Multi-Party Computation

[IK00] Randomizing Polynomials: A New Representation with Applications to Round-Efficient Secure Computation

动机：

如果函数可以用低次数的多项式表示，则可以进行low round complexity MPC. 正常来说，大部分boolean函数不能低次数多项式表示，因此考虑加入randomness。

结论：

- 低次数randomizing polynomials的存在性和局限性：
  - Degree-3 randomizing polynomials are sufficient to represent any function f over any finite field; the number of outputs, s, and the number of random inputs, m, are at most quadratic in the branching program size of f.
  - Degree-2 randomizing polynomials cannot represent any Boolean function, except those defined by systems of linear equations and those which can be represented by standard degree-2 polynomials.
- round efficient protocol存在性：
  - Two (respectively, three) communicationrounds are suf- ficient to evaluate any k-argument function f with per- fectinformation-theoretic $\lceil\frac{k-1}{3}\rceil$-privacy (resp., $\lceil\frac{k-1}{2}\rceil$-privacy), probabilistic correctness, and communication complexity which is at most quadratic in the branching program size of f and the number of parties k.
  - 或者，也可以得到zero error probability (perfect correctness), perfect privacy, *expected* $2+\epsilon$ (resp. $3+\epsilon$) rounds, 对任意小的ϵ.

constant secure computation protocols:

- J. Bar-Ilan and D. Beaver. Non-cryptographic fault-tolerant computinginaconstantnumberofrounds. 1989.
  对于NC^1函数构造了高效的，（expected）常数轮，信息论安全，optimal security threshold 的协议。

- U. Feige, J. Kilian, and M. Naor. A minimal model for se- curecomputation(extendedabstract). 1994
  Y. Ishai and E. Kushilevitz. Private simultaneous messages protocols with applications. 1997
  R. Cramer and I. Damgbrd. Securedistributed linear algebra in a constant number of rounds. 2000.
  三篇文章扩展了可以计算的函数类。
- [BB89] 和 [FKN94] 都将f的计算规约为有限群中元素的乘积。但是需要群元素和它们的逆元的分布式生成，用到interactive inversion subprotocol，造成额外的round complexity。
- In the current work we utilize a different, inversion-free, randomization approach, extending a technique from [IK97]. 并且不需要其中的私密可逆矩阵分布式生成步骤。

Randomized Polynominials 定义：

向量p(x, r)被称为函数$f : \{0,1\}^n \to \{0,1\}$ 的randomized polynomial，如果存在概率分布 D_0, D_1 使得：一方面(privacy)，输出分布只取决于 $f(x)$ 即 对任意输入 $x \in \{0,1\}^n, P(x) = D_{f(x)}$, 另一方面(correctness)，分布 D_0 & D_1 统计距离远（statistically far）本文中要求 $SD(D_0, D_1) \geq 1/2$

三个参数：

1. degree of the polynomial vactor p: 次数就是 p 的每一个分量 p_i 中，关于 x & r 的分量 x_i 和 r_i
2. output complexity: 向量 p 的长度
3. randomness complexity: 向量 r 的长度

remark：

- 虽然没说，但是本文中的构造都是可以高效计算的
- 如randomizing polynomial的计算和efficient distinguisher
- 常数 $1/2$ 是随便选的，可以amplified to $1-\epsilon$
- **问题**：为什么统计距离大就可以高效的reconstruct？

MPC

Some Points

- 有两种模型（settings）： the secure channel model and the computational model.
- 安全性定义（simulation）： 直观上，"This is formalized by requiring that whatever an adversary can achieve (and learn) in the “real-life” execution of the pro- tocol, it could have also achieved in an ideal model, where a trusted party is being used to perform the computation.",
  "This can be formalizedby requiringthe existenceof a probabilistic simulator algorithm S, satisfying the following condition. For any input y and collusion B of at most t parties, the

output generated by S on input $(B, y_B, f(y))$, where y_B denotes y restricted to its B-entries, is distributed identically to the joint view of parties from B in the execution of F on input y (including the inputs, random inputs, and communication)"

- An *active* adversary is allowed to maliciously alter the behavior of the parties it corrupts, whereas a *passive* adversary only learns their view of the protocol.
- 本文中考虑passive adversary，ϵ-correctness，t-privacy

Branching Programs

Definition of non-deterministic mod-p branching program:

A(non-deterministic) mod-p branching program is defined by a quadruple $BP = (G, \Phi, s, t)$, where $G = (V, E)$ is a directed acyclic graph, Φ is a labeling function assigning each edge a negative literal x_i^0, a positive literal x_i^1 the constant 1, and s, t are two special vertices.

Each input assignment $x = (x_1, ..., x_n)$ naturally induces an unlabeled subgraph G_x , whose edges include everye $e \in E$ such that $\Phi(e)$ is satisjied by x.

The function computed by BP is the Boolean function f satisfying $f(x) = 1\, iff.$ the number of "accepting" $s - t$ paths in G_x is nonzero $modulo - p$.

Finally, define the size of BP to be $|V|$,the number of vertices in G

Definition of deterministic Branching Program:

The better known model of deterministic branching programs may be defined as the special case in which for every input x , the out-degree of every vertex in G, is at most 1 (note that in this case, the choice of p does not make a difference,and it can be fixed to be as small as 2).

