arXiv:2302.05520v1 [cs.DS] 10 Feb 2023

同步/异步与静态/移动？后者更胜一筹……至少在理论上。

ELI GAFNI，加州大学洛杉矶分校

VASILEIOS ZIKAS，普渡大学

与异步性一样，故障的移动性 也会阻碍共识。然而，对于一种能够实现共识的模型 M，存在一种放宽条件的类比模型，在其中无法实现共识；我们可以进一步探讨：如果系统起初表现得如同这种放宽条件的类比模型，但最终转变为 M，那么共识是否可实现。我们考虑 M 的两种放宽条件的类比模型。第一种是传统的异步模型，第二种则是下文将定义的移动类比模型。对于某些 M，我们证明，在异步类比模型中无法实现共识，而在所有移动类比模型中均可实现共识。因此，从这个角度看，移动性优于异步性。

我们所设想的理想关系是：当且仅当二元提交-采纳在移动类比模型中可实现时，共识在 M中可实现。

“仅当”部分很简单。下面我们逐一说明，对于所有常见的故障类型，“若”部分都成立。

致谢

此版本已提交至 PODC 2020。

1 引言

1.1 移动性与消息对手

宽容性这一概念，通常带有贬义，如今在分布式计算领域得到了认可，用来形容一种算法：它能在一个起初异步、但最终表现为同步的环境中解决任务[14]。算法在异步阶段保持安全性，因此可容忍这一阶段。进入同步阶段后，它便能实现活性。显然，这一概念也可以扩展到任务 T，即如果存在一个宽容的算法来解决 T，那么 T就是宽容的。更准确地说，如果任务 T能在具有特定类型和数量故障的任意模型中得到解决，那么它最终也能得到解决，则 T是宽容的。

因此，宽容性涉及一个任务，以及所有能够完成该任务的模型。对于每个模型，我们假设都定义了一个宽松模型。如果系统从相应的宽松模型开始运行，但最终表现得如同（未宽松的）模型，那么该任务就是宽容的。

因此，宽容性有三个参数：任务 T、故障类型，以及未宽松模型/宽松模型对。

在本文中，我们将任务设为共识，并针对常见的故障类型，将模型对设为同步/异步。

本文的动机源于一个令人不快的观察：对于两种故障类型，共识并不具有宽容性。 t:

(1) 发送遗漏故障：在同步情形下，对于 $t < n$可以实现共识，其中 t 是故障数量， n 是处理器数量。相比之下，众所周知，在发生网络分区时，异步性下无法维持 $t \geq n/2$的安全性。(2) 认证拜占庭：在同步情形下，对于 $t < n/2$ [7, 10]可以实现共识。相比之下，在异步性下无法维持 $t \geq n/3$ [8]的安全性。

作者地址：Eli Gafni，加州大学洛杉矶分校，eli@ucla.edu；Vasileios Zikas，普渡大学，vzikas@purdue.edu。

为了避免引起不快，我们研究用静态/移动这一对概念取代同步/异步这一对概念。系统是同步的，每个处理器都会收到一条轮次结束信号，此后不会再有消息在传输途中发往该处理器。故障具有移动性时会发生错误行为，而我们期望的是故障变为静态。若故障数量为 t，在静态情形下，只有固定的 t 个处理器会发生故障。在移动情形下，每一轮可能有t 个不同的处理器表现出故障行为。

通常，故障归因于处理器。在这里，我们将讨论范围限定为这样一种对手：它通过控制处理器发送消息的接口来攻击处理器。因此，1-遗漏容错移动系统是一个同步 系统，其中所有处理器都向所有其他处理器发送消息；在每一轮中，对手选择一个处理器，并可丢弃该处理器发送的任意数量的消息（参见 [15]）。在拜占庭情形中，对手不仅可以丢弃消息，还可以篡改消息。最后，我们关注认证拜占庭情形；它在定义移动性概念以及证明相应结果方面带来了最主要的技术难题。

定义认证时，人们通常会想到通过数字签名实现密码学认证。但“认证”在英语词典中的定义并不依赖数字签名，它指的是核实某项陈述是否成立。核实一幅画是否出自毕加索、一块化石是否来自旧石器时代，或一本日记是否由希特勒所写，都称为认证。因此，本文中的认证拜占庭是一种抽象假设，涉及哪些陈述可以（因而无法伪造地）得到验证，哪些不能。显然，在我们的情形中，问题是：接收者能否核实如下陈述是否属实——“处理器 p_i 声称它在第 j 轮从 p_k 收到了消息 m”。因此，这对对手施加了限制，使其只能作出无法被证伪的陈述。撇开验证手段不谈，我们将认证视为一组有序对 $(p_i,\ j)$，其中 p_i 是处理器，j 是轮次编号。p_i 在第 j 轮发送的消息可以伪造，而由不在该集合中的有序对所生成的消息则不能伪造。

我们关于认证拜占庭的结果针对的是这一抽象约束。据我们所知，在移动情形下，认证拜占庭从未在如此清晰的抽象层面上被定义过。本文提出的一个挑战是：在适当假设下，例如使用密码学，实现我们抽象定义中的功能。不过，证明仍然可以在抽象功能层面上完成。

我们最初接触移动性这一思想，是通过 Santoro 和 Widmayer 的精彩观察 [19]： FLP 证明中的共识不可能性可以原封不动地转换到移动情形中；后来，这一思想又被应用于主动安全 [17]背景下侵犯隐私的腐化。

