原文：《SolutionstoDelayAttacksonRollups》作者：OffchainLabs联合创始人Edfelten

编译：隔夜粥，DeFi之道。

rollup协议的设计者面临的一个微妙问题是如何应对延迟攻击。在本文中，我将讨论它们是什么以及Arbitrum如何防止它们。，从而带来一些激动人心的新发展。

延迟攻击是试图阻止汇总协议取得进展的恶意行为。他们不会攻击协议的安全性，也就是说，他们不会试图强行确认不正确的结果。相反的延迟攻击通过试图阻止或延迟任何结果的确认来攻击协议的活性。

这些问题可能比较微妙。老实说，协议设计者通常不会&#039；我不想谈论延迟攻击，但是每个第2层系统，不管是&#039；s最优卷积、ZK卷积等等，都需要处理延迟和协议进度的问题。

本文深入探讨了延迟攻击的问题。并讨论了在各种版本的Arbitrumrollup协议中如何处理这个问题。

什么是延迟攻击？

在延迟攻击中，恶意方(或一组恶意方)在汇总协议内采取行动。，遵循旨在防止或延迟向L1连锁店返回确认结果的政策。

这不同于拒绝服务(DoS)攻击，在拒绝服务攻击中，攻击者试图阻止协议中的任何操作。相反，在延迟攻击中，动作会继续发生。然而，攻击者&#039；美国的行为将阻碍或推迟结果的确认(即推迟从L1提取资产)，并迫使诚实的核查人员烧煤气。

任何看似合理的汇总协议都需要参与者承诺。因此，拖延攻击者必然会失去一项或多项质权。我们在这里假设攻击者愿意在一定限度内牺牲质押权，以追求他们的攻击。

我们还将保守地假设攻击者在绕过交易方面具有优势。因此，每当攻击者与诚实的一方在链式交易中争夺优先权时，攻击者总是会赢。

最后，我们假设攻击者可以检查对底层L1区块链的访问，以排除汇总事务，但仅限于有限的一段时间。我们称之为&quot挑战期&quot。特别是，攻击者可以启用和禁用概念性的&quot审查模式&quot。启用审核模式后，攻击者可以完全控制哪些交易可以到达L1。但是，攻击者只能在质询期间启用审核模式。。(我们假设任何一

组审查模式周期加起来超过一个挑战周期，这将触发L1社区的社会反应以防止审查尝试。）

评估协议抗延迟攻击

在评估协议时，我们可以提出六个问题：

此协议是否有防欺诈机制？(否则，延迟攻击没有实际意义，因为参与者不能延迟任何结果的确认，即使是欺诈结果。)是否有集中的操作员或认证人员可以通过简单地停止或扣留数据来停止进度？？如果是这样，那方可能会造成无限期的延误。协议是否为没有信任的最终进展提供了保障？换句话说，不管攻击者做了什么，一个诚实的参与者能强迫最后的进展吗？如果协议保证一个没有信任的过程攻击者能够造成的延迟上限是多少？袭击者怎么样了&#039；成本与延误成正比？如何衡量诚实回应的总成本？

确定这些标准后，让&#039；评估Arbitrumrollup协议的两个版本历史。

协议一：学术论文协议

2018年Arbitrum学术论文？一般使用以下协议(忽略不相关的一致模式)。任何出质人都可以坚持提议的结果。我们称之为断言。在一个时间窗口内，其他宣誓者的任何子集都可以挑战该断言，并且声明者必须向每个挑战者捍卫他的断言，一次一个挑战者。在每个挑战结束时，失败的一方将失去他们的质押权利。

(请注意，需要允许多个质权人对该主张提出异议，并给每个异议人推翻该主张的机会。这是必要的，因为恶意方可能会故意输掉挑战应该&quot赢了。给每个挑战者一个单独的挑战。它可以确保诚实的挑战者能够击败不正确的断言，无论有多少恶意方故意输掉挑战。）

如果没有挑战，或者断言者赢得了所有挑战，那么断言将被确认，协议将继续。但是如果断言者输掉了任何挑战，其断言将被拒绝，并且协议状态将回滚到做出断言之前的状态。

评估

协议有有效的欺诈证明，但不保证有进展，因为恶意参与者可以无休止地做出不正确的断言，每次都牺牲质押权利。，但是会导致无休止的循环做出和拒绝同一个断言，导致不断的回滚和缺乏进展。

