预处理模型下安全多方计算协议的相关随机性研究

Correlated Randomness for MPC in the Preprocessing Model

杨鹏

哈尔滨工业大学（深圳）

21s151080@stu.hit.edu.cn

2022/07/22

摘要

预处理模型被提出，特别是在不诚实大多数的情形下，以降低安全多方计算协议的计算和通信开销。预处理模型下的安全多方计算协议在预处理阶段生成大量相关随机性，然后用于在线阶段，使得在线阶段无需复杂的密码学操作，且只需要几轮交互便可完成电路评估。在大多数文献中，相关随机性被考虑为 Beaver 三元组，但更丰富形式被注意到可以用于不同的应用场景，来构造具体有效的安全多方计算协议。本文针对预处理模型下安全多方计算协议进行分析，总结了现有文献在预处理阶段和在线阶段的问题，阐明了相关随机性设计和生成对于预处理阶段和在线阶段的影响，并提出相应的研究内容和研究方案。

1 研究背景与意义

安全多方计算 (Secure Multi-Party Computation, MPC) 起源于姚期智院士提出的 “百万富翁问题”，经过四十多年的发展，已经成为隐私计算领域的一项重要技术工具。随着隐私保护应用需求的日益高涨，基于 MPC 的隐私保护技术发挥着越来越重要的作用。然而，在 MPC 协议保障用户隐私的同时，也给隐私保护应用带来了额外开销，实用安全多方计算协议依然面临着巨大的挑战。

在实用安全多方计算协议中，会面临各种各样的攻击者（也称敌手）。目前，有两种主流的安全模型被提出以刻画敌手的能力，分别是半诚实安全模型和恶意安全模型。在半诚实模型下，敌手遵循安全多方计算协议的规则，试图通过检查计算过程中的交互记录，以得到更多超出允许范围的信息。而在恶意模型下，敌手可以任意偏离协议的规则，以破坏安全多方计算通用协议的安全性。因此，恶意敌手比半诚实敌手的行为更为复杂且更难以应对，这使得构造恶意安全多方计算协议比构造半诚实安全多方计算协议的难度更大，也更具有挑战性。

当考虑存在恶意敌手的不诚实大多数情形时，安全多方计算协议的通信开销和计算开销一直是实用化过程中的瓶颈。目前有两条主流的技术路线来解决这个问题：

- 基于混淆电路（Garbled Circuit）的协议：这方面的工作有非常好的理论结果，可以实现常数轮的协议。不过这类工作大多只针对布尔电路，而且由于要针对电路的每个门构造混淆表，使得协议往往要求很高的带宽；
- 基于秘密共享（Secret Sharing）的协议：这类方案需要逐门计算电路（Gate-by-gate Circuit Evaluation），因此协议轮数和电路深度成正比，但该协往往具有较低的通信开销。

通常来说，基于混淆电路的协议适合广域网（Wide Area Network，WAN）的设定，因为该协议由于是常数轮的，因此延迟可以很低；而基于秘密共享的协议更适合局域网（Local Area Network，LAN）的情形，因为局域网下该协议的通信代价较低 [1]。不过，基于秘密共享的协议通常基于域或者环，其上的算术运算在现实应用场景更为契合，例如机器学习算法由大量的加法和乘法构造，基于混淆电路的协议采用布尔运算很难实现算术操作。

在基于秘密共享的协议中，有一类协议非常适合构造不诚实大多数的恶意安全多方计算协议，即基于预处理模型的安全多方计算协议[1]。在预处理模型中，协议被分为以下两个阶段：

- 预处理阶段（Preprocessing Phase）：在预处理阶段，参与方共同运行一个与输入独立的协议来安全地生成相关随机性（Correlated Randomness），这些随机的相关数据可以辅助完成计算任务。该阶段也称为设置阶段（Setup Phase）、离线阶段（Offline Phase）。
- 在线阶段（Online Phase）：一旦参与方输入已知，在线阶段就利用预处理阶段提供的相关随机性来完成目标计算任务。

预处理模型下的 MPC 协议将大量的通信和计算移到了预处理阶段，从而获得了十分高效且安全的在线阶段。根据已有的工作，如果给定相关随机性的话，在线阶段的通信开销仅为对应明文计算过程的常数倍。另外，在线阶段甚至可以不使用任何密码学操作，从而实现信息论安全。对于预处理阶段，该阶段不需要知道参与方的输入，因此预处理阶段可以在目标计算任务之前的任何时间执行。总而言之，预处理模型下的 MPC 协议能够大大加快在线阶段的速度，并且能很好地扩展到恶意安全模型下。

尽管预处理模型被广泛使用在恶意安全多方计算协议中，特别是不诚实大多数（Dishonest Majority）的情况，但是预处理模型的实际效率离理论结果有很大的差距。一方面，预处理模型存在与应用场景不符的假设，例如预处理模型下的 MPC 协议往往基于较大的整数域或者整数环，但大多数应用场景下都需要处理小数或者浮点数，这使得预处理模型不能被直接应用于现实场景；另一方面，大部分工作的预处理模型针对通用安全多方计算协议，因此该协议可以处理任何计算任务，但这也使得所处应用场景中的一些特点被忽略。

因此针对特定隐私保护应用场景下，设计预处理模型下切实有效（Concretely Efficient）的 MPC 协议尤为重要，切实有效的 MPC 协议指该协议具有良好的现实安全性和效率，具有更低通信复杂度，并

[1]尽管基于混淆电路的协议也可以结合预处理模型 [2]，使得基于混淆电路的方案和基于秘密共享的方案之间的性能差距有所缩小，但这种方法将混淆电路看成相关随机性，使协议并不通用，而且在不同的网络设置下，两种方法还是像之前的讨论一样各具优势。

且是半诚实安全的或是恶意安全的，其计算代价和存储开销越小越好。

2 研究现状及分析

预处理的概念最早源于 Beaver 的工作 [3]。在 1991 年，Beaver 针对 BGW 协议的复杂乘法运算，提出了电路随机化（Circuit Randomization）的概念，该技术允许提前准备乘法三元组（Multiplication Triples）用于乘法运算。通过利用乘法三元组（后也称 Beaver 三元组），用一轮交互便可完成一次乘法运算，大大提高了乘法运算的效率。

