Notes

Begining: *2021 年 2 月 18 日*
Updating: *2021 年 7 月 16 日*
Author: *Mykrobin*

Abstract

This note is a simple summary of the Papers, books, and important websites. This summary focus on system virtualization, operating system, and system security knowledge. I will write down the vital erudition I learned during the Ph.D. interval. Though this is the slowest way to pick the information, I still want to write them down.

The rest of this summary is organized as follows. In Chapter Section 1, we introduce some vital security of the system (系统安全). In Chapter Section 2, we declare the system virtualization (系统虚拟化). In Chapter Section 3, we summary the way to protect the VMs using a trusted hypervisor (基于可信虚拟机管理器保护客户机). In Chapter Section 4, we declare the existing way to protect the security of system virtualization (在非可信虚拟机管理器环境中构建可信). In Chapter Section 5, we introduce the famous attacks and the way they bypass the system detection, and the typical way to mitigate them (系统中攻击、漏洞以及缓解方式). In Chapter Section 8, we briefly introduce the system detection (系统检测). In Chapter Section 9, we state the trusted computing base (可信计算基). In Chapter Section 6, we explain the trusted platform module (可信计算模块). Finally, we write down the summary of searches (小方向调研报告) and the essential basic knowledge (专业必备基础知识) in Chapter Section 10 and Chapter Section 11, respectively. In Chapter Section 12, Table Table 12.1 illustrates the vital conferences and journals of the relevant search fields.

Keywords: System Security, Virtualization, Hypervisor, Xen, Attack, TCB, TPM, SGX, TEE.

Chapter 1. 系统安全

In Chapter Section 1, I'll introduce the System Security.

1.1 概念

This is the first question, what is System security? What does it mean?

1.2 方向一

1.3 方向二

1.4 精读论文总结

Table 1.1 系统安全精读论文表

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Table 1.2 系统安全精读论文表

序号	刊物	论文全称	时间	session
[12]	TDSC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	2016	Virtualization and Security Virtualization and Security
[12]	CCS	TSAC：Enforcing Isolation of Virtual Machines in Clouds		

1.5 历年系统安全论文总览

Table 1.3 USENIX Security'2020 论文总览

Session	序号 & 论文名称 & 论文核心
CCS	[12] TSAC：Enforcing Isolation of Virtual Machines in Cloud 基于可信Hypervisor为客户机构造可信的隔离计算环境
	[11] Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments 本文聚焦在ARM TrustZone中的一个冲突矛盾点：在开发者眼中，想要通过在可信执行环境中约束第三方软件的安装起到最小化可信计算基的目的，而第三方软想要将其安装在可信环境中。已有的解决方案： IEE 为了解决上述的矛盾点，研究者们在普通的执行环境中创建了一个隔离的执行环境，这样可以为安全敏感型应用提供保护。本文系统的研究了 IEE 的数据保护模型和 ARM cache的特征，发现了三种可以成为CITM的攻击能够泄露在IEE中保护的敏感数据。具体地，由于映射到映射到IEE内存的cache效率低以及不一致的问题存在，普通世界的攻击者可以威胁IEE中数据的安全。比如：通过并发执行操控IEE的内存，当安全敏感型应用挂起或结束时旁路掉安全机制，或者在IEE上下文切换的时候滥用不完整的安全机制。作者在三种有名的IEE系统中测试了CITM攻击，最后分析了CITM攻击的根本原因，并提出了一种防御机制。

