拜占庭将军问题

作者：LESLIE LAMPORT、ROBERT SHOSTAK 和 MARSHALL PEASE @ 斯坦福国际研究院
译者&校对：闵敏 & 裴奇, Elisa @ EthFans.org

可靠的计算机系统必须具备处理故障组件的能力，以防它们向系统中的其它组件传递冲突信息。这种情况可以抽象地表述成一群拜占庭军队的将军各自带领部队驻扎在一座敌城周围。在只能依靠信使进行通信的情况下，这些将军必须就同一个作战计划达成共识。然而，他们中间可能会出现一个或多个叛将，试图混淆其他人的视听。问题在于如何找到一种算法来确保忠将之间达成共识。经论证，如果仅使用口信交流，当且仅当忠将人数超过 2/3 时，此问题有解；因此，一位叛将可以迷惑两位忠将。如果使用无法伪造的书信交流，无论将军总人数和潜在叛将的人数是多少，这个问题都有解。本文最后讨论了如何应用这些解法构建可靠的计算机系统。

分类和关键词：C.2.4. [**计算机通信网络**]：分布式系统——网络操作系统；D.4.4 [**操作系统**]：通信管理——网络通信；D.4.5 [**操作系统**]：可靠性——容错

通用术语：算法，可靠性

其它关键词：交互一致性

1.引言

可靠的计算机系统必须具备处理一个或多个故障组件的能力。故障组件可能会表现出一类常常被忽视的行为——向系统中的其它组件发送冲突信息。处理这类故障的问题可抽象地表述成拜占庭将军问题。本文主要讨论了这一抽象问题，并在结论中提出了如何用这些解决方案实现可靠的计算机系统。

假设拜占庭军队中有几个分队驻扎在一座敌城周围，每个分队都有一个将军。这些将军之间只能依靠信使进行交流。侦查过敌军之后，他们必须制定一个统一的作战计划。然而，一些将军可能会叛变，企图阻止忠将达成共识。这些将军必须通过一种算法来保证：

A. 所有忠将都达成一致的作战计划。

忠将会遵从算法的一切指令，叛将则会随心所欲。算法必须保证无论叛将怎么捣乱条件 A 都成立。

忠将不仅应该达成共识，还应该就合理的计划达成一致。因此，我们还想要保证：

B. 少数叛将无法致使忠将采纳不良计划。

要将条件 B 形式化很难，因为这需要精确定义何为不良计划，我们也不打算尝试给出该定义，而是把各位将军达成共识的方式作为思考的切入点。每位将军都会侦查敌方情况，并将侦查结果传达给其他将军。假设第 i 位将军传达的消息为 $v(i)$，将军总人数为 n，每位将军通过某种方法将 $v(1),\ldots,v(n)$ 的值结合起来形成一个作战计划。要实现条件 A，所有将军必须采用同一种方法结合信息。要实现情况 B，必须采取一种稳健的方法。例如，如果只需决定是进攻还是撤退，$v(i)$ 则可以代表将军 i 认为哪种选项更好的观点，最终决定可以采用多数票决的方式作出。只有在忠将投出的正反票数几乎相等的情况下，少数叛将才能左右最终决定，在这种情况下，进攻或撤退都不算是不良方案。

虽然这种方式可能不是满足条件 A 和 B 的唯一方法，但这是我们了解的唯一方法。该方法假设各位将军通过某种方法彼此传达 $v(i)$ 值。最直接的方式是让将军 i 通过信使将 $v(i)$ 传达给其他将军。然而，这是行不通的，因为满足条件 A 需要每位忠将获得相同的 $v(1), \ldots, v(n)$ 值，而叛将可能会向不同的将军传递不同的值。若要满足条件 A，必须做到：

1. **每位忠将必须获得相同的信息 $v(1), \ldots, v(n)$。**

条件 1 意味着其它将军不一定要使用直接从将军 i 处获得的 $v(i)$ 值，因为如果叛变的是将军 i，他可能会向不同的将军发送不同的值。此即表明，除非非常谨慎，否则为了满足条件 1 ，还需考虑一种可能性，哪怕将军 i 是忠诚的，其他将军们还是使用了一个不同于将军 i 发送的 $v(i)$ 值（校注：即接收 $v(i)$ 的将军中有叛变）。若要满足条件 B ，必须禁止这种情况发生。例如，在所有忠将发送的值均为 “进攻” 的情况下，不能允许少数叛将诱导忠将在 “撤退”, … , “撤退” 这些值中作出决定。因此，将军 i 还必须遵从下列要求：

2. **如果将军 i 是忠诚的，那么其它忠将必须将他发送的值作为 $v(i)$ 值。**

我们可以将条件 1 改为针对所有将军 i 的条件（不管将军 i 忠诚与否），

1'. 任意两位忠将使用的 $v(i)$ 值是相同的。

条件 1' 和 2 均针对将军 i 发送的值。因此，我们现在仅需考虑一位将军如何将他的值发送给其他将军。这种情况可以描述成一位指挥官向副官下达命令，引出了下述问题。

*拜占庭将军问题。*一位指挥官必须向他的 $n-1$ 位副官发送一个命令，这个命令要满足以下两个条件：

IC1. 所有忠诚的副官都服从同一个命令。

IC2. 如果指挥官是忠诚的，则每位忠诚的副官都会服从他下达的命令。

IC1 和 IC2 称为 *交互一致性* 条件。请注意，如果指挥官是忠诚的，那么满足了 IC2 势必会满足 IC1 。**然而，指挥官不一定是忠诚的。**

若要解决最初的问题，将军 i 在发送 $v(i)$ 值之时使用拜占庭将军问题的解法，发送命令“将 $v(i)$ 作为我的值”，其他将军则充当副官。

2. 无解之解 （反证法）

拜占庭将军问题看似简单，实则难在：如果将军只能发送口信的话，除非忠将人数达 2/3 以上，否则无解。尤其是在仅有三位将军的情况下，一旦出现一位叛将，这个问题就无解。由于口信的内容是完全由发送者控制的，因此叛变的发送者可以传达任意消息。这类消息类似于计算机通常情况下互相发送的那类消息。我们在第 4 节提出的签名书面消息属于另一种情况。

如上所述，如果以口信作为通信方式，一旦三位将军中有一位叛变 ，则该问题无解。为了简化问题，我们只考虑“进攻”和“撤退”两种选择。首先探究图 1 所示场景，即指挥官是忠诚的，并发送了“进攻”的命令，然而副官 2 叛变了，向副官 1 谎报自己收到的是“撤退”的命令。若要满足条件 IC2 ，副官 1 必定要服从进攻的命令。

现在考虑图 2 所示的另一个场景，即指挥官叛变了，并向副官 1 发送了“进攻”的命令，向副官 2 发送了“撤退”的命令。副官 1 不知道谁是叛将，而且没法辨别指挥官实际向副官 2 发送了什么消息。因此，图 1 和图 2 所示场景对副官 1 来说没有区别。如果叛将坚持谎报军令，副官 1 没法辨别是哪种场景，就必须服从“进攻”的命令。因此副官 1 只要从指挥官处收到 “进攻” 的命令，必定会服从。

