**2022年全国大学生信息安全竞赛**

**作品报告**

**作品名称：内容加密还不够：基于SGX的匿名通信网络系统**

**电子邮箱：**

**提交日期：2022年8月12日**

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考，作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目录

**[摘要 4](#_Toc29937)**

**[第一章 作品概述 5](#_Toc5120)**

**[1.1. 背景分析 5](#_Toc2334)**

**[1.2. 相关工作 6](#_Toc25300)**

**[1.3. 遇到的困难 11](#_Toc23223)**

**[1.4. 特色分析 12](#_Toc4973)**

**[1.5. 应用前景分析 14](#_Toc14808)**

**[第二章 作品设计与实现 16](#_Toc32734)**

**[2.1. 系统方案概述 16](#_Toc16682)**

**[2.2. 系统实现细节 17](#_Toc17822)**

**[第三章 作品测试与分析 21](#_Toc6126)**

**[3.1. 安全性说明 21](#_Toc6119)**

**[3.2. 延迟测试方案 21](#_Toc12654)**

**[3.3. 测试环境 22](#_Toc24049)**

**[3.4. 测试设备 22](#_Toc23761)**

**[3.5. 测试数据 23](#_Toc17387)**

**[3.6. 结果分析 23](#_Toc20639)**

**[3.7运行流程 24](#_Toc23786)**

**[第四章 创新性与实用性说明 26](#_Toc30599)**

**[4.1. 创新性 26](#_Toc9508)**

**[4.2. 实用性 29](#_Toc12317)**

**[第五章 总结 31](#_Toc28466)**

**[5.1. 项目总结 31](#_Toc32675)**

**[5.2. 心得体会 32](#_Toc23598)**

**[5.3. 未来工作 33](#_Toc30759)**

**[参考文献 35](#_Toc15840)**

# 摘要

通信是我们日常生活中必不可少的一部分，在通信过程中，个人隐私的保护渐渐成为了一个热点问题。然而，现有的通信加密技术往往只关注于怎么对通信内容进行加密，却忽略了通信双方之间的连接关系也会泄露通信双方的个人隐私。比如说A经常和心脏病医生B通信联系，则我们可以推断A患有心脏病；司令员在多兵种作战时和海军部队以及坦克部队联络过，则敌人可以推断出主要的出战兵种是这两种。可以看到，尽管我们没有透露出通信的具体内容，但知道这样的通信关系就已经会导致严重的隐私泄露。我们把这种谁和谁在通信的通信关系称为通信的元隐私数据。

近年来，通信元隐私数据保护备受关注，学术界也在各大顶级会议上发表了多篇有关可保护元隐私数据的匿名通信网络的工作。但是这些工作都有着一些不足，一部分基于差分隐私的工作可以支持大量用户在秒级延迟内实时通信，但是安全性还不够强；另一部分达到可证明安全性的工作尽管足够安全但却通信延迟太高或者无法支持大量用户。现在急需一种既能够达到可证明安全性又能够在秒级延迟内支持大量用户的匿名通信网络来满足人们日益增长的隐私保护需求。在这种背景下，我们的元守匿名通信网络应运而生。

我们首次基于CPU指令扩展——SGX创造的可信执行环境设计元守匿名通信网络，减少了传统方法在服务器端验证洗牌的时间开销，兼顾通信的安全性与效率，在1台服务器上，对于6万用户，通信延迟仅为1.45秒。而此前最好的工作Addra[15]在与我们相同的实验条件下，通信延迟为84.529秒。我们设计的匿名通信网络系统主要贡献有（1）安全性强。在所有服务器都向攻击者妥协的条件下仍不会泄露通信双方之间的联系（2）通信延迟低。比同等安全假设的工作都要更快速（3）可扩展到大量用户使用

我们的匿名通信网络应用场景广泛，可应用于军方指挥系统，医患交流系统，匿名举报系统等多种需要强隐私保护的领域。随着大数据时代的到来，我们相信元隐私数据的保护将会越来越受到重视。我们期待在不久的将来，我们的通信系统融入每个人的日常通信，做您元数据的忠实守卫者。

# 作品概述

## 背景分析

大数据时代，用户逐渐注重保护个人隐私，通信隐私就是其中重要的一方面。但是目前人们大多把保护的目光集中在通信内容上，但却遗漏了同样重要的通信元数据[3]，只有做到保护通信元数据，才能真正实现用户通信隐私的保护。

目前在法律层面，我国对于通信状态是否作为通信秘密的保护范围的定义仍较为模糊，对于通信元数据尚未有一个明确的定义；欧盟在“隐私和电子通信条例”中对电子通信元数据进行了定义：电子通信元数据是指在电子通信网络中传输、分发、交换的电子通信内容；包括追踪、确定通信来源和目的的数据，电信服务中设备的位置数据，以及通信的日期、时间、时常和类型等的数据。本文所提到的元数据与该定义相同。

通信元数据的保护看似无关紧要，但我们可以用一个例子来证明通信元数据泄露不比通信内容泄露危害小，想象一下，公司雇佣两个员工对一位销售代表进行监控：员工1查阅了代表的电子邮件、通信历史和电话记录等，而员工2则记录代表的外出情况、客户拜访地点、出差行程等。虽然员工1没有查看邮件和通信内容，但他们获得了代表的通信元数据。同时，员工2对代表的活动进行了监视，得到了所有不涉及通信内容的行为信息。掌握了这些通信元数据和行为信息，公司可以轻松了解代表的工作流程和客户沟通情况。

从中我们可以看出通信元数据泄露的危害并不逊色于通信内容泄露。正如美国国家安全局的前法律总顾问斯图尔特·贝克所言，“元数据绝对能告诉你关于某人生活的一切。如果你有足够的元数据，你并不真的需要内容。”2014 年，前美国国家安全局和中央情报局局长迈克尔海登评论说：“我们根据元数据杀人。”

再举个例子，通信内容相当于传统邮政通信的信件内容，而通信元数据相当于信件上面的地址、邮编等相关信息。当知道了一个人的邮编、地址等相关信息，那么想要窥探他的生活就变得轻而易举。相较于数据，元数据更具有启发性、揭示性与信息解读可能性。

遗憾的是，在当前整个通信网络中，对于通信元数据的保护还不够健全。从技术层面来看，大多数通信提供商往往会保留通信元数据，对于许多用户而言，这些数据都是高度敏感的，尤其是一些需要隐藏消息来源、保护自身安全的记者、热心公民等。指望通信提供商删除通信元数据并不足以解决问题，因为攻击者仍然可以通过网络监控实时了解通信的情况。在当今高度互连的网络环境下，保护通信元数据的重要性已经迫在眉睫。

为了解决此类问题，我们提出了名为“元守”的匿名通信网络。该网络可以在提供延迟较低的通信的同时保护通信元数据，从而为艰难的元数据隐私保护之路上迈出了新的一步。

## 相关工作

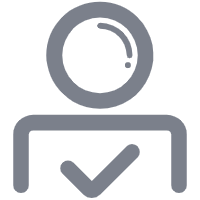
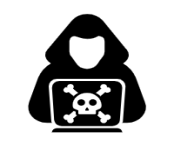
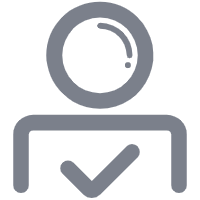
匿名通信指采取一定的措施隐蔽通信流中的通信关系，使窃听者难以获取或推知通信双方的关系及内容。匿名通信的目的就是隐蔽通信双方的身份或通信关系，保护网络用户的个人通信隐私。

如今的匿名通信系统有很多，但它们的性能和应用场景大有不同。我们可以将其分为两种类别，一类是安全性能高的匿名通信网络系统，这类通信系统的突出优点在于，当攻击者有较强的攻击能力，比如攻击者可以获得服务器的权限、可以观察全局网络情况时，此类通信系统仍然能保证通信双方的元隐私数据不被窃取或推理得出，但是它们往往需要较大的带宽和较长时间的通信延迟并且不能给大量用户提供服务。这类匿名通信网络系统适用于有高安全要求的小群体，群体人数不宜过大。另一类是开销成本小的匿名通信网络系统，这类通信系统能够支持大量用户并且通信延迟低。但是它们的安全性在较强攻击的情况下无法保证，当攻击者进行监控全局网络，通信量统计分析时，总会有方法能够得到或推理出通信双方的元隐私数据，比如知道谁和谁在通信。这类通信系统适用于大群体，但是无法防御拥有较强能力的攻击者，无法满足大群体中某些需要高安全度保护的用户。

### 工业界的Tor（洋葱路由）及其缺陷

说到匿名通信网络，大多数人第一反应一定是Tor。的确，Tor是当前普及最广，使用人数最多的匿名通信网络，它属于我们上文提到的第二类匿名通信网络，即能支持大量用户但安全性不足。Tor是第二代洋葱路由（onion routing）的一种实现，用户通过Tor可以在因特网上进行匿名交流。Tor可以给网络流量进行三重加密，并将用户流量在世界各地的电脑终端里跳跃传递，这样就很难去追踪它的来源。从而实现让攻击者不知道谁和谁在进行通信即达到了保护元隐私数据的目的。

它的实现原理是Tor用户在本机运行一个洋葱代理服务器（onion proxy），这个代理周期性地与其他Tor交流，从而在Tor网络中构成虚电路（virtual circuit）。Tor是在5层协议栈中的应用层进行加密（也就是按照'onion'的模式）。而它之所以被称为洋葱路由，是因为它的结构就跟洋葱相同，你只能看出它的外表，而想要看到核心，就必须把它层层的剥开。即每个路由器间的传输都经过对等密钥（symmetric key）来加密，形成有层次的结构。它中间所经过的各节点，都好像洋葱的一层皮，把客户端包在里面，算是保护信息来源的一种方式，这样在洋葱路由器之间进行通信，如果攻击者想要追溯信息来源，就必须对洋葱的每一层进行解密，只要有一层攻击者无法解开，信息来源就得以保护。



Where does it go???

Where does this message from???