Table 1.4 S&P'2020 论文总览

Session	序号 & 论文名称 & 论文核心
CCS	[12] TSAC：Enforcing Isolation of Virtual Machines in Cloud 基于可信Hypervisor为客户机构造可信的隔离计算环境
	[11] Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments 本文聚焦在ARM TrustZone中的一个冲突矛盾点：在开发者眼中，想要通过在可信执行环境中约束第三方软件的安装起到最小化可信计算基的目的，而第三方软想要将其安装在可信环境中。已有的解决方案： IEE 为了解决上述的矛盾点，研究者们在普通的执行环境中创建了一个隔离的执行环境，这样可以为安全敏感型应用提供保护。本文系统的研究了 IEE 的数据保护模型和 ARM cache的特征，发现了三种可以成为CITM的攻击能够泄露在IEE中保护的敏感数据。具体地，由于映射到映射到IEE内存的cache效率低以及不一致的问题存在，普通世界的攻击者可以威胁IEE中数据的安全。比如：通过并发执行操控IEE的内存，当安全敏感型应用挂起或结束时旁路掉安全机制，或者在IEE上下文切换的时候滥用不完整的安全机制。作者在三种有名的IEE系统中测试了CITM攻击，最后分析了CITM攻击的根本原因，并提出了一种防御机制。

Table 1.5 CCS'2020 论文总览

Session	序号 & 论文名称 & 论文核心
CCS	[12] TSAC：Enforcing Isolation of Virtual Machines in Cloud 基于可信Hypervisor为客户机构造可信的隔离计算环境
	[11] Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments 本文聚焦在ARM TrustZone中的一个冲突矛盾点：在开发者眼中，想要通过在可信执行环境中约束第三方软件的安装起到最小化可信计算基的目的，而第三方软想要将其安装在可信环境中。已有的解决方案： IEE 为了解决上述的矛盾点，研究者们在普通的执行环境中创建了一个隔离的执行环境，这样可以为安全敏感型应用提供保护。本文系统的研究了 IEE 的数据保护模型和 ARM cache的特征，发现了三种可以成为CITM的攻击能够泄露在IEE中保护的敏感数据。具体地，由于映射到映射到IEE内存的cache效率低以及不一致的问题存在，普通世界的攻击者可以威胁IEE中数据的安全。比如：通过并发执行操控IEE的内存，当安全敏感型应用挂起或结束时旁路掉安全机制，或者在IEE上下文切换的时候滥用不完整的安全机制。作者在三种有名的IEE系统中测试了CITM攻击，最后分析了CITM攻击的根本原因，并提出了一种防御机制。

Table 1.6 NDSS'2020 论文总览

Session	序号 & 论文名称 & 论文核心
CCS	[12] TSAC：Enforcing Isolation of Virtual Machines in Cloud 基于可信Hypervisor为客户机构造可信的隔离计算环境
	[11] Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments 本文聚焦在ARM TrustZone中的一个冲突矛盾点：在开发者眼中，想要通过在可信执行环境中约束第三方软件的安装起到最小化可信计算基的目的，而第三方软想要将其安装在可信环境中。已有的解决方案： IEE 为了解决上述的矛盾点，研究者们在普通的执行环境中创建了一个隔离的执行环境，这样可以为安全敏感型应用提供保护。本文系统的研究了 IEE 的数据保护模型和 ARM cache的特征，发现了三种可以成为CITM的攻击能够泄露在IEE中保护的敏感数据。具体地，由于映射到映射到IEE内存的cache效率低以及不一致的问题存在，普通世界的攻击者可以威胁IEE中数据的安全。比如：通过并发执行操控IEE的内存，当安全敏感型应用挂起或结束时旁路掉安全机制，或者在IEE上下文切换的时候滥用不完整的安全机制。作者在三种有名的IEE系统中测试了CITM攻击，最后分析了CITM攻击的根本原因，并提出了一种防御机制。

Chapter 2. 系统虚拟化

2.1 概念

2.2 Xen

2.2.1 Xen启动及其加载虚拟机的过程分析

https://mp.weixin.qq.com/s/rDNH2poDscFyLyMMhEhc_Q

2.3 KVM

2.4 Xen 与 KVM 对比分析

2.5 工业界虚拟化技术总结

2.6 精读论文总结

Table 2.1 系统虚拟化论文总览

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 3. 可信Hypervisor

3.1 概念

3.2 方向一

3.3 方向二

3.4 精读论文总结

Table 3.1 可信Hypervisor 论文总览

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

[12] 基于可信Hypervisor 为客户机构造可信的隔离计算环境。

Chapter 4. 非可信Hypervisor

4.1 概念

hypervisor 保护综述 [20]
hypervisor 代码完整性检测机制 [19]