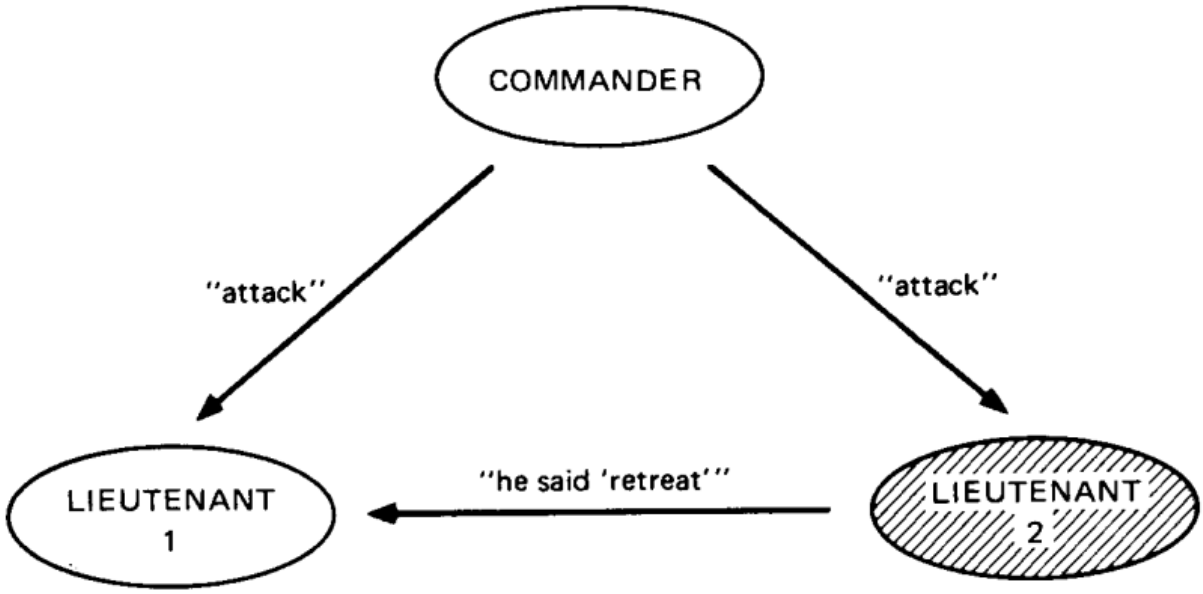

图 1.副官 2 叛变

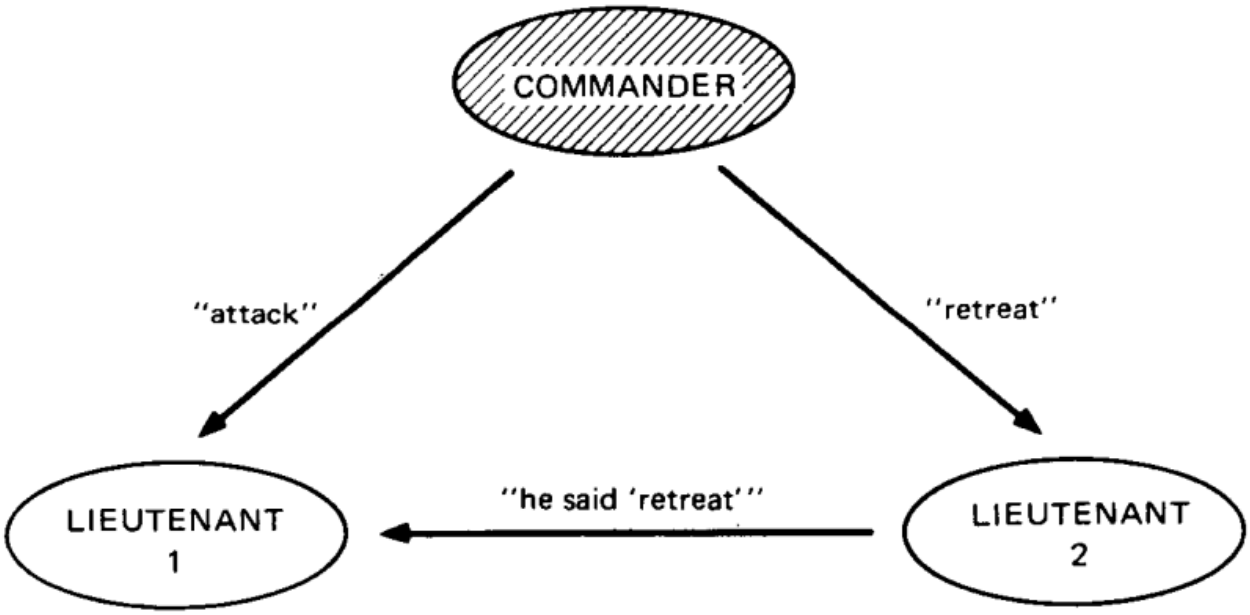

图 2. 指挥官叛变

然而，类似上述论断，如果副官 2 从指挥官处收到的命令是“撤退”，即使副官 1 告诉他指挥官下达的命令是“进攻”，他也会服从“撤退”命令。因此，在图 2 所示场景下，副官 2 必须服从“撤退”的命令，而副官 1 必须服从“进攻”的命令，从而违反了条件 IC1。因此，三位将军中只要有一位叛变，这个问题就无解。

这个论点似乎很有说服力，不过我们强烈建议读者对这种未经严格论证的推理持怀疑态度。虽然所得结论确实是对的，但是我们也见过对无效结论做出的类似的“证明”。我们知道，在计算机或数学领域中，用非形式推理来研究这类算法的出错率是最高的。关于“两忠一叛”问题无解的严密论证，我们建议阅读第 3 节 。

由此可得，如果叛将人数为 m，将军总人数少于 $3m+1$ 时，则该问题无解[1]。该证明过程使用反证法——先假设在将军人数不超过 $3m$ 的情况下该问题有解，并用它来构造我们已知无解的“两忠一叛”问题的解。为避免混淆两种算法，我们将假设有解的问题中的将军称为阿尔巴尼亚将军，构造求解的问题中的将军称为拜占庭将军。因此，通过先假设一个“不超过 $3m$ 位阿尔巴尼亚将军中出现 m 位叛将”的算法，就可以构造出“两忠一叛”的拜占庭将军问题的解。

可将每位拜占庭将军模拟为大约 1/3 的阿尔巴尼亚将军，来得到“两忠一叛”问题的解，即每位拜占庭将军相当于最多 m 位阿尔巴尼亚将军。其中，拜占庭指挥官相当于一位阿尔巴尼亚指挥官加上最多 $m-1$ 位阿尔巴尼亚副官，剩下两位拜占庭副官各相当于最多 m 位阿尔巴尼亚副官。拜占庭将军中仅有一位会叛变，相当于最多有 m 位阿尔巴尼亚将军会叛变。因此，该模拟下假定的解可以保证阿尔巴尼亚将军问题同样满足条件 IC1 和 IC2 。根据 IC1，由一位忠诚的拜占庭副官模拟的所有阿尔巴尼亚副官都会服从同一个命令。很容易可以验证出，阿尔巴尼亚将军问题像拜占庭将军问题一样满足条件 IC1 和 IC2，由此我们已经找到了我们需要的无解之解。

有人可能认为解决拜占庭将军问题的难点在于需要达成某个确切的共识。为证明事实并非如此，我们现在要论证达成近似共识与确切共识的难度相当。假设将军只试图就大致的作战时间，而非确切的作战计划达成共识。更准确地说，我们假设指挥官下达了进攻时间的命令，且以下两个条件需成立：

IC1'. 所有忠诚的副官发动进攻的时间差在 10 分钟之内。

IC2'. 如果指挥官是忠诚的，所有忠诚的副官会按照指挥官命令的时间在 10 分钟内发动进攻。

（假设命令是在进攻前一天下达并处理的，而命令何时送达无关紧要——重要的只有命令中给出的进攻时间。）

与拜占庭将军问题一样，除非忠将人数超过 2/3 ，否则该问题无解。我们的证明需要一个假设，令该问题的“两忠一叛”问题有解，那么我们就能够用这个解构建出一个拜占庭将军问题的“两忠一叛”的解。假设指挥官想要下令“进攻”或“撤退”。根据假定算法，如果指挥官发送的进攻时间为 1:00 ，则代表进攻的命令，如果发送的进攻时间为 2:00 ，则代表撤退的命令。每位副官通过下列步骤获取命令。