图 1 描述了两方计算中基于 Beaver 三元组的预处理模型框架，该框架基于加性秘密共享（Additive Secret Sharing）方案，可以计算算术电路和布尔电路。

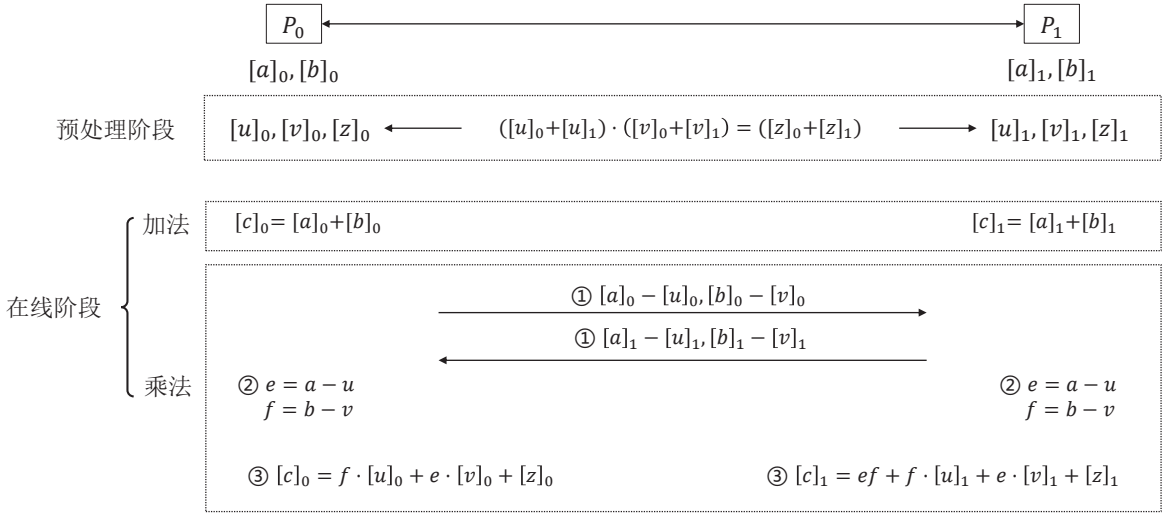

图 1: 基于 Beaver 三元组的预处理模型

在后续的研究中，预处理模型被大量工作广泛使用，它们分别聚焦于不同的方面，总得来说有以下几个方面：

1. 关注预处理阶段的开销：尽管预处理模型可以获得一个高效安全的在线阶段，但是预处理阶段是整个协议的瓶颈。在预处理阶段，许多工作关注下面几点：
   - 相关随机性的形式：大部分工作仅考虑图 1 的 Beaver 三元组及其可认证形式，但是更多形式的相关随机性被讨论，而且已经得到应用。
   - 相关随机性的生成：目前大多数工作都聚焦于生成相关随机性，有基于同态加密的，有基于不经意传输协议的等等，这个方向是一个非常活跃的研究方向。
2. 关注在线阶段的开销：高效安全的在线阶段是大多数工作的目标，也是预处理模型需要不断探索的方向。

2.1 预处理阶段

2.1.1 预处理模型下的相关随机性

预处理模型下的相关随机性是指在未知参与方输入（和待计算功能函数）的情况下在预处理阶段生成的随机的相关数据，图 1 的 Beaver 三元组 $[u], [v], [z]$ 即为最经典的一种相关随机性形式。讨论相关随机性形式的动机有以下几点：

1. 相关随机性预处理模型的关键，研究不同形式的相关随机性可以对其本质有更加深入的了解。

2. 当考虑专用预处理（Dedicated Preprocessing）时，待计算功能函数可以提前已知（当特定的计算场景被考虑时这个假设是实际的），我们可以针对不同的基本运算（加法、乘法、矩阵加法、矩阵乘法等），甚至更复杂的运算（线性系统求解）设计专用的相关随机性来提高在线阶段和离线阶段的效率。

正如前面提到的，在现有的工作中大部分都直接沿用了 Beaver 三元组这一基本形式的相关随机性，当现在的工作显示了探索相关随机性更多形式的趋势。

2013 年，Damgård 和 Zakarias [4] 提出 MiniMac 协议的同时，注意到 BDOZ 协议和 SPDZ 类协议使用的通用预处理（Universal Preprocessing）既不知道参与方输入也不知道待计算函数，该工作进一步放松了这一限制，提出了专用预处理（Dedicated Preprocessing）的概念[2]。专用预处理模型的预处理阶段可以提前知道计算电路（例如 AES 电路）以便预先计算更多的结果。

2013 年，Keller 等人 [5] 考虑到了相关随机性的形式，提出了除常见的 Beaver 三元组外的 Square 元组、Bit 元组、Input 元组和 Inv 元组，从而提高某些算术操作的效率。不过工作 [4,5] 都没有继续深入研究。

2017 年，Mohassel 等人 [6] 提出了一个隐私保护机器学习系统 SecureML，该系统设计中将 Beaver 三元组矢量化，实际上提出了矩阵形式的乘法三元组。2020 年，Chen 等人 [7] 提出了隐私保护深度学习系统，该系统将 Beaver 三元组扩展到矩阵形式和卷积形式的乘法三元组。

2022 年，Reisert 等人 [8] 则进一步扩展了 Beaver 三元组和二项式元组（Binomial Tuples），提出了算术元组（Arithmetic Tuples）的概念去计算多元多项式。

分析与总结 目前，已提出的相关随机性有 Beaver 三元组及其可认证版本、矩阵形式和卷积形式的乘法三元组、混淆电路相关随机性、不经意传输和不经意线性函数评估相关随机性和一次性真值表相关随机性 [9]。除了乘法三元组形式的相关随机性针对乘法运算，其他相关随机性都针对特定的函数计算，这也是相关随机性发展的趋势。因此，上述相关随机性的优化以及进一步探索更多形式的相关随机性是未来的方向。