Santoro 和 Widmayer 谈到遗漏故障时，究竟将什么视为故障的起因，并不清楚。遗漏故障为什么会发生？对此可以有两种看法。一种是处理器行为失常，另一种是对手截获了所发送的消息，并将其丢弃。概念上，选择哪一种看法会带来很大的差异。在拜占庭故障情形下，前一种看法的推广是病毒控制了处理器；后一种看法的推广则是守护进程只篡改了消息。

从理论上讲，后一种将守护进程视为对手而非病毒的观点更令人满意——事实上，Afek 和 Gafni 正是由此提出了消息对手 (MAd)的概念 []。在这种方法中，消息对手统一了许多看似毫不相关的概念。例如，在此将 MAd 推广到拜占庭情形时，处理器始终都是无辜且良好的。因此，我们可以避免诸如“受损处理器是否必须输出？如果是，它们应该输出什么？”之类的问题，也无需将问题规定为涉及“正确”和“错误”的形式；这两个概念都与执行有关，从而违背了将任务仅视为一种数学关系、且独立于其所处求解环境的初衷。这相当于集中式计算中的函数。

唉，尤其是在拜占庭情形下，将对手视为消息对手，或许并不符合现实。本文讨论的并非现实，而是我们能否让数学表达更优美。

本文遵循集合共识 [5]的研究范式。集合共识没有杀手级应用。或许永远也不会有。尽管如此，数学表明，对于分布式计算而言，共识并不优于集合共识。本文的情况也类似。然而，事实上，有些主动安全方案正朝着我们理论上的方向发展，即 MAd[17]。

1.2 共识 vs. 提交-采用

提交-采用（CA） [13] 是一种放宽了共识要求的任务。只有当所有处理器都以相同的值启动时，它才要求满足“活性”。否则，它应当保证安全：如果某个处理器输出了一个值，那么所有处理器都采用该输出值作为自己的新输入值。

本文提出了一个非正式猜想：共识之于静态性，犹如 CA 之于移动性。从理论上带有偏见的同步/异步这一对概念中，人们或许已窥见这一点。在拜占庭情形下，对应的是共识与可靠广播 [4]。而可靠广播不就是共享内存（经适当定义），进而也就是 CA 吗？（仔细思考后便会明白。）对于同步/异步这一对模型域而言，共识/可靠广播可解的范围相同，即 $< n/3$。

上述猜想有两个方向。较容易的方向是：对于任何能够用二元 CA 求解的移动设置，其静态对应情形中的共识也是可解的。这可以由 [3, 11] 中通过 CA 构造共识算法的方法推出。反方向则是该猜想的核心。例如，它应当意味着：对于静态设置中的任意共识算法，若在对应的移动设置中执行，且所有处理器都以相同比特 b 开始，那么它们都应输出 b。我们尚未成功证明这一点。

相反，我们逐个情形进行了分析，并证明了以下结论：对于所有常见的故障类型，静态情形中共识所能容忍的故障数量，等于对应移动情形中 CA 所能容忍的故障数量。

2 模型

2.1 共识作为任务

在 n 个处理器上定义的一个任务 是一个二元关系 $R \subseteq \mathrm{I}^n \times \mathrm{O}^n$，其定义域和值域分别是由元组构成的集合：第一个元组的各项取自输入空间 I，第二个元组的各项（即输出）取自输出空间 O。每个 $(x_1, \ldots, x_n) \in \mathrm{I}^n$（或 $(y_1, \ldots, y_n) \in \mathrm{O}^n$）的含义是，$x_i$（或 y_i）是 p_i 的输入（或输出）值。

对于共识任务，相应的关系 R_C 满足以下条件，其中每个处理器 p_i 都有输入v_i：

(1) 如果某个处理器输出 v，那么必定有某个处理器的输入为 v。
(2) 如果某个处理器输出 v，那么没有处理器输出 $v' \neq v$。

在本文中，我们只关注二元共识——也就是说，输入和输出都是比特。

请注意，上述共识的定义不同于拜占庭环境中的传统共识定义 [16, 18], ，后者不考虑输入，也不对被攻陷/拜占庭参与方的输出作出任何保证。

交互式处理器 是指除本地计算外，还能够通过某个网络向其他处理器发送消息并与之通信的处理器。由 n 个处理器组成的协议 是一组 n交互式处理器规范。[1]

一个模型 M由两个组件组成，即系统模型 和通信模型。系统模型规定处理器的类型和能力，以及它们之间通信的性质。故障（即对手）模型 规定处理器可能遭受的不同故障的类型（例如故障停止、遗漏、拜占庭等）及其组合（例如最多 t 个故障）。

具有相应关系 R_T的任务 T在模型 M中是可解的，如果在 M中存在一个协议，使得当处理器以某个输入元组开始运行时，它们都输出一个输出元组，并且该输出元组通过 R_T与该输入元组相关。

定义 1。 一个由 n个处理器组成、由关系$\mathcal{R}_T \subseteq \mathcal{I}^n \times O^n$ 描述的任务 T，如果存在一个在模型 M 中运行的协议 Π，使得 M中每个处理器 p_i在输入 x_i上运行（其代码） Π时，每个处理器都输出 $y_i \in O$，并且满足$((x_1, \ldots, x_n), (y_1, \ldots, y_n)) \in \mathcal{R}_T$，则称该任务在模型 M 中可解。

提交-采用。 除了共识之外，我们还将使用提交-采用（CA）任务 [13]。CA（也称为分级共识 [20]）在理念上与等级广播 [1, 2, 9] 相似。

其规范如下：
每个处理器都持有一个输入值 v_i。每个处理器 p_j输出一个值，对于某个值 v_j，该输出要么是 $commit(v_j)$，要么是 $adopt(v_j)$；此值等于某个处理器的输入。

