素数(如 \(p = a^{b\_1}q, q \equiv 1 (\mod a^{b\_1})\))失效.

3. 最终验证:

对可能通过初步测试的候选数 \(N\),执行额外的欧几里得算法测试(GCD测试),来确认 \(r = z \cdot N, z = r \cdot (p+q1)\) 是否成立,以排除最后一类特殊非双素数.

4. 安全性与效率:

协议通过 \(F\_{Biprime}\) 在 \(F\_{ComCompute}\) 混合模型下实现,提供了统计安全保证,有效地对抗静态、半诚实的对手,最多可处理 \(n1\) 个不诚实参与方.

尽管 GCD 测试可能引入一些假阴性(由 BFGM 建模),这通常不会对协议总体效率产生显著影响.

综合分析:

文档提供的协议设计说明了在密钥生成过程中确保双素数 \(N\) 真实性的重要性,同时展示了在现有密码学研究基础上如何有效地整合和优化复杂的数学测试,以提高系统的安全性和实用性.通过结合不同的数学工具和密码学协议,该方法能够有效地满足高安全标准的需求.

## 5.2.恶意设置

这段文档介绍了一种复杂的双素性检验协议,它利用一系列密码学功能性来确保检验的准确性和安全性,尤其是在涉及恶意行为的分布式环境中.该协议的核心是结合使用 \(F\_{ComCompute}\)、\(F\_{Zero}\) 和 \(F\_{CT}\) 功能来执行核心运算和验证.

协议概述:

1. 输入承诺阶段:

每个参与方采样其 \(p\) 和 \(q\) 的份额,并通过 \(F\_{ComCompute}\) 提交承诺,这样可以在后续步骤中验证输入的一致性和完整性.

2. BonehFranklin测试:

执行改良的MillerRabin素性测试,包括对选定的基 \(y\_j\) 进行多次计算,确保 \(N\) 的可能素性.

3. 一致性检查和最大公约数 (GCD) 测试:

在完成初步的双素性测试后,进行一致性检查,以确保所有参与方提交的 \(N\) 的份额能正确重构出 \(N\).

如果在一致性测试后 \(N\) 仍然可能是双素数,则进行 GCD 测试以排除最后的错误情况.

安全性和效率分析:

安全性:通过使用多个功能性进行核心计算和验证,该协议能够防止和识别恶意行为,即使是在参与方中有多数是不诚实的情况下.同时,确保了数据和计算的隐私性不被泄露.

效率:尽管协议包含多个计算步骤和安全检查,它旨在通过分布式计算减少单一点的计算负担,优化整体性能.

应对恶意行为:协议特别关注防止恶意参与方通过提交错误数据来破坏或误导双素性测试的结果.通过一系列检查和平衡措施(如输入承诺和一致性验证),增强了整体的协议弹性.

结论:

这种高度结构化和安全性强化的双素性检验协议为分布式系统中的密钥生成提供了一种可靠的解决方案,适用于需要高安全保证的应用场景.通过细致的设计和对潜在安全威胁的全面考虑,该协议展示了现代密码学在实际应用中的复杂性和深度.

分布式双素性检验协议（VerifyBiprime）的详细实现过程以及其统计保证。此算法为复杂的双素性检测提供了一种基于密码学功能的方法，涉及多方计算和校验机制来确保结果的准确性和安全性.

算法和理论分析概述:

算法 \(VerifyBiprime(N, M, c, \{(p\_i, q\_i, t\_{i,j}, s\_{i,j})\}\_{i\in [n]})\):

1. 初始化步骤:

随机采样 \(r \in \mathbb{Z}\_N\) 并计算 \(z = r \cdot (1 + \sum\_{i \in [n]} (p\_i + q\_i)) \mod N\).

检验 \(N, p\_i, q\_i\) 是否符合特定条件,如 \(p\_i, q\_i\) 必须在 [0, M) 内,并验证通过其他数学关系.

2. 复合验证:

确保 \(N = \sum p\_i \cdot \sum q\_i\) 并且所有的 \(t\_{i,j}\) 和 \(s\_{i,j}\) 都能够满足相关数学关系的条件,如 \(t\_{1,j} = s\_{1,j} + c\_j \cdot (p\_1 + q\_1 6)/4\) 等.

如果以上条件不满足,则输出失败.