Definition of standard non-deterministic branching program

A definition of standard non-deterministic branching programs may be obtained from the above by counting the number of accepting paths over the integers, rather than modulo p.

Points

- 每个函数都能用BP计算
- 有些函数效率很高（linear-size BP）
- BP的大小不会超过Boolean formula大小
- 并不是能高效计算所有polynomial-time computable函数

本文主要结果

构造 Low-Degree Randomizing Polynomials

Theorem 3.1 For a matrix $A \in K^{w,n}$ and vector $b \in K^w$, the Boolean function $f_{A,b}$ testing whether $Ax = b$ can be randomized by degree-2 polynomials, with output complex- ity 1 and randomness

complexity w .
proof: $p(x,r)=r^t(Ax-b)$ 是0或者uniform distribution。

这个定理可以给一些简单的函数实现二次的randomizing Polynomials，但是不像普通的多项式表示，可以相加由简单的来构造复杂的，randomizing polynomials做不到这一点（ 为什么？ ）。

下面是主要结果，就是要：given a modular branching program of size $l+1$ computing f , an input x can be transformed via a simple *affine transformation* into an $l\times l$ matrix M_x , such that the output value $f(x)$ directly corresponds to the rank of M.

Lemma 3.2 Let $K=GF(p)$, where p is prime, and suppose that BP is a $mod-p$ branching program of size $l+1$ computing a Boolean function $f:\{0,1\}^n\rightarrow\{0,1\}^n$. Then, there exists an affine (degree-1) transformation $L:K^n\rightarrow K^{l,l}$, such that for any $x\in\{0,1\}^n$:

- If $f(x)=1$ then the matrix $M_x =^{def} L(x)$ is of full rank (i.e., $rank(M_x)=l$}.
- If $f(x)=0$ then the matrix $M_x =^{def} L(x)$ is one less than full rank (i.e., $rank(M_x)=l-1$}.

proof 参见文章appendix A。

然后，下面两个引理告诉我们如何把这个矩阵变成一个随机矩阵，并且能区分不同的情况（即randomized encoding）：

Lemma 3.3 Let R_1,R_2 be two independent random matrices, each uniformly distributed over $K^{l,l}$. Then, for any $M,M'\in K^{l,l}$ such that $rank(M)=rank(M’)$, we have: $R_1MR_2\equiv R_1M'R_2$.

Lemma 3.4 Let R_1,R_2 be distributed as above, and suppose that $rank(M)>rank(M’)$. Then, $SD(R_1MR_2,R_1M'R_2)>0.08$.

两个结合起来得到：如果有一个boolean函数的$mod-p$ BP，大小为 l，则有degree-3 randomizing polynomial，output complexity $(l-1)^2$, randomness complexity $2(l-1)^2$

问题：为什么有efficient distinguisher ?

由此构造Round-Efficient Secure Computation

简单来说是两部reduction：

- the ϵ-correct private computation of $f(z)$ reduces to a perfectly-correct private computation of the randomized function $P(x)$
- which in turn reduces to a perfectly-correct private computation of a vector of deterministic polynomials of the same degree as p .

注意：如果要用到multi party协议中，随机输入 r 必须要分发到各个parties手中，整个 r 必须要保密，否则，知道随机输入和randomized output可能会给出不止于$f(x)$ 的信息。
为了做到这一点，我们把随机向量 r 拆分成 $r_1+\cdots+r_k$, 每个人就拿着自己的那部分输入，和一部分的随机，$f'(y'_1,\cdots,y'_k)=^{def} p(x,r_1+\cdots,r_{t+1})$ 那么次数、输入、输出复杂性，基本上都不变。这样我们有：We now argue that a private evaluation of f reduces to a private evaluation of the $degree-d$ polynomial vector f' ; that is, any $t-private$ perfectly-correct protocol for computing f' immediately translates into a $t-private\ \epsilon-correct$ protocol for computing f.
协议的构造和证明参看论文定理4.1及appendix A。

上诉结果结合起来得到：
Corollary4.2 Suppose that $f:\{0,1\}^n \to f\{0,1\}$ can be computed by a mod-p branching program of size l. Then, a *t-private k-party* evaluation of f (with a constant one-sided error) is no more expensive than a *t-private k-party* evaluation of $O(l^2)$ degree-3 polynomials on $n+O(tl^2)$ inputs over $GF(p)$.

然后再使用 Optimized variants of standard infoxmation-theoretically private protocols（BGW等？）即我们已知有如下结果：

Lemma4.3 A degree-3 polynomial vector with n inputs and s outputs over a field K can be computed t-privately by k parties with either:

- 2 rounds, $t=[\frac{k-1}{3}]$, and communication of $O(k(n+ks))$ field elements;
- 3 rounds, (optimal) privacy threshold of $t=[\frac{k-1}{2}]$, and communication of $O(k^2(n^2+s))$ field elements.

这就得到最终结论：

Corollary4.4 Let f be a k-argument function which can be computed by a mod-p branching program of size l. Then, two (respectively, three) communication rounds are suffcient to $[\frac{k-1}{3}]$ -privately (resp., $[\frac{k-1}{2}]$-privately) compute f with a one-sided error: This can be done while communicating $O(k^2l^2log1/\epsilon)$ fielde elements $(O(log(max\{k,p\}))$ - bits each), where E is the one-sided error probability.