图1-1 Tor 示意图

但是，Tor同样具有着很大的不足，Tor的安全性并不够，这个安全性不足是针对我们的应用前景提出的。我们希望我们的匿名通信网络可以为军方服务，具有最小的受攻击面，因此Tor离我们的要求有以下几点不足：

1. Tor网络的每个节点都是由志愿者运行的，所以攻击者很容易成为恶意Tor节点之一。当这些恶意节点数量增加时，可能会导致所有通信链路都是由恶意节点组成，那么就有可能泄露通信的元数据信息。
2. Tor无法抵御主动攻击。主动攻击是指攻击者直接对数据通信过程进行干扰，例如修改、重放、丢弃或延迟数据包等操作，以便更有效地进行流量特征分析和抓取。这种攻击方法使Tor更难以保护通信双方的隐私并避免泄露其联系方式。

### 学术界相关匿名通信网络研究

到目前为止，国迄今为止，国际顶级会议上的通讯研究工作可以大致分为两类。第一类工作致力于严格实现密码学上的安全性，并已被可证明安全性理论成功证明，比如量子密钥分发协议。这类方法通常需要多次加密和解密，因此导致通信延迟比较高。而第二类工作则采用宽泛的安全性要求，仅仅通过在数据上加噪声的方式实现差分隐私保护，这种方法的通信网络中至少需要一个服务器是可信的，因此当攻击者控制了所有服务器时，数据仍然有被攻击者窃取的风险，但这类方法的效率通常较高。

#### 基于差分隐私技术的匿名通信网络

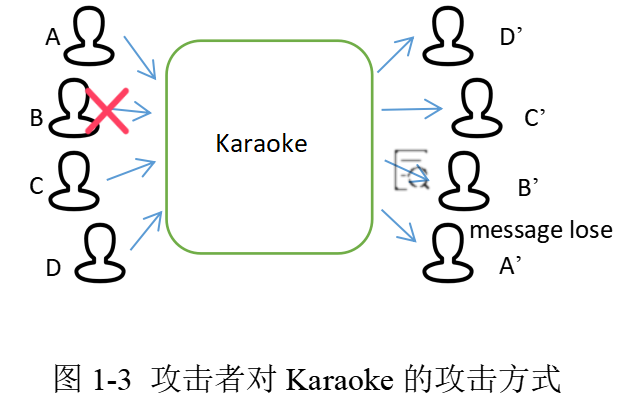
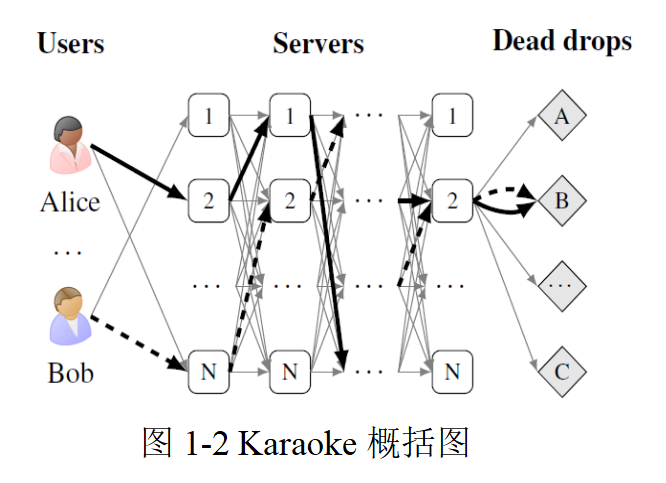
差分隐私是指通过添加噪声的方式使得攻击者无法得到确定的有意义结果。原本是为了防止差分攻击而提出的，主要是通过加入噪声的方式把查询结果模糊化。假如我们把一个查询表示为 f(x):x→R （这里暂时只考虑输出结果为一维的情况），随机噪声可以用 r 表示，最终得到的查询结果就是 M(x)=f(x)+r ，对于两个汉明距离为1的数据集 x,x’（即相邻数据集：两个数据集中只有一个数据不一样），对于任意的输出集合 S ，应该有：

|  |  |
| --- | --- |
|  | () |

如果将用户与谁通信或者不通信的状态看作是不同的输出值（即攻击者观察到的信息），应用到通信系统中就可以在一定程度上保护用户元数据。这一方法主要是通过服务器产生通信噪音，降低攻击者得到用户正确通信状态的概率，使该概率满足差分隐私的公式(1)。

2015年SOSP会议上，David Lazar的Vuvuzela[6]引起了人们对匿名通信领域的关注，Vuvuzela通过在一组链式相连的服务器上层层加密，并且每个服务器给通信加噪使达到差分隐私，也就是使通信的每一个可能对象在概率统计上是无差别的。尽管这种方法使得系统能够以更好的性能扩展到更多的用户，但其实每条消息都泄漏了少量有限的元数据，因此都有“隐私预算”的概念。这些系统中的用户在每次发送敏感消息时都会花费少量的隐私预算，最终无法保证强隐私。通信量大的用户可能会很快耗尽这笔预算，而且一旦用完，也没有明确的机制来增加隐私预算。在2018年 OSDI 会议上，David又发表了Karaoke[2]，尽管他改进了验证噪音的技术，通过虚假信息减少了服务器必须要生成的噪音。但是在安全性上，Karaoke依然处于差分隐私层面的安全级别。

差分隐私相关的工作不够安全，主要体现在要在安全性与数据可用性之间做一个权衡，隐私预算越小越安全，但同时数据的可用性也会降低；同时差分隐私工作的隐私预算一旦消耗完，服务器将会拒绝查询。



#### 可证明密码学安全性的匿名通信网络

在密码学中，评估一个密码学方法的安全性是极为重要的。其中，有条件安全性、计算安全性和可证明安全性等方法皆可用于评估密码学方法的安全性，其中可证明安全性被认为是一种相对实用的安全性证实方式。它的基本思想是将密码系统的安全性归约为某个已知的数学难题（如大整数质因子分解、计算离散对数等），而这些困难数学问题已经被深入研究，并被证明难以计算。由于这些难题的计算难度已经被证明，因此这些密码方法的安全性也就被证明了。

可证明安全性最早被应用于公钥密码学中，如RSA和ElGamal等算法就是利用了这种方式来保证安全性。此外，这种方法也被广泛应用于基于身份的密码学、属性加密等领域。并且，在可证明安全性的保障下，许多安全性更为高效和先进的方法不断被提出和应用于实际生活中。

可证明安全性理论是基于可证明安全性的规约思想发展而来的一系列理论，其中，私有信息检索等技术已经被可证明安全性理论[16]确认了它的安全性。

从2016年 OSDI 会议的Pung [7]这个方法的提出开始，被可证明安全性理论成功证明的PIR、CPIR等方法[8] [9] [10] [11]开始引起关注，基于此类方法的匿名通信网络的安全性更上一层楼。Pung用私有信息检索(PIR)技术来保护元隐私数据，它假定任何一个服务器都可以是向攻击者妥协的，也就是说，每个服务器都可以是不安全的。尽管Pung等工作安全性有保证但是通信延迟太高了，对于2M用户来说需要2000秒的通信延迟，之所以有这么高的延迟是因为其在验证噪音时需要有高额的时间开销。2020年发表在会议 NSDI 上的XRD [12]提出了聚合杂交洗牌（aggregate hybrid shuffle）的方法来提高验证洗牌正确性的速度。不过带宽代价较大，并且对于2M用户来说仍然有200s的延迟。可以看到，相比之前pung的工作，XRD、Express[4]和Yodel[13]等近几年的匿名通信系统的性能大大提高，不过对于现实生活中即时通信的需求，效率仍然是远远不够的。

图1-4是此类基于可证明安全性理论的系统的实现方法的基本结构：

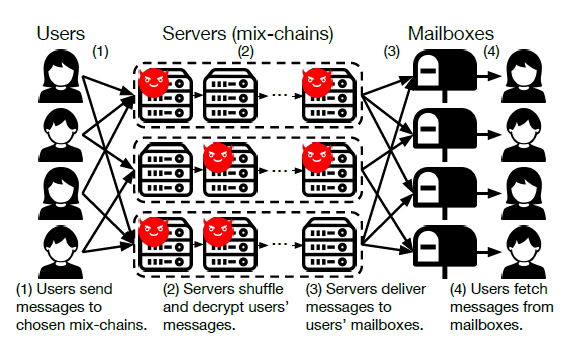


图1-4 基于可证明安全性理论的系统实现方法的结构

用户向选定的服务器链发送信息，无论服务器是否向攻击者妥协，都需要对信息进行洗牌（shuffle），然后我们用零知识证明（ZKP）等技术在不提供具体数据内容的情况下来验证洗牌后的数据是否是合法的，即有没有增、删、改我们的信息。最后将信息传递到接收者的信箱中。

可以看到，通过验证洗牌的方法，此类结构的匿名通信网络在安全性上是要强于仅达到差分隐私的方法的。但缺点也不言而喻，因为要验证洗牌，所以会引入较大的通信延迟。

### 基于SGX的匿名通信网络

在2017年IEEE TON上的这篇SGX-Tor工作，提出了利用Intel SGX实现Tor的方法。这样，就可以通过一种硬件辅助的可信执行环境，即CPU边界，解决广泛的安全漏洞和系统受损的问题。具体而言，Intel SGX提供了17种新的指令架构，可以为代码和数据设置私有区域，以及防止对执行中代码和内存中数据的直接攻击。虽然与当前几年内的匿名通信网络相比，安全性上还不足，但这种结合SGX的方法是一种新的思路，为我们提供了很大的启发。

最近，在2021年的IEEE TDSC上又提出了一篇将P2P网络与SGX相结合的工作。在该方法中，每个用户都可以参与通信网络，并负责转发消息，从而实现更安全的通信。在实验中，该方法不仅达到了高水平的密码安全性，而且在8000个用户的试验中仅有2.2秒的通信延迟。然而，这种方法的一个限制是它要求每个用户都能使用SGX。这样的限制实际上是一个很强的假设，因为在实际应用中，有很多用户的电脑不支持SGX。可见，虽然SGX在匿名通信网络中具有许多潜力，它仍然需要更多的改进和完善，以便帮助我们更好地保护我们的隐私和安全。

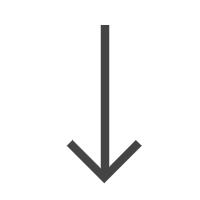
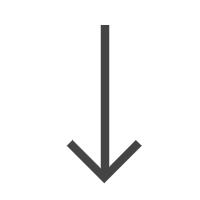
## 遇到的困难

从之前所述的相关顶会工作可以看出，现在的匿名通信网络系统要么可以大规模使用但安全度不够高，要么达到可证明安全性理论的要求，却有较高通信延迟。根据可证明安全性理论，我们的系统需要检验洗牌的合法性，但洗牌又会带来很高的时间延迟，这个矛盾一直是相关研究的难点。