4.2 方向一

4.3 方向二

4.4 精读论文总结

Table 4.1 非可信Hypervisor 论文总览

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 5. 系统攻击

5.1 概念

5.2 CFI

5.3 TOCTTOU

5.4 ROP

5.5 Runtime Page Modify

5.6 精读论文总结

Table 5.1 系统攻击论文总览

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 6. 硬件提供安全保障

利用硬件技术提供可信执行环境，从而构造TCB.

搞清楚的问题：

1. TPM是什么？

2. Intel SGX 与TPM 的区别？

3. ARM’s TrustZone？

6.1 概念

6.2 TPM

6.3 Interl SGX

6.4 Enclave 飞地

6.5 ARM TrustZone

6.6 精读论文总结

Table 6.1 硬件提供安全保障

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 7. 软件件提供安全保障

7.1 概念

序号	刊物	论文全称	时间	session

Table 7.1 硬件提供安全保障

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 8. 系统检测

8.1 概念

8.2 代码完整性检测

代码完整性检测，这里着重分为应用层代码完整性，系统层级的代码完整性和hypervisor层级的代码完整性。

检测专利 [17]

系统层代码检测 [15] 和 hypervisor代码检测 [19]

[18] 本文中的度量仅仅对符号描述表中的部分字段进行度量，通过度量信息可以保证系统的完整性!

8.3 控制流完整性检测

8.4 入侵检测

8.5 分布式检测

8.6 精读论文总结

Table 8.1 系统检测论文总览

序号	刊物	论文名称	核心思想
[12]	TC	TSAC：Enforcing Isolation of Virtual Machines in Clouds	基于可信Hypervisor为客户机构造可信的隔离计算环境

Chapter 9. 可信执行环境

9.1 概念

9.2 方向一

9.3 方向二

9.4 精读论文总结

Table 9.1 可信执行环境论文总览

序号	刊物	论文名称	核心思想
[11]	CCS	Cache-in-the-Middle (CITM) Attacks: Manipulating Sensitive Data in Isolated Execution Environments	基于Arm处理器构造可信执行环境
		SoK: Understanding the Prevailing Security Vulnerabilities in TrustZone-assisted TEE Systems	

Chapter 10. 简单调研总结

10.1 信任链

10.2 多核处理器间同步协议

10.3 Distributed Protocol

分布式计算原理会议（ACM Symposium on Principles of Distributed Computing）是分布式计算领域的国际顶级学术会议，也是中国计算机学会（CCF）推荐的B类国际学术会议。以下是对PODC重点session 的简介 [1]：

Consensus 是分布式计算理论中的核心问题。针对不同的计算模型，研究人员一直在努力提高Consensus算法的效率或者探索Consensus问题本身的难度。根据著名的FLP定理，在异步模型下，不存在确定性的、具有容错能力且满足wait-free性质的Consensus算法。为了绕开这个不可能性定理，研究人员开始考虑随机算法或者非严格（exact）的Consensus问题。

Concurrency并发理论是多个领域共同的研究热点，如分布式计算领域、多处理器领域、数据库领域、程序设计语言领域等。论文“Locking Timestamps versus Locking Objects”关注传统的数据库事务处理问题，提出了“为时间戳上锁而不是为对象上锁”的基本思想，并据此设计了一类新的多版本并发控制协议。论文“Relaxed Schedulers Can Efficiently Parallelize Iterative Algorithms”考察了非严格的（relaxed）调度器对迭代算法的并行度的影响。结果表明存在一类非严格的调度器可以高效地执行某些经典的迭代算法，例如求极大独立集或者极大匹配的贪心算法。