(1) 从指挥官处收到进攻时间的命令后，副官采取以下某个步骤：

(a) 如果进攻时间不晚于 1:10 ，进攻。

(b) 如果进攻时间不早于 1:50 ，撤退。

(c) 其它情况则执行步骤（2）。

(2) 询问另一个副官在步骤（1）中做出了什么决定。

(a) 如果另一个副官做出了决定，做出与他相同的决定。

(b) 其它情况则撤退。

由 IC2' 可推知，如果指挥官是忠诚的，忠诚的副官会在步骤（1）得到正确的命令，从而满足 IC2 。如果指挥官是忠诚的，满足了 IC2 势必会满足 IC1，因此只需要证明在假设指挥官是叛将的情况下，仍能满足 IC1 即可。 因为叛将人数最多只有一位，所以可推知两位副官都是忠将。由 IC1' 可推知，如果一位副官在步骤（1）中作出了进攻的决定，那么另一位副官不能在步骤（1）中得出撤退的决定。因此存在两种可能性，即两位副官执行步骤（1）得出了相同的决定，或是至少有一位将决定推迟到了步骤（2）。进入步骤（2）后，他们显然会做出相同的决定，从而满足 IC1 。因此，我们已经为无解的拜占庭将军问题的“两忠一叛”构建出了解法。综上所述，解决“两忠一叛”问题的算法不可能同时满足 IC1' 和 IC2' 。

现在可使用上述将一位将军模拟成 m 位将军的方法来证明：如果叛将人数为 m，将军人数少于 $3m+1$ 时，则该问题无解。论证过程与原始的拜占庭将军问题类似，留待读者自行论证。

3.口信型拜占庭问题之解

经上述证明，在将口信作为通信手段的情况下，如果叛将人数为 m，则将军人数不能少于 $3m+1$ ，否则拜占庭将军问题无解。现在，我们要求解将军人数不少于 $3m+1$ 的拜占庭将军问题。不过，首先要明确定义何为“口信”。每位将军应该通过执行某种算法向其它将军发送消息，且我们假设忠将会正确执行算法。口信的定义具体体现在下列关于将军的通信系统的假设中：

A1. 但凡发送出去的消息都会正确传达给将军。

A2. 消息的接收者知道发送者是谁。

A3. 一旦消息缺失，会被发现。

假设A1 和 A2 能够防止叛将干扰其他两位忠将的通信，因为根据 A1 ，叛将无法干涉忠将间发送的信息；根据 A2 ，叛将无法通过传递虚假信息来混淆两位忠将的视听。A3 可防止叛将通过不发送信息来阻碍决策。关于上述假设条件的实际运用，在第 6 节讨论。

本节及下一节所述算法要求每位将军都能直接向其它将军发送信息。在第 5 节中，我们将讨论无需满足上述要求的算法。

如果叛变的是指挥官，他可能会决定不下达任何命令。由于副官必须遵守某个命令，在这种情况下，他们需要服从某个默认命令。我们将 **RETREAT（撤退）**设定为默认命令。

综上，我们归纳定义了一种 *口信* 算法 $OM(m)$ ，且 m 为非负整数，指挥官据此向 $n-1$ 位副官发送命令。经证明，在将军总数不少于 $3m+1$ 且叛将人数不超过 m 的情况下，可通过 $OM(m)$ 解决拜占庭将军问题。我们发现在描述这个算法时，表述成副官"获取一个值"而非"遵守命令"更方便。

该算法假设了一个函数 $majority$ ，满足：如果大多数 v_i 的值等于 v ，则 $majority(v_1,\ldots,v_{n-1})$ 的值等于 v 。（实际上，该算法假设了一系列这样的函数——对每一个取值 n ，都有一个） $majority(v_1,\ldots,v_{n-1})$ 的值有两种很自然的选择：

1. 如果 v_i 中存在多数值，则取该值，反之则使用默认值**RETREAT**；
2. 假设 v_i 值组成一个有序集合，取其中位数。

以下算法只要求满足上述 $majority$ 函数的性质。

算法 $OM(0)$。

（1）指挥官将他的值发送给每位副官。

（2）每位副官都使用他从指挥官处得到的值，或者在没有收到值的情况下使用默认值 **RETREAT**。

算法 $OM(m), m>0$。

（1）指挥官将他的值发送给每位副官。

（2）对于任意 i ， v_i 表示副官 i 从指挥官处收到的值，如果没有收到值，则使用默认值 **RETREAT**。副官 i 在算法 $OM(m-1)$ 中担任指挥官，将 v_i 值发送给其他 $n-2$ 位副官。

（3）对于任意 i 和 j $(j\neq i)$，用 v_j 表示步骤（2）（使用算法 $OM(m-1)$ ）中副官 i 从副官 j 处收到的值，如果没有收到值，则使用默认值 **RETREAT**。副官 i 使用 $majority(v_1,\ldots,v_{n-1})$ 的值。

为便于读者理解上述算法是如何运作的，我们以 $m=1, n=4$ 为例。图 3 阐明了当指挥官发送的值是 v ，且副官 3 为叛将时，副官 2 收到的信息。在 $OM(1)$ 的步骤（1）中，指挥官向其他所有 3 位副官发送 v 。在步骤（2）中，副官 1 通过简单的算法 $OM(0)$ 向副官 2 发送 v 值。同样在步骤（2）中，叛变的副官 3 向副官 2 发送其它值 x 。在步骤（3）中，副官 2 收到的值分别是 $v_1=v_2=v$ 以及 $v_3=x$，因此他会得出正确的值 $v=majority(v,v,x)$。

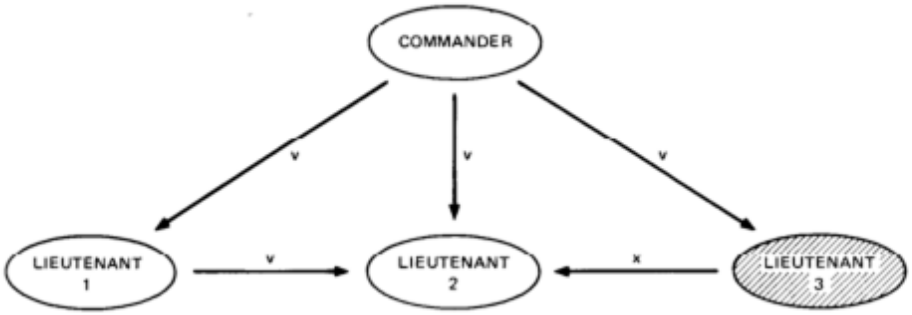

Fig. 3. Algorithm OM(1); Lieutenant 3 a traitor.

图 3. 算法 OM(1)；副官 3 是叛将

接下来看看指挥官叛变的情况。图 4 显示了当指挥官为叛将，分别向三位副官发送三个随机值 x、y 和 z 时，每位副官分别收到的值。每位副官获得的值分别是 $v_1 = x$、$v_2 = y$ 和 $v_3 = z$，因此，不管 x、y 和 z 这三个值是否相等，他们在步骤（3）中获得的值均为 $majority(x, y, z)$。

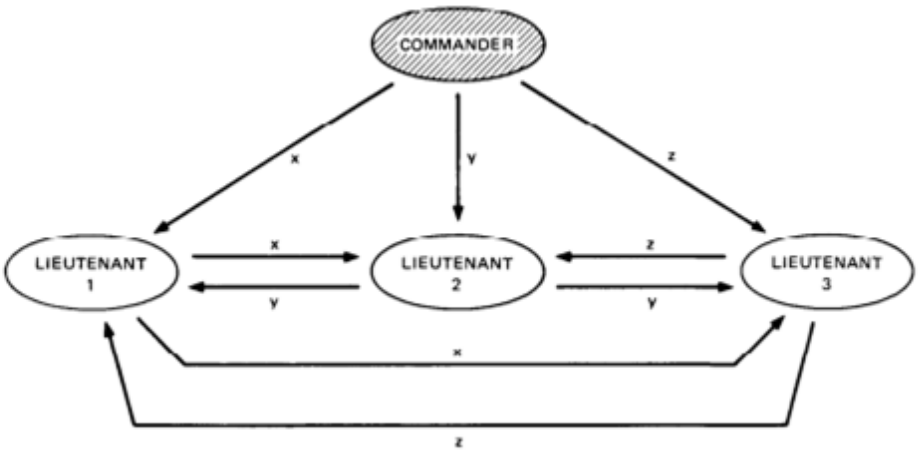

Fig. 4. Algorithm OM(1); the commander a traitor.