而我们的思考是之所以要验证洗牌，是因为服务器有可能是向攻击者妥协的。那如果有一段可信的内存空间留给我们进行加密，在这种情况下不就不用耗费时间进行安全验证了吗？于是我们想到了用Intel公司的SGX这一绝对安全的内存区域来实现匿名通信网络。但是这其中有着诸多的困难不易解决。

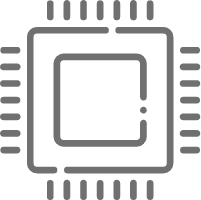
密码学意义安全

慢！



验证洗牌

不验证的情况下保证洗牌的合法



**SGX**

**CPU**

图1-5 idea思考过程

### 1.3.1 主动攻击者假设

首先在以往的研究中，往往是假设攻击者只能被动地监听通信，比如攻击者可以通过观察数据包的来路和去向、进行流量分析等被动攻击手段来嗅探信息而不能主动地进行消息劫持。其次以往的工作中只假设部分服务器向攻击者妥协而不能所有的服务器都向攻击者妥协。随着隐私保护重要程度的提高，我们需要更强的攻击者假设来达到更高的安全性，这种攻击者假设是很有必要的。举个例子，假设在军方的通信系统中，网络管理员被劫持或者网络管理员本身就是间谍，那么这时候就相当于所有的服务器都向攻击者妥协，我们希望我们的元守网络在这种情况下，依然能够保护通信双方的元隐私数据。因此，我们作以下两点攻击者假设。

1. 攻击者可以进行拦截通信的主动攻击（这种攻击方式更加难以防御）
2. 所有服务器都可以向攻击者妥协

### 1.3.2 SGX的局限性

在SGX中我们将它的可信安全内存空间叫作飞地（enclave）。即便有着这样一片安全的内存空间，攻击者也并非无计可施。SGX需要一个物理机，在我们的假设中，这台计算机是不可信的。攻击者可以查看服务端收到的消息包的大小、来源等相关信息，因此攻击者可以通过客户端收到与发送消息之间的对应关系来判断通信双方。其次，针对飞地本身也会有一些独特的攻击方式。比如安全内存空间可以提供数据的匿名性，但是攻击者可以通过RAM或者外存来推断进行操作的地址。即飞地无法保护数据所在地址的匿名性。通过这些攻击者同样可以建立服务器收发消息之间的对应关系来判断用户之间的通信情况。我们系统的设计目标是即使攻击者掌握了这种攻击方式，依然无法判断通信元数据。

## 特色分析

### 系统功能特点

与其他匿名通信网络系统相比，我们的基于SGX的匿名通信网络系统有以下特点：

1. 在保证安全性同时具有更快的速度。

相比之前工作的安全性与即时性无法兼得，我们的匿名通信网络系统在达到可证明安全性的同时，由于基于SGX这一安全硬件，所以能够取得较之前的工作更快的速度。在一台服务器下，对于6万用户，通信延迟仅为1.45秒。

1. 主动攻击者威胁下的强隐私保护能力。

我们的匿名通信网络系统，即使在所有服务器都受到主动攻击者的劫持的条件下，也能保证用户的元隐私数据不被泄露，即是一个能够抵御极强主动攻击者的匿名通信系统。

1. 使用门槛低，用户友好。

对于使用用户来说，我们不要求用户的设备必须配置SGX，因为我们本身在服务器端有SGX 的配置，所以用户不需要额外的硬件需求和额外的设备配置调整，就可以使用匿名通信网络进行通信。这为用户提供了一个非常方便和快捷的通信途径，而且更加友好实用。

### 实现特色

在实现过程中，我们对消息的处理具有以下特点：

1. 采取消息批处理和用户分组策略，解决SGX空间有限问题。

在理想的安全环境中，空间是足够大的，我们可以一次性处理所有用户输入队列的请求，然后发送输出即可。但是SGX的空间非常有限，所以我们采用消息批处理和用户分组的策略：将很长的输入消息按批划分，将所有用户进行分组；在一轮通信中，仅将一批消息加载到SGX的输入队列中去，然后，将这批消息按用户组分类，加载到对应的输出队列。这有效解决了SGX空间有限的问题。

1. 采取按组发送消息的策略，有效降低了通信的成本。

要完成匿名通信的工作，朴素的思路是SGX将所有的加密消息都发送给每个组，然后让组中的用户对所有消息进行解密。这样虽然可以达到匿名通信的目的，但是将所有的消息发送给每个组的成本太高。我们采取了仅仅将该用户组需要的消息发送给每个组的方式，用户只需要对该组内的消息进行解密，这使得通信效率得到了很大的提高。

1. 进行消息长度补齐，防止攻击者窥探通信双方。

在SGX中，我们采用实现按组内最长的有效消息，进行消息补齐的技术，使得服务端发出、用户端接收的消息长度保持一致，避免了攻击者通过消息长度的长短来推测通信双方，保护了用户的通信元数据。

## 应用前景分析

随着网络的广泛应用，网络环境下的信息安全成为网络用户关注的问题。虽然现在有很多系统可以保护用户元隐私数据，但是都由于一些缺陷无法大规模应用。本作致力于研究在不可信环境下的元数据隐私保护系统，并取得了一定的成果。可应用于以下领域：

### 医患交流平台

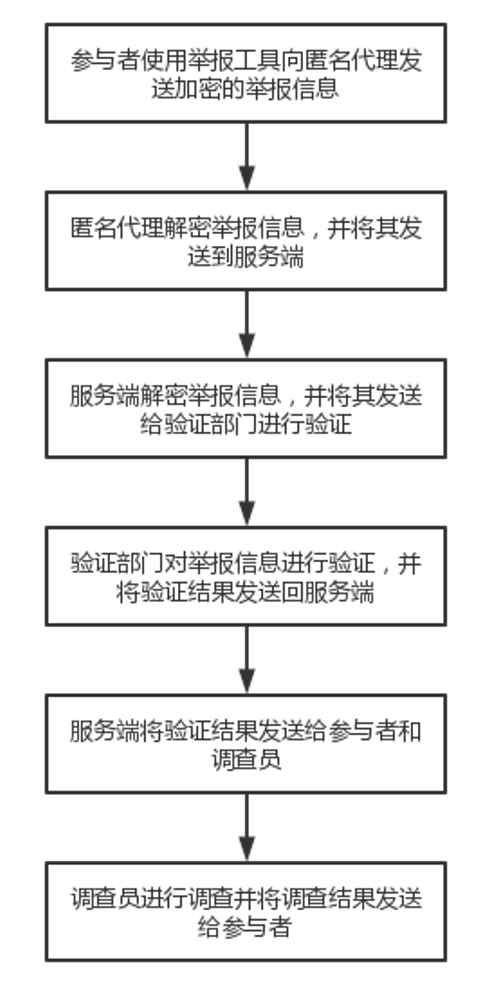
现在，医疗机构开通了医患线上交流平台，这为慢性病患者、术后康复患者、家庭医生服务包会员和其他无法线下前往医院的患者提供了很好的服务。这些用户可能需要长期与某个医生或医生团队进行沟通，而信息加密技术可以保护他们的具体交流内容不被攻击者获得。但攻击者仍然可以通过观察用户的参与通信情况来推断出患者可能的身体问题，甚至在了解到医生的身份和专业领域后，可以推断出患者患了哪种病。这样的信息会使攻击者有意地利用这些信息，让用户遭受名誉和人身攻击，如恶意贩卖病情信息给患者的竞争对手等等。举一个具体的例子，有一个65岁的女士，她患上了严重的心脏病，每周需要定期与她的心脏病专家进行在线沟通，在这些沟通中医生会询问她的身体状况、症状和药物反应等问题，以便更好地了解她的病情和调整治疗方案。攻击者可以通过检查这位女士的在线沟通记录来猜测她患有心脏病，甚至可以推测出她的症状和药物治疗情况。这使得攻击者有机会泄露此信息，对患者造成名誉损失或人身攻击。

我们的产品可以解决这些问题。它可以避免攻击者借助用户的参与通信情况来推断患者的身体状况，保障用户的元数据隐私。我们的产品可以确保用户与医生之间的通信在进行加密的同时，同时确保攻击者无法得知用户是否与医生通信、通信频率以及通信时间。这种技术可以让患者在使用医患在线交流平台时更加放心，不用担心自己的身体状况和交流对象被攻击者了解。从而使用户获得更加安全和私密的医疗线上交流服务体验。

所以，我们的产品可以为慢性病患者、术后康复患者、家庭医生服务包会员和其他线上医疗服务用户提供更高层次的保障，在隐私性、私密性、安全性上得到更加全面和完美的保护。这样，用户就可以更加方便、安心、舒适地使用线上医疗服务体系，享受更好的健康服务，获得更快速度的医疗专业帮助。

### 举报平台

在举报等敏感场景中需要保护参与者的隐私信息以免参与者被报复或者骚扰。如果他们的信息遭到泄露，比较幸运的人可以享有法律保护，但是也有可能面临着报复。为了降低这些风险，许多匿名举报系统比如Signal 和 SecureDrop应运而生，但是仍不能抵御足够强大的对手。我们的系统假设攻击者足够强，可以观察到整个系统中所有人的收发信息，并可以复制或阻断任意一个包的发送与接受。同时服务端也可以是全部不可信的。只要服务端还在工作，攻击者就不能获得想要的信息。因此，我们的系统可以适用于匿名举报系统等其他需要高隐私保护的环境，并提供足够的安全性。



### 客户沟通平台

在涉及到举报等敏感场景中，保护参与者的隐私信息非常重要，以免遭到报复或骚扰。即使具有法律保护，也可能会面临着报复的风险。为了解决这个问题，许多匿名举报系统如 Signal 和 SecureDrop 出现了，但是即便如此仍然难以抵挡足够强大的攻击者。

我们的系统假设攻击者非常强大，可以观察到整个系统中所有人的收发信息，并且可以复制或阻止任何一个数据包的发送和接收。这意味着我们的系统在任何时候都可以为参与者的隐私信息提供必要的保护。即使服务端不是百分之百可信的，只要系统保持正常工作，攻击者就无法获得他们想要的信息。