3. 一致性检查和GCD测试:

确保所有参与方提交的数据在逻辑和数学上一致.

使用 \(VerifyBiprime\) 算法的电路版本在 \(F\_{ComCompute}\) 上执行,检查 \(gcd(z, N) = 1\) 确保 \(N\) 是双素数.

理论支持和安全保障:

统计保证:该协议利用一系列密码学功能性,如 \(F\_{CT}\)、\(F\_{ComCompute}\) 和 \(F\_{Zero}\),以确保在恶意模型下的安全性.此外,算法内嵌的数学验证和逻辑校验进一步强化了对抗恶意行为的能力.

安全模型:论证基于能够处理多达 \(n1\) 个恶意参与方的假设,展示了即使在高度不信任的环境中,协议也能保持其完整性和有效性.

效率和实用性:尽管涉及复杂的计算步骤,协议设计考虑了计算和通信效率,通过分布式计算降低了单个参与方的负担,提高了总体性能.

结论:

这种双素性检验协议的设计展示了在现代密码学应用中处理复杂问题的能力,通过结合理论严谨的数学检验与先进的密码学技术,提供了一种既安全又高效的解决方案.通过详细的安全模型和统计保证,这种方法适用于需要高度安全保障的关键应用场景,如公钥基础设施和安全通信协议.

# 6.效率分析

作者讨论了他们的协议在半诚实和恶意安全模型下的性能分析.特别是,这包括理想功能 \(F\_{RSAGen}\) 的成功概率分析,这是通过对其使用的采样函数 \(CRTSample\) 进行分析得出的.文档还探讨了如何通过组合多次调用 \(F\_{RSAGen}\) 来增加生成双素数的概率,以及实现 \(F\_{AugMul}\) 和 \(F\_{Biprime}\) 协议的成本分析.

协议分析概述:

1. 成功概率分析:

通过对采样函数 \(CRTSample\) 的性能进行详细分析,评估理想功能 \(F\_{RSAGen}\) 生成双素数的成功概率.

提出不同的策略来组合 \(F\_{RSAGen}\) 的多次调用,以提高双素数成功生成的可能性.

2. 成本分析:

给出了实现 \(F\_{AugMul}\) 和 \(F\_{Biprime}\) 功能时的具体成本计算.

分析了这些功能性在整个协议中所占的成本比例,并提出了减少成本的方法.

3. 组合成本与回合数:

在节 6.3 中,作者提供了将成本分析合成整个 \(TRSAGen\) 协议调用的成本估算.

节 6.4 探讨了一种构成策略,该策略旨在通过减少总回合数来优化协议效率,计算了实际所需的回合数.

4. 性能比较:

节 6.5 提供了与 Frederiksen 等人的协议的性能比较,突出显示了新协议在安全性、效率和成本方面的优势和潜在的不足.

理解协议的理论基础和实际应用提供了重要视角,特别是在考虑安全性和成本效率的平衡时.通过详细的数学和算法分析,文档不仅证明了协议的可行性,也强调了在设计高安全性密钥生成协议时需要考虑的多个关键因素.

## 6.1.每个实例都有成功的概率

提供了关于协议 \(F\_{RSAGen}\) 和其关键组件 \(CRTSample\) 的成功概率的深入分析,包括具体的成功概率计算和所需的调用次数以生成一个有效的双素数.文档通过理论分析和具体的参数实例来评估协议在不同的安全参数设置下的表现.

主要内容和理论分析:

成功概率的界定

Lemma 6.1: 提供了一个具体的成功概率下限,这是基于一个随机选择的值是质数的条件概率,考虑到它与最大元素 \(m\) 的 \(m\)互质性.

Lemma 6.2 和 6.3: 分别讨论了 \(m\) 的最大元素和 \(M\)(最大数的乘积)的增长率,这些都是分析 \(CRTSample\) 成功概率的关键因素.

参数实例和成功率

表格中列出了不同的安全参数 \(k\) 下 \(CRTSample\) 的具体迭代参数和对应的成功概率.例如,对于 \(k=1024\),每次迭代成功的概率至少为 \(1/3607\).

组合策略

定理 6.4 (Samplingefficiency theorem): 描述了为了达到足够的成功概率所需的 \(F\_{RSAGen}\) 调用次数.根据此定理,期望的调用次数与 \(k\) 的平方成正比,这反映了双素数生成的困难度.