图 4. 算法OM(1)；指挥官是叛将

递归算法 $OM(m)$ 会调用算法 $OM(m-1)$ 进行 $n-1$ 次独立运算，每一次 $OM(m-1)$ 运算又会调用算法 $OM(m-2)$ 进行 $n-2$ 次运算，以此类推。此即表明， $m>1$，一位副官会向其他每一位副官发送很多独立信息。必须找到某种方法将这些不同的信息给予区分。读者可以验证：如果每位副官 i 将步骤（2）中发送的 v_i 值前面加上数字 i 作为前缀，这样就可以消除所有歧义。随着递归算法的反复调用，算法 $OM(m-k)$ 将被调用 $(n-1)\dots(n-k)$ 次，发送以 k 个副官序号为前缀的值。

要证明算法 $OM(m)$（m 可取任意值）的正确性，首先要证明以下引理。

引理 1. *对于任意 m 和 k，如果将军总数超过 $2k+m$，叛将人数不超过 k，则算法 $OM(m)$ 满足条件 IC2 。*

论证：该论证对 m 进行归纳推导。IC2 仅仅规定了指挥官是忠将时的情况。根据 A1 可知, 如果指挥官是忠将，简单的算法 $OM(0)$ 显然有效，因此当 $m=0$ 时引理 1 成立。现在我们假设当 $m>0$ 之时，引理 1 对 $m-1$ 成立，并求证其对 m 同样成立。

在步骤（1）中，忠诚的指挥官向所有 $n-1$ 位副官发送一个值 v。在步骤（2）中，每位忠诚的副官运行 $OM(m-1)$ 算法，此时的 n 为 $n-1$ 位将军。根据 $n>2k+m$ 的假设，可推出 $n-1>2k+(m-1)$，由上述假设可推知，每位忠诚的副官都能从忠诚的副官 j 处收到 $v_j=v$。因为叛将人数最多为 k，且 $n-1>2k+(m-1)\geq 2k$，所以 $n-1$ 位副官中多数人是忠诚的。因为 $n-1$ 位副官中多数人收到的值都是 i，所以对于每位忠诚的副官，都有 $v_i=v$，则他们在步骤（3）中可以得到 $majority(v_1,\dots,v_{n-1})=v$，证实了条件 IC2 。

以下定理断言算法 $OM(m)$ 可解决拜占庭将军问题。

定理 1. 对于任意 m，如果将军人数超过 $3m$，且叛将人数不超过 m，算法 $OM(m)$ 满足条件 IC1 和 IC2。

论证：该论证对 m 进行归纳推导。如果叛将人数为 0，算法 $OM(0)$ 显然满足条件 IC1 和 IC2 。因此，当 $m>0$, 我们假设定理 1 对 $OM(m-1)$ 成立，并求证其对 $OM(m)$ 同样成立。

首先考虑指挥官是忠将的情况。使引理 1中的 $k=m$，可知 $OM(m)$ 满足条件 IC2 。如果指挥官是忠诚的，满足了 IC2 势必会满足 IC1 ，因此，我们只需要验证在指挥官叛变的情况下，条件 IC1是否满足即可。

叛将人数不超过 m，而且指挥官也是其中之一，因此至多有 $m-1$ 位副官是叛将。因为将军的人数大于 $3m$，则副官人数大于 $3m-1$，且 $3m-1>3(m-1)$ 。由上述假设可推知，$OM(m-1)$ 满足条件 IC1 和 IC2 。因此，对每个 j，任意两位忠诚的副官在步骤（3）中都能得到相同的 v_j 值。（如果两位副官中有一位是副官 j，可从条件 IC2 推知，否则可从条件 IC1 推知。）因此，任意两位忠诚的副官可以得到以 $v_1,\ldots,v_{n-1}$ 为值的相同向量，由此在步骤（3）中获得相同的 $majority(v_1,\ldots,v_{n-1})$ 值，证实了 IC1 。

4.签名信息型拜占庭问题之解

如图 1 和图 2 的场景所示，拜占庭问题的难点在于叛将会撒谎。如果能够约束叛将撒谎的能力，解决这个问题就会容易很多。一种方式是允许将军发送不可伪造的签名信息。更确切来说，我们在 A1 - A3 的基础上新增了下列假设：

A4

(a) 忠将的签名无法伪造，而且对他的签名消息的内容进行任何更改都会被发现。

(b) 任何人都能验证将军签名的真伪。

请注意我们并未对叛将的签名进行任何假设。也就是说，我们允许叛将之间互相伪造签名，以此允许他们之间相互勾结。

由于我们已经引入了签名消息，之前关于 4 位将军才能容许 1 位叛徒的论点已经站不住脚了 。实际上，“两忠一叛”问题确实有解。现在，我们要给出无论将军人数有多少都能容许 m 位叛将的算法。（如果将军总数少于 $m+2$，这个问题就没有意义。）

在此算法中，指挥官向每位副官发送一个签名命令。之后，每位副官会将自己的签名添加到这个命令上，把命令发送给其他副官，其他副官加上自己的签名后再发送给其他人，以此类推。这意味着，副官必须先收到一个签名消息，复制几份，署上签名，再将这些副本发送出去。这些副本的产生方式并不重要；收到的消息或许是直接影印的，或许是经过签名按需分发的许多相同消息的堆栈。

该算法假设了一个函数 $choice$，用来从一组命令中选出一个命令。我们对这个函数的要求是：

1. 如果集合 V 只包含一个元素 v，则 $choice(V)=v$ 。
2. $choice(\emptyset)$ = **RETREAT**，其中 $\emptyset$ 代表空集。

请注意可以将 $choice(V)$ 定义成取集合 V 的中位数（假设集合中的元素按一定顺序排列）。

在下述算法中，我们用 $x:i$ 表示由将军 i 签署的 x 值。因此，$v:j:i$ 代表先由将军 j 签署，再由将军 i 签署的 v 值。设将军 0 为指挥官。在这个算法中，每位副官 i 都有一个集合 V_i，包括了他目前为止收到的经有效签署的所有命令。（如果指挥官是忠诚的，则该集合内的元素不应该超过一个。）不要将他收到的 *命令* 集合 V_i 与他收到的消息集合混淆。因为同一个命令可能会对应多个不同的消息。

算法 $SM(m)$

初始值 $V_i=\emptyset$

（1）指挥官将自己签署过的值发送给每位副官。

(2) 对任意的 i：

(A) 如果副官 i 从指挥官处收到形式为 $v:0$ 的信息，且尚未收到任何命令，则

(i) 使 V_i 等于 $\{v\}$;