举例来讲，工作 [10] 为求解线性系统（Linear Systems），设计了加法、乘法和除法的子协议，该协议开销最大的是除法部分，而除法的引入是为了解决矩阵求逆问题，因此如果能针对除法运算或者矩阵求逆运算设计相应的相关随机性，那么协议的在线阶段的开销将大大减少。不过这十分具有挑战性，原

[2]值得注意的是，针对通用电路的专用预处理本质上还是通用预处理。

因在于相关随机性与参与方输入无关，而很多问题的基本运算（如矩阵求逆）和参与方输入关联性很大，因此针对这个问题作出相应转换十分重要。

2.1.2 相关随机性生成方案

生成相关随机性很微妙的一点是该生成方案必须是隐私保护（Privacy Preserving）的，即必须保证生成的相关随机性是正确的，各方的相关随机性不能泄露给其他参与方，因为相关随机性的安全决定着参与方输入的安全性。相关随机性生成方案是预处理模型最活跃的领域之一，大部分工作集中于如何高效安全地生成相关随机性，图 2 总结了主要的相关随机性生成方案。

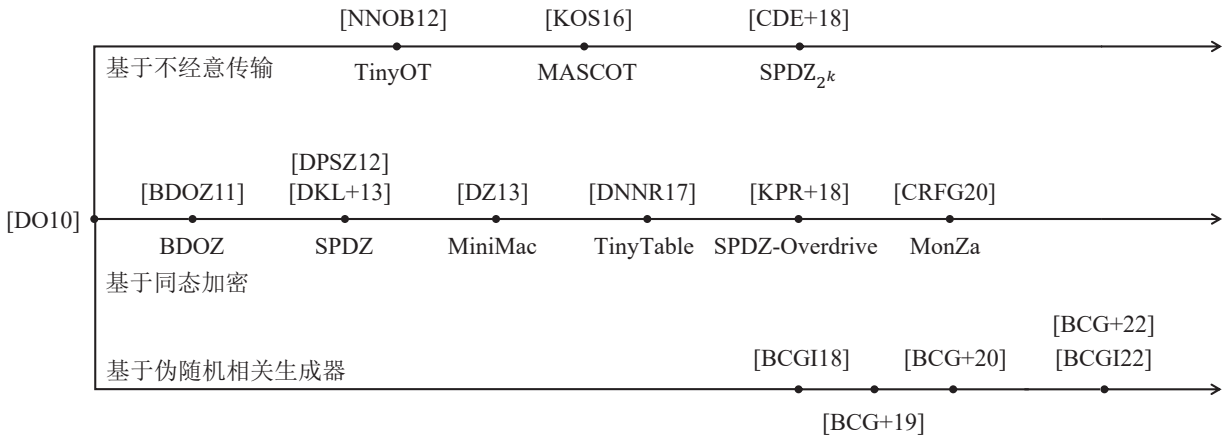

图 2: 相关随机性生成方案

早期的研究往往关注于让一个可信第三方为协议的在线阶段提供相关随机性，并讨论如何用一个高效且安全的协议来实现这个可信第三方 [11, 12]。2010 年，Damgård 和 Orlandi [12] 提出了不诚实大多数下的恶意安全多方计算协议，并在协议中采用了预处理模型。在该协议中，采用的乘法三元组是带承诺的 Beaver 三元组，例如 $[a], [b], [c]$ 和 $\mathsf{Comm}(a), \mathsf{Comm}(b), \mathsf{Comm}(c)$，并利用基于 Shamir 秘密共享的安全协议来生成乘法三元组，同时用零知识证明来验证三元组的正确性。

Damgård 和 Orlandi 的工作是很多后续工作的基础，特别是基于同态加密的相关随机性生成方案，这类方案的背后动机是同态加密的突破性发展。

用同态加密实现预处理阶段 2009 年，Gentry [13] 基于理想格提出了第一个全同态加密（Fully Homomorphic Encryption，FHE）方案。这使得许多工作将安全多方计算和同态加密结合起来，即用同态加密来生成乘法三元组。

2011 年，Bendlin 等人 [14] 利用半同态加密（Semi-Homomorphic Encryption，SHE）来实现高效的预处理阶段，构造了不诚实大多数的恶意安全多方计算协议（称为 BDOZ 协议）。与工作 [12] 不同的是，BDOZ 协议的预处理阶段不再依赖于离散对数问题的困难性，仅仅基于 Paillier 加密系统 [15] 的安全性。

2012 年，Damgård 等人 [16] 也实现了一个不诚实大多数的恶意安全多方计算协议来计算 $\mathbb{F}_{p^k}$ 上的算术电路，该协议优化了 BDOZ 协议的可认证秘密共享方案，使用部分同态加密（SomeWhat Homomorphic

Encryption，SWHE）来生成乘法三元组，进一步降低了在线阶段的计算复杂度和通信复杂度，使其与参与者数量呈线性关系，是对应明文计算的常数倍。工作 [17] 则具体实现了该协议。Damgård 等人 [18] 在 2013 年提出了进一步的优化和扩展，并且提供了高效的实现。相比于工作 [16] 只考虑恶意敌手，该协议还考虑了隐蔽敌手（Covert Adversaries）。具体来说，在基本假设方面，工作 [16] 使用标准模型，而该协议基于 RO 模型；针对在线阶段，该协议优化了消息认证码的验证，使得在线阶段可以“重用”相关随机性，并将工作 [16] 中 sacrificing 步骤移至离线阶段进行；针对离线阶段，该协议设计了针对隐蔽敌手的同态加密方案，不过相比于工作 [16] 使用基于 Schnorr 类协议的零知识证明来实现恶意安全，该协议则使用类似于 Cut-and-Choose 的技术来代替 Schnorr 类协议，解决了工作 [16] 中的许多问题 [19]。为了在它们之间进行区分，这两个协议分别称为 SPDZ-1 [16] 和 SPDZ-2 [18]。