为了提高效率,协议可以通过并行批处理来执行,这减少了总的等待时间并优化了资源使用.

性能优化

讨论了通过调整并行调用的数量来平衡调用次数和响应时间的方法,提供了不同 \(p\) 值下所需的具体调用次数,从而为实现期望的成功概率提供了具体的操作指导.

通过理论推导和具体实例展示了在实际应用中如何有效地管理和预测 \(F\_{RSAGen}\) 的性能,特别是在涉及高安全需求的密钥生成过程中.通过这种方法,可以在保证安全的同时,有效控制操作的复杂性和成本.

## 6.2.实例化FBiprime和FAugMul的成本

实现不同密码学功能性时的具体成本,包括 \(F\_{Biprime}\) 和 \(F\_{AugMul}\).文档详细讨论了在半诚实和恶意安全模型下,这些功能性如何被实现,并分析了相关的资源消耗.

主要内容和分析:

1. 实现 \(F\_{Biprime}\) 的成本:

文档指出,为了在恶意环境下实现 \(F\_{Biprime}\),采用了 BonehFranklin 的双素性测试方法.这个过程需要在每个阶段进行数据提交和解除提交,以确保消息在被揭示前已经得到承诺.

GCD测试和一致性检查是双素性测试的关键步骤,只有在 \(N\) 不是双素数时才会触发成本,这样可以优化在 \(N\) 是双素数的情况下的运行效率.

2. 实现 \(F\_{AugMul}\) 的成本:

\(F\_{AugMul}\) 通常用于多方计算场景中实现乘法操作,可以选择使用基于义务转移(OT)的乘法或其他方法.这些操作通常涉及大量的数据传输和处理,因此成本分析主要集中在数据传输和OT协议的执行上.

对于半诚实设置,通过简化的协议和优化的OT扩展协议(如 KOS OT)来减少传输成本和提高效率.

3. 运算和通信成本的详细分析:

表格中列出了不同算术操作的成本,包括模加、模乘等,这对于理解整体协议的资源消耗至关重要.

通过改进的 OT 协议和高效的消息处理策略,能够在保证安全性的同时,降低通信和计算成本.

4. 效率改进和安全性权衡:

讨论了如何在保证安全性的基础上通过技术手段降低成本,例如使用 Silent OT 来减少 OT 协议的开销.

同时,还强调了在实际部署中需要根据具体情况调整参数和选择技术的重要性.

对高安全性密钥生成和验证协议中涉及的具体计算和通信成本的深入洞察.通过这种成本分析,可以更好地理解在不同安全需求下实现这些功能性的资源需求和潜在的优化空间.这有助于在实际应用中做出合理的技术选择和资源配置.

## 6.3综合上述内容

在本节中,我们将详细解析我们的主要协议 \(TRSAGen\) 的成本,这包括对 \(F\_{AugMul}\) 和 \(F\_{Biprime}\) 的调用成本,以及与这些功能性交互较少的协议组件的成本.这一分析(除功能性调用的一致性检查阶段外)在恶意和半诚实设置中是相同的.我们随后将插入我们在6.2节中得到的各个组件的成本,并得出单个协议实例的具体成本数字.然后我们使用6.1节的分析来确定当我们的采样协议被顺序调用时采样一个双素数的具体成本.我们将讨论实现恒定或预期恒定轮次数的替代组合策略(并在下一节提供确切的轮次数).我们从成本细分开始:

1. 候选筛选:对于每个 \(j \in [2, \ell]\),各方调用 \(F\_{AugMul}\) 的 sample 指令使用模数 \(m\_j\).这些调用是并发的.接着,对于 \(j \in [\ell+1, \ell']\),各方调用 \(F\_{AugMul}\) 的 input 指令两次,并使用模数 \(m\_j\) 调用 multiply 指令.如附录 B.3 讨论,如果按此模式调用,input 指令是免费的.

2. 双素性测试:各方最多发送一次 check-biprimality 到 \(F\_{Biprime}\).由于我们的协议包括局部试除(见 \(TRSAGen\) 的第5步),有些协议实例可能完全跳过双素性测试.为了计算具体成本,我们假设这种局部试除没有执行.换句话说,我们假设每个协议实例都精确地运行了一次双素性测试,如果局部试除会拒绝一个候选者,那么双素性测试肯定失败.