协议2:分叉和切割

当前的Arbitrum协议(自2020年以来部署在Arbitrum的每个版本上)通过引入分支改进了以前的协议。这个想法是允许多个质权人作出相互竞争的主张。并将相互竞争的断言视为分叉的链。然后一系列的挑战让分支互相对抗，最后所有的分支都被砍掉，只留下一个分支，这样剩下的一个就可以确认了。

具体方式如下。。区块链中的每个汇总块跟踪其第一个子块(即后续块)由出质器创建时的时间戳。其他出质器可以创建额外的子块。每个子断言都隐含地声明它所有的旧块都是不正确的。

创设该主张的质权人需要质押，其他质权人也可以选择质押。

如果两个出质人赌姐妹&#039；断言，并且两者都不在挑战中，则可以在两个宣誓者之间发起挑战。两个姐妹断言中较早的宣誓者正在捍卫姐姐断言的正确性，而另一个宣誓者正在挑战其正确性。挑战失败的质押人将失去其质押权利，并从质押集中删除。协议包括行动的最后期限。第一创建父断言的子断言的最后期限是在创建第一个子断言之后的一个挑战期。其次，保证断言的最后期限是在创建断言之后的挑战期。

如果保证断言的最后期限已过，，且无出质人对该主张进行质押，该主张将被删除。被删除断言的任何后代、孙辈或其他后代也被删除。

如果一个断言早于一个挑战期，并且没有未修剪的姐妹断言，则可以确认。，从而代表了协议的进步。

评估

该协议具有有效的欺诈证明，任何集中的操作员或证明者都不能停止该过程，因此任何参与者都可以推进该过程。为了推进这一进程一个诚实的宣誓者可以发布一个正确的子断言(如果它没有&#039；t已经存在)。在此之后，在截止日期之前将经过一段有限的时间，以确保不会再创建更多的姐妹断言，并且不会再有出质者可以在现有的姐妹断言上下注。从那时起诚实的宣誓者将参加一系列挑战，以一次性击败并清除错误的宣誓者(如果有多个诚实的宣誓者，他们可以同时击败他们的对手)。一旦移除了所有这些参与者，就可以确认正确的子断言。

对该协议最有效的延迟攻击是让一个恶意出质者保证一个不正确的姐妹断言，并让N-1个恶意出质者保证正确的姐妹断言。不管有多少诚实的宣誓者赌上了正确的姐妹断言。攻击者可以在诚实方之前安排不正确的宣誓者挑战他们的盟友(悲观地说，

这假设攻击者总是可以在诚实方之前进入交易)。盟友会故意放弃挑战，尽可能拖延时间(由于拖延规则每个盟友可能需要一个或两个挑战期)。只有在所有N-1个盟友都牺牲了自己之后，才会要求押错誓的一方挑战诚实方，诚实方获胜并最终消灭押错誓的一方。

这次攻击实现了大约n个挑战期的延迟，攻击者付出了n个质押权的代价。

诚实策略对这种攻击的成本与诚实方的数量成线性关系，因为每个诚实方需要在承诺期限之前承诺。

总结：

确保流程攻击的代价与造成的延迟成线性关系，诚实方的代价与诚实方的数量成线性关系。权限验证

目前部署在Arbitrum上的协议2面临的延迟攻击问题。这就是为什么我们选择暂时将验证者的角色限制在一组许可方，而不是使验证完全免于许可。实现免许可证验证的最后一步，需要一个能最大程度抵抗延迟攻击的协议版本。你可能已经猜到有这样一个协议版本。而我们目前正在努力实现。

即将到来的协议3

这篇文章已经很长了，所以我赢了&#039；t提供新协议的详细描述，这将在后面的文章中介绍。

简单来说，Arbitrum协议的下一个版本对断言和挑战的工作方式做了一些微小的改变，但(在我看来)它做了优雅的改变，这样最糟糕的延迟攻击只能导致挑战期的延迟(无论攻击者愿意损失多少质押权)。

这是基于Arbitrum研究团队的技术突破，它使&quot全体反对全体&quot挑战可行且高效。这允许诚实的宣誓者有效地击败一大群发布恶意分支断言的攻击者。

新协议的规范已经完全确定，目前正在实施。当然，我们赢了&#039；在代码被彻底测试和审计之前，不要在主网络上发布它。

在以后的文章中，我将深入讨论这个新协议。