Damgård 等人的工作创造性地将预处理模型用于解决不诚实大多数的恶意安全多方计算协议，并引入消息认证码、同态加密和零知识证明等技术。后续许多工作遵循了 SPDZ 协议的范式 [4,14,16,18,20–22]，目前最好的结果是 Keller 等人在 2018 年提出的 SPDZ-Overdrive 协议。

2018 年，Keller 等人 [21] 提出了 SPDZ-Overdrive 协议，其中分别包括 Low Gear 协议和 High Gear 协议，前者用基于 BGV 的加法同态加密来替换 SPDZ 离线阶段的部分同态加密，这使得 ciphertext modulus 更小，从而减少了通信量和计算量；而后者则用全局的知识的零知识证明代替了 SPDZ 协议中各方的零知识证明，这可以大大减少计算量。

总的来说，用同态加密实现预处理阶段的好处在于只需参与方提供输入便可得到输出，大大减少了通信量。不过，目前的全同态方案相较于 MPC 协议效率较低，只能用于计算小规模电路，因此大部分工作使用的是半同态加密（Semi-Homomorphic Encryption）和部分同态加密（Somewhat Homomorphic Encryption）。特别地，对于 SPDZ 类协议来说，其电路深度很小（在 SPDZ 的原始论文中电路深度为 1），而且可以使用并行化技术来提高效率，因此用同态加密实现预处理阶段比用 MPC 协议来实现会更加高效。

用 OT 协议实现预处理阶段 2012 年，Nielsen 等人 [23] 提出了 TinyOT 协议，该协议基于不经意传输（Oblivious Transfer，OT）构造了恶意安全两方计算协议[3]来计算布尔电路（Boolean Circuits），其在线阶段和工作 [14] 类似，但在预处理阶段则是基于 OT 协议和 OT 扩展（OT Extension）。2014 年，Larraia 等人 [24] 将 TinyOT 协议扩展到多方，并保持不诚实大多数的设置。

2015 年，Frederiksen 等人 [25] 基于相关 OT 扩展（Correlated OT Extension）协议，实现了预处理模型下的带 MACs 的 MPC 协议（MPC with MACs），该工作关注于有限域 $\mathbb{F}_{2^k}$，其结果显示基于相关 OT 扩展的预处理阶段比 SPDZ 协议 [16,18]、TinyOT 协议 [23,24] 和 MiniMac 协议 [4] 都要高效。

2016 年，Keller 等人 [26] 提出 MASCOT 协议扩展了工作 [25]，其预处理阶段基于相关不经意传输协议（Oblivious Transfer，OT），而不是 SPDZ 协议采用的部分同态加密技术，其性能相比于 SPDZ-2 协议提升了两个数量级。

2018 年，Cramer 等人 [27] 提出了 $\mathbb{Z}_{2^k}$ 上（而不是域上）的 SPDZ 类协议，称为 SPD$\mathbb{Z}_{2^k}$，该协议采用了 SPDZ-1 协议的预处理模型，但在预处理阶段使用 MASCOT 协议的 OT 技术来实现。该协议是

[3]注意到，恶意的两方计算属于不诚实大多数情形。

基于模 $\mathbb{Z}_{2^k}$ 的运算，这和现代计算机系统的 CPU 计算方式十分契合。

基于 OT 协议的相关随机性生成方案最好的结果是 Keller 等人提出的 MASCOT 协议，我们在表 1 中对比了 MASCOT 协议和 SPDZ-Overdrive 协议（Low Gear 协议和 High Gear 协议）生成乘法三元组的效率 [21]。

表 1: MASCOT 协议和 SPDZ-Overdrive 协议的乘法三元组生成效率

协议	元组数/秒	网络设置	所在的域
MASCOT	5100	1Gbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	214	50Mbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	5100	1Gbit/s	域 $\mathbb{F}_{2^{128}}$
LowGear	**30000**	1Gbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	3200	50Mbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	117	1Gbit/s	域 $\mathbb{F}_{2^{128}}$
HighGear	5600	1Gbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	1300	50Mbit/s	素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$
	67	1Gbit/s	域 $\mathbb{F}_{2^{128}}$

可以看到 MASCOT 在二进制域（Binary Fields）$\mathbb{F}_{2^{128}}$ 上十分高效，而 LowGear 在素数域 $\log_2 \lvert\mathbb{F}\rvert = 128$ 上十分高效，工作 [21] 表明当参与方数量大于 7 时，HighGear 比 LowGear 效率更高。

用伪随机相关生成器实现预处理阶段 2018 年，Boyle 等人 [28] 考虑到预处理模型中相关随机性逐渐成为瓶颈，因此从一种新的角度看待相关随机性，与 BDOZ 协议和 SPDZ 类协议将相关随机性看成乘法三元组（Beaver 三元组），该工作利用伪随机相关生成器（Pseudorandom Correlation Generator，PCG）来扩展一对相同的秘密随机串（Secret Random Strings）来获得相关随机性，并构造了针对 VOLE（Vector Oblivious Transfer Evaluation）相关随机性的相关生成器，实现了 “silent preprocessing” 的效果。值得一提的是，该协议是基于域上的 Learning Parity with Noise (LPN) 假设变体来保证安全性。

进一步地，Boyle 等人 [29] 在 2019 年则对 PCG 进行了系统的研究，并提出了针对 MPC 相关性的构造协议。不过工作 [28, 29] 需要较多的通信轮数，并且只能保证半诚实安全，因此 Boyle 等人 [30] 在 2019 年针对性地解决了这些问题。2020 年，Boyle 等人 [31] 基于环上 LPN（ring-LPN）假设设计了新的 PCG 方案，该 PCG 能够生成 OLE 相关性（OLE Correlations）、可认证乘法三元组（Authenticated Multiplication Triples）、矩阵乘法相关性（Matrix Product Correlations）等。

