Faster Secure Two-Party Computation Using Garbled Circuits

Yan Huang David Evans Jonathan Katz Lior Malka
University of Virginia *University of Maryland* *Intel**
http://MightBeEvil.org

Abstract

Secure two-party computation enables two parties to evaluate a function cooperatively without revealing to either party anything beyond the function's output. The *garbled-circuit technique*, a generic approach to secure two-party computation for semi-honest participants, was developed by Yao in the 1980s, but has been viewed as being of limited practical significance due to its inefficiency. We demonstrate several techniques for improving the running time and memory requirements of the garbled-circuit technique, resulting in an implementation of generic secure two-party computation that is significantly faster than any previously reported while also scaling to arbitrarily large circuits. We validate our approach by demonstrating secure computation of cir-

The second approach relies on completeness theorems for secure computation [7, 8, 34] which give protocols for computing *any* function f starting from a Boolean-circuit representation of f. This generic approach to secure computation has traditionally been viewed as being of theoretical interest only since the protocols that result require several symmetric-key operations per gate of the circuit being executed and the circuit corresponding to even a very simple function can be quite large.

Beginning with Fairplay [22], several implementations of generic secure two-party computation have been developed in the past few years [11, 21, 27] and used to build privacy-preserving protocols for various functions (e.g., [4, 13, 16, 26, 29]). Fairplay and its successors demonstrated that Yao's technique could be implemented to run in a reasonable amount of time for small circuits

这是一篇关于pRIDE所用技术的检索结果

1.路网嵌入（ROAD NETWORK ENBEDDING）

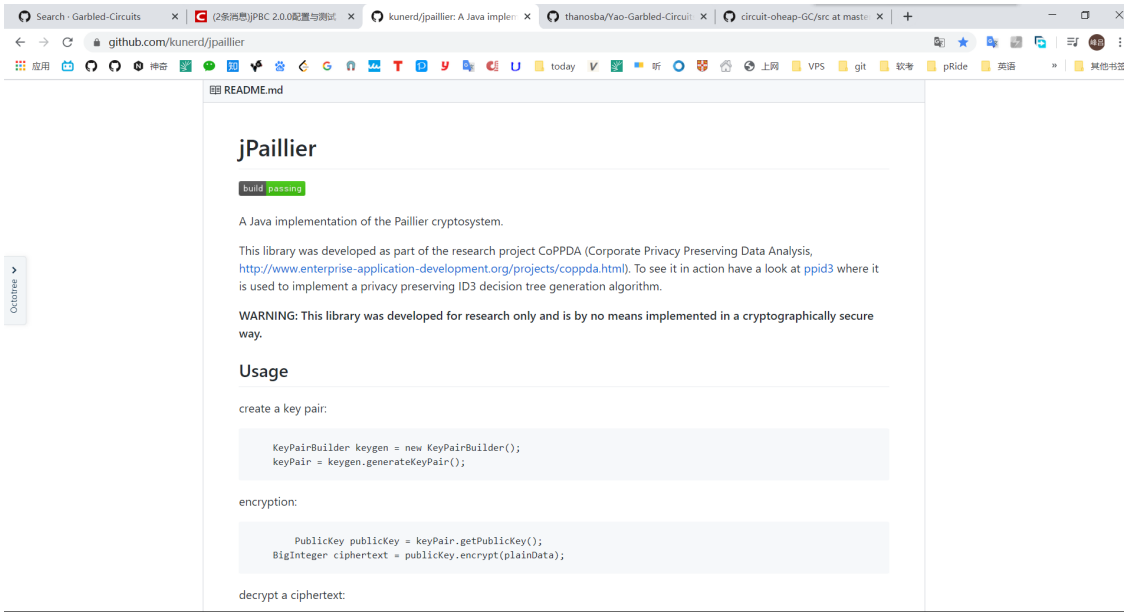
目前的思路是自己从百度地图(google地图)查，然后存下来

至于用什么形式存，555我也不清楚

2.同态加密

pRIDE论文有提到两个链接，我觉得挺好的，应该可以直接用

- github-上的代码-jPaillier https://github.com/kunerd/jpaillier



- 论文还提到一个JDBC（不知道是用来干啥的）

jPBC 2.0.0配置与测试（补充版）

原创 刘巍然-BUAA 最后发布于2014-04-12 14:10:02 阅读数 15974 ☆ 收藏 展开

题注

随着技术博客中的文章越写越多，越写越有经验，我也越来越喜欢把各种各样自己做的有意思的东西公开给大家了~通过技术博客也认识了全国各地的朋友们，他们涉及到的领域真是包罗万象：有做设计的，有做算法的，有做密码学的，还有很多很多好玩的，有趣的人。这也是我希望看到的：大家把自己的技术，自己的心得与其他人分享，一起努力一起进步！

这一篇博客实际上是一位技术朋友希望我能撰写的。他和他团队的小伙伴们在使用jPBC这个Library的时候遇到了很多的问题，希望如果有可能的话，能得到我的帮助。帮助倒是谈不上，能互相交流一下，玩一玩技术也是不错的~ jPBC我以前也配置过，不过那个时候其稳定版本是1.2.1，还并不是特别好用。这次再配置一遍，发现2.0.0的使用比1.2.1方便得多啊，而且竟然可以在Windows下面配置了！所以我赶快做了几个测试，并把我配置的全部过程分享给大家，希望能对大家的配置和开发有所帮助。