(1) 如果所有处理器以相同的值 $v_i = v$ 开始，那么它们都输出 $commit(v)$；(2) 如果某个处理器输出 $commit(v)$，那么所有处理器输出的要么是 $commit(v)$，要么是 $adopt(v)$。

在本文中，我们着重研究二元 CA，其中输入值和输出值 v 都是比特。

在本节余下部分，我们将介绍所讨论的模型。这里考虑的所有模型都具有相同的系统模型组件，但对手模型各不相同。

2.2 系统模型

我们假设有 n 个交互式处理器 $\mathrm{P} = \{p_1, p_2, \ldots\ldots, p_n\}$，也称为参与方。它们可以是交互式计算机，例如交互式图灵机（ITM）。通常，我们假设处理器可以执行多项式长度（关于 n）的计算（并且可以通信多项式长度的消息），不过，即使处理器的计算能力不受限制，我们的否定性结果仍然成立。处理器通过一个完全可靠的通信网络连接，任意两个处理器之间都有一条专用信道，采用 LSP[16, 18] 的方式。这里，可靠是指：如果 p_j（通过连接这两个处理器的专用信道）收到来自 p_i 的消息，那么 p_j 就知道该消息确实由 p_i 发送，或者在 p_i 被攻陷时由对手代表其发送。我们假设协议（及通信）是同步的。具体而言，协议按轮次进行，每一轮中所有处理器都会向其他所有处理器发送一条（可能各不相同的）消息。任一轮次 ρ 发送的所有消息都会在第 $\rho + 1$ 轮开始时送达。

2.3 （MAd-）对手模型

请注意，我们对解决任务的协议所作的定义要求，即使是被攻陷方/处理器也必须输出一个值，并且我们为它们提供与诚实方/未被攻陷方相同的输出保证（例如，在共识的语境下保持一致）[1]我们将注意力限制在非反应式任务上，在这类任务中，处理器仅在开始时接收一次输入，并产生单一本地输出（见下文）。

这类协议足以支持我们的结果；不过，我们可以将此定义扩展到反应式任务。

参与方。这使我们的腐化模型更适合描述这样一种移动对手：它可能在协议执行过程中腐化每个处理器。

为了实现更强的保证，我们考虑在参与方的“网络接口”（例如通信带）上运作的对手，而不是攻陷参与方的内部状态。这使我们能够将生成输出定义为：把输出写入一条特殊的只写、只追加输出带，而该输出带不在对手的权限范围内。下面介绍对手的定义。

我们考虑一个中心化对手，它可能会影响其攻陷的参与方所发送的消息；我们将这种对手称为消息对手，简称*MAd*（对手）。顺便指出，某一轮中的被攻陷的处理器这一概念，只是为了描述性地界定 MAd 对手的能力。更具体地说，MAd 对手可能会截获处理器的发出消息，而非其内部状态。简而言之，每一轮中，每个被攻陷的参与方都会根据前几轮收到的消息，遵循其协议指令来准备当前轮次的消息；根据底层通信模型的隐私假设，MAd 对手可以篡改这些消息。

为尽可能得出最强的结论，我们将在此考虑完全信息模型 [12]，在该模型中，对手可以查看协议中交换的所有消息。我们假设，在满足其约束的前提下，对手是非确定性的，并且可以生成垃圾信息，或生成对算法运行最具破坏性的消息集合。因此，我们不会将对手描述为根据其历史运行的算法，而只会描述在给定腐化模式的情况下，哪些消息无法由非确定性方式生成。例如，对于对手无法访问的密钥（如未被攻陷参与方 p_i 的密钥），数字签名的不可伪造性意味着，对手可能代表 p_i 向协议注入的消息仅限于实际由 p_i生成的消息。

对手的描述依据参与方何时被攻陷——静态、移动或最终静态（移动）——以及腐化类型——遗漏、拜占庭、和认证拜占庭，具体如下文所述。

静态消息对手。 静态 t-MAd 对手是在协议执行过程中最多只能腐化 t 个处理器的对手。[2]

移动消息对手。 移动 t-MAd 对手每轮最多只能腐化 t 个处理器。因此，随着时间推移，所有处理器都可能遭遇消息篡改，尽管发生在不同轮次。

最终静态（移动）消息对手。 这种对手在有限（但协议未知）的 ρ 轮内表现为移动对手，但从第 $\rho + 1$ 轮开始转为静态。更具体地说，最终静态 *t-MAd* 对手会先采用移动 t-MAd 对手策略执行有限轮次，然后选择一个固定集合，并从某个时刻起将腐化限制在该集合内，集合中最多包含 t 个处理器。

2.3.1 论文中考虑的 *MAd*-腐化类型。

发送遗漏故障。 对手仅限于删除消息，但在删除某处理器的部分消息后，并不受限于必须删除该处理器后续轮次中的所有消息。

拜占庭故障。对手可以篡改消息，将其替换为自己任意选择的内容 g.