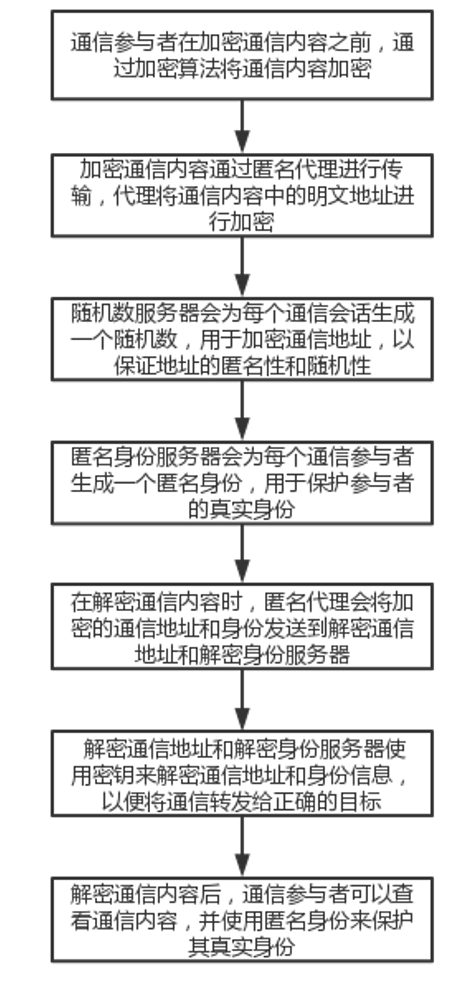
因此，我们的系统可以应用于各种匿名举报系统等需要高度保密性的环境，为参与者的隐私信息提供充分的保护，提高相关人员参与这类系统的安全系数，使他们得到必要的自我保护。比如说，那些担心报复行为的举报员工可以在我们的系统中按照匿名的方式提交线报，这将极大地减少他们面临的风险。在信息安全方面，我们的系统提供了一种非常实用而强大的解决方案。

### 军事网络

在军用网络中，采用成熟的加密技术可以保护通信的内容不被窃听者破译。但是，由于网络通信协议需要明确目标的源地址和目的地址，通信参与者的身份很难通过加密算法来隐藏。因此，网络攻击者可以通过分析网络流量和数据分组来获取通信参与者的角色和位置，以对活跃的网络节点和军事中心进行攻击和破坏。此外，即使军官得到了驻扎的任务，他们可能不知道任务的其他详细信息或执行人员。但是，如果监视长官的通信情况，结合通信双方的身份信息，攻击者可能会推断出任务的规模和其他关键信息，从而威胁军事行动的安全。

我们的产品吸取了之前已有的匿名通信技术的优点，并可以使攻击者即使知道某个用户参与了通信，也无法推测出通信目标是谁。这种保护用户隐私的方式可以将整个通信过程全面保护，使得攻击者无法获得有用的信息并加以利用。在这种情况下，通信参与者可以在不泄漏个人身份信息的情况下进行通信，从而保障通信的安全和可靠性。

同样的，在军事行动中，我们的产品可以保护军官的隐私信息，确保攻击者无法获取任务的细节和执行人员的身份。即使攻击者知道某个军官参与了通信，他们也无法推断出任务细节的相关信息。这样可以最大程度上保护军事行动的安全性和保密性，确保顺利完成任务。我们的产品具有非常强大的隐私保护功能，可以有效保护通信参与者的隐私信息，从而提高整个军事网络系统的安全性和保密性。



# 作品设计与实现

## 系统方案概述

表2-1 符号说明

|  |  |  |
| --- | --- | --- |
| **符号** | **说明** | **单位** |
|  | 总用户人数 | 人 |
| ***GroupCnt*** | 总组数 | 组 |
| ***Data*** | 发送数据包 | 个 |
| ***To\_UID*** | 接收者id | 无需单位 |
| ***SerPubKey*** | 服务器公钥 | bit |
| ***RecPubKey*** | 接收者公钥 | bit |
| ***m*** | 发送消息明文 | 字节 |
| ***SerPriKey*** | 服务器私钥 | bit |
| ***EID*** | 用接收者公钥加密的id | 无需单位 |
| ***RecPriKey*** | 接收者私钥 | bit |
| ***M*** | 补齐后一个组的所有信息 | 字节 |

本通信系统借助SGX这一绝对安全的内存空间，实现保护元隐私数据的匿名通信网络系统。通信过程以轮次进行，也就是在一个信息发送轮到来时，所有人可以发信息进行通信，但当这一轮次结束，任何人不得发送消息，此外一个人在一轮中可以发多条消息。为了在保证安全性的同时，确保速度足够的快。我们采用用户分组的方式进行设计。为了解决SGX内存空间太小的问题，我们还在服务器SGX之外的内存空间中维护了两种队列，分别为接收消息队列和组发送消息队列，整体的系统方案为：

1. 进行分组。组数与组内人数我们进行提前规定，在第三章的测试中，我们将对不同的人数进行测试，所以针对不同的人数n，分组策略为256个用户为一组，组数GroupCnt为总人数除以256向上取整。

|  |  |
| --- | --- |
|  | (2) |

1. 服务器端在内存中维护每个组的接收消息队列用于接收发送方所发信息，发送方所发消息数据包Data包含一个用接收方公钥RecPubKey加密的密文消息块和一个用服务器公钥SerPubKey加密的接收方To\_UID。用户将这两块合并为一个数据包传递到接收消息队列中。

|  |  |
| --- | --- |
|  | (3) |

1. 服务器按照接收消息队列顺序处理消息，由于enclave的内存空间有限，每次服务器用SGX指令仅读取接收消息队列里的一批消息（256MB），然后服务器在SGX内部用secure enclave中存储的服务器私钥SerPriKey对这批消息的id部分进行解密，查明接收方是谁。

|  |  |
| --- | --- |
|  | (4) |

1. 在enclave空间中，由于已经直到了接收者是谁，于是服务器根据接收方所在的用户组，将正在处理的消息一组一组的分开，一个消息的接收方是谁就分到对应的组别。然后再用接收方的公钥对属于每个人的id进行加密。

|  |  |
| --- | --- |
|  | (5) |

1. 然后找出消息长度最长的那个组，将其余组的消息长度都补齐到跟最长的一样长，这里用来补齐的就是虚假信息。我们现在对每个用户组都有了一个同样长度的明文下的消息块。紧接着把这一组的消息发到对应的组发送消息队列中。

|  |  |
| --- | --- |
|  | (6) |
|  | (7) |

1. 接收方从属于自己组别的发送消息队列中取出所有消息到客户端，用自己的私钥对每一个消息id部分进行解密，如果成功解密是发送给自己的消息，那么就继续用私钥对消息内容进行解密，得到属于自己的消息。

|  |  |
| --- | --- |
|  | (8) |
|  | (9) |

1. 服务器继续处理下一批消息，重复上述流程直到处理完成一轮的所有信息，然后等待下一通信轮到来。通信过程结束。

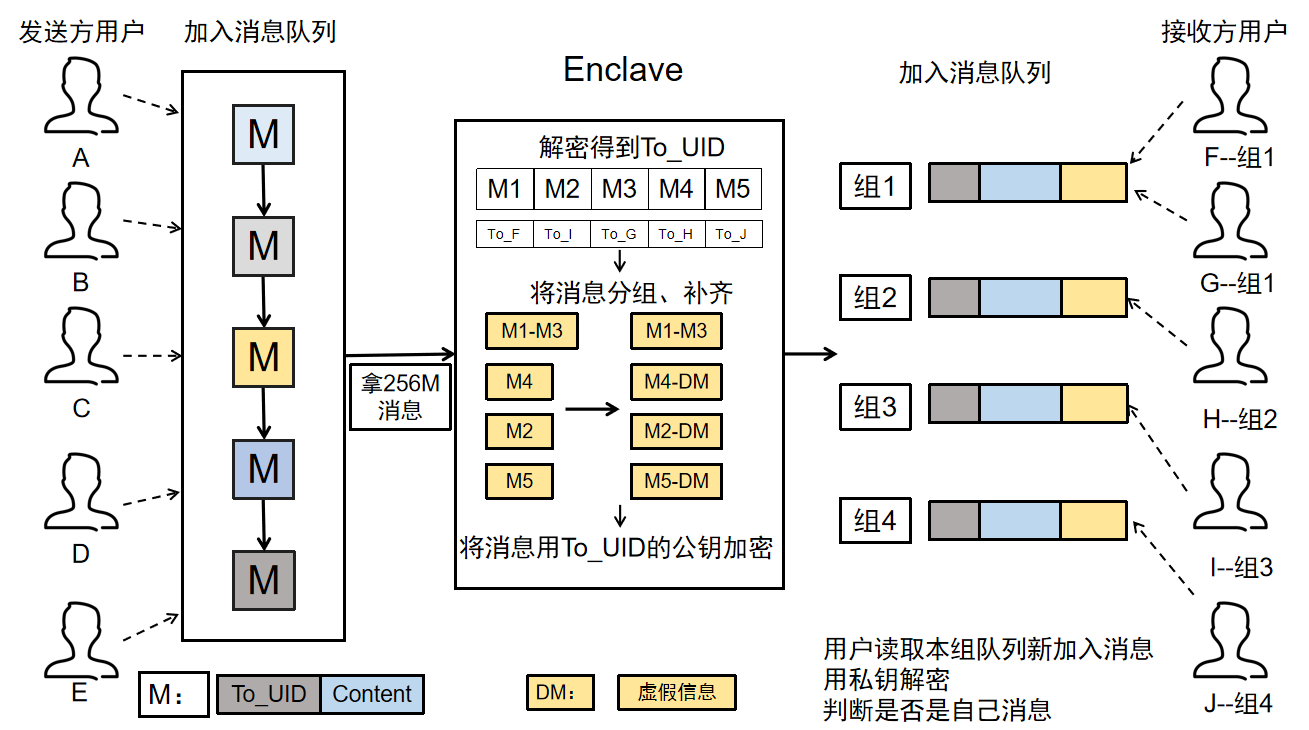


图2-1 系统方案概述流程图

## 系统实现细节

在本章节的实现细节中我们将按照通信的步骤，依照先后顺序介绍每一步操作并详细说明我们系统如此设计是为了什么。

### 分组策略

我们的分组策略是将所有用户分为256人一组，并向上取整总用户数除以256的结果作为组数。这是因为SGX内存空间有限，只能处理一定数量的消息（128MB）。如果不进行设计，SGX处理消息会存在先后差异，比如先处理了第一个人发的消息，放到了SGX外部的内存中，再处理第二个人的消息。由于我们假设服务器可能会遭到攻击者妥协，攻击者可以看到SGX外部的内存情况，并基于消息处理顺序得知每条消息存放在哪个内存位置。这样，攻击者可以轻易跟踪出通信双方的身份信息。