3. 一致性检查:这一阶段在半诚实设置中无成本.在恶意设置中,其成本取决于候选双素数是否通过双素性测试和局部试除测试.在失败的情况下,各方改为发送 \(check, sids, f\),其中 \(sids\) 是在筛选阶段采样或加载到 \(F\_{AugMul}\) 的会话 ID 向量,\(f\) 是在执行一系列 \(CRTRecon\) 调用后得到的值,对应于小模数 \(m\_j\) 上的扩展向量.在每种情况下,我们使用 \(CRTRecon\) 在不需要额外模数输入的情况下完成这些调用,并且其总门数如下所述:

\[

(\ell' - 1) \cdot (2k + \max(m\_j) + \ell' - \log\_2 \ell') + 3k - \log\_2 n + \sum\_{j \in [\ell']} \text{intmul2}(2k, m\_j) + \text{modred}(2k) + \text{intmul}(k) - 2

\]

4. 通信复杂性:为了探讨我们协议在 \(k, s, n\) 方面的渐进网络成本,我们考虑 \(\lambda\) 为 \(O(s)\) 目的.我们还假设 \(s = O(k) \cap O(\log k)\).我们所有的渐进数字都是按每方传输的数据量给出的.

## 6.4.严格恒定和预期恒定的回合

在前面的章节中,我们讨论了在假设 $\pi\text{RSAgen}$ 顺序迭代的情况下生成双素数的成本.这样确保不会浪费任何工作或通讯资源,因为在第一次成功之后不会再运行额外的 $\pi\text{RSAgen}$ 实例.然而,这也导致了预期轮次复杂性的增加,其增长至少为 $\Omega(2^{-k/2} \log^2 k)$.

我们还尝试了并行生成候选值的方法,但这会导致通信复杂性相对于顺序组合增加,这是不令人满意的.因此,我们提出了一种非黑盒组合策略,通过调整批量大小参数,使用批处理采样程序并行采样,以确保至少有 $x$ 次成功,并以期望的恒定轮次采样双素数.

通过具体的计算例子,我们发现,如果我们按批次采样 9899 个候选值,至少 2500 个将是非零的,成功概率为 $1 - 2^{-S}$.这种方法的带宽开销大约是顺序策略的 1.65 倍.

最后,我们确定了精确的轮次计数,这与 $\pi\text{RSAgen}$ 的单个实例的轮次计数相同.如果使用 $\pi\text{AugMul}$ 并行采样,则总共需要 5 轮采样.通过调整和组合轮次,我们显著减少了总轮次数,如果双方是首次互动,他们只需要进行一次额外的 OT 扩展初始化.

## 6.5.与前期工作的比较

在这一节中,我们深入比较了我们的方法和Frederiksen等人的先前工作.不幸的是,Frederiksen等人没有报告具体的通信成本,所以我们必须尽可能重新推导.在讨论具体的成本之前,我们首先描述了他们方法的两种不同的方案,并且我们的比较方法也基于这两种协议只支持两方的事实.

我们的采样机制和他们的采样机制在结构上有很大的不同.我们的CRT形式采样具有复杂性在$O(k \cdot \log k)$或$O(k \cdot s)$范围内,适用于半诚实或恶意设置.相比之下,他们的方法是从其因子的标准形式中取样整数份额,然后使用自设计的OT乘法器计算$N = p \cdot q$.值得注意的是,他们的乘法操作没有在CRT形式下执行,因此计算复杂性达到了$O(k^2)$,并且在数据泄露方面,我们的方法没有比Boneh和Franklin更多的泄露.

在恶意环境下,Frederiksen等人的协议采用了证明诚实性的方法,主要是一个由一般MPC功能评估的电路,类似于我们$\pi\text{RSAgen}$协议的$\pi\text{Biprime MPC}$构成.不同的是,他们需要运行他们的电路两次,因为他们的协议仅支持两方,而且已经有泄露.相比之下,我们必须使用一个$n$-方方法来评估我们的电路,以实现完全的恶意安全性.

总的来说,尽管我们的方法在成本上较高,但在实际和渐进上我们的协议都显示出了明显的优势,特别是在采样双素数的预期成本上.这直接体现了我们使用CRT形式采样策略的优势,因为两种半诚实协议在其他方面非常相似.在恶意设置中,我们的协议和Frederiksen等人的协议之间的关系更为复杂,但我们的协议在理论和具体实施上都更为优越.