(ii) 将形式为 $v:0:i$ 的信息发送给其它副官。

(B) 如果副官 i 收到形式为 $v:0:j_1:\ldots:j_k$ 的信息，且 v 不包含在集合 V_i 内，则

(i) 将 v 加入集合 V_i；

(ii) 如果 $k<m$，则向除了 $j_i,\ldots,j_k$ 的其他副官发送形式为 $v:0:j_1:\ldots:j_k:i$ 的消息 。

(3) 对任意的 i： 如果副官 i 不再收到更多消息，则遵守命令 $choice(V_i)$ 。

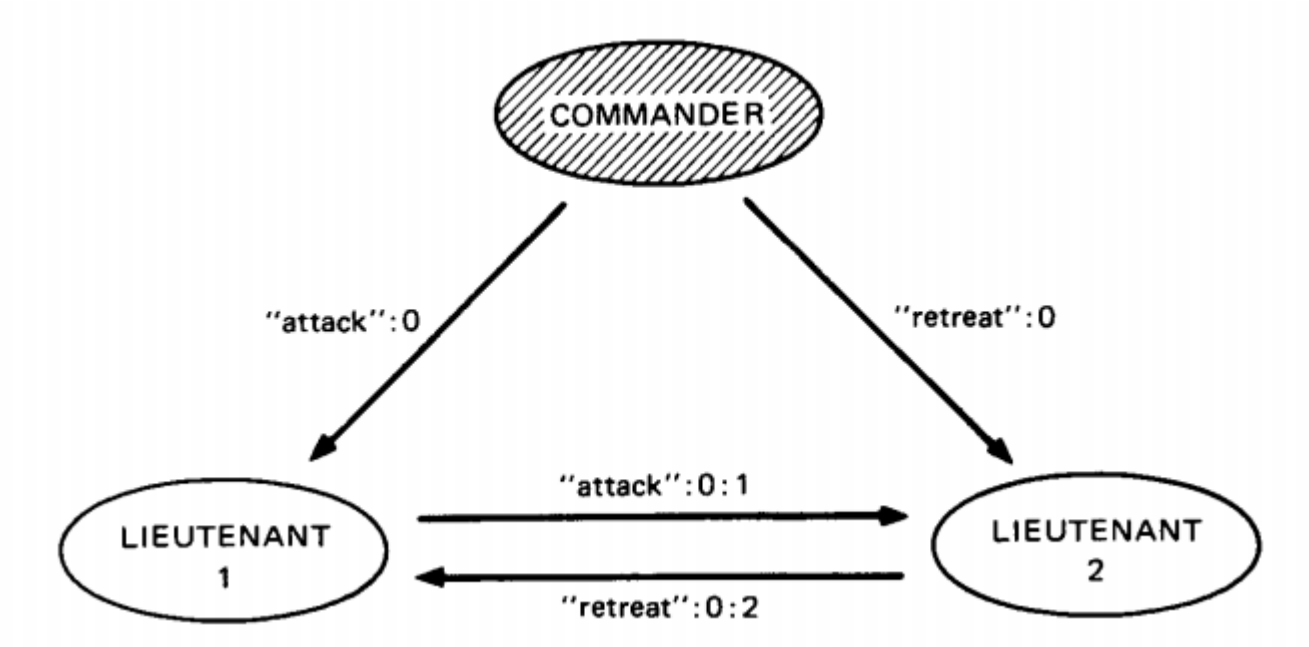

图5. 算法 SM(1)；指挥官是叛将。

请注意在步骤（2）中，副官 i 会忽略任何已经包含在集合 V_i 中命令 v 的消息。

我们尚未规定副官在步骤（3）中如何判定自己不会收到更多消息。通过 k 进行规约推导，很容易就能证明对于任意副官序列 $j_1,\ldots,j_k(k\leq m)$，每位副官在步骤（2）中最多会收到一条形式为 $v:0:j_1,\ldots,j_k$ 的消息。如果要求副官 j_k 必须发送该消息或是发消息表示自己不会发送该消息，很容易就能判断消息将于何时接收完毕。（根据假设 A3，其他副官可以判定叛变的副官 j_k 是否两种消息都没有发。）另一种方法是通过设定时限来判定何时开始无法接收消息。关于设定时限的讨论见第 6 节。

请注意在步骤（2）中，凡是消息内的值形式不对，没有紧跟一连串签名的，副官 i 都会忽略。如果采用将相同的消息打包的方式免去复制信息之需，此即表明消息包内只要没有足够多的、经过有效签名的相同消息，副官就会将其丢弃。（如果由 k 位副官签名，该消息的副本数应为 $(n-k-2)(n-k-3)\ldots(n-m-2)$ 。）

图 5 显示了算法 $SM(1)$ 是如何解决指挥官为叛将的 “两忠一叛” 问题。指挥官向一位副官发送 “进攻” 的命令，向另一位副官发送 “撤退” 的命令。两位副官在步骤（2）中分别收到了命令，得出 V_1 = V_2 = {“attack” , “retreat”} ，因此两方均遵守 $choice$({“attack” , “retreat”}) 的命令。请注意图 5 所示情况与图 2 不同，副官会发现指挥官叛变了，因为他的签名出现在两个不同的命令上，而且根据 A4 可知，这些签名不可能是伪造的。

在算法 $SM(m)$ 中，副官通过签署自己的姓名来确认已收到命令。如果他是第 m 位在命令上签名的副官，消息接收方不会将他的签名转发给其他人，因此签名是多余的。（更确切地说，假设A2 消除了该情况下签名的必要性）特别地，在算法 $SM(1)$ 中，副官无需签署他们的消息。

接下来要证明该算法的正确性。

定理 2. 取任意正整数 m，如果叛将人数不超过 m，则算法 $SM(m)$ 可以解决拜占庭将军问题。

论证：首先证明 IC2 。如果指挥官是忠诚的，他会在步骤（1）中将自己的签名命令 $v:0$ 发送给每位副官。因此，在步骤（2）中，每位忠诚的副官都会收到命令 v（A）。此外，由于叛变的副官无法伪造任何形式为 $v':0$ 的消息，忠诚的副官在步骤（2）中不会收到其他命令（B）。因此，每位忠诚的副官 i 在步骤（2）中得到的集合 V_i 只包含一个命令 v，根据函数 $choice$ 的性质 1 可知，他们都会在步骤（3）中服从这个命令。因此，IC2得到了论证 。

综上，如果指挥官是忠诚的，满足了 IC2 势必会满足 IC1 。因此，要证明 IC1，只需考虑指挥官叛变的情况。如果两位忠诚的副官 i 和 j 在步骤（2）中收到的命令集合 V_i 和 V_j 是相同的，他们在步骤（3）也会服从相同的命令。因此，一旦证明了：如果 i 在步骤（2）中将命令 v 放入 V_i，则 j 肯定会在步骤（2）中将同一个命令 v 放入 V_j，即可证明 ICl 。为此，必须证明 j 收到了包含这个命令且经过有效签署的信息。如果 i 在步骤（2）中收到了命令 v（A），会在步骤（2）中将它发送给 j（A）（ii）；因此 j 会收到这个命令（根据 A1 可知）。如果 i 在步骤（2）中将这个命令添加进了 V_i（B），他收到的第一条消息肯定是形式为 $v:0:j_1:\ldots:j_k$ 的消息。 如果 j 属于 j_r，则根据 A4 可知，j 肯定已经收到了命令 v 。反之，则有两种情况：

1. $k<m$。在这种情况下，i 会向 j 发送形式为 $v:0:j_1,\ldots,j_k:i$ 的消息；因此 j 肯定会收到命令 v 。
2. $k=m$。由于指挥官叛变了，最多有 $m-1$ 位副官是叛将。 因此，副官 $j_1,\ldots,j_k$ 中至少有一位是忠诚的。这位忠诚的副官肯定一收到 v 值就发送给了 j，因此 j 肯定收到了这个值。

