Coded DPF Proofs

Albert Gao / sixiangg

02/27/2022

1 Context

We explore the problem of retrieving an item from a server without revealing which item is retrieved. This general class of problems is generally referred to as private information retrieval (PIR), first introduced in [1]. One class of approaches to this problem makes use of distributed point functions (DPF), first introduced in [2], with improvements in [3] and [4]. Another direction to the problem of PIR is the storage model of servers. Instead of assuming multiple servers that replicate the same data set, some recent work such as [5] looks at PIR involving servers with coded storage. We show that the DPF technique can be applied to the coded setting while maintaining polylogarithmic communication complexity.

Before adapting our approach to coded storage, we first discuss DPF in the setting of replicated storage as well as some of its extensions. Roughly speaking, suppose a client desires item index $\alpha \in \{0,1\}^m$; each server $j \in [n]$ has access to two functions $h, g_j : \{0,1\}^m \to \{0,1\}^r$ such that $h(\gamma)$ is the data at index γ and $f = \sum_j g_j$ is a coordinate function that vanishes everwhere but α, where $f(\alpha) = 1$. Then as long as each g_j by itself does not reveal α and servers cannot collude, we can compute $h(\alpha) = (h \cdot f)(\alpha) = \sum_\alpha h(\alpha) \sum_j g_j(\alpha) = \sum_j \sum_\alpha h(\alpha) g_j(\alpha)$, where each server j only communicates $\sum_\alpha h(\alpha) g_j(\alpha)$ to the client. We now make our notion precise.

Definition 1.1. (Point Function). For $m, r \in \mathbb{Z}^+, \alpha \in \{0,1\}^m, \mu \in \{0,1\}^r$, the point function $f_{\alpha,\mu} : \{0,1\}^m \to \{0,1\}^r$ is defined via

$$f_{\alpha,\mu}(\beta) = \begin{cases} \mu, & \beta = \alpha \\ 0^r, & \beta \neq \alpha \end{cases}.$$

In order to share point functions among n servers, the client generates n relatively short keys so the servers can evaluate each of the shared functions without too much trouble. Here, we assume one-way functions exist, which implies pseudorandom generators (PRGs) exist.

Definition 1.2. (Distributed Point Function: Syntax). An n-party distributed point function (DPF) is a tuple of algorithms (Gen, Eval, Rec) with the following syntax.

- $\mathsf{Gen}(1^\lambda, \alpha, \mu)$ is a PPT algorithm that outputs n keys $(k_1, \cdots, k_n)$, where 1^λ is the security parameter, $\alpha \in \{0,1\}^m$, and $\mu \in \{0,1\}^r$.
- $\mathsf{Eval}(k, \beta)$ is a PPT algorithm that outputs some $\kappa \in \{0,1\}^r$, where $\beta \in \{0,1\}^m$.
- $\mathsf{Rec}(\{(i, \kappa_i) : i \in [n]\})$ is a PPT algorithm that outputs some $\mu' \in \{0,1\}^r$.

DPF under the setting of at most a unresponsive failures (or b Byzantine servers) is where Eval may instead return empty string (or arbitrary strings) for at most a (or b) servers. We now specify what it means for a DPF to be secure.

Definition 1.3. (Distributed Point Function: Security). An n-party t-secure DPF is a tuple of algorithms ($\mathsf{Gen}, \mathsf{Eval}, \mathsf{Rec}$) with the following properties.

- **Correctness:** For any nonempty $\alpha, \mu \in \{0,1\}^*$, if $(k_1, \cdots, k_n) \leftarrow \mathsf{Gen}(1^\lambda, \alpha, \mu)$, then for any $\beta \in \{0,1\}^{|\alpha|}$ we have $\Pr[\mathsf{Rec}(\{(i, \mathsf{Eval}(k_i, \beta)) : i \in [n]\}) = f_{\alpha,\mu}(\beta)] = 1$.
- **Secrecy:** For any set $S \subseteq [n]$ of size t, there exists a PPT algorithm Sim such that for every sequence of $(\alpha_\lambda \in \{0,1\}^\lambda, \mu_\lambda \in \{0,1\}^* \setminus \{\epsilon\})$ where $\lambda \in \mathbb{N}$, the outputs of the following experiments Real and Ideal are computationally indistinguishable.
  - $\mathsf{Real}(1^\lambda) : (k_1, \cdots, k_n) \leftarrow \mathsf{Gen}(1^\lambda, \alpha_\lambda, \mu_\lambda)$. Output $(k_j)_{j \in S}$.
  - $\mathsf{Ideal}(1^\lambda)$: Output $\mathsf{Sim}(1^\lambda, |\mu_\lambda|)$.

2 Multi-Party DPF

We extend two-party DPF construction from [4] to the n-party setting.

Theorem 2.1. *An n-party 1-secure DPF under the setting of at most $n-2$ unresponsive failures (or, less than $\frac{n}{2}$ Byzantine failures) exists.*

Proof. The tuple of algorithms is specified in Algorithm 1

Algorithm 1 n-Party Distributed Point Function

Suppose $G : \{0,1\}^\lambda \to \{0,1\}^{2(\lambda+n-1)}$ is a pseudorandom generator. Notation-wise, if b is a bit, we use $\bar{b}$ to denote $b \oplus 1$.

$\mathsf{Gen}(1^\lambda, \alpha, \mu)$:

Let $\alpha_1, \cdots, \alpha_m$ be the bit decomposition of α.
Sample $s_j^{(0)} \leftarrow \{0,1\}^\lambda$ for each $j \in [n]$.
Let $t_{j,0}^{(0)} = 0, t_{j,1}^{(0)} = 1$ for $j \in [n] \setminus \{1\}$.
for $i = 1$ to n **do**
    $s_j^0 || t_{j,2}^0 || \cdots || t_{j,n}^0 || s_j^1 || t_{j,2}^1 || \cdots || t_{j,n}^1 \leftarrow G(s_j^{(i-1)})$, for each $j \in [n]$.
    $CW_{s_j} \leftarrow s_j^{\overline{\alpha_0}} \oplus s_1^{\overline{\alpha_0}}$, for each $j \in [n] \setminus \{1\}$.
    CW_{t_j}
end for

□

References

[1] Benny Chor, Oded Goldreich, Eyal Kushilevitz, and Madhu Sudan. Private information retrieval. In *Proceedings of IEEE 36th Annual Foundations of Computer Science*, pages 41–50. IEEE, 1995.

[2] Niv Gilboa and Yuval Ishai. Distributed point functions and their applications. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 640–658. Springer, 2014.

[3] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *Annual international conference on the theory and applications of cryptographic techniques*, pages 337–367. Springer, 2015.

[4] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing: Improvements and extensions. Cryptology ePrint Archive, Report 2018/707, 2018.

[5] Razane Tajeddine, Oliver W Gnilke, David Karpuk, Ragnar Freij-Hollanti, and Camilla Hollanti. Private information retrieval from coded storage systems with colluding, byzantine, and unresponsive servers. *IEEE Transactions on information theory*, 65(6):3898–3906, 2019.