2021 年，Boyle 等人 [32] 提供了一个编译器，该编译器可以将任何满足温和的安全和结构要求的预处理模型下的 MPC 协议（即大多数半诚实安全的标准协议），编译为恶意安全多方计算协议，其中额外的附加存储及在线通信成本和电路大小呈对数关系，从而平衡了在线阶段的通信成本和相关随机性存储代价。

2022 年，Boyle 等人 [33] 基于 Expand-Accumulate Codes 设计了新的 PCG 构造方案，通信代价更低。同样是在 2022 年，Boyle 等人 [34] 基于工作 [32] 实现了预处理模型下的 MPC 协议的亚线性预处理阶段（Sublinear Preprocessing），其离线阶段的通信和电路大小呈亚线性关系。该工作也有效地结合了 SPDZ 类协议，高效地实现了 SPDZ 类协议可认证乘法三元组生成。

其他范式 除了上述范式，还有一些其他的相关随机性生成方案。2021 年浙江大学隐私计算团队的 Lu 等人 [35] 将可信执行环境（Trusted Execution Environment，TEE）引入到相关随机性的生成过程，该工作提出半可信硬件模型（Semi-Trusted Hardwares Model），利用半可信硬件（Semi-Trusted Hardwares）来生成混淆电路这种相关随机性，从而提高协议的效率。

分析与总结 基于 OT 和同态加密的范式被广泛研究，已经是非常成熟的技术，很多实际协议都使用 OT 和同态加密来实现预处理阶段 [6,36,37]。工作 [6] 认为在使用 OT 协议生成乘法三元组的通信量大，但计算量下，适合局域网环境，而使用同态加密生成乘法三元组的通信量小，但计算量大，适用于广域网环境。而基于伪随机相关生成器的方案非常有前景，它允许大幅度减少预处理阶段的通信量和存储量，但该方法基于 LPN 假设，而且目前并不成熟，许多方案仍在设计之中。

目前预处理模型的性能瓶颈在相关随机性生成方案效率这一方面，大部分工作都在探索更低通信量、计算量和存储量的预处理阶段。另外，当新的相关随机性被设计出来，如何生成该相关随机性也是一个难题，通用的基于 OT 和同态加密的范式可以直接使用，但表现出低的效率，而基于伪随机相关生成器的方案并不一定能够直接使用，因为该方案还并不通用。不过，关于预处理阶段效率问题是十分微妙的，在 Boyle 等人的工作之前很多工作忽略预处理阶段的效率，认为低效的预处理阶段不影响在线阶段的效率，甚至一些工作在相关随机性模型[4]下进行讨论并设计相关协议。不过，随着越来越多的工作关注切实有效（Concretely Efficient）的安全多方计算协议 [38]，预处理阶段的效率也越来越受到重视。

受到工作 [39] 将预处理阶段进行外包的启发，当考虑资源受限的参与方时，其计算能力和存储能力有限甚至不足以执行预处理阶段，将预处理阶段外包的方案将在很大程度上帮助参与方。可以看到，外包的预处理阶段是上述生成方案的扩展，可以获得很高的效率。不过，外包方案需要考虑新的安全模型，例如参与方和外包服务器之间的合谋，而且外包方案还必须是可验证的，即能够验证生成的相关随机性确实是按照预定函数进行计算的。

2.2 在线阶段

预处理模型最吸引人的就是高效的、信息论安全的在线阶段。预处理阶段生成相关随机性后，在线阶段会利用这些相关随机性来计算功能函数，例如图 1 中在线阶段的加法和乘法运算。讨论在线阶段的动机有以下几点：

1. 在给定相关随机性的条件下，需要去明确在线阶段开销的理论下界，而且更重要的是需要知道如何设计实际协议使其效率和和理论结果一致。

[4]相关随机性模型指给定相关随机性，将相关随机性生成方案当成黑盒。

2. 在线阶段利用相关随机性计算功能函数，而在特定的应用场景下，在线阶段的协议可以巧妙地利用好相关随机性来计算功能函数，并对相关随机性的生成有所影响。

在理论层面，Ishai 等人 [40] 在 2013 年研究了相关随机性能在多大程度上帮助构造不诚实大多数的安全多方计算协议，给出了很多关于通信复杂度和计算复杂度理论的结果，并提出基于 OTTT（One-Time Truth Table）的半诚实和恶意安全计算协议，该协议实现了相关随机性模型（Correlated Randomness Model）下的无条件安全性，对于计算功能函数 $f:\{0,1\}^n \to \{0,1\}^m$ 需要 $O(n+m)$ 的通信复杂度，但其相关随机性的存储量是指数时间的（n 的函数）。

2016 年，Damgård 等人 [41] 的结果显示当采用逐门计算电路设计模式（"gate-by-gate"design pattern）时，所有无条件安全协议（诚实大多数情况下，或预处理模型下的不诚实大多数情况）的通信复杂度都和电路大小成比例，而且这是固有（inherent）的结果。

2019 年，Couteau [42] 基于工作 [40, 41] 构造了相关随机性模型下亚线性的通信复杂度，并且相关随机性的存储量是多项式大小的。不过该工作只针对结构化电路，特别地，该协议只考虑了分层布尔电路（Layered Boolean Circuits，LBC）。

在 SPDZ 类协议及其实现中，已经达到了上述理论结果的下界，这意味着在线阶段很难再被改进。不过，在特定的应用场景下这一结果还是存在不断改进的空间。

例如，在 2017 年，Mohassel 等人 [6] 提出的隐私保护机器学习系统 SecureML 就表明在线阶段可以利用不独立的相关随机性。在该工作中，其使用 Beaver 三元组计算乘法，每评估一个乘法门便需要"消耗"一组相关随机性。而当计算类似于 $\mathbf{X}^T \times (\mathbf{X} \times \mathbf{w} - b)$ 的功能函数是，可以使用一组相关随机性 $[u], [v], [z]$ 计算乘法 $\mathbf{X} \times \mathbf{w}$，并得到结果 $\mathbf{Y}$，当计算 $\mathbf{X}^T \times (\mathbf{Y} - b)$ 时，则可以使用另一组相关随机性 $[u], [v'], [z']$ 来计算，这两组相关随机性是相关联的（重复使用了相关随机性 $[u]$），使得减少了需要生成的相关随机性数量，同时也改变了相关随机性的生成方案设计。