至此，论证已经圆满结束。

5. 消失的通信路径

在上文中，我们假设的都是所有将军之间可以直接互传消息的情况。现在抛开这一假设，换成另一种假设，即有物理障碍限制将军互通消息。将这些将军想象成节点，组成了一个简单[2]、有限的无向图 G，其中两个节点之间的弧线代表相应的两位将军可以直接互通消息。算法 $OM(m)$ 和 $SM(m)$ 假定图 G 是完全互联的，现将它们扩展成更一般的图。

要扩展口信算法 $OM(m)$，需要作出以下定义，其中由一条弧线相连的两位将军互称为 *邻居* 。

定义 1：

（a）如果

（i）每个 i_j 都是 i 的邻居，且

（ii）对于任意一个非 i 的节点 k 来说，i_j 与 k 之间都存在一些不经过 i 的路径 $\gamma_{j,k}$，且任意两条不同的路径 $\gamma_{j,k}$ 仅有 k 一个共同节点，

则节点集合 $\{i_1,\ldots,i_p\}$ 称为节点 i 的 *正则邻居集合*（*a regular set of neighbors*）

（b）如果每个节点都有一个包含 p 个不同节点的正则邻居集合，则称图 G 为 *p* 度正则图（ p-regular）。

图 6 属于简单的 3度正则图。图 7 不是3度正则图，因为中心节点的正则邻居集合内包含不止 3 个节点。

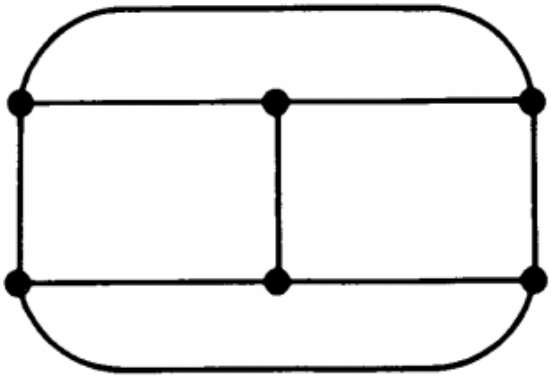

Fig. 6. A 3-regular graph.

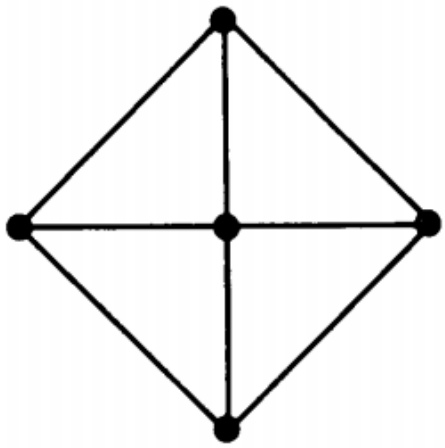

Fig. 7. A graph that is not 3-regular.

我们在算法 $OM(m)$ 的基础上进行扩展，用 3*m* 度正则图 *G* 解决叛将人数为 m 的拜占庭问题。（请注意 3*m* 度正则图 G 必须包含至少 $3m+1$ 个节点。）取任意正整数 m 和 p，图 *G* 是 *p* 度正则图，我们定义了算法 $OM(m,p)$（当图 G不是 *p* 度正则图时，算法 $OM(m,p)$ 是未定义。）该定义对 m 进行规约推导。

算法 $OM(m,p)$

（0）选取指挥官的正则邻居集合 N，其中包含 p 位副官。

（1）指挥官将他的值发送给 N 中的每一位副官。

（2）对于 N 中的任意一位副官 i 来说，从指挥官处收到的值均为 v_i，如果未收到值则使用默认值 **RETREAT**。副官 i 向其它副官 k 发送 v_i 之时应遵循以下规则：

（A）如果 $m=1$，则通过路径 $\gamma_{j,k}$ 发送值 v_i（根据定义 1 中的 (a)(ii) 部分可知，路径 $\gamma_{j,k}$ *必然存在*）。

（B）如果 $m>1$，则充当算法 $OM(m-1,p-1)$ 中的指挥官，这个算法的将军节点图是将原始指挥官从 *G* 中移除后得到的。

（3）对于 N 中的任意一位 k 和 $i(i\neq k)$ 来说，副官 k 在步骤（2）中从副官 i 处收到的值为 v_i，如果没有收到值，则使用默认值 **RETREAT**。副官 k 使用 $majority(v_{i_1},\ldots,v_{i_p})$ 的值，其中 $N=\{i_1,\ldots,i_p\}$。

请注意：将某个节点从 *p* 度正则图 G 中移除之后，图 G 就会变成 $(p-1)$ 度 正则图 。因此，可以在步骤（2）中运用算法 OM(*m* - 1, *p* - 1)（B）。

现在开始论证叛将人数不超过 m 的情况下，$OM(m,3m)$ 是如何解决拜占庭将军问题的。其论证过程与算法 $OM(m)$ 类似，此处不再作详细阐述。先从以下引理 1 的扩展引理开始。

引理 2. *取任意* $m(m>0)$ *和* $p(p\geq 2k+m)$，*如果叛将人数不超过* k，*则算法* $OM(m,p)$ *满足 IC2* 。

论证：如果 $m=1$，请注意副官得到的值是 $majority(v_1,\ldots,v_p)$，其中指挥官发送 v_i 值的路径与其它值的发送路径互不相交。由于叛将人数不超过 k 且 $p\geq 2k+1$，忠诚的副官占了一半以上的路径。因此，如果指挥官是忠诚的，则大部分 v_i 值都与指挥官发送的值相同，因此满足了 IC2 。

现在假设 $m-1,m>1$ 的情况。如果指挥官是忠诚的，则 N 中的 p 位副官都会收到正确的值。由于 $p>2k$，大多数副官是忠诚的，而且根据假设可推知，他们都会向忠诚的副官发送正确的值。因此，每位忠诚的副官收到的值多数都是正确的，因此会在步骤（3）中会得出正确的值。

以下结论可直接证明算法 $OM(m,3m)$ 的正确性。

定理 3. *取任意正整数 $m(m>0)$ 和 $p(p \geq 3m)$，如果叛将人数不超过 m，则算法 $OM(m,p)$ 可以解决拜占庭将军问题。*

论证：根据引理 2 可知，假设 $k=m$，则 $OM(m,p)$ 满足 IC2 。如果指挥官是忠诚的，则满足了 IC2 势必会满足 IC1 ，因此只需要证明在指挥官叛变的情况下， IC1 成立即可。为此，我们要证明每位忠诚的副官在步骤（3）中都会得到同一个 v_i 值集合。如果 $m=1$ ，则可证明这一点，因为所有副官，包括 N 中的副官，都是忠诚的，而且这些路径 $\gamma_{j,k}$ 都不经过指挥官。如果 $m>1$ ，可通过一则简单的归纳推导论证，因为根据 $p \geq 3m$ 可推知 $p-1 \geq 3(m-1)$ 。

扩展后的算法 $OM(m)$ 要求图 *G* 是 3*m* 度正则图，需要强连通性假设。[3] 事实上，如果将军人数只有 $3m+1$（最低人数要求），则 3*m* 度正则意味着全连通性，且算法 $OM(m,3m)$ 等同于算法 $OM(m)$。相比之下，扩展后的算法 $SM(m)$ 根据假设对连通性的要求可能是最低的。首先考虑的问题是，要解决拜占庭将军问题，需要什么程度的连通性。IC2 要求忠诚的副官服从忠诚的指挥官。如果指挥官无法与副官交流，显然不可能实现这一点。尤其是在指挥官发送给副官的每条消息都会经由叛将转发的情况下，无法确保副官收到指挥官的命令。同样地，如果两位副官只能经由叛将进行交流，则无法保证能满足 IC1 。