经典的进程通信模型有两种，一种是消息传递模型，另一种是共享内存模型。在共享内存模型中，多个进程通过访问共享对象进行通信，而无需关心更底层的消息传递细节。因此，共享内存模型具有抽象层次高、易于理解、易于使用等特点。Hagit Attiya等人的论文“Separating Lock-Freedom from Wait-Freedom”关注共享内存模型中的一个由来已久的重要理论问题：**是否所有的可以用lock-free方式实现的共享对象都存在wait-free的算法？对此，该论文给出了否定的回答**。这意味着lock-freedom与wait-freedom有着本质区别。Faith Ellen等人的论文“Revisionist Simulations: A New Approach to Proving Space Lower Bounds”研究共享内存模型中的空间复杂度问题。具体而言，论文给出了解决 obstruction-free-set agreement 问题所需共享读写寄存器（read/write register）数量的下界，这是对已知下界结果的重大改进。论文“On the Classification of Deterministic Objects via Set-Agreement Power”研究Set-Agreement问题是否能完全刻画确定性共享对象的计算能力。对此，论文给出了否定的回答。论文“Passing Messages while Sharing Memory”则另辟蹊径，将两种经典的通信模型融合起来，提出了一种混合通信模型。通过解决**经典的Consensus问题与Leader Election**问题，论文展现了混合模型相对于单个模型在可扩展性、容错性以及对异步的容忍性等方面的优势。

Algorithm Graph 分布式图算法一直以来都是分布式计算领域的研究重点，尤其是近几年，更是成为一大研究热点。今年的PODC会议议程中至少有四个以分布式图算法为主题的Sessions，涉及的图算法问题包括极大独立集、图匹配、顶点覆盖、点着色、最短路径、图的直径、平面图判定、最小割、最小生成树等。

区块链技术并非起源于分布式计算领域，但是正如Maurice Herlihy在PODC2017会议主题报告中指出的那样，围绕区块链技术产生的很多问题——比如Consensus、数据复制（replication）、容错、隐私、安全等——都是分布式计算领域中的经典问题。

Table 10.1: **PODC'2020 分布式协议**

Session	序号 & 论文名称 & 论文核心
Concurrency (并发)	[2] Brief Announcement: Why Extension-Based Proofs Fail
	Continued on next page

