SoftSpokenOT protocol

Silence Laboratories

March 2025

This protocol refers to the protocol described in Fig.10 [1] with changes in the reduction of communication by a factor of k due to an increase in computation by a factor of $2^{k-1}/k$.

The protocol uses an arbitrary stretch pseudorandom generator PRG, and a hash functions $\mathsf{H}_1 : \{0,1\}^* \to \{0,1\}^s$, $\mathsf{H}_2 : \{0,1\}^* \to \{0,1\}^\kappa$, modeled as a random oracle.
$s = 128$, $k = 4$.

Initialize:

1. R samples κ pairs of random κ-bit seed, $\{(\boldsymbol{k}_0^i, \boldsymbol{k}_1^i)\}_{i=1}^{\kappa}$
2. S samples a random $\Delta = (\Delta_1, \ldots, \Delta_\kappa) \in \mathbb{F}_2^\kappa$.
3. The parties call $\kappa \times OT_\kappa$ with inputs Δ and k_0, k_1.
4. S receives $\boldsymbol{k}_{\Delta_i}^i$, for $i = 1, \ldots, \kappa$.
5. R and S run protocols described in Fig.13 and Fig.14 [2] to convert their $\kappa \times \begin{pmatrix} 2 \\ 1 \end{pmatrix}$-OT to all-but-one $(\kappa/k) \times \begin{pmatrix} 2^k \\ 2^k - 1 \end{pmatrix}$-OT. In $\begin{pmatrix} 2^k \\ 2^k - 1 \end{pmatrix}$-OT a random function $F : \mathbb{F}_{2^k} \to \mathbb{F}_2^l$ is known to R, while S has a random point Δ and the restriction F^* of F to $\mathbb{F}_{2^k} \setminus \{\Delta\}$.

Extend:

1. According to Fig.7 [2], R computes:
$$r_x^i = \mathsf{PRG}(F^i(x)) \in \mathbb{F}_2^{l'} \quad \text{for } x \in \mathbb{F}_{2^k},$$
$$u^i = \sum_{x \in \mathbb{F}_{2^k}} r_x^i, \quad \text{and} \quad v^i = -\sum_{x \in \mathbb{F}_{2^k}} r_x^i \cdot x \quad \text{for } i \in 1, \ldots, \kappa/k.$$
2. S computes:
$$r_x^i = \mathsf{PRG}(F^{*i}(x)) \in \mathbb{F}_2^{l'} \quad \text{for } x \in \mathbb{F}_{2^k} \setminus \{\Delta\},$$

$$w^i = \sum_{x \in \mathbb{F}_{2^k} \setminus \{\Delta\}} r_x^i \cdot (\Delta^i - x) \quad \text{for } i \in 1, ..., \kappa/k.$$

3. R inputs the choice bits $x_1, \ldots, x_l \in \mathbb{F}_2$. Let $l' = l + s$, and assume that $s|l$. R picks random $x_{l+1}, \ldots, x_{l+s} \in \mathbb{F}_2$ and sets $\mathbf{x} = (x_1, ..., x_{l'})$.

   R computes:
$$u^i = u^i + \mathbf{x} \quad \text{for } i \in 1, \ldots, \kappa/k\,,$$
$$\chi_j = \mathsf{H}_1(j||u) \quad \text{for } j \in 1, \ldots, m.$$

   Consistency check: Let $m = l/s$. We divide l' OTs into $m+1$ blocks of s bits, writing $\mathbf{x} = (\hat{x}_1, \ldots, \hat{x}_{m+1}) \in \mathbb{F}_{2^s}^{m+1}$, and similarly $v^i = (\hat{v}_1^i, \ldots, \hat{v}_{m+1}^i) \in \mathbb{F}_{2^s}^{m+1}$. Then R computes the following values over $\mathbb{F}_{2^s}$:

$$x = \sum_{j=1}^{m} \hat{x}_j \cdot \chi_j + \hat{x}_{m+1}, \quad t^i = \sum_{j=1}^{m} \hat{v}_j^i \cdot \chi_j + \hat{v}_{m+1}^i \quad \text{for } i \in 1, \ldots, \kappa/k$$

   and sends u^i, x, t^i to S.

4. S computes:
$$\mathbf{q}^i = \Delta^i \cdot u^i + w^i, \quad \text{for } i \in 1, \ldots, \kappa/k$$
$$\mathbf{q}^i = (\hat{q}_1^i, ..., \hat{q}_{m+1}^i) \in \mathbb{F}_{2^s}^{m+1}$$
$$\chi_j = \mathsf{H}_1(j||u), \quad \text{for } j \in 1, \ldots, m$$
$$q^i = \sum_{j=1}^{m} \hat{q}_j^i \cdot \chi_j + \hat{q}_{m+1}^i, \quad \text{for } i \in 1, \ldots, \kappa/k$$

   and checks that $q^i = t^i + \Delta^i \cdot x$, for all $i \in 1, \ldots, \kappa/k$. If any check fails, output **AbortAndBanParty.**

Transpose and randomize:

1. Let $\mathbf{q}_j$ denote the j-th row of the $l' \times \kappa$ bit matrix $[\mathbf{q}^1| \ldots |\mathbf{q}^\kappa]$ held by S, and similarly let v_j be the j-th row of $[v^1| \ldots |v^\kappa]$, held by R.

2. R outputs
$$out_{x_j,j} = \mathsf{H}_2(j||v_j), \quad j \in [l].$$

3. S outputs
$$out_{0,j} = \mathsf{H}_2(j||\mathbf{q}_j) \quad \text{and} \quad out_{1,j} = \mathsf{H}_2(j||(\mathbf{q}_j + \Delta)), \quad j \in [l].$$

♡

References

[1] Marcel Keller, Emmanuela Orsini, and Peter Scholl. Actively secure ot extension with optimal overhead. In *Annual Cryptology Conference*, pages 724–741. Springer, 2015.

[2] Lawrence Roy. Softspokenot: Communication–computation tradeoffs in ot extension. *Cryptology ePrint Archive*, 2022.