[2]这包括密码学文献中定义的静态腐化和自适应腐化。

认证拜占庭故障。与拜占庭故障相同，唯一的区别是：如果在第 j 轮，处理器 p_i 未被攻陷，那么在之后的轮次中 $k > j$，没有处理器可以声称在第 j 轮由 p_i 发送了实际上并未发送的消息，但可以假装没有收到消息。更形式化地说，在同步认证拜占庭环境中，我们不失一般性地假设，在协议中，任何消息若在第 ρ 轮由 p_i 发送，其形式为 (p_i, ρ, m)，其中$m \in \{0,1\}^*$是消息内容，p_i和ρ是关联元数据。于是，认证拜占庭对手模型规定，对于任何消息 (p, ρ, m)（其中 $p \in \mathrm{P}$），以及任何参与方 $p' \in \mathrm{P}$，若 $(p'\, \rho' m')$ 是 m的一个子串（或其编码），其中 $\rho' \leq \rho$，则要么 p' 在第 ρ' 轮被攻陷，要么其未被攻陷，并发送了 $(p'\, \rho' m')$。

备注1（不放弃被攻陷方——即使被攻陷方也会产生输出）。人们可能会考虑一种标准拜占庭和遗漏腐化的自然 MAd 类比，同时篡改入站和出站通信。然而，这样定义会使广播和共识的定义变得复杂。尤其是，传统的密码学定义会放弃被攻陷方，也就是说，不对被攻陷方的输出作任何保证。这意味着，这些任务不仅不能被定义为只考虑处理器输入的简单函数，还必须考虑对手；而且，定义还可能放弃一个所有操作都正确执行的参与方，仅仅因为它被“标记”为已被攻陷。相反，在这里，我们希望考察上述基于任务的自然定义下共识任务的可行性，在这种定义中不会区别对待被腐化的处理器。显然，如果对手能够篡改或阻断入站通信，就无法避免被攻陷方不产生输出（例如，$\perp$），甚至根据设置和对手的能力产生错误输出。因此，我们限制 MAd 对手只能篡改出站通信。请注意，该对手仍可能以被攻陷方的出站消息的形式注入消息。

3 跨模型归约

设 M 是上述任意一种模型，在该模型的静态情形中，共识是可解的；在移动情形中，元胞自动机是可解的。

由 [14] 可直接得到一个用于共识的宽容协议：取任意一个能在 M 中解决共识的协议 Π_C，以及任意一个能在 M的移动类比中解决元胞自动机的协议 Π_{CA}，交替运行二者。处理器在 Π_{CA}中提交时输出。尽管如此，请注意，我们模型中的处理器会永远运行（因为共识通常用于实现账本）。因此，我们没有停止的概念。

不过，宽容性只是一个引出话题的旁枝。我们想要证明，共识和元胞自动机是一对孪生问题：对于相同的 M，只要分别具备静止性和移动性，它们就总是可解的。

一个方向可由下文第4节开头（定理3和推论 4）以及以下事实推出：如果某项任务在移动模型中可解，那么它在静态模型中也可解。

因此，我们得到以下定理:

定理 1. 如果元胞自动机在移动模型 M 中可解，那么共识在静态 M 中可解。

将 [14] 应用于上述定理，可得以下推论。

推论2. 如果元胞自动机在移动模型 M 中可解，那么共识在 M 中具有宽容性。

我们希望得到如下定理：如果在 M 中可以实现共识，那么在移动 M中可以实现二元 CA。我们猜想存在一种通用的证明方法，但目前我们按具体情况逐一证明。

4 面向静态对手的提交-采用

作为热身，我们从静态对手开始，即对手在整个协议过程中最多腐化 t 个处理器，且腐化对象始终不变。本节的大多数结果都可以从现有文献中轻易得到。尽管如此，为了内容完整，并便于在下一节中引用这些结果，我们仍将其列出。

为了建立共识与提交-采用之间的联系，我们使用如下简单归约，该归约来自 [10, 20]（而其本身依赖于 [3, 11] 中的思想）。设 CA 是一个用于静态环境下提交-采用的协议，可抵御 t 次腐化。那么，下面这个简单的阶段国王协议（我们称之为 Consensus$_{Stat}$）使我们能够通过 CA 构造共识。

- 令 x_i 为处理器 p_i的输入。每个参与方设置 $temp_i := x_i$
- 对于 $i = 1, \ldots, n$(1) 处理器在输入 $temp_1, \ldots, temp_n$上执行 CA；将 p_i在 CA 中的输出记为 y_i。根据 CA 的定义，对于每个 p_j，存在某个 b_j，使得 $y_j \in \{commit(b_j)\ adopt(b_j)\}$ (2) p_i向每个 p_j发送 b_i，后者将收到的值记为 $b_{i\to j}$。(3) 每个 p_j设置 $temp_j$:=(若 $y_j = commit(b_j)$，则为 b_j；否则

- 每个处理器都输出 $temp_n$

定理 3（[10, 20]）。如果 元胞自动机 在静态拜占庭 MAd 对手模型中解决提交-采用问题，那么 共识$_{Stat}$ 就在相同模型中解决共识问题。

不难验证，上述定理可原样应用于发送遗漏腐化和认证拜占庭环境。为完整起见，我们在以下推论中陈述这一点。

推论 4。如果 元胞自动机 在静态（发送）遗漏或认证拜占庭 MAd 对手模型中解决提交-采用问题，那么共识$_{Stat}$ 就在相应模型中解决共识问题。

此外 ，以下定理可直接由提交-采用到共识的平凡归约得出：

(1) 处理器运行共识；用 y_i 表示处理器 p_i的输出。(2) 每个 p_i 都输出 $commit(y_i)$

定理 5. 如果存在一个协议，能够在静态（发送）遗漏、拜占庭或认证拜占庭MAd 对手模型中解决二元共识，那么就存在一个协议 元胞自动机 ，能够在相应模型中解决二元提交-采纳。