Table 10.1 – continued from previous page

Session	序号 & 论文名称 & 论文核心
	作者引入基于扩展的证明，这是一类不可能性证明，包括价态参数。它们被建模为一个证明者和一个协议之间的交互。使用基于组合拓扑学，它已经被证明是不可能的。确定性地求解 $n > k \geqslant 2$个过程之间的k集协约问题以免等待的方式。然而，不知道证明是否基于更简单的技术是可能的。我们解释为什么这不可能的结果不能通过扩展证明来获得。并且，基于扩展的证明在功率上是有限的。【本文具体在讲什么不清楚】
Graph Algorithms I	[4] Truly Tight-in-Delta Bounds for Bipartite Maximal Matching and Variants 本文为二部最大匹配的复杂性和许多自然变量（包括加和常数在内）提供了真正的 严格边界。本文的结果产生了Balliu等人的证明的相当简化的版本。我们证明了可以通过有界自动回合消除来获得我们的结果，该自动回合消除是Brandt [PODC’19]最新的自动回合消除技术的一种版本，从实际的角度来看，它特别适合于自动化。【图最大匹配算法】
	[3] Lower Bounds for Distributed Sketching of Maximal Matchings and Maximal Independent Sets 分布式图草绘模型：在无向图G中有一个裁判和n个顶点共享公共随机性。每个顶点v仅知道其在G中的邻域，并且裁判最初不接收任何输入。每个顶点同时向裁判发送一条消息，称为草图，然后裁判根据接收到的草图输出G上某个组合问题（例如最小生成树问题）的解决方案。以前在图形草绘上的工作表明，许多问题，包括连通性，最小生成树，边或顶点连通性，切割或频谱稀疏化以及（ Δ+1）顶点着色，都在该模型中承认高效的算法，这些算法只需要绘制草图即可。每个顶点的大小polylog（n）。本文证明了最大匹配和最大独立集这两个基本问题不承认这种有效的解决方案：我们通过在允许有限数量的参与者之间共享输入的通信模型中分析这些问题的通信复杂性来证明的结果，因此该通信模型位于标准的手头人数与头顶人数多方通信模型之间。我们的证明基于一系列使用Ruzsa-Szemerédi图和信息理论论证的硬实例，以建立通信的下界。【这篇论文比较符合当前思路】
	[9] Computing shortest paths and diameter in the hybrid network model 本文证明，所有对最短路径问题可以在 $\widetilde{O}(\sqrt{n})$ 轮中精确求解，这改进了以前的$\widetilde{O}\left(n^{2/3}\right)$轮算法，缩小了与现有算法的差距已知下限（最多n个多段对数因子）。此外，在k足够大的条件下，本文给出了k源最短路径问题（k-SSP）的常数逼近。在单源的情况下，改进了直径为D的图的算法。【这篇文章的价值在于，其描述是K源，可能会扩展到个人研究中】
Consensus	[8] Brief Announcement: Byzantine Agreement with Unknown Participants and Failures 尽管存在恶意参与者或拜占庭参与者，但仍希望就共同意见达成共识的一组参与者需要解决一个拜占庭协议问题的实例。这个经典问题已经得到了很好的研究，但是大多数现有解决方案都假定参与者知道n-系统中的参与者总数-和f-拜占庭式参与者人数的上限。在本文中，研究了一个带有拜占庭式故障的同步系统，其中参与者既不知道n也不知道f。参与者具有唯一的标识符，这些标识符不一定是连续的。对于这样的系统，本文给出了转子协调器和共识算法，两者的弹性都为n 大于 3f，这也是当参与者知道n和f时解决共识的最佳弹性。因此，即使拜占庭参与者可以说谎大约n和f，弹性也不会受到影响。
Concurrency, Self-* Algorithms and more	[6] Self-stabilizing leader election in regular graphs 在K-正则图中选主问题

Continued on next page

Table 10.1 – continued from previous page

Session	序号 & 论文名称 & 论文核心
Wireless Protocols and Graph Models	
Graph Algorithms II	
Byzantine Attacks and Consensus (拜占庭式攻击和共识)	
Coordination (协调)	[10] Perigee: Efficient Peer-to-Peer Network Design for Blockchains 区块链中的一个关键性能指标是交易被广播和确认之间的延迟(确认延迟)。虽然共识技术的改进可以导致更低的确认延迟，但确认延迟的基本下限是消息通过底层点对点（p2p）网络的传播延迟（在比特币中，传播延迟是几十秒）。比特币和其他区块链使用的事实上的p2p协议是基于随机连接的：每个节点连接到一个随机的节点子集。诱导的p2p网络拓扑结构可能是高度次优的，因为它忽略了地理距离、带宽、散列功率和计算能力在不同对等体之间的差异。作者提出一个去中心化的算法，命名为Perigee，它可以自动学习一个高效的p2p拓扑结构，纯粹基于对等体与邻居的交互来调整上述网络异质性。受多臂强盗问题的文献启发，Perigee在保留与已知关系良好的邻居的连接和探索与之前未见过的邻居的新连接之间进行了最佳平衡。 **改善传播延迟会直接改善系统的关键性能指标：交易吞吐量，确认交易的等待时间和安全性。**【本文构造了一个新型的P2P协议，但粗略看来这种协议是在一个大规模的节点情况下构造的通讯协议。】
	[7] DConstructor: Efficient and Robust Network Construction with Polylogarithmic Overhead 本文构建了一个根据节点的加入与退出，可以动态调整网络拓扑的P-2-P网络【节点的加入与退出可以借鉴一下】
Graph Algorithms CONGEST Model	[5] Single-Source Shortest Paths in the CONGEST Model with Improved Bound 单源带权值最小路径问题，讨论了分布式时间复杂度的下限问题。分布式环境中，每个节点拥有一个处理器的场景【有可能用的到，可以借鉴一下】