拜占庭将军问题可解所需要的最弱连通性假设是忠将之间形成的子图是连通的。在这种假设下我们可以证明，算法 $SM(n-2)$ 是一种解法，其中 n 代表的是将军人数——不管叛将的数目。当然，这个算法必须经过修改，以此限制将军只能将消息发送至可以发送之处。更确切来说，在步骤（1）中，指挥官只能将已签署的命令发送给相邻的副官；在步骤（2）（B）中，副官 i 只能将该消息发送至不属于 j_r 的 *相邻* 副官处。

下面先证明一个较为普遍的结论。假设将军节点图 *直径* 的最小值为 d ，此即表明任意两个节点之间的路径最多包含 d 条弧线。

定理 4. *取任意 m 和 d ，如果叛将人数不超过 m，且忠将子图的直径为 d，则算法 $SM(m+d-1)$（根据上述内容修改）可解决拜占庭将军问题。*

论证：该论证与定理 2 的论证非常相似，此处不作详述。若要证明 IC2 ，首先可以观察到的是：根据假设，从忠诚的指挥官到副官 i 的传令路径会经过 $d-1$ 或更少忠诚的副官。这些副官会准确无误地将命令转达至副官 *i* 处。如上所述，假设 A4 会防止叛将伪造命令。

为了证明 IC1 ，我们假设指挥官叛变了，而且必须证明凡是忠诚的副官 i 收到的命令，忠诚的副官 j 肯定也收到了。假设 i 收到了未经 j 签署的命令 $v:0:j_1:\ldots:j_k$ 。如果 $k<m$，则 i 会将它发送给还未收到该命令的相邻副官处，而且会在 $d-1$ 步之内将命令转发给 j 。如果 $k \geq m$ ，则前 m 位签署者中必有一位是忠诚的，并将命令发送至所有相邻的副官处，从而在忠将之间转发，并在 $d-1$ 步之内发送至 j 处。

推论：*如果忠将图是连通的，则 $SM(n-2)$ (根据上述内容修改) 可以解决将军人数为 n 的拜占庭将军问题。*

论证：假设忠将图的直径为 d 。因为连通图的直径小于节点数量，所以忠将人数肯定超过 d ，且叛将人数肯定少于 $n-d$ 。假设 $m=n-d-1$ ，可以根据定理得出上述推论。

定理 4 假设忠将子图是连通的。根据其论证可以很容易推出，即使将军子图不连通，只要叛将人数不超过 m , 则算法 $SM(m+d-1)$ 具备以下两个性质：

1. 如果任意两位忠将之间的路径只经过最多 d 位忠将，则二者遵守同一个命令。
2. 如果指挥官是忠诚的，对于忠诚的副官来说，如果指挥官的传令路径最多只经过 $m+d$ 位忠将，这些副官就会遵守指挥官的命令。

6. 可靠的系统

据我们所知，除了使用本质上可靠的电路元件之外，实现可靠计算机系统的唯一方法是使用几个不同的“处理器”来计算相同的结果，之后再通过多数票决的方式从它们的输出值中得出唯一值。（投票可能是由系统内部执行，也可能是由使用输出值的外部用户执行。）这不论是在计算机系统中使用冗余电路防止单个芯片出现故障来实现可靠性，还是在弹道导弹防御系统中使用冗余的计算站点防止单个站点被核弹摧毁，都是成立的。二者之间唯一的差别在于被复制的“处理器”的大小。

使用多数票决的方式来实现可靠性是基于所有无故障处理器都会产生同一个输出值的假设。只要这些处理器使用的都是同一个输入值，这个假设就成立。然而，任意一个输入数据均来自某个物理元件——例如，可靠计算机中的另一条电路，或是导弹防御系统中的某个雷达站点——而且一个出故障的元件会向不同的处理器发送不同的值。而且，如果是读取变化中的值，即使值来自无故障的输入装置，不同的处理器也会得到不同的值。例如，如果两个处理器在时钟走动的过程中读取时间，二者得到的时间难免会有先后之差。只有在时钟走动过程中同步读取时间才能防止这种情况。

要通过多数票决的方式来创建可靠系统，应该满足以下两个条件：

1. 所有无故障处理器必须使用相同的输入值（因此会产生相同的输出值）。
2. 如果输入装置没有故障，则所有无故障进程将该装置提供的值作为输入值（因此会产生正确的输出值）。

上述就是我们提出的交互一致性条件 IC1 和 IC2 ，其中“指挥官”代表的是产生输入值的装置，“副官”代表的是处理器，“忠诚”即无故障。

人们会倾向于从“硬件”入手绕开这个问题。例如，一种方法是让所有处理器都从统一条电线读取输入值，从而确保它们获得的输入值相同。然而，一个故障的输入装置可能会通过这条电线发送边际信号——一些处理器会将其解释成 0 ，另一些则会解释成 1 。除非让处理器之间互相通信来解决拜占庭将军问题，否则没法保证不同的处理器从可能出故障的输入设备处获得相同的值。

当然，一个故障的输入设备可能会提供无意义的输入值。拜占庭将军问题的解决方案只能保证所有处理器使用相同的输入值。如果这个输入值非常重要，应该存在几个独立的输入设备提供冗余的值。例如，导弹防御系统中应该布置冗余的雷达和处理站点。然而，输入值冗余无法实现可靠性；而且依然有必要确保无故障的处理器使用冗余的数据来产生相同的输出值。

输入设备在未出故障的情况下也有可能提供不同的值，因为这些值是在变化过程中被读取的，在这种情况下，我们依然想要无故障的处理器获得合理的输入值。由此可见，如果函数 $majority$ 和 $choice$ 被看做中位数函数，则二者都具备的一个特性是：无故障处理器获得的值会落在输入装置提供的值的范围内。因此，只要输入装置生成了一系列合理的值，无故障处理器就会获得一个合理的值。

我们已经提出了一些解决方案，不过是就拜占庭将军问题而非计算系统阐述的。现在要探讨的是如何将这些解决方案应用于可靠的计算系统。当然，用处理器来实行拜占庭将军问题的算法完全可以。问题在于如何在满足假设 A1-A3（对于算法 $SM(m)$ 来说要满足假设 A1-A4）的情况下实现消息传递系统。现在依次考虑以下假设。

A1. 该假设规定无故障的处理器发送的每条消息都会被正确传递。在真实系统中，通信线路是会出故障的。对于口信算法 $OM(m)$ 和 $OM(m,p)$ 来说，无法辨别是连接两个处理器的通信线路还是其中一个处理器出了故障。因此，我们只能保证这些算法在故障（不管是处理器还是通信线路故障）数量不超过 m 的情况下有效。（当然，如果同一个处理器对应的几条通信线路出故障的话，相当于这个处理器出了故障。）如果假设出故障的通信线路无法伪造签名信息——从下文可知该假设是非常合理的，则签名信息算法 $SM(m)$ 对通信线路故障并不敏感。更确切地说，即使存在通信线路故障，定理 4 依然有效。通信线路故障堪比线路报废——这会降低处理器节点图的连通性。

A2. 该假设规定对于接受到的任何消息，处理器都可以判别出发送者是谁。出故障的处理器无法模仿未出故障的处理器。实际上，这意味着处理器之间采用的是固定的通信线路，而非经由某个信息交换网络。（如果采用了交换网络，就必须考虑故障的网络节点，而且会再度面临拜占庭将军问题。）请注意，如果有了假设条件 A4，且所有信息都被签署的话，则无需假设条件 A2 ，因为模仿另一个处理器的签名就是在伪造它的消息。