4.1 （发送）遗漏型腐化

对于发送遗漏故障，在可容忍任意数量 $t < n$ 被腐化处理器的情况下，存在相应协议，这可由 [20]推出。这个界显然是紧的。值得注意的是，在我们的模型中——每个参与方都需要输出——类似的正面结果可直接由移动对手的相应界推出（参见定理 9）。根据定理 5 和推论 4 可知，这个界对于元胞自动机也是紧的。

定理 6. 当且仅当 $t < n$.时，存在一个协议可在静态（发送）遗漏 t-MAd 对手模型中解决元胞自动机。

4.2 拜占庭

Lamport 等人 [16, 18] 证明，只有当 $t < n/3$ 的参与方被腐化时，拜占庭共识才有可能实现。随后，Berman 等人 [3]给出了一个达到该界限的高效协议。我们将他们的协议称为 BGP-Consensus。尽管他们对共识的定义并未保证被腐化处理器的输出，但只需额外增加一轮，让每个参与方都将自己在 BGP-Consensus 中的输出发送给所有人，并让每个人输出至少收到 2n/3 个处理器发送的相同值，就能以黑盒方式轻松使用他们的协议来添加这一保证。由于在 BGP-Consensus 中，所有未被攻陷的处理器都会输出相同的值 v，因此所有处理器都会从其中所有处理器处收到 v，而从至多 $t < n/3$ 个参与方处收到 $v' \neq v$，所以它们都会输出 v。根据上述结果以及拜占庭模型中共识与提交-采用的等价性（定理 5 和推论 4），我们得到以下结论：

定理 7（[3]）。当且仅当 $t < n/3$，存在一种协议，可在静态拜占庭 *t-MAd* 对手模型中解决元胞自动机。

4.3 认证拜占庭

在认证拜占庭环境中，Fitzi 证明了腐化数量下界为 $t < n/2$ [10, 命题 3.1]。认证环境下的协议源于这样一个观察：在静态模型中，通过假设数字签名方案（完全）不可伪造，并让每个参与方使用标准的存在性不可伪造数字签名方案为其消息进行数字签名，就能满足我们对认证拜占庭 MAd 对手的限制。[3] 事实上，在这种实现下，对手将无法冒充任何未被攻陷的处理器创建消息。

上述观察表明，将共识归约为广播的经典归约对于 $t < n/2$ 在我们的模型中产生了一个共识协议：让每个参与方使用 Dolev-Strong 广播 [7] 向所有人广播其输入，然后对广播的值取多数。

定理 8（[7, 10]）。当且仅当 $t < n/2$时，在静态认证拜占庭 *t-MAd* 对手模型中存在一个能够求解二元 *CA* 的协议。

5 面向移动对手的提交-采用

5.1 （发送）遗漏型腐化

我们证明，即使在每一轮中，MAd 对手都可以选择任意 $n-1$ 个处理器，丢弃其希望丢弃的该处理器发送的任意消息，并在下一轮更换为另一组 $n-1$ 个处理器，二元 CA 仍然是可求解的。

该算法分两轮进行：

（1）每个处理器向所有处理器发送其输入 （2）每个处理器 p_i：如果存在一个比特 b，使得仅收到 b，则向所有处理器发送 $propose\text{-}commit(b)$。否则发送 *no-commit*（3）每个处理器 p_i：如果对于比特 b，仅收到 $propose\text{-}commit(b)$，则输出 $commit(b)$；否则，如果对于唯一的 b，任意参与方收到 $commit(b)$，则输出 $adopt(b)$；否则输出 $adopt(0)$。

每一轮中，至少有一个处理器会被所有处理器听到。因此，在处理器交换输入的一轮之后，只有当某个处理器只听到 b 时，它才会在下一轮提出提交 b。由于所有处理器都从某个处理器那里听到了相同的比特，因此下一轮中只能有一个比特被提议提交。

[3]确实，我们对 认证拜占庭的定义等价于假设存在一种完美的虚构签名方案。请注意，现实中，这种签名 s
并不存在，因此 在实际实现中，该协议只能以压倒性概率达成共识。

在第二轮中，只收到提议提交 b 的处理器会提交 b；如果收到提议提交 b，则会采用 b。如果某个处理器提交了，那么它就收到了来自至少一个所有处理器都能听到的处理器所提出的提交 b。因此，如果有一个处理器提交，所有处理器都会收到提议提交，进而至少会采用该提议。

定理 9。 对于任意 $t <$，存在一个协议，能够在移动 $MAd\ t$ 对手下解决提交-采用问题 $n.$

5.2 拜占庭

根据定理 7，以及移动对手至少与静态对手同样强这一事实，可以直接得出：提交-采用问题最多能够容忍 $t < n/3$ 个拜占庭移动腐化。下面我们描述一种达到这一界限的构造。我们的提交-采用协议遵循 [10,20], 中分级共识构造的结构，其中使用了一个弱共识原语。

协议 WeakConsensus(P, ® = (x_1, ..., x_n))。

(1) 每个 $p_i \in$P 向每个 p_j发送 x_i； p_j表示向其发送 0（分别为 1）的参与者集合，即 $P_{(0)}j$（分别为 $P_{(1)}j$）。

(2) 各 p_j组设置 y_j := { 0 如果 $|P_j^{(0)}| > 2n/3$，否则; 1 如果 $|P_j^{(1)}| > 2n/3$，否则; “n/v” }

定理 10. 协议 弱共识 针对一个最终静态的移动 $MADt$-对手（$t < n/3$）满足以下性质：（弱一致性）存在某个 $y \in \{0, 1\}$，使得每个 $p_j \in \mathcal{P}$ 组都设置 $y_j \in \{y, \text{“n/v”}\}$。（持久性）如果每个 $p_i \in \mathcal{P}$ 的输入 x 都相同，那么它们都会设置 $y_j := y = x$。（终止性）所有参与方在一轮之后设置其 y_i 值

证明。 （终止性）由于所有参与方都在各自唯一的一轮结束时设置值，因此终止性显然成立 .

