Group Meeting

Members: Yong Li, Depeng Liu, Weizhi Feng, Xie Li, Shizhen Yu, Yutian Zhu, Zongxin Liu

2021 年 7 月 28 日

差分隐私——刘德鹏

这周:

- Pufferfish 杂志：基于 POPL 文章修改，当时对比 CCS' 18 的测试工具，不知是否加入新工具的比较（20 年的新工具扩展性强，但没有处理离散机制，比较可能不占优势）；
- 读完 Proving that Programs Are Differentially Private (APLAS '19)：加入任一噪声机制后，攻击者猜测的损失函数值比起不加扰动时反而变小，即更加能猜出原 secret 的值，有点不合常理；初步原因在于攻击者知道数据分布、所加机制、以及观察对应的数据分布，发现反例不符仍需进一步证明；
- 参加讨论班，阅读 cav 文章报告。

计划:

- Pufferfish 期刊整理投稿，新实验？期刊未定（FAC, I&C 等..）；
- 继续研究文章中的问题，与隐私参数选取，utility 有关；
- DP 模型：RL 相关 model free/ model learning 内容学习。

内存安全

本周:

- 继续对函数调用的符号执行进行 Debug，能跑通一个函数调用的例子。
- 阅读 Principles of Program Analysis，进行了报告和讨论。
- CAV 报告分享。
- 其他事务性工作。

TODO:

- 根据 SV-COMP 加入库函数语义的支持。
- 代码的重构。
- 继续跑例子进行测试和 Debug。
- 下周三报告以及其他文献阅读和调研。

冯维直

- 期刊文章：
  - 上周计划是将 SDBA 取补算法正确性证明写完，然后增加一个 determinization (确定化，给定 SDBA，构建一个 DRA) 的章节，目前完成进度：没有写完证明，但是增加了一些 intuition 和定义，方便证明里面叙述更清楚；determinization 部分把构造写了，但是还没有写证明.
  - 实验部分之前做了一些简单实验，现在和 Andrea 一起对更多自动机的例子进行实验，和 Andrea 讨论用那些工具和命令进行比较，输入的 benchmark 中自动机的形式等.
- 阅读
  - 看视频读 CAV 文章：1. 决策树学习生成 ranking function 这篇进行了报告；2. 另一篇是用 decoupled search 做 composed Büchi automata 的 liveness verification, 感觉难一点，看了一部分，还没看懂.
- 计划
  - 期刊文章第 6 章写完，和 Andrea 继续做实验，根据实验情况开始写第七章.
  - 没读懂的 CAV 文章继续读完.

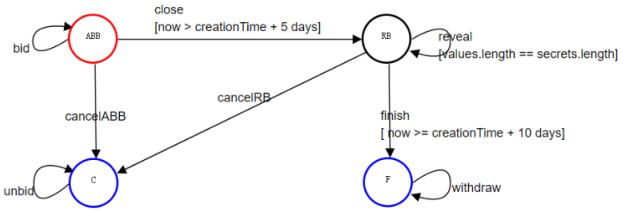
Shizhen Yu

进展：

- 上周略读了一篇 CAV 文章《Reflections on Termination of Linear Loops》，已经分享过一个简单版本
- 正在读李老师安排我读的一篇 DBA active learning 的文章《Learning Deterministic Automata on InfiniteWords》，尽量这周读完与李老师讨论
- 计划：
  - 精读一下《Reflections on Termination of Linear Loops》主要是看一下文章中几个结果的证明
  - 跟进一下静态分析讨论班的进度，补一下已经讲过的《Principles of Program Analysis》章节

朱雨田

- 调研 FSolidM 工具（用于建模）

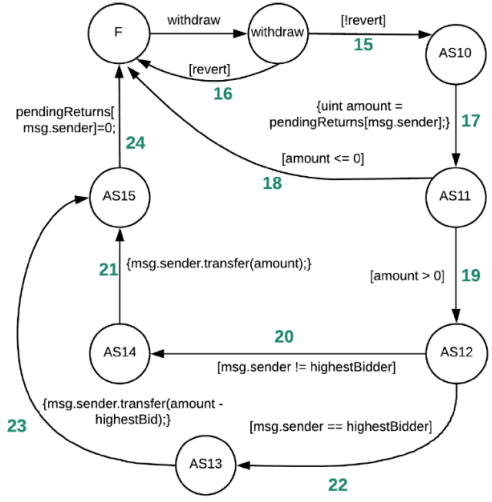



```
//Transition withdraw
function withdraw (
//Insert function input

) public
//Insert tags

//Insert function output


{
  require(state == States.F);

  //Insert guards




  //State change
  state = States.InTransition;
  //Insert statements

      uint amount = pendingReturns[msg.sender];
      if (amount > 0) {
         if (msg.sender!= highestBidder)
         msg.sender.transfer(amount);
      else
         msg.sender.transfer(amount - highestBid);
      pendingReturns[msg.sender] = 0;
  }



//State change
state = States.F;
```

朱雨田

- Augmented Contract(VeriSolid)

```
//Transition withdraw
function withdraw {
  //insert function input

} public
  //insert tags

  //insert function output


{
  require(state == States.F);

  //insert guards




  //insert statements
  state = States.ToTransition;
  //insert statements

      uint amount = pendingReturns[msg.sender];
      if (amount > 0) {
        if (msg.sender!= highestBidder)
          msg.sender.transfer(amount);
      else
          msg.sender.transfer(amount - highestBid);
      pendingReturns[msg.sender] = 0;
  }


//State change
state = States.F;
```



朱雨田

- 与锦龙师兄讨论了重用去年协议工程自动机建模部分的可行性。发现与 FSolidM 基本一致。

Plan:

- 继续调研 VeriSolid，考虑用自动机对智能合约建模的具体做法（方案）。
- 学习 Solidity 相关内容。
- 继续阅读 memory repair 相关文章 (option)。