10.4 Code Integrity Detection

10.5 精读论文总结

Chapter 11. 必备基础知识总结

11.1 计算机体系结构

学习资源：

1. 深入理解计算机系统
2. 计算机体系结构-量化研究方法

11.2 操作系统

学习资源：

1. ECNU-OS-lab 配套实验完成
2. 操作系统导论课本学习
3. MIT-6828 课程学习，包括课程实验
4. 清华大学高级操作系统

11.2.1 分布式一致性协议

11.3 计算机网络

11.3.1 TCP/IP 协议

11.4 图论

正则图 如果图中所有点的度一样，那么就是正则图；

最大独立集 在图中选出的点都不相邻的最大集合；

最大匹配 这是二分图中的一个匹配问题，将图分为两个独立集，集合内部无边，集合之间有边，最大程度的实现两个集合的一一对应，则称为最大匹配问题。

11.5 时间自衰减信任

[16] [14]

11.6 概率图模型

概率图模型研究进展概述 [13] 中提到的贝叶斯网络比较符合当前的信任传播模型。

概率图模型作为一类有力的工具,能够简洁地表示复杂的概率分布,有效地(近似)计算边缘分布和条件分布,方便地学习概率模型中的参数和超参数。因此,它作为一种处理不确定性的形式化方法,被广泛应用于需要进行自动的概率推理的场合。

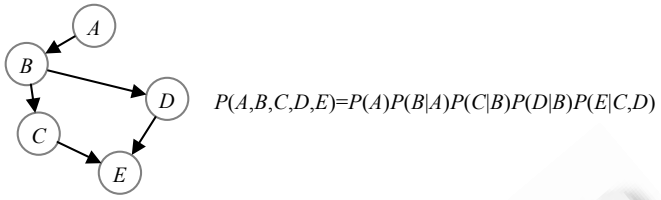

Figure 11.1 5节点贝叶斯网络模型及其联合分布

11.7 基本概念罗列

NP-hard 问题

Chapter 12. Appendix

Table 12.1 系统和安全方向的顶刊、顶会

方向	类型	刊物名称	刊物全称
安全	期刊	TDSC	IEEE Transactions on Dependable and Secure Computing
	会议	USENIX Security	Usenix Security Symposium
	会议	S&P	IEEE Symposium on Security and Privacy
	会议	CCS	ACM Conference on Computer and Communications Security
	会议	NDSS	Network and Distributed System Security Symposium
体系结构	期刊	TC	IEEE Transactions on Computers
	会议	PPoPP	ACM SIGPLAN Symposium on Principles & Practice of Parallel Programming
	会议	FAST	Conference on File and Storage Technologies
	会议	ASPLOS	International Conference on Architectural Support for Programming Languages and Operating Systems
	会议	USENIX ATC	USENIX Annul Technical Conference
	会议	VEE	International Conference on Virtual Execution Environments
	会议	EuroSys	European Conference on Computer Systems
	会议	HPDC	International Symposium on High Performance Distributed Computing
	会议	SOCC	ACM Symposium on Cloud Computing
	会议	ISCA	International Symposium on Computer Architecture
系统	期刊	JSAC	IEEE Journal on Selected Areas in Communications
	会议	NSDI	Symposium on Network System Design and Implementation
	会议	SOSP	ACM Symposium on Operating Systems Principles
	会议	OSDI	USENIX Symposium on Operating Systems Design and Implementations
数据库	期刊	TKDE	IEEE Transactions on Knowledge and Data Engineering
	会议	SIGMOD	ACM Conference on Management of Data
	会议	ICDE	IEEE International Conference on Data Engineering
	会议	VLDB	International Conference on Very Large Data Bases

Chapter 13. Sample

长表格示例

Table 13.1: A sample of long table.