我们可以通过分组的方案来降低时间复杂度。朴素的想法是，在每个接收者处都发送完整的消息，这与私密信息检索（PIR）有些类似。但显然这会增加通信开销，无法很好地处理大量通信。因此，我们将用户分为n组后，直接向每个组发送消息。这样可以将发送消息的数量减少n倍，因为其他组的信息不会被发送到该组的消息队列中。这种优化方案有效地降低了通信复杂度，延迟也得到了有效控制，保障了用户的通信隐私和安全。

在这样的方法之下，我们是否存在隐私泄露问题呢？是有的，由于我们定向取接收者，所以知道是哪一组人在跟发消息的人通信，按照我们128的分组人数，也就是说，猜到通信双方的概率是1/128，我们希望降低此概率，于是加入了服务器端数据补齐处理。这点将在后面2.2.3阐述。

### 发送方发送

1. 发送数据包格式：每个人发送的数据包应包括两个东西：（1）以接收者公钥加密的通信内容（2）以服务器公钥加密的接收者id
2. 数据送到哪里：由于SGX内存空间有限，我们无法将所有信息全部装入enclave，所以必定要在服务器SGX之外的内存空间暂存数据。为了适应分组策略，我们在服务器的内存中维护一个接收消息队列来存放服务器收到的消息包，以及每组一个的组接收消息队列存放对应分组的用户接受的消息。所以，用户在发送轮直接将整理好的数据包发送给接收消息队列中，想发送多条消息给多个人也是可以的。

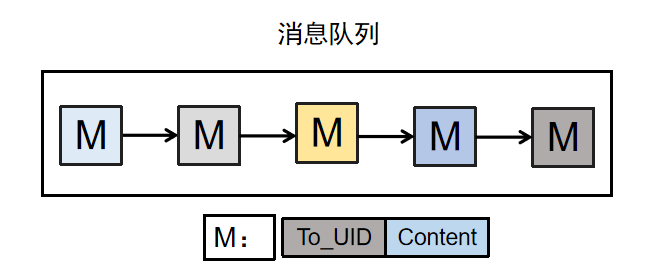


图2-2 发送方消息队列及消息格式

### 服务器端数据补齐及处理

当所有消息都发到了对应的接收消息队列后，服务器就要开始对这些消息进行处理了。处理的目的是得到接收者是谁以及进行一个虚假信息补齐。这里就要使用到SGX，为什么呢？想要保证安全性，元隐私数据不被泄露，就一定要使得服务器不知道接收方是谁，因为我们假设攻击者是可以控制服务器的。怎么做到让服务器不知道谁是接收者呢？借用SGX特权指令，SGX使得即使有服务器的root权限也无法访问CPU的运算过程和SGX的内部空间，我们在SGX的enclave空间中对消息的id部分进行解密从而得到接收方是谁。整体的流程如下：

1. 服务器按照256MB为一个批次读取接收消息队列中的消息，调用SGX指令将消息读入enclave中
2. 在enclave里对接收方id部分用服务器私钥进行解密，得到全部消息的接收方是谁后按照接收方的组别分组，由于各个人、各个组别接收信息的数量肯定是不同的，我们为了防止攻击者通过发送消息数量和接收消息数量的联系推断出通信双方的关系，我们要对每一个批次的消息进行一个补齐，使得发送到各个发送消息队列中的消息是一样长度的。

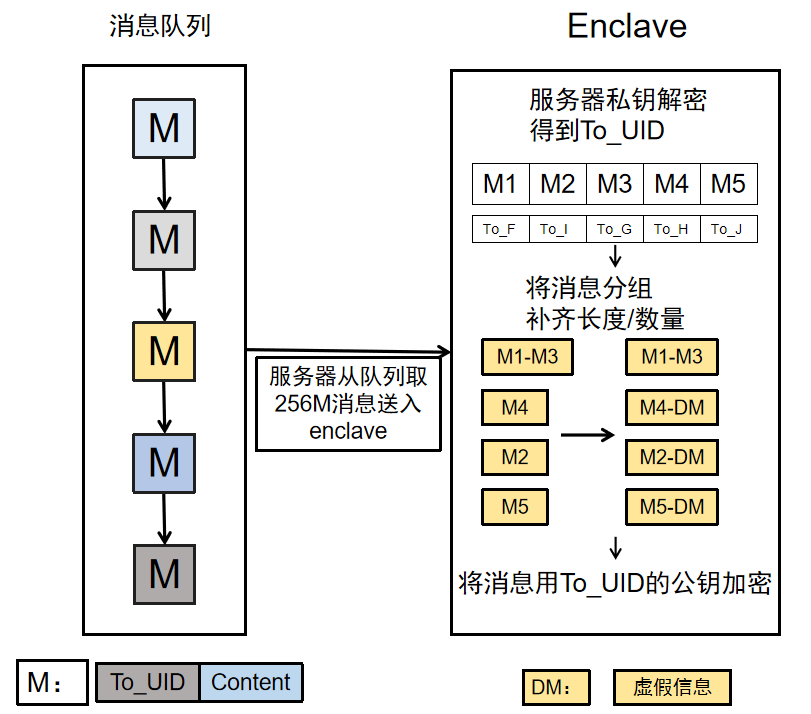


图2-3 Enclave获取消息及处理概述图

1. 补齐之后，使用接收者公钥再次对接收者id进行加密，然后就把消息从enclave中移出到服务器维护的组发送消息队列中，同样是一个用户组对应一个发送消息队列，接收者将从队列中读取信息。

### 接收方读取接收

接收方将属于自己组别的消息读入客户端，遍历全部消息，对每一个数据包解密，如果是属于自己的消息，则读取并对消息内容的密文用私钥进行解密。至此通信过程结束，接下来服务器继续进行消息批处理直到当前轮次所有消息发送完毕，等待下一轮的开始。

# 作品测试与分析

## 安全性说明

当前会议中匿名通信相关主题的作品通常采用零知识证明或者私有信息检索技术（PIR）来保障安全性，我们选择用SGX这一理论上绝对安全的硬件来实现我们的匿名通信网络。在安全性上，由于SGX将合法软件的安全操作封装在enclave中，可以保护其不受恶意软件的攻击，并且特权或者非特权软件都无法访问enclave。一旦软件和数据位于enclave中，即使操作系统也无法影响enclave里的代码和数据。Enclave的安全边际只包含CPU和它自身，这一设计避免了基于软件的可信执行环境自身存在软件安全漏洞与威胁的缺陷，有效地提升了系统安全保障。同时，Intel官网中已经对SGX安全性进行了证明。因此我们选择通过速度来与其他相关工作进行对比，来证明我们的方案的价值。

## 延迟测试方案

在完成了基于SGX的匿名通信系统的原理设计与实现后，接下来对我们的系统'元守'进行测试。接下来主要测试我们系统在速度方面的优势，即在相同环境下测试我们的系统与其他匿名网络系统的端到端延迟，并进行对比分析。

### 对比系统

我们选择了Addra和XRD这两种匿名网络通信系统进行对比实验。这是因为我们的匿名系统可以在完全不可信的服务器环境下仍然保护元数据，而有些匿名系统，如Stadium和vuvuzela等，在安全性方面较低，不能对付我们假设的攻击者。而虽然Tor的安全性较高，但如果对手控制网络中的某些关键点，则很容易受到流量分析的攻击。因此，Tor在完全不可信的环境中也不能抵御攻击者。与此相比，在网络环境完全不可信的情况下仍然可以保护用户通信元数据隐私的系统代表是Addra和XRD。

Addra是一种利用类私人信箱方式改造PIR技术的匿名通信系统。它第一次实现了在完全不受信任的基础设施上保护元数据的语音通信系统，并可以支持数万级别的用户。Addra设计了一种完全分散的体系结构，用户的彼此之间的通信不需要网络操作，完全依赖于本地计算机。这种分散的方式有效地保护了参与者的隐私信息，使攻击者无法获取到有关通信的任何数据。

XRD则通过将负载分布在多台服务器上实现了扩展，并使用聚合混合洗牌技术来保护元数据，可以在100台服务器的下以228秒的延迟支持200万用户。并且还可以向控制整个网络、固定比例服务器和任意数量用户的对手隐藏所有通信元数据。XRD根据服务器的可靠性对多个组内的负载进行聚合，同时使用洗牌、更换标识符等技术来实现元数据的保护。这有效地掩盖了参与者的身份信息。

这两种匿名网络通信系统在保护用户元数据和隐私方面非常出色，可以在不可信任的网络环境中保护用户的隐私数据。这些性能使Addra和XRD成为广泛应用于匿名通信的系统，能够有效解决各种网络攻击问题。

### 具体测试方案：

我们编写程序时没有选择编写为用户可以一对一通信的形式，因为测试时直接找大量的用户是不现实的，因此我们选择构建一台虚拟机同时发送大量的消息来模拟大量用户同时使用我们的匿名通信系统。考虑到虚拟机开销的问题，我们采用docker容器技术来模拟虚拟机，同时也对docker进行了网络通信的测试确保可以正常通信。

1. 我们通过本机环境创建docker镜像，启动多个docker虚拟机模拟服务端与客户端，然后计算端到端的平均延迟。通过单个客户端模拟大量用户向服务端发送消息来模拟大量用户访问，在模拟足够多的轮次后，得到端到端平均延时。
2. XRD与Addra的通信延迟都跟服务器数量有关，通常来讲服务器数量越多延迟越低，在单服务器情况下XRD与Addra运行延迟过大，因此我们选择16台服务器。同时测试发现元守只需要一台服务器就已经可以达到很好的效果，所以在测试中’元守’配置为单服务器，而XRD与Addra是16台服务器。
3. XRD测试时使用16条链，整体服务器网络为16×16，服务器配置为8核，客户端配置为16核，其他配置均为默认值。
4. Addra测试配置：16台8核服务器，master一台16核服务器，客户端一台16核服务器，其他配置为默认值。