背景

Pairing-Based Cryptography

只要是最近10年做密码学的人，就不可能不知道传说中的Pairing-Based Cryptograhphy（PBC）。在现代密码学中，人们常常用两种群来构造安全的密码学算法，即类似RSA的\phi (N)群，以及离散对数群。因为各种各样的原因（具体原因太理论了，在此我不详细展开，有兴趣的朋友们可以单独联系我，我们可以讨论一下~），类似RSA的\phi (N)群基本已经淘汰不用，而离散对数群得到了广泛的应用。

大家可能有所了解，\phi (N)群安全的最大基础是，N=p*q，其中p和q是两个大质数，且给定N，很难分解为p*q。这也就是我们常说的

3.混淆电路Garbled Circuit

- 混淆电路介绍，知乎上的，讲的海星

  https://zhuanlan.zhihu.com/p/41172002

- pRIDE论文里提到了他们用的是一篇论文提出的加速版安全双方计算混淆电路

  Faster Secure Two-Party Computation Using Garbled Circuits

Faster Secure Two-Party Computation Using Garbled Circuits

Yan Huang *University of Virginia* David Evans *University of Virginia* Jonathan Katz *University of Maryland* Lior Malka *Intel**

http://MightBeEvil.org

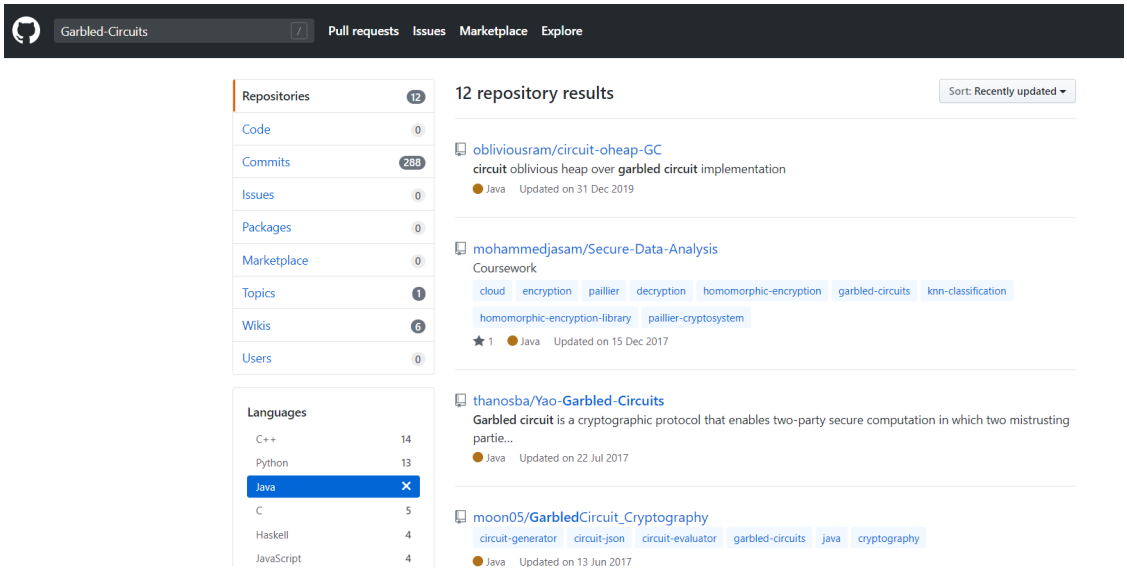
Abstract

Secure two-party computation enables two parties to evaluate a function cooperatively without revealing to either party anything beyond the function's output. The *garbled-circuit technique*, a generic approach to secure two-party computation for semi-honest participants, was developed by Yao in the 1980s, but has been viewed as being of limited practical significance due to its inefficiency. We demonstrate several techniques for improving the running time and memory requirements of the garbled-circuit technique, resulting in an implementation of generic secure two-party computation that is significantly faster than any previously reported while also scaling to arbitrarily large circuits. We validate our approach by demonstrating secure computation of cir-

The second approach relies on completeness theorems for secure computation [7, 8, 34] which give protocols for computing *any* function f starting from a Boolean-circuit representation of f. This generic approach to secure computation has traditionally been viewed as being of theoretical interest only since the protocols that result require several symmetric-key operations per gate of the circuit being executed and the circuit corresponding to even a very simple function can be quite large.

Beginning with Fairplay [22], several implementations of generic secure two-party computation have been developed in the past few years [11, 21, 27] and used to build privacy-preserving protocols for various functions (e.g., [4, 13, 16, 26, 29]). Fairplay and its successors demonstrated that Yao's technique could be implemented to run in a reasonable amount of time for small circuits

- github的搜索结果 https://github.com/search?q=Garbled-Circuits



- github-某个java仓库，我觉得挺好的

  https://github.com/thanosba/Yao-Garbled-Circuits

Yao-Garbled-Circuits JAVA

Garbled circuit is a cryptographic protocol that enables two-party secure computation in which two mistrusting parties can jointly evaluate a function over their private inputs without the presence of a trusted third party. Andrew Yao of Stanford University first proposed the protocol in 1986 to solve his famous Millionaires' Problem.The protocol is also known as Yao's protocol, Yao's garbled circuit protocol or simply the GC protocol in the cryptography literature. In the garbled circuit protocol, the function has to be described as a Boolean circuit.