分析与总结 事实上，工作 [6, 8] 已经证明预处理模型的预处理阶段和在线阶段可以通过相关随机性相互影响，这表明虽然在线阶段可能有理论界限，但是这一结果并未很好地体现到具体的应用中，这也是实用安全多方计算协议需要努力的方向。

2.3 研究现状分析与总结

从预处理模型下的 MPC 协议发展来看，早期研究关注于不诚实大多数情况下的恶意安全多方计算协议，其在预处理模型下能够得到很好的结果，其中以 BDOZ 协议、SPDZ 协议为代表的基于同态加密的工作和以 TinyOT 协议和 MASCOT 协议为代表的基于不经意传输的工作分别给出了通用协议设计与高效实现。这些工作的主要区别在于预处理阶段的实现上，用 MPC 协议进行实现是一个自然的想法，基于不经意传输和同态加密的乘法三元组生成方案有不同的优势，而基于服务器辅助或可信硬件的方案则有更多很强的假设，需要在安全性和效率上有所权衡。

许多工作给出了预处理模型下无条件安全的在线阶段通信复杂度下界，随着研究的深入，关于相关随机性的研究逐渐进入人们的视野，伪随机相关生成器的提出针对性地解决了预处理阶段的效率瓶颈，但

也引入了更多的假设。不过，基于伪随机相关生成器的相关随机性生成方案给预处理模型下的 MPC 协议带来了新的启发。

最新的结果显示，预处理模型的安全多方计算协议在不诚实大多数的设定下可以实现线性通信复杂度的无条件安全的在线阶段，和亚线性的预处理阶段（在 LPN 假设下），但是这仅仅是理论的结果，获得实际有效（Concretely Efficient）安全多方计算协议仍具挑战。当处理现实场景下的具体应用时，待计算的函数及其所需的基本运算是提前可知的，因此可以考虑函数相关的预处理模型，这是和工作 [43] 类似的，不过进一步地，我们可以像工作 [6,8] 一样引入更多形式的相关随机性，为特定应用设计专用的相关随机性。这使得预处理模型的预处理阶段和在线阶段都变得不同，而且导致了一些开放的问题：

1. 相关随机性能否和 MPC 协议基本运算模块相结合，从而提高更复杂运算的通信和计算效率?
2. 与基本运算模块相结合的相关随机性如何生成?

第一个问题的可行性在 SecureML [6] 中已经得到验证，已有的混淆电路相关随机性、OT 和 OLE 相关随机性、（可认证）乘法三元组等都显示该问题是一个有趣的方向 [9]。而第二个问题也充满挑战，目前的方案在效率上和通用性上进行权衡，针对新的相关随机性设计相关随机性生成方案可以结合目前最新的结果 [34]，并考虑实际的性能。本质上，关于相关随机性的基本内涵也并不清晰，有关于它的许多问题（例如相关随机性在相同/不同的函数计算能不能重复使用）都值得进一步研究。

3 主要研究内容与研究计划

3.1 研究内容

下面，我们则针对上述两个开放的问题明确我们的研究内容，我们的目标是具体高效（Concretely Efficient）的预处理模型下安全多方计算协议，我们考虑不诚实大多数的半诚实敌手和恶意敌手，并考虑带中止的安全（Security with Abort）[5]。

研究内容一：预处理模型下面向复杂运算的相关随机性设计与应用

在预处理模型下的安全多方计算协议中，尽管在线阶段的大部分计算都被转移到预处理阶段，在线阶段的加法运算可以本地进行，乘法运算仅需要一轮交互，但是协议的轮数是和电路深度成正比的。可以看到的是，SPDZ 类协议使用了基于秘密共享的安全多方计算协议，因此使得协议轮数和电路深度成正比，这是 SPDZ 类协议固有的。但是如果我们牺牲一些通用性并可以提前获知待计算功能函数的信息，针对基本运算（矩阵运算、比较运算、线性方程系统等）设计专用的相关随机性，这些相关随机性可以被用于在线阶段，以提高在线阶段的协议计算效率和通信效率。因此，我们可以面向特定场景下的复杂运算设计专用的相关随机性，以使预处理模型变得实用。

此外，设计的相关随机性还可以根据在线阶段的功能函数计算进行优化，以针对部分输入进行重用相关随机性，减少相关随机性数量。

研究内容二：相关随机性生成方案设计及相关随机性的可验证外包计算方案

[5]理论已经证明，不诚实大多数的安全多方计算无法实现保证输出交付和公平性，协议会因为敌手的偏离协议而中止，这是不可避免的。

相关随机性被用于在线阶段的，其主要作用是掩蔽参与方输入，因此相关随机性不涉及参与方输入，但是相关随机性的泄露会影响参与方输入的隐私性，因此相关随机性的生成方案也应该被认为是隐私保护的。针对研究内容一，我们需要生成特定的相关随机性，目前基于不经意传输、同态加密的方案是比较成熟且安全的，而基于伪随机相关生成器（Pseudorandom Correlation Generator，PCG）的方案在基于 LPN 假设的情况下是十分高效的。我们需要针对应用场景和相关随机性设计实现预处理阶段。

另外，将相关随机性外包给云服务器进行计算可以释放参与方的大量资源，构成了一种新的相关随机性生成方案。不过，相关随机性的外包方案提出了新的安全模型，而且还需要验证计算好的相关随机性，因此需要研究和设计相关随机性的可验证外包计算方案。