测试变量如下：

自变量：用户数量，分别为2^10、2^11.......2^20。

因变量：端到端延迟（单位ms）、系统类型。

## 测试环境及设备

1. 网络通信框架：

'元守'使用tcp短链接通信框架进行物理机通信。

XRD与ADDRA均使用官网源代码所使用的框架通信。

1. 网络环境：

我们保证本机与服务器在测试的所有时刻都有良好的WIFI网络或者是

4G/5G网络。

1. 测试设备：

物理机：Ubuntu 18.04 16core

虚拟机：Ubuntu 18.04 16core

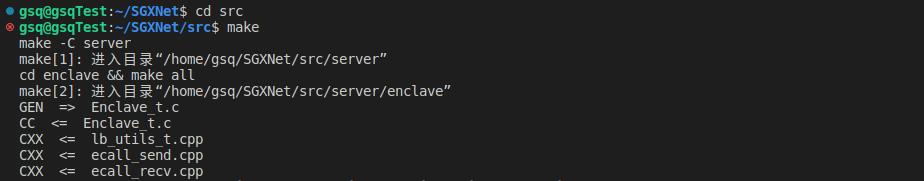
## 测试数据

测试时发送信息大小为256bit。并且由于测试都是通过一个模拟的客户端来发送信息的，所以不需要网络包数据集。

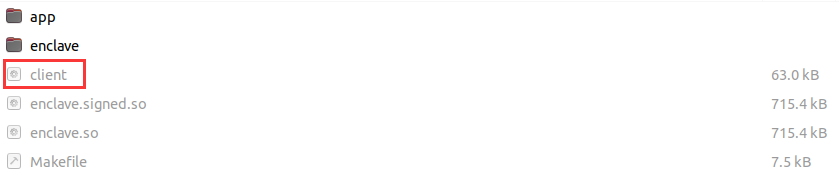
## 测试流程

源码主要使用C++编写，使用协程libgo解决异步问题，使用cmake进行项目管理，编写环境为vscode，网络通信采用tcp连接。源码解压之后有.vscode，src两个文件夹，.vscode是vscode的配置文件夹，src是源码文件夹，编写的程序都存放到这里。由于使用了cmake管理项目，编译时只需要进入src文件夹执行命令make即可。若无报错，会分别在client，server文件夹下生成程序client，server两个可执行文件。之后新建两个窗口，分别运行client与server两个可执行文件。

进入源码文件夹并执行make编译：

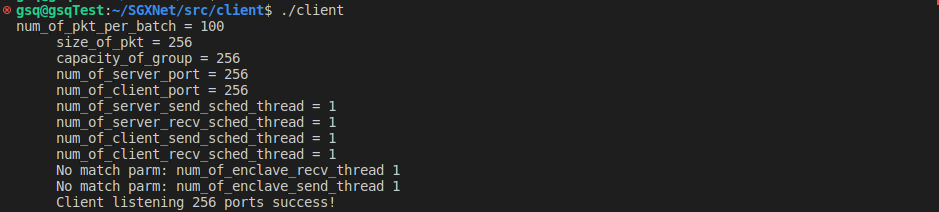


编译后生成中间文件以及可执行文件：





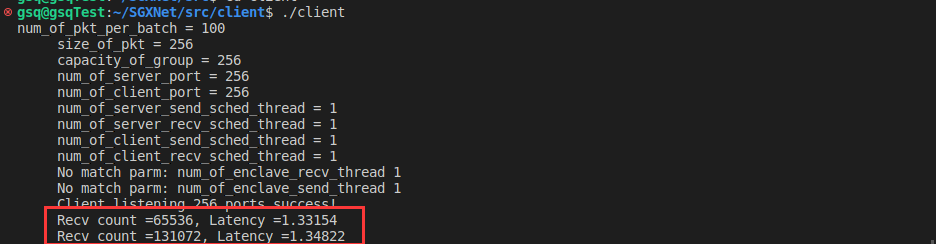
新建窗口进入文件夹/src/client执行./client：



新建窗口进入文件夹/src/server执行./server：



开始测试获得延迟（图中以人数65536为例）：



可以更改用户数量来获得不同用户数下元守所对应的延迟。

## 结果分析

### 测试结果数据

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 表3-1 元守不同用户数下的延迟 | | | | | | | | | | | |
| 总user数量（2^x） | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 元守  latency(s) | 0.027 | 0.045 | 0.104 | 0.167 | 0.414 | 0.698 | 1.45 | 2.843 | 10.062 | 18.565 | 34.734 |
| XRD  latency(s) | 0.84 | 1.647 | 2.505 | 5.794 | 11.344 | 22.426 | 44.427 | 89.608 | 184.377 | 371.083 | 751.622 |

注：元守测试clt = 1 server = 1；XRD测试16×16 chain(8 core) clt=1(16 core)

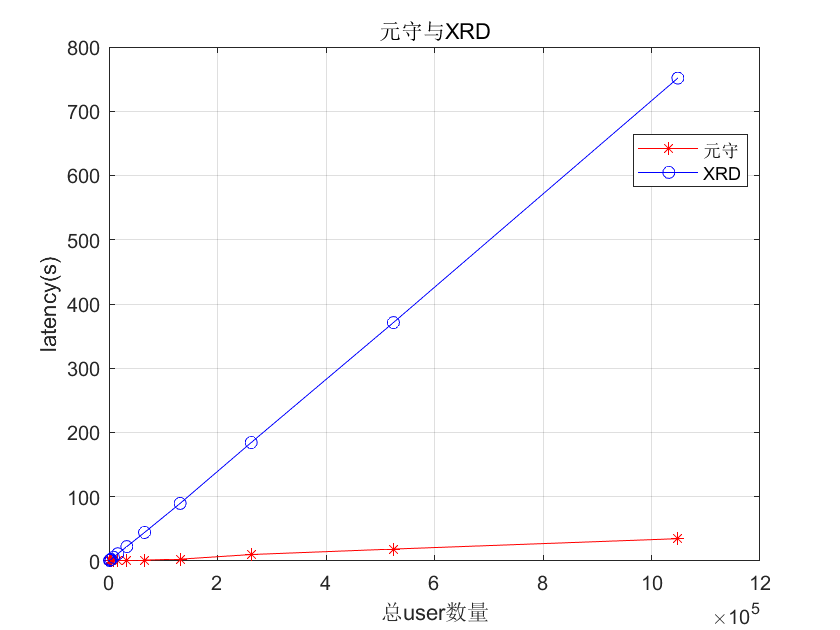


图 3-1 元守与XRD延迟比较

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 表3-2 addra worker=16(8 core) master=1(16core) clt=1(16core) | | | | | |
| 总user数量（2^x） | 12 | 13 | 14 | 15 | 16 |
| 元守  latency(s) | 0.104 | 0.167 | 0.414 | 0.698 | 1.45 |
| ADDRA  latency(s) | 1.769 | 3.946 | 9.74 | 26.993 | 84.529 |

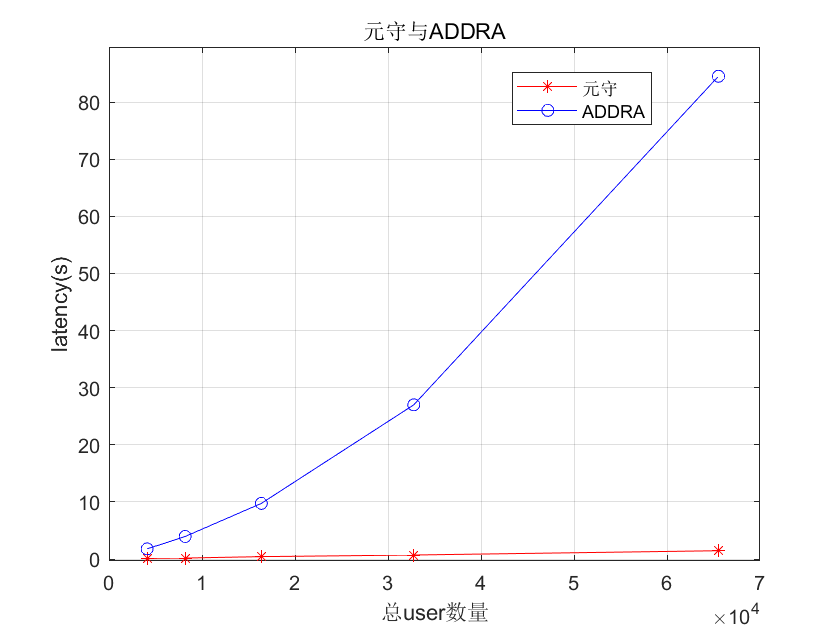


图 3-2 元守与ADDRA延迟比较

### 测试结果分析

如下图所示（图3-3），元守即便在没有实现多线程的情况下与现在速度较快的系统，比如XRD与Addra相比，在同配置下有更低的延迟。实验中我们选择的XRD为16×16，配置更多的服务器可以使延迟进一步降低，但是花费也就越多。

实验结果如图3-4所示：在2^16（65536）个用户的参与下，Addra、Xrd和元守的延迟分别为84.5s、44.4s和1.5s；在2^20（约105万）个用户的参与下，Xrd通信系统的延迟达到了751s，而元守通信系统的延迟仅仅有34.7s。元守通信系统的速度相比Xrd通信系统提升了约22倍，相比Addra通信系统提升了约40倍。