（弱一致性）假设玩家 p_i 设置了 $y_i = 0$。这意味着 $y_i = 0$： $|P_i^{(0)}| > 2n/3$。但由于 $|P_i^{(0)}|$ 中被攻陷的参与方少于 $1/3n$，这意味着在协议唯一的一轮中，超过 1/3 的参与方未被攻陷，并且也向其他每个 p_j 发送 0。因此 $|P_{(1)i}| \leq 2n/3$，这意味着没有任何 p_j 会决定取值 1。

（持久性）如果所有未被主动攻陷的参与者的输入都是 0（事先一致同意取 1 的情形可对称地处理），那么每个 p_i： P 至少会从所有这些参与方收到 0，即 $y_i = 0$： $|P_i^{(0)}| > 2n/3$，因此输出 0。□

CA 协议(P, ® = ($x_1, \ldots, x_n$)). (1) 参与者调用弱共识(P, ®)，并令 y_i 表示由 p_i 设置的值（注意， $y_i \in${0, 1, “n/v”}）。(2) 每个参与者 $p_i \in$P 向每个 p_j 发送 y_i。 p_j 将向其发送 0、1 和“n/v” 的参与者集合分别记为 $P(0)\,j$、 $P(1)\,j$。(3) 每个参与者 p_j 设置 b_j： = { 0 （若 1 ，否则.....）。(4) 每个参与者 p_j 输出 o_j： =($commit\ b_j$ $|P_j| > 2n\ 3$( )（若 (z_j) / ( )，否则 $adopt\ b_j$

定理 M 11. 该协议 CA 可在移动拜占庭 t-MAd 对手下解决提交-采用问题，其中 $t < n/3$。

证明。我们需要证明以下性质：

- *(性质 1)* 如果对于某个 $z \in \{0,1\}$，某个处理器 $p_i \in$ P 输出 $commit(b)$，那么每个处理器 $p_j \in$ P 都输出 $o_j \in \{commit(b)\ adopt(b)\}$。
- *(性质 2)* 如果每个 $p_i \in$ P 的输入 x 都相同，那么它们的输出为 $commit(x)$。

以下给出这些性质的证明：

性质 1。假设玩家 p_i 输出 $o_j = commit(0)$（$o_j = commit(1)$ 的情况可对称地处理）。这意味着 p_i 在第二轮从超过 $2n/3$ 个参与方处收到了 0，即 $|P_i^{(0)}| > 2n/3$。由于这些参与方中少于 $n/3$ 个在第 2 轮被攻陷，这意味着 p_i 从第 2 轮未被攻陷的参与方中超过 $n/3$ 个处收到了 0，而这些参与方也因此向其他每个 p_j 发送了 0。因此，在第 2 轮中，每个 p_j 都收到了超过 $n/3$ 次 0，即 $|P_i^{(0)}| > n/3$。此外，由于第 2 轮发送的消息是在第 1 轮设定的消息（即弱共识的结果），参与方在第 1 轮的输出必定属于 {0，“n/v”}，因此第 2 轮中未被攻陷的参与方不可能发送 1；于是，由于每轮至多有 $n/3$ 个被攻陷方，$|P_{(1)_j}| \le n/3 < |P_{(0)_j}|$，且每个参与方都设置 $z_j = 1$。

性质 2。如果所有非主动受损玩家的输入都是 0（预先就 1 达成一致的情况可以对称地处理），那么根据弱共识的持久性，每个人在第 1 轮都会设置 $y_i = 0$，因此在第 2 轮，每个未被攻陷的参与方都会发送 0；所以每个参与方至少有 $2/3n$ 次收到 0，因此输出 $commit(0)$ □

5.3 认证拜占庭

同样，直接由定理 8 以及移动对手至少与静态对手一样强这一事实可知，提交-采用最多可以容忍 $t < n/2$ 次认证拜占庭移动腐化。下面我们将介绍一种达到这一上界的构造。

我们引入一个称为 t-MAd-认证-拜占庭-SM 的任务：每个处理器 p_i 都有一个来自某个域 V 的输入值 $x(i)$，并输出一个向量 $\mathbb{y}_i = (y(1)_i, \ldots, y(n)_i)$，使得每个 $y(\ j)_i \in V \cup \{\perp\}$，并且满足以下条件：

- 存在一个索引集合 $Ind \subseteq [n]$，其中 $|Ind| \ge n - t$，使得对于每个 $j \in Ind$，$y(j)_i = x(j)$ 对所有 $i \in [n]$均成立
- 如果 p_i 且 p_j 分别输出 $y(\ell)_i \neq \perp$ 和 $y(\ell)_j \neq \perp$，那么 $y(\ell)_i = y(\ell)_j$