图 3 总结了上述研究内容。

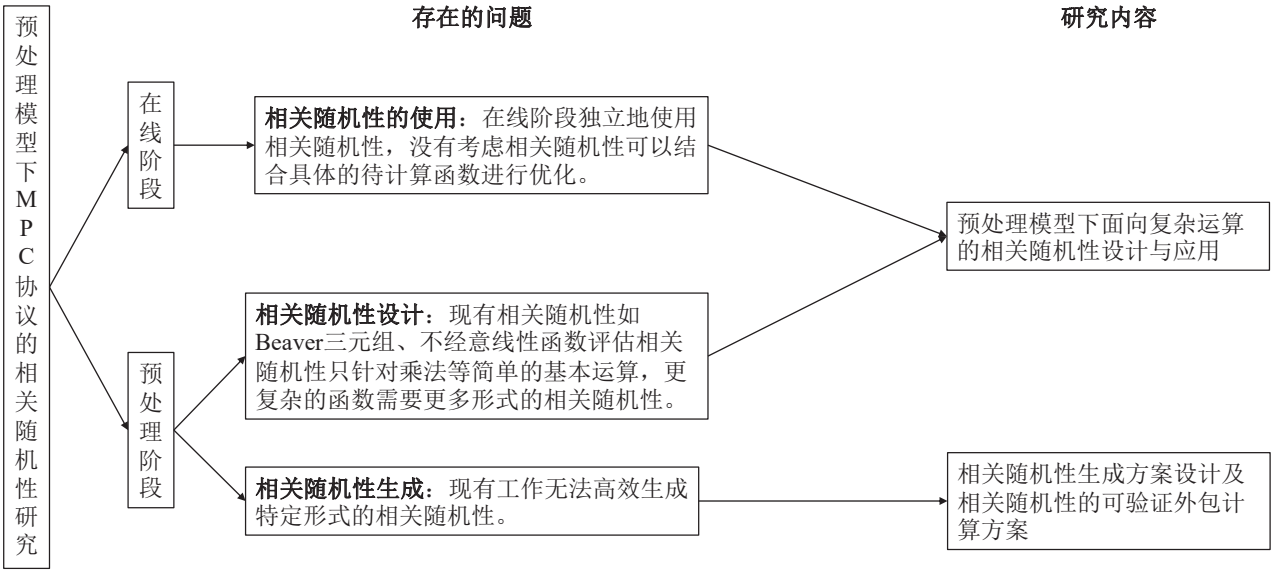

图 3: 主要研究内容

3.2 研究计划

针对上述研究内容，我们提出以下的研究计划，主要任务如下：

1. 进一步理解现有工作中相关随机性基本性质和工作原理，为进一步设计新的相关随机性巩固理论基础。

2. 分析隐私保护应用场景下的基本运算，从数据类型（整数、定点小数等）和运算操作（算术运算、布尔运算等）入手，设计利于电路计算的相关随机性，并分析新设计的相关随机性在电路计算的通信复杂度，与现有工作进行对比确定是否有理论改进。

3. 分析隐私保护应用的性能瓶颈，如果该性能瓶颈体现在现有的基本运算上面，则将新设计的相关随机性应用到针对性的隐私保护应用中，并和现有工作进行对比，确定是否在通信开销和计算开销上有改进。

4. 针对上述的新的相关随机性，研究和设计基于现有工作的相关随机性生成方案。

5. 分析和研究相关随机性的外包方案的可行性，如果外包方案是可行的，进一步设计可验证的相关随机性外包计算方案。

参考文献

[1] Emmanuela Orsini. Efficient, actively secure mpc with a dishonest majority: A survey. In *International Workshop on the Arithmetic of Finite Fields*, pages 42–71. Springer, 2020.

[2] Xiao Wang, Samuel Ranellucci, and Jonathan Katz. Global-scale secure multiparty computation. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 39–56, 2017.

[3] Donald Beaver. Efficient Multiparty Protocols using Circuit Randomization. In *Annual International Cryptology Conference*, pages 420–432. Springer, 1991.

[4] Ivan Damgård and Sarah Zakarias. Constant-overhead Secure Computation of Boolean Circuits Using Preprocessing. In *Theory of Cryptography Conference*, pages 621–641. Springer, 2013.

[5] Marcel Keller, Peter Scholl, and Nigel P Smart. An architecture for practical actively secure mpc with dishonest majority. In *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*, pages 549–560, 2013.

[6] Payman Mohassel and Yupeng Zhang. Secureml: A system for scalable privacy-preserving machine learning. In *2017 IEEE symposium on security and privacy (SP)*, pages 19–38. IEEE, 2017.

[7] Hao Chen, Miran Kim, Ilya Razenshteyn, Dragos Rotaru, Yongsoo Song, and Sameer Wagh. Maliciously Secure Matrix Multiplication with Applications to Private Deep Learning. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 31–59. Springer, 2020.

[8] Pascal Reisert, Marc Rivinius, Toomas Krips, and Ralf Kuesters. Arithmetic tuples for mpc. *Cryptology ePrint Archive*, 2022.

[9] Elette Boyle, Niv Gilboa, and Yuval Ishai. Secure Computation with Preprocessing via Function Secret Sharing. In *Theory of Cryptography Conference*, pages 341–371. Springer, 2019.

[10] Linpeng Lu and Ning Ding. Horizontal Privacy-Preserving Linear Regression Which is Highly Efficient for Dataset of Low Dimension. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, pages 604–615, 2021.

[11] Ivan Damgård, Martin Geisler, Mikkel Krøigaard, and Jesper Buus Nielsen. Asynchronous Multiparty Computation: Theory and Implementation. In *International workshop on public key cryptography*, pages 160–179. Springer, 2009.

[12] Ivan Damgård and Claudio Orlandi. Multiparty Computation for Dishonest Majority: From Passive to Active Security at Low Cost. In *Annual cryptology conference*, pages 558–576. Springer, 2010.

[13] Craig Gentry. Fully Homomorphic Encryption using Ideal Lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178, 2009.

[14] Rikke Bendlin, Ivan Damgård, Claudio Orlandi, and Sarah Zakarias. Semi-homomorphic Encryption and Multiparty Computation. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 169–188. Springer, 2011.

[15] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In *International conference on the theory and applications of cryptographic techniques*, pages 223–238. Springer, 1999.