A3. 该假设规定信息缺失会被发现。消息缺失只有超出一定的时间范围才会被发现——换言之，使用某种时限约定。通过设定时限来满足 A3 需要遵循以下两个假设条件：

1. 生成并传播一条消息所需的时间有固定上限。
2. 发送方和接收方的时钟必须同步，不能超出固定的最大误差。

第一个假设条件显然是有必要的，因为接收者必须知道消息需要等待多久。（生成时间指的是处理器在接收到生成消息所必需的输入值之后，需要多长时间才能将消息发送出去。）第二个假设条件的必要性没那么明显。然而，可以证明的是，要解决拜占庭将军问题，必须满足这个或类似的假设条件。更确切地说，假设算法只允许将军在以下情况采取行动：

1. 在某个固定的初始时间（对所有将军都一样）。
2. 收到一条消息之时。
3. 随机选取的一段时间过去之后。（即，一位将军可以将计时器设定成一个随机值，并且在到点时采取行动。）

（由此能设想出的最普遍的一类算法是不允许构建同步时钟的。）可以证明的是，如果消息可以快速地随意传播，即使消息传递时延设有上限，这类算法也无法解决拜占庭将军问题。即使限制叛将只能通过不发送消息来造成干扰，也不可能找到解决方案。关于这个结论的论证超出了本文的讨论范围。要注意的是设定消息传递延时的上下限允许处理器通过往返发送信息来实现时钟的功能。

上述两个假设条件能够降低发现未发送信息的难度。假设消息生成和传输的最大延时为 μ ，且无故障的处理器之间的时钟误差在任意时刻均不超过 τ 。如果一个无故障处理器根据自己的时钟于时间 T 开始生成消息，消息送达之时接收方的时钟上显示的时间应为 $T+\mu+\tau$ 。因此，如果接收者没有按时收到消息，则认定这条消息并未发送。（如果消息送达迟了，发送者必定出了故障，因此算法的正确与否不取决于被发送的信息。）让输入处理器在固定时间发送值，可以计算出接收消息的处理器根据自己的时钟必须等待多久。例如，在算法 $SM(m)$ 中，为使任意一条消息经历 k 次签名，处理器必须等到 $T_0+k(\mu+\tau)$ 的单位时间，其中 T_0 指的是指挥官（根据自己的时钟）开始执行算法的时间。

时钟频率或多或少都有差异，因此无论多么精确地同步所有处理器的初始时间，最后都会产生时差，除非定期进行再同步。因此，问题在于如何将所有处理器的时钟同步保持在某个固定的误差范围内，即使是在一些处理器出故障的情况下。这个问题的难度本身不亚于拜占庭将军问题。时钟同步问题的解决方法与拜占庭将军问题的解决方法联系紧密。我们会在后续的论文中进行论述。

A4. 该假设规定了无故障的处理器的签名不能被伪造。签名指的是数据项 M 中由处理器 i 生成的一条冗余消息 $S_i(M)$ 。由 i 签署的信息包括一个配对 $(M, S_i(M))$ 。要满足 A4 的 (a) 和 (b) 部分，函数 S_i 必须具备以下两个性质：

(a) 如果处理器 i 未出故障，则所有出故障的处理器都不能生成 $S_i(M)$ 。

(b) 给定 M 和 X ，所有处理器都能判定 X 是否等于 $S_i(M)$ 。

性质 (a) 是无法被保证的，因为 $S_i(M)$ 只是一个数据项，而一个出故障的处理器可以生成任意数据项。然而，算法违反性质 (a) 的可能性是可以如愿降低的，从而增强这个系统的可靠性。如何实现这一点取决于预期会出现哪类故障。以下是我们感兴趣的两种情况：

1.*随机故障* 。如果将 $S_i(M)$ 变成一个适当的"随机化"函数，处理器出现随机故障时生成正确签名的概率与通过一个随机选择程序生成正确签名的概率基本相等——是可能生成的签名数量的倒数。下面来介绍一种方法。假设消息都被编码成小于 P 的正整数，其中 P 是 2 的幂次方。假设 $S_i(M)$ 与 $M*K_i$ 关于模 P 同余，其中 K_i 是随机选择的小于 P 的奇数。假设 K_i^{-1} 是唯一一个小于 P 且满足 $K_i * K_i^{-1} \equiv 1 \bmod P$ （$K_i * K_i^{-1}$ 与 1 关于模 P 同余）的数，可以通过测试 $M \equiv X * K_i^{-1} \bmod P$ （M 与 $X * K_i^{-1}$ 关于模 P 同余）来判定 X 是否等于 $S_i(M)$ 。如果另一个处理器的内存中没有 K_i，它为一个（非零的）消息 M 生成正确签名 $M*K_i$ 的概率应为 $1/P$ ：与通过随机选择程序生成正确签名的概率相等。（要注意的是，如果处理器可以通过某个简单的程序得到 K_i ，则出故障的处理器 j 有更大的概率会在计算 $S_j(M)$ 之时用 K_i 代替 K_j 来伪造 i 的签名。）

2.*恶意操控*。如果一个处理器受恶意操控的影响出了故障——例如，这个处理器本身是完好的，却受到一个打算扰乱系统的人操控——构建签名函数 S_i 就成了一个密码学问题。建议读者阅读 [1] 和 [4] 中关于如何解决该问题的讨论。

要注意的是，在处理器已经见过 $S_i(M)$ 签名的前提下，生成该签名是很容易的。因此，重点在于确保相同的消息不被签署两次。此即表明，在反复使用 $SM(m)$ 分发一系列值的时候，应该为每个值加上序列号来确保其唯一性。

7.结语

我们已经提出了拜占庭将军问题在不同假设情况之下的几种解法，并且展示了如何通过他们来实现可靠的计算机系统。这些解法需要很高的时间和消息成本。算法 $OM(m)$ 和 $SM(m)$ 都需要长达 $m+1$ 的消息路径。换言之，每位副官必须等指挥官先生成消息，再通过其他 m 位副官转达。Fischer 和 Lynch 已经证明了所有能应对 m 位叛将的解法都是如此，因此我们提出的解法在这个方面是最优解。对于不具有完全全连通性的节点图来说，我们提出的算法要求消息路径长达 $m+d$，其中 d 代表的是忠将子图的直径。我们认为这可能也是最优解。

算法 $OM(m)$ 和 $SM(m)$ 需要发送多达 $(n-1)(n-2)\ldots(n-m-1)$ 条消息。可以通过合并消息来减少单条信息的发送数量。减少消息的转发量也是有可能的，不过尚未经过详细研究。但是，我们认为依然需要传送大量消息。

存在随机故障的情况下如何实现可靠性是一大难题，而且其解法本质上成本高昂。降低成本的唯一方法是假设可能出现的故障类型。例如，通常的一种假设是计算机可能响应不了，但是绝对不会返回错误响应。然而，如果对可靠性的要求非常高，就不能通过假设法来节省拜占庭将军问题的成本了。

参考文献

1. DIFFIE, W., AND HELLMAN, M.E. New directions in cryptography. IEEE Trans. Inf. Theory IT-22 (Nov. 1976), 644-654.
2. DOLEV, D. The Byzantine generals strike again. J. Algorithms 3, 1 (Jan. 1982).
3. PEASE, M., SHOSTAK, R., AND LAMPORT, L. Reaching agreement in the presence of faults. J. ACM 27, 2 (Apr. 1980), 228-234.
4. RIVEST, R.L., SHAMIR, A., AND ADLEMAN, L. A method for obtaining digital signatures and public-key cryptosystems. Cornrnun. ACM 21, 2 (Feb. 1978), 120-126.

1. 更准确地说是将军人数在 3 人或以上时，则该问题无解，因为在将军人数仅有 2 人的情况下，这个问题无意义。↩

2. 简单图是指任意两个节点间最多只有一条弧相连，且每条弧都连接两个不同的节点的图。↩

3. Dolev [2] 最近提出的一种算法只需弱连通性。↩