协议 $t - MAdABSM(\mathrm{P}, \mathbb{x} = (x(1) \ldots, x(n)))$. (1) 每个 p_i 向每个 p_j 发送 $x(i)$，后者将接收到的值记为 $v(i)_j$（$v(i)_j$ 若未接收到任何值，则将其设为默认值，例如 0）。令 $\mathbb{v}_j = (v(1)_j, \ldots, v(n)_j)$。(2) 每个 p_i 向每个 p_j 发送 $\mathbb{v}_i$。p_j 将接收到的向量记为 $\mathbb{v}_{i\to j} = (v(1)_{i\to j}, \ldots, v(n)_{i\to j})$，其中如果接收到 $\mathbb{v}_{i\to j} \notin V^n$，则 $\mathbb{v}_{i\to j} := \perp^n$。(3) 每个 p_j 以及每个 $\ell \in [n]$：如果存在某个 $b \in V$ 和某个集合 $I_i \subseteq [n]$，满足 $|I_i| \ge n - t$，且对所有 $i \in I_i$ 都有 $v(\ell)_{i\to j} = b$，并且对所有 $i \in [n] \setminus I_i$ 都有 $v(\ell)_{i\to j} = \perp$，则设置 $y(\ell)_j = b$；否则设置 $y(\ell)_j = \perp$。输出 $\mathbb{y} = (y(1)_j, \ldots, y(n)_j)$。

定理 12。协议 $t - MAdABSM$ 在移动 t-MAd 认证拜占庭对手模型中，为 $t < n/2$ 解决了 t-MAd-认证拜占庭-SM。

协议 元胞自动机(P, ő$x = (x(1) \ldots, x(n))$). (1) 执行 $t - MAdABSM(P, ®)$；每个处理器 p_j将其输出记为 $®^z j$。 (2) 对每个 p_j：如果存在某个值 $v \neq \perp$，使得 $®^z j$中的多数元素等于 v，则设置 $o(j) := propose.commit(v)$；否则设置 $o(j) := propose.no.commit$。 (3) 执行 $t - MAdABSM(P, ®) = (o(1) \ldots, o(n))$；每个处理器 p_j将其输出记为 $® d_j = (d(1)j, \ldots, d(n)j)$。 (4) 每个 p_j：如果对于某个比特 b，满足 $d(\ell)_j = commit(b)$ 的索引 ℓ数量多于 $n/2$，则输出$commit(b)$；否则，如果对于某个比特 b'：$|commit(b')| > |commit(1 - b')| > 0$，则 $adopt(b')$；否则 $adopt(x(j))$。

定理 1 3. 协议 元胞自动机 在移动 *t-MAd* 认证拜占庭对手模型中，以 $t < n/2$求解二元元胞自动机 .

定理 12 的证明（概要）。考虑处理器 p_i，它在第 1 行（即第一轮）未被攻陷。所有处理器 p_j都会具有 $v(i)_j = x(i)$；因此，由于第二轮（第 2 行）被攻陷的是少数处理器，多数处理器将在第二轮发送 $x(i)$。另一方面，在第二轮被攻陷的处理器无法伪造 $x(i)$，因此它们对 p_i的取值将是 $x(i)$ 或 $\perp$。

现在设 p_i是一个在第一轮被攻陷的处理器。若要输出一个不同于 $\perp$（第3行）的值，处理器 p_j必须从索引 k接收到相同的值多数 $v(i)_k$，且这些索引的数量至少达到多数。这个多数中至多有少数处理器被攻陷，因此，这个相同的值 $v(i)_k$是任何处理器在输出值 $y(i)j$ 时唯一的候选值（而非 $\perp$）；该值对所有 j都相同。

□

定理 13 的证明（概要）。首先，我们证明，如果所有处理器一开始的值相同，那么它们都会提交该值。根据 t-MAd-Authenticated-拜占庭-SM 的定义，所有处理器都会拥有一个集合 I，其中包含 $|I| > n/2$；这些处理器返回的输入值在此情形下都相同。因此，所有处理器都会提议提交该值v。于是，根据第二次使用的 t-MAd-Authenticated-拜占庭-SM 的性质，它们都会针对该值输出$commit(v)$。

为了完成证明，我们说明，如果某个处理器 p_i输出 $commit(v)$，那么所有处理器都会输出 $adopt(v)$。注意，在第2行中只能输出一个 $propose - commit(v)$。这是因为， p_i 的提交提议 v 要求其输出中多数索引上的值为 v。根据 t-MAd-Authenticated- 拜占庭 -SM 的性质，没有其他处理器会在这些多数索引上输出与之冲突的不同值，因此，任何其他处理器都不可能在多数索引上得到另一个值。

现在假设 p_i提交了 v。那么，它从其输出向量的第3行返回，该行的多数值为 $propose.commit(v)$。令此多数值为 m。由于处理器从 t-MAd-Authenticated-拜占庭-SM返回时，其输出项之间不存在冲突，因此很容易看出，如果另一个处理器 p_j在某个项中有 $\perp$，而 p_i在该项中有 $propose.commit(v)$，那么这是因为 p_j 在 t-MAd-Authenticated-拜占庭-SM 的第一轮中被攻陷。因此，对于 p_i具有 $propose.commit(v)$、而 p_j未能获知该值的每个索引，对手都必须付出一次第一轮腐化。假设 p_j有 q个索引，其中 p_i输出 $propose.commit(v)$，并且它输出了 $\perp$，那么它还可以再腐化 $t - q$个索引，使其具有 $propose.commit(\bar{v})$。因此， $propose.commit(\bar{v})$的数量为 $t - q$，而它拥有的 $propose.commit(v)$数量为 $m - q$。由于 $m > t$，它将采用 v。

□

6 结论

我们考察了用静态/移动取代同步/异步的情形，并表明，在 MAd 对手模型下，前者对于 commonmodels 是宽容的，而后者则不然。

我们认为，若按照本文的猜想，在静态/移动的背景下考察 MAd 对手，分布式理论将会变得更加丰富且明晰。证明我们的逐案分析其实是多余的，是一项很有意义的挑战。移动良性遗漏的情形也一样：将二元 CA 改为多值 CA。