[16] Ivan Damgård, Valerio Pastro, Nigel Smart, and Sarah Zakarias. Multiparty Computation from Somewhat Homomorphic Encryption. In *Annual Cryptology Conference*, pages 643–662. Springer, 2012.

[17] Ivan Damgård, Marcel Keller, Enrique Larraia, Christian Miles, and Nigel P Smart. Implementing AES via an Actively/Covertly Secure Dishonest-Majority MPC Protocol. In *International Conference on Security and Cryptography for Networks*, pages 241–263. Springer, 2012.

[18] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P Smart. Practical Covertly Secure MPC for Dishonest Majority–or: Breaking the SPDZ Limits. In *European Symposium on Research in Computer Security*, pages 1–18. Springer, 2013.

[19] 李艳斌, 刘瑜, 李木舟, 吴韧韬, and 王鹏达. MASCOT 协议的参与方自适应变体. 计算机科学, 47(11A):380–387, 2020.

[20] Ivan Damgård, Jesper Buus Nielsen, Michael Nielsen, and Samuel Ranellucci. The TinyTable Protocol for 2-party Secure Computation, or: Gate-scrambling Revisited. In *Annual International Cryptology Conference*, pages 167–187. Springer, 2017.

[21] Marcel Keller, Valerio Pastro, and Dragos Rotaru. Overdrive: Making spdz great again. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 158–189. Springer, 2018.

[22] Dario Catalano, Mario Di Raimondo, Dario Fiore, and Irene Giacomelli. MonZa: Fast Maliciously Secure Two Party Computation on $\mathbb{Z}_{2^k}$. In *Public-Key Cryptography - PKC 2020*, pages 357–386. Springer, 2020.

[23] Jesper Buus Nielsen, Peter Sebastian Nordholt, Claudio Orlandi, and Sai Sheshank Burra. A New Approach to Practical Active-secure Two-party Computation. In *Annual Cryptology Conference*, pages 681–700. Springer, 2012.

[24] Enrique Larraia, Emmanuela Orsini, and Nigel P Smart. Dishonest majority multi-party computation for binary circuits. In *Annual Cryptology Conference*, pages 495–512. Springer, 2014.

[25] Tore Kasper Frederiksen, Marcel Keller, Emmanuela Orsini, and Peter Scholl. A Unified Approach to MPC with Preprocessing using OT. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 711–735. Springer, 2015.

[26] Marcel Keller, Emmanuela Orsini, and Peter Scholl. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 830–842, 2016.

[27] Ronald Cramer, Ivan Damgård, Daniel Escudero, Peter Scholl, and Chaoping Xing. SPD$\mathbb{Z}_{2^k}$: Efficient MPC mod 2^k for Dishonest Majority. In *Advances in Cryptology–CRYPTO*, 2018.

[28] Elette Boyle, Geoffroy Couteau, Niv Gilboa, and Yuval Ishai. Compressing vector ole. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 896–912, 2018.

[29] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient Pseudorandom Correlation Generators: Silent OT Extension and More. In *Annual International Cryptology Conference*, pages 489–518. Springer, 2019.

[30] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Peter Rindal, and Peter Scholl. Efficient two-round ot extension and silent non-interactive secure computation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 291–308, 2019.

[31] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Efficient Pseudorandom Correlation Generators from Ring-LPN. In *Annual International Cryptology Conference*, pages 387–416. Springer, 2020.

[32] Elette Boyle, Niv Gilboa, Yuval Ishai, and Ariel Nof. Sublinear GMW-Style Compiler for MPC with Preprocessing. In *Annual International Cryptology Conference*, pages 457–485. Springer, 2021.

[33] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, Nicolas Resch, and Peter Scholl. Correlated Pseudorandomness from Expand-Accumulate Codes. In *Annual International Cryptology Conference*, pages xxx–xxx. Springer, 2022.

[34] Elette Boyle, Niv Gilboa, Yuval Ishai, and Ariel Nof. Secure Multiparty Computation with Sublinear Preprocessing. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 427–457. Springer, 2022.

[35] Yibiao Lu, Bingsheng Zhang, Hong-Sheng Zhou, Weiran Liu, Lei Zhang, and Kui Ren. Correlated Randomness Teleportation via Semi-trusted Hardware—Enabling Silent Multi-party Computation. In *European Symposium on Research in Computer Security*, pages 699–720. Springer, 2021.

[36] Daniel Demmler, Thomas Schneider, and Michael Zohner. Aby-a framework for efficient mixed-protocol secure two-party computation. In *NDSS*, 2015.

[37] Arpita Patra, Thomas Schneider, Ajith Suresh, and Hossein Yalame. ABY2.0: Improved Mixed-Protocol Secure Two-Party Computation. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 2165–2182, 2021.

[38] Dengguo Feng and Kang Yang. Concretely efficient secure multi-party computation protocols: survey and more. *Security and Safety*, 1:2021001, 2022.

[39] Peter Scholl, Nigel P Smart, and Tim Wood. When it' s all just too much: outsourcing mpc-preprocessing. In *IMA International Conference on Cryptography and Coding*, pages 77–99. Springer, 2017.

[40] Yuval Ishai, Eyal Kushilevitz, Sigurd Meldgaard, Claudio Orlandi, and Anat Paskin-Cherniavsky. On the Power of Correlated Randomness in Secure Computation. In *Theory of Cryptography Conference*, pages 600–620. Springer, 2013.

[41] Ivan Damgård, Jesper Buus Nielsen, Antigoni Polychroniadou, and Michael Raskin. On the communication required for unconditionally secure multiplication. In *Annual International Cryptology Conference*, pages 459–488. Springer, 2016.

[42] Geoffroy Couteau. A Note on the Communication Complexity of Multiparty Computation in the Correlated Randomness Model. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 473–503. Springer, 2019.

[43] Aner Ben-Efraim, Michael Nielsen, and Eran Omri. Turbospeedz: double your online spdz! improving spdz using function dependent preprocessing. In *International Conference on Applied Cryptography and Network Security*, pages 530–549. Springer, 2019.