元守目前没有实现多线程处理，配置为单服务器单客户端。即便如此在延迟上依旧优于XRD与Addra。

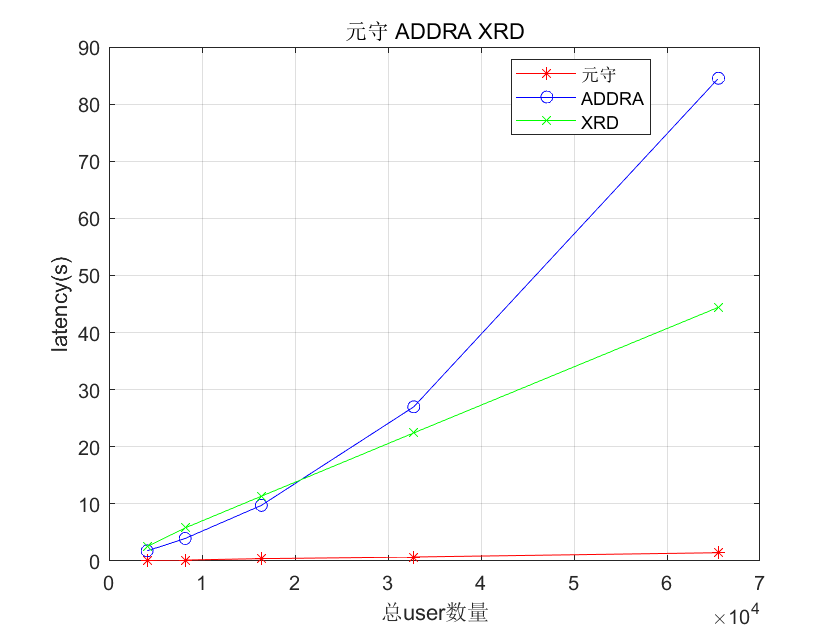


图3-3 元守、XRD和ADDRA延迟比较

# 创新性与实用性说明

## 创新性

### 现有匿名通信系统的缺陷

匿名通信系统是指采取一定的措施隐蔽通信流中的通信关系，使窃听者难以获取或推知通信双方的关系及内容。匿名通信的目的就是隐蔽通信双方的身份或通信关系，保护网络用户的个人通信隐私。作为一种新兴的通信系统，匿名通信系统所提供的元数据保护也正越来越受到人们的重视。而当前的匿名通信系统的缺陷在于：

1. 匿名通信系统的支持人数扩展的成本较大。Pung[OSDI’16]达到一百万用户时，通信延迟达到了较高的272s, XRD[NSDI’20]的一百万用户延迟也达到了114s。虽然近年来的匿名通信技术已经有了一定数量级的改善，但是仍然在扩展到一定用户时，其延迟仍然难以达到最终的商业化目的。即便是在安全性方面做了一些让步，采用了差分隐私策略的Karaoke[OSDI’18]，在达到了八百万用户时延迟也已经达到了22s，难以和当今流行的实时通信工具，例如QQ、微信等，进行用户市场竞争。
2. 匿名通信系统难以保证在不可信的服务器上的安全性。在以往的匿名通信系统中，XRD[NSDI’20]中要求有20%及以下的不可信服务器，而Karaoke[OSDI’18]要求至少一个的可信服务器。在这些匿名通信中，我们可以设想，如果攻击者本身就是这些服务器的管理员，比如军队中有叛徒，那么它们可以控制所有的服务器，从而发现通信的元数据等隐私信息，而这是存在一定安全风险的。我们在此基础上设想的是一个任何服务器都不可信的环境，这样可以更好地保证安全。
3. 满足SGX配置的用户端数量较少、安全性仍然有待改进。 对于匿名通信中的SGX结构，之前也做过一些研究，在2017年IEEE TON上的一篇SGX-Tor的工作提出了用SGX实现Tor的方法，2021年的IEEE TDSC中，又有一篇工作提出将P2P网络与SGX相结合，等等。前者将SGX与Tor结合令人耳目一新，然而还是由于一些Tor的固有安全隐患，存在安全性不足的问题。后者虽然在实验上达到了密码上的安全性而且通信延迟较低，但是需要每个用户端都使用SGX。这个要求的条件较难达到，这是因为有部分用户的计算机内核并不是Intel架构、SGX的环境需要部分激活步骤才能使用等原因。

### 基于SGX的匿名通信系统

我们的主要创新点在于，基于一个足够可信的硬件环境SGX，在以往的匿名通信工作的基础上，设计具有安全性和并发性的算法，来完成我们的匿名通信网络系统。Intel 公司出品的SGX实际上是一组 CPU 指令扩展，可以创造出一个可信执行环境来保护代码和数据，即使攻击者拥有root 权限也无法访问。应用程序可以使用它来隔离代码和数据的特定可信区域，可以防止底层OS被攻击成功后所受到的攻击，同时在软件的管理上也可以不信任任何云供应商，我们基于以上研究， 提出了SGX是“绝对安全的”这一假设。SGX最关键的技术是Enclave，这种技术可以理解为一个盒子，在运行过程中，只能外部读取，当权限拥有者从外部写入后，非权限拥有者便无法对其更改。 但同时也伴随着一些缺点比如由于enclave处于用户态, 增大了安全风险和系统开销。其次SGX提供的enclave可使用内存太小, 当程序数量和规模增大时, 需要换进换出页面导致系统开销大。目前SGX主要是用于云安全、可信计算等领域。

对于匿名通信系统的研究面临着两个问题，安全性与通信延迟，前者是匿名通信的要求，后者是实现即时通信的必要条件。想要实现安全性就需要对服务端进行验证洗牌，但这样会增大通信延迟。不进行验证洗牌可以达到低延迟，但是就很容易被攻击者攻击。并且为了抵抗流量分析与活性攻击，需要添加虚假信息，使用差分隐私可以是一种方法，但是安全性达不到密码安全，并且也不好控制添加噪声的量。添加大量的虚假信息可以提高对这些攻击的抵抗能力，但是会进一步的增加通信延迟。使用SGX，我们可以同时解决安全性与延迟问题，使用SGX的安全空间可以一定程度地简化安全验证问题；由于简化了验证安全的步骤，通信延迟也会降低。

值得注意的是，我们设计的系统仅在服务器端具有SGX需求而用户无需配置SGX，通过服务端的安全空间对信息进行分配，可以达到密码上的安全，并且实验上对于6万用户，在一台服务机上，仅有1.45秒的通信延迟。

### 适用于SGX小内存的信息处理算法

我们通过消息批处理和用户分组的信息分配算法，有效解决了SGX小内存和匿名通信中大消息量的矛盾，扩大了可支持用户量。在匿名通信系统的设计中，可支持用户量也是一个重要的评判标准。当支持用户量比较小，比如只能支持几百几千用户时即便可以达到密码安全实用意义也不高。支持大量用户就会面临数据处理的问题。SGX的enclave空间很小，不可能同时把数据放到安全空间中处理，每次遍历全部用户又会导致较高的资源开销。因此我们设计了一个信息分配算法，主要思路是采用消息批处理和用户分组的方法，将用户分为若干组，每次只处理一组用户的消息，并且由于没有发送消息数量的限制，还需要在安全空间外维护加密消息队列。

### 支持大量用户的安全实时通信

我们在通信双方同时上线的情况下，将一定数量的用户（例如256个）分为一组，只需要扩展组的数量，可以实现大量用户的通信。在我们的测试数据中，对于2^15的用户数量，延迟仅仅为0.698s；对于2^17的用户数量，也仅仅有1s左右的延迟。而在一轮次的通信时，对于该组的两个用户发出的消息，我们用在第二章中介绍的方法，实现通信双方的安全实时通信。在我们的匿名通信架构下，对于发送和接收消息的双方，接收的消息是几乎实时的，这保证了接收到的消息的时效性。这在原理上类似我们日常生活中的电话通信，对于一些需要实时交流，而不太希望等待别人应答的用户，可能会得到非常好的体验。

### 更强的攻击抵抗能力

我们具有的强攻击抵抗能力，体现在我们的设计能够很好地解决服务器的可信问题和流量分析、劫持攻击问题。

对于服务器的可信问题，我们基于全服务器恶意假设，成功进行了匿名通信测试实验，可以保证在所有的服务器都被攻击者劫持的情况下，仍然具有保护元数据的匿名通信的能力。这主要归因于SGX的不可探测性。SGX作为一组 CPU 指令扩展，创造出的可信执行环境，可以用来保护代码和数据，即便攻击者拥有root 权限也无法访问。SGX最为重要的意义在于，它可以隔离代码和数据的特定可信区域，可以防止底层OS被攻击成功后所受到的攻击，同时在软件的管理上也可以不信任任何云供应商。而以往的匿名通信系统之中，最好的隐私要求也是Addra[15]提出的只需要一个服务器是未受攻击的，我们基于这一要求，提出了全服务器恶意假设这一较为严格的假设，并且最终取得了一定成果。

对于流量分析、劫持攻击等攻击策略，我们的服务器在信息分配时，会在安全空间内部添加虚假消息、长度补齐等加密手段，使得攻击者无法通过观察网络流量得到任何有用信息。在信息分配的过程中也会对信息进行再次加密，因此并不能通过对消息添加标签来判断消息流向。由于使用了SGX，即使所有服务器都被劫持，由于攻击者无法访问SGX内的内容，所以通过已知的常规手段，几乎无法通过任何手段得到任何有效的元数据信息。我们对被动的流量分析攻击和主动的劫持攻击都有较强的抵抗能力。

## 实用性

### 支持广播通信和多消息同步发送

由于我们对用户进行分组，输入消息队列仅仅要求提供接收者的身份，而没有限制发送者的身份是否互斥，所以该系统可以在一个轮次内，在本机完成一个用户往多个用户发送消息的动作，也就是广播通信。同时，一个发送者也可以选择注册多个用户，同时出现在不同的组之中，从而实现多消息同步发送的功能。这有效提高了发送者发送消息的手段和效率，使得发送者不再需要类似于打电话，逐个通知接收者群体；而只需要输入消息，一键群发，就可以使得所有接收者都实时地得到相应的信息；对于一个有多种身份的人，也可以同时以不同的角色给不同的接收者群体发送信息，例如以父亲的身份给子女发送 “放学了吗？”，同时以经理的身份给若干员工发送 “今天的工作情况如何？”

### 高安全性与低延迟

我们的匿名通信网络系统在一台服务器上对于6万用户仅有1.45秒的通信延迟。相比于当前的匿名通信系统如Karaoke[OSDI’18]、XRD[NSDI’20]、Addra[OSDI’21]等,我们的产品同时达到了可支持大量用户、密码级的安全性与即时通信可接受的延迟，比这些工作还要优异。并且我们的安全性是建立在即使所有服务端均不可信，即使所有服务端均被攻击成功，我们的系统仍然有较高的安全性保障以及较低的延迟。不仅可以应用于匿名通信系统，也可以应用于其他攻击者假设较高的系统中。

### 对小内存大数据的处理算法

我们设计的此算法可以在SGX有限的安全空间中保障安全性的同时对大量用户进行处理分配。对于之后使用SGX的其他系统有着重要的实际意义。在除了匿名通信系统之外的方面，随着信息技术的迅速发展与广泛应用, 人类社会已经进入了一个崭新的互联网时代，根据摩尔定律，数据量也在短期内以指数级提升。随着大数据的到来，如何在有限的安全空间中高效地处理远超安全空间的数据量是一个挑战。我们设计的算法在此方面有着很好的效果。

