Diving into the zk-snarks setup phase

2022年9月20日 星期二

15:37

- https://zkproof.org/2021/06/30/setup-ceremonies/#
- https://electriccoin.co/blog/the-design-of-the-ceremony/
- https://secbit.io/blog/2020/01/08/nizk-by-crs/

Background:

- CRS (Common Reference String): prover 在构造 NIZK(non-interactive zero-knowledge) 证明之前由一个受信任的第三方产生的随机字符串，CRS 必须由一个受信任的第三方来完成，同时共享给 prover 和 验证者 verifier。第三方不直接参与证明，但要确保随机字符产生过程的可信。
- Trusted-Setup, simply implies that an entity that is trusted should generate the keys.
- SRS (Structured Reference String): creating a pre-processing phase that encodes the relation into the CRS, called a SRS, which is available to both the prover and the verifier.

三种不同的方法：确保 verifier 知道什么被证明了。

1. **Verifier gets the full statement**

- The verifier uses an explicit representation of the relation, which means that the verifier running time is at least linear in the relation size.
- 例如：Bulletproofs: Best for small statements.
- Uniform Reference String (URS).
- **It gets the prover and verifier to agree on some randomness and public parameters**

2. **Verifier gets a succinct representation of the statement**

- The verifier uses a succinct representation of a uniform relation, meaning that there is a basic representation of a computation that is repeated many times.
- Random oracle: https://www.4hou.com/posts/E6Dv
- 随机预言机模型中的安全性证明。 所有参与者(Encryptor、 Decryptor 和对手)都可以调用随机预言，后者为它们计算哈希函数并向所有参与者提供一致的响应。

- 哈希是由一个叫做“预言”的新魔法团体完成的。 如果任何一方需要对某些内容进行哈希操作，他们需要将消息(通过一个虚构的安全通道)发送给预言模型，由 预言模型计算哈希并将其发回给他们。
- 随机语言的两个限制：1. 它必须实现一个真正的随机函数；2. 语言给出的回答必须是一致的。

3. **Pre-processing**

- Snarks achieve succinctness for any relation, even if it does not have a repetitive structure and is not a short statement.
- Pre-processing stage digests all the relation into a short string of bits.
- **Groth16 算法：**
- https://eprint.iacr.org/2016/260.pdf
- https://learnblockchain.cn/2019/05/27/groth16

4. **So what is the problem?**

- Zero-knowledge systems require the use of some randomness to represent the challenge that the verifier sends to the prover.
- **The randomness must be established in advance and in advance and its generation be publicly verifiable.**
- **Toxic waste:**

Jens Groth 在 2010 年基于 KEA（Knowledge of Exponent Assumption） 假设与 Pairing 提出了一种新的 NIZK Arguments 方案[Gorth10b]，这也是后续许许多多 zkSNARKs 方案的起点。*这里的 CRS 由一对对的 （g^x^n, g^αx^n） 构成，被用来实现「知识承诺」。其中 x 与 α 是两个随机数，在产生完 CRS 之后，必须被「遗忘」*。有些人把这部分需要遗忘的随机数叫做 toxic waste

- Multiparty Computation (MPC):

https://en.wikipedia.org/wiki/Secure_multi-party_computation
a subfield of cryptography with the goal of creating methods for parties to jointly compute a function over their inputs while keeping those inputs private.

- Implementation Track Proceeding:

Implementation Track proceeding

5. **The Zcash MPC Ceremonies**

5.1. Sprout:

- 一组六人参与者；在世界的不同地方；在线三天生成 random toxic waste in a round robin protocol
- Vulnerability：
  - there were extra elements in the SRS that were not needed and that actually enabled the prover to trick the verifier into unknowingly verifying a different statement.
  - The vulnerability is not present in：

[PGHR13] (which underlies [BCTV14]), nor in [BCGTV13], [GM17] or [BG18], （因为这些算法没有依赖 SRS）

5.2. Sapling:

- **Phase1:Powers of Tau:**
  - The first ceremony generates most of the random toxic waste, independently of the circuit to be used, which enables re-usability of the powers of tau. (more than 80 parties)
  - Each of the participants wrote an attestation describing what was done during their computation.
- **Phase2:The circuit-dependent MPC key generation**

6. **Subversion Resistance:**

- Subversion soundness resistance (S-SND):

which means that the proof scheme is such that there is no way (even in theory) to generate a **backdoored SRS that would allow creating fake proofs.**

- Subversion zero-knowledge resistance (S-ZK)

which is the fact that there is no way to generate a malicious SRS that would cause provers **to inadvertently leak information to a verifier about their private inputs to the statement** (violating zero-knowledge).

7. **The Real Trade-Off**

- One needs to compute a new pair of keys for every new relation that

Is used. (在部署和升级系统的时候会带来问题)

- 方法一：

https://eprint.iacr.org/2013/507.pdf （BCGTV13）

[BCTV14]

- 方法二：Universal Circuit（UC）

A UC can embed any circuit up to a fixed sized, representing any proving relation of that size.

- 方法三：the updateable CRS

This model allows for a structured CRS to be updated dynamically, even after generating some keys. co