发展一种目前尚无实际应用的优美理论，值得吗？这就是所谓的基础研究（BR），而我们仍然支持基础研究。

在 [6], Dolev 和 Gafni 的分析中，静态故障与移动故障混合出现。结合本文的观点重新审视 [] ，会很有意思。

最后，如果能找出其他可以定义宽容性的自然“配对”，将会很有意义。

参考文献

[1] James Aspnes. 一种共享内存共识的模块化方法及其在概率写入模型中的应用。载于 Andréa W. Richa 和 Rachid Guerraoui 编，《第29届 ACM 分布式计算原理年会论文集》，PODC 2010，瑞士苏黎世，2010年7月25—28日，第460–467页。ACM，2010。[2] Michael Ben-Or、Danny Dolev 和 Ezra N. Hoch。简短报告：基于等级广播的简单算法。载于 Nancy A. Lynch 和 Alexander A. Shvartsman 编，《分布式计算：第24届国际研讨会》，DISC 2010，美国马萨诸塞州剑桥，2010年9月13—15日。论文集，《计算机科学讲义》第6343卷，第194–197页。Springer，2010。[3] Piotr Berman、Juan A. Garay 和 Kenneth J. Perry。迈向最优分布式共识（扩展摘要）。载于《第30届计算机科学基础年会》，美国北卡罗来纳州研究三角园，1989年10月30日—11月1日，第410–415页。IEEE 计算机学会，1989。[4] Gabriel Bracha。异步拜占庭一致协议。《信息与计算》，75(2):130–143，1987。[5] Soma Chaudhuri。一致比共识更难：完全异步系统中的集合共识问题。载于 Cynthia Dwork 编，《第九届 ACM 分布式计算原理年会论文集》，加拿大魁北克省魁北克市，1990年8月22—24日，第311–324页。ACM，1990。[6] Danny Dolev 和 Eli Gafni。同步混合消息对手。CoRR，abs/1605.02279，2016。[7] Danny Dolev 和 H. Raymond Strong。用于拜占庭一致的认证算法。《SIAM 计算杂志》，12(4):656–666，1983。[8] Cynthia Dwork、Nancy A. Lynch 和 Larry J. Stockmeyer。部分同步条件下的共识（初步版本）。载于 Tiko Kameda、Jayadev Misra、Joseph G. Peters 和 Nicola Santoro 编，《第三届 ACM 分布式计算原理年会论文集》，加拿大不列颠哥伦比亚省温哥华，1984年8月27—29日，第103–118页。ACM，1984。[9] Paul Feldman 和 Silvio Micali。拜占庭一致的最优算法。载于 Janos Simon 编，《第20届 ACM 计算理论年会论文集》，1988年5月2—4日，美国伊利诺伊州芝加哥，第148–161页。ACM，1988。[10] Matthias Fitzi。拜占庭一致中的广义通信与安全模型。博士论文，苏黎世联邦理工学院，2002。[11] Matthias Fitzi 和 Ueli M. Maurer。可抵御一般对手的高效拜占庭一致。载于 Shay Kutten 编，《分布式计算：第12届国际研讨会》，DISC ’98，希腊安德罗斯岛，1998年9月24—26日，论文集，《计算机科学讲义》第1499卷，第134–148页。Springer，1998。[12] Greg N. Frederickson 和 Nancy A. Lynch。在同步环中选举领导者。《ACM 杂志》，34(1):98–115，1987。[13] Eli Gafni。逐轮故障检测器：统一同步性与异步性（扩展摘要）。载于 Brian A. Coan 和 Yehuda Afek 编，《第十七届 ACM 分布式计算原理年会论文集》，PODC ’98，墨西哥巴亚尔塔港，1998年6月28日—7月2日，第143–152页。ACM，1998。[14] Rachid Guerraoui。宽容算法（初步版本）。载于 Gil Neiger 编，《第十九届 ACM 分布式计算原理年会论文集》，2000年7月16—19日，美国俄勒冈州波特兰，第289–297页。ACM，2000。[15] Prasad Jayanti、Tushar Deepak Chandra 和 Sam Toueg。遗漏故障下优雅降级的代价。《信息处理快报》，71(3-4):167–172，1999。[16] Leslie Lamport、Robert E. Shostak 和 Marshall C. Pease。拜占庭将军问题。《ACM 编程语言与系统汇刊》，4(3):382–401，1982。[17] Rafail Ostrovsky 和 Moti Yung。如何抵御移动病毒攻击（扩展摘要）。载于 Luigi Logrippo 编，《第十届 ACM 分布式计算原理年会论文集》，加拿大魁北克省蒙特利尔，1991年8月19—21日，第51–59页。ACM，1991。[18] Marshall C. Pease、Robert E. Shostak 和 Leslie Lamport。存在故障时达成一致。《ACM 杂志》，27(2):228–234，1980。[19] Nicola Santoro 和 Peter Widmayer。时间并非良药。载于 Burkhard Monien 和 Robert Cori 编，《STACS 89：第六届计算机科学理论方面年会》，联邦德国帕德博恩，1989年2月16—18日，论文集，《计算机科学讲义》第349卷，第

304–313。Springer，1989。[20] Vassilis Zikas、Sarah Hauser 和 Ueli M. Maurer。安全多方计算中的现实故障。载于 Omer Reingold 编，密码学理论，第六届密码学理论会议（*TCC 2009*），美国加利福尼亚州旧金山，*2009*年*3*月*15*日至*17*日。论文集，第5444卷，计算机科学讲义，第274–293页。Springer，2009年。