### 系统的可靠性、移植性与可扩展性

经过第三章测试，我们系统的安全性高、通信延迟少，可靠性能够得到较为充分的保障。整个系统能够实现发送用户消息，数据上传到服务端进行分配和处理，易于移植。由于该系统具备如上所述的移植性和扩展性，因此其推广方便可行，适合在现实生活中应用。开发者能够方便地在目前系统的基础上进行改进和新功能的开发，系统的更新和维护也易于操作。

### 高适用性与用户体验

在当下的信息化社会中，安全性越来越被用户所注意，每个人都不想自己的信息遭到泄露，尤其是在通信中，但是对于一个通信系统而言，用户体验也尤为重要。对于以前的匿名通信系统，难以保证消息的时效性和实时相应的能力，同时由于可能需要每台机器上拥有SGX，使用场景也非常有限，可能你进行匿名通信拨号，等待了很久才接通了拨号，拨号的另一端传来的却还是好几分钟之前对方的信息。这会让大部分用户很快就失去了耐心。而我们的匿名通信系统的相应时间大大缩短（测试结果详见第三章），同时不需要任何一个用户终端拥有SGX技术，只需要服务器端拥有即可。这能保证足够高的用户体验质量和商业化能力，向真正的匿名通信在现实世界中得以实现迈出了关键的一步。

# 总结

## 项目总结

基于SGX的匿名通信网络系统是一个具有高安全性的通信网络系统。在保护用户通信内容的同时还能够保护用户的元隐私数据，使得攻击者在窃取信息时既不知道你的通信内容是什么，还不知道你在跟谁通话，以及什么时候进行了通话。我们最终测试此通信系统，在一台服务器下，对于6万用户，通信延迟仅为1.45秒，即使在所有服务器都向攻击者妥协的情况下，都不会暴露任何通信双方的通信关系。这使得对隐私保护有高需求的用户可以使用我们的通信网络进行实时交流并且保护住他们的隐私。

我们首先接受客户端发送来的消息按顺序放到外存中构成消息队列，然后我们每次取一定数量的消息包放入enclave这片理论上完全安全的内存空间进行以下操作：解密获取接收方UID，得到对应的组号，然后根据这个组好对消息进行分组补齐，然后添加到对应组的消息队列里面。通过以上操作，我们实现了安全性的同时不需要验证洗牌。这不仅去除了之前工作由于验证洗牌导致的通信延迟，还使得攻击者完全无法知道每个人发的消息究竟是发给谁的，达到了我们的目的。攻击者就算是在不断尝试注入、窃听等攻击方式来攻击服务器，都无法得到他们想要的通讯元数据，即到底是哪两个人在通信。

我们的匿名通信系统完全可以应用在对隐私保护有高要求的领域，比如用在医疗领域保护病人的患病信息不从元隐私数据中泄露；用在军事领域防止高级特工从通信元数据得知军队的调度信息；用在举报平台使得被举报的高官无法查出举报者是谁，保护举报者的隐私；用于保密网络通信系统，使任务仅仅让最关键的人知道。想要达到以上目的，都只需使用我们的匿名通信网络即可，无任何其他花费和设施需求。

## 心得体会

经过四位组员的共同努力和指导老师的悉心指导，我们最终完成了基于SGX的匿名通信网络相关项目研究与实现。经过测试，我们的匿名通信网络在达到可证明安全性的同时通信延迟很低，各项指标都达到了预期，不过由于我们无法找到百万级以上的志愿者进行系统测试，所以无法从实际上得知我们是否能支持大量用户同时实用。在理论上，我们的开销很小，完全可以提供给百万级以上人数同时使用。

在项目过程中，我们四位组员通力合作，分工明确，计划合理。在线上每日积极讨论的同时还会进行每周两次的线下交流，确定项目进度、讨论面临问题。我们从一起收集论文开始，总结相关工作，提出观点看法，定义待解问题，进行相关实验，讨论测评方案。在这过程中，我们的知识储备和动手能力都有了很大的提高，更重要的是我们在处理问题时，学会了如何去寻找问题的突破点，尽快地抓住问题的核心。

当然，项目进行的过程并不是一帆风顺的，无论是在实验，写代码，找解决方案，还是测评，部署，我们都遇到了很多的技术难题。在这，我们也要感谢我们的指导老师给予了我们专业的技术指导和相关资源。老师的耐心指点和帮助使得我们的项目能够如期推进。

## 未来工作

### 待实现目标

现阶段，我们已经初步完成了基于SGX的匿名通信网络的相关开发。但为了更好的投入社会使用，我们还需要进一步的测试和完善：

（1）测试志愿者的数量我们现阶段是没有达到百万级以上的，所以我们需要一些激励措施或合作项目来为我们的通信系统进行测试。测试中我们需要计算详细的开销数据，以此进行市场分析，来决定最后我们的应用场景可能有哪些。

（2）假设能够投入使用，由于我们初期的使用人数很少，将会导致攻击者有很大的概率猜出通信双方是谁。举一个极端的例子，攻击者一旦得到了匿名通信系统中的用户总人数，如果使用系统的只有两个人，那么只要有消息发送，那么一定是这两人在联系。所以，安全性是要基于大量的使用群体的。于是我们想在应用程序中加入“虚假用户”，比如做一个浏览器插件，获取一些我们招募到的志愿者在打开浏览器时的ip地址作为发送源。这点需要我们后续的实现。

（3）目前我们的通信系统只针对文字消息这一种交流介质，而不包含语音、视频等介质的交流。为满足更多用户的更多交流需求，我们预计在未来将进行语音、视频通信的匿名通信网络实现。

（4）多服务器扩展：虽然我们进行了分组来使元守能够支持较多的用户，但是一台服务器的性能总归是有限的，因此我们预计在未来进行多服务器扩展，同时也要保证安全性不会因为服务器数量增多而降低。

### 展望未来

随着大数据时代的到来与通信领域的发展，人们对于自己的通信数据隐私保护将会越来越重视。在这种背景下，仅仅对通信内容加密当然不够，元隐私数据的保护将会越来越重要。

因此，我们认为我们的匿名通信网络系统在未来将不仅仅只应用于军事、医疗、举报平台等某些需要强隐私保护的领域，而是应该推广到每个人的日常通信中，做您最忠实的隐私保卫者。

# 参考文献

[1] N. Tyagi, Y. Gilad, D. Leung, M. Zaharia, and N. Zeldovich, 2017. “Stadium: A Distributed Metadata-Private Messaging System,” in Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, pp. 423–440.

1. D. Lazar et al., 2018. “Karaoke : Distributed Private Messaging Immune to Passive Traffic Analysis. ”in Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’18),Carlbad,CA,USA, pp. 711-725.
2. D. Lazar and N. Zeldovich, 2016.“Alpenhorn: Bootstrapping Secure Communication without Leaking Metadata. ” in Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation(OSDI '16), Savannah, GA, USA, pp. 571–586.
3. S. Eskandarian, H. Corrigan-Gibbs, M. Zaharia, and D. Boneh, 2021.“Express: Lowering the cost of metadata-hiding communication with cryptographic privacy,” in Proceedings of the 30th USENIX Security Symposium., pp. 1775–1792.

[5] M. Bailey and R. Greenstadt, Eds., 2021. “30th USENIX Security Symposium, USENIX Security 2021, August 11-13, 2021”.

[6] J. Van Den Hooff, D. Lazar, M. Zaharia, and N. Zeldovich, 2015 .“Vuvuzela: Scalable private messaging resistant to traffic analysis,” In Proceedings of the 25th Symposium on Operating Systems Principles(SOSP '2015). Association for Computing Machinery, New York, NY, USA , pp. 137–152.

[7] S.Angel, I. Osdi, S. Angel, and U. T. Austin, 2016. “Unobservable Communication over Fully Untrusted Infrastructure. ”in the Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI ’16), Savannah, GA, pp. 551-569.

[8] B. J. Bowe, R. Blom, and E. Freedman, 2012. “Cyber-Dissent and Power: Negotiating Online Boundaries in Repressitarian Regimes.” Int. J. Inf. Commun. Technol. Hum. Dev., vol. 4, no. 2, pp. 1–19.

[9] H. Corrigan-Gibbs, D. Boneh, and D. Mazières, 2015.“Riposte: An Anonymous Messaging System Handling Millions of Users.” in 2015 IEEE' Symposium on Security and Privacy (SP ' 2015). San Jose, CA, USA , pp. 321–338.

[10] A. Kwon, H. Corrigan-Gibbs, S. Devadas, and B. Ford, 2017.“Atom: Horizontally Scaling Strong Anonymity.” in Proceedings of the 26th Symposium on Operating Systems Principles (SOSP ' 2017) , Shanghai, China, pp. 406–422.

[11] A. Bittau et al., 2017. “Prochlo: Strong Privacy for Analytics in the Crowd.” in Proceedings of the 26th Symposium on Operating Systems Principles(SOSP ' 2017), Shanghai, China , pp. 441–459.

[12] A. Kwon, D. Lu, and S. Devadas , 2020. “XRD: Scalable messaging system with cryptographic privacy.” in Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI ’20), Santa Clara , CA , USA , pp. 759–776.

[13] D. Lazar, Y. Gilad, and N. Zeldovich, 2019. “Yodel: Strong metadata security for voice calls,” in Proceedings of the 27th ACM Sympsium on Operating Systems Principles (SOSP ' 2019), Huntsville , ON , Canada , pp. 211–224.

[14] T. K. and D. H. S. Kim, J. Han, J. Ha, 2018. “SGX-Tor: A Secure and Practical Tor Anonymity Network With SGX Enclaves,” IEEE/ACM Trans. Netw., vol. 26, no. 5 , pp. 2174–2187.

[15] I. Ahmad, Y. Yang, D. Agrawal, A. El Abbadi, and T. Gupta, 2021. “Addra: Metadata-private voice communication over fully untrusted infrastructure,” in  Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation(OSDI ' 2021). pp. 313–329.

[16] [Mahdi Jafari Siavoshani](https://dblp.uni-trier.de/pid/57/7800.html), [Seyed Pooya Shariatpanahi](https://dblp.uni-trier.de/pid/55/10543.html), [Mohammad Ali Maddah-Ali](https://dblp.uni-trier.de/pid/18/3725.html), 2021. “Private Information Retrieval for a Multi-Message Scenario With Private Side Information.” IEEE Trans., vol. 69,no. 5, pp.3235-3244.