First column	Second column	Third column
名称	语义解释	案例说明
名称	语义解释	案例说明

References

[1] Introduction to podc conference. `https://zhuanlan.zhihu.com/p/47208633`, 2018. Section 10.3

[2] Dan Alistarh, James Aspnes, Faith Ellen, Rati Gelashvili, and Leqi Zhu. Brief announcement: Why extension-based proofs fail. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 54–56, 2020. Table 10.1

[3] Sepehr Assadi, Gillat Kol, and Rotem Oshman. Lower bounds for distributed sketching of maximal matchings and maximal independent sets. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 79–88, 2020. Table 10.1

[4] Sebastian Brandt and Dennis Olivetti. Truly tight-in-δ bounds for bipartite maximal matching and variants. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 69–78, 2020. Table 10.1

[5] Shiri Chechik and Doron Mukhtar. Single-source shortest paths in the congest model with improved bound. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 464–473, 2020. Table 10.1

[6] Hsueh-Ping Chen and Ho-Lin Chen. Self-stabilizing leader election in regular graphs. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 210–217, 2020. Table 10.1

[7] Seth Gilbert, Gopal Pandurangan, Peter Robinson, and Amitabh Trehan. Dconstructor: Efficient and robust network construction with polylogarithmic overhead. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 438–447, 2020. Table 10.1

[8] Pankaj Khanchandani and Roger Wattenhofer. Brief announcement: Byzantine agreement with unknown participants and failures. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 178–180, 2020. Table 10.1

[9] Fabian Kuhn and Philipp Schneider. Computing shortest paths and diameter in the hybrid network model. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 109–118, 2020. Table 10.1

[10] Yifan Mao, Soubhik Deb, Shaileshh Bojja Venkatakrishnan, Sreeram Kannan, and Kannan Srinivasan. Perigee: Efficient peer-to-peer network design for blockchains. In *Proceedings of the 39th Symposium on Principles of Distributed Computing*, pages 428–437, 2020. Table 10.1

[11] Jie Wang, Kun Sun, Lingguang Lei, Shengye Wan, Yuewu Wang, and Jiwu Jing. Cache-in-the-middle (citm) attacks: Manipulating sensitive data in isolated execution environments. In *proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1001–1015, 2020. Table 1.3, Table 1.4, Table 1.5, Table 1.6, Table 9.1

[12] Chuliang Weng, Jianfeng Zhan, and Yuan Luo. TSAC: Enforcing isolation of virtual machines in clouds. *IEEE Transactions on Computers*, 64(5):1470–1482, 2014. Table 1.1, Table 1.2, Table 1.2, Table 1.3, Table 1.4, Table 1.5, Table 1.6, Table 2.1, Table 3.1, Section 3.4, Table 4.1, Table 5.1, Table 6.1, Table 7.1, Table 8.1

[13] 张宏毅, 王立威, and 陈瑜希. 概率图模型研究进展综述. *软件学报*, 24(11):2476–2497, 2013. Section 11.6

[14] 李佳伦, 谷利泽, and 杨义先. 一种具有时间衰减和主观预期的 p2p 网络信任管理模型. *电子与信息学报*, 31(11):2786–2790, 2009. Section 11.5

[15] 王昌舒. 安卓系统敏感事件截获与系统层代码完整性校验方法研究. Master's thesis, 西安理工大学, 2019. Section 8.2

[16] 石志国, 贺也平, and 张宏. 一种对等计算安全性的时间自衰减信任管理算法. *计算机研究与发展*, 44(1):1, 2007. Section 11.5

[17] 翁楚良 and 顾佳男. 面向虚拟化系统的hypervisor完整性检测方法, 2019. Section 8.2

[18] 陈志锋, 李清宝, 张平, and 王炜. 基于内存取证的内核完整性度量方法. *Journal of Software*, 27(9), 2016. Section 8.2

[19] 顾佳男. 面向 hypervisor 代码完整性的分布式检测机制. Master's thesis, 华东师范大学, 2020. Section 4.1, Section 8.2

[20] 顾佳男, 郑蓓蕾, and 翁楚良. 面向云平台非可信hypervisor的保护机制综述. *计算机科学与探索*, (2):200–214, 2020. Section 4.1