Optimal PSO from Commutative Weak Pseudorandom Functions

Abstract

XXX

Keywords: XXX

Contents

1 Preliminaries

1.1 MPC in the Semi-honest Model

We use the standard notion of security in the presence of semi-honest adversaries. Let Π be a protocol for computing the function $f(x_1, x_2)$, where party P_i has input x_i. We define security in the following way. For each party P, let $\mathrm{View}_P(x_1, x_2)$ denote the view of party P during an honest execution of Π on inputs x_1 and x_2. The view consists of P's input, random tape, and all messages exchanged as part of the Π protocol.

Definition 1.1. 2-party protocol Π securely realizes f in the presence of semi-honest adversaries if there exists a simulator Sim such that for all inputs x_1, x_2 and all $i \in \{1, 2\}$:

$$\mathsf{Sim}(i, x_i, f(x_1, x_2)) \approx_c \mathrm{View}_{P_i}(x_1, x_2)$$

1.2 Private Set Union

PSU is a special case of secure two-party computation. We call the two parties engaging in PSU the *sender* and the *receiver*. The sender holds a set X of size n_{x}, and the receiver holds a set Y of size n_{y}. Both sets consist of σ-bit strings. We always assume the set sizes n_{x} and n_{y} are public. The ideal PSU functionality (depicted in Figure 1) computes the union, outputs nothing to the sender, and $X \cup Y$ to the receiver.

Parameters:

- Sender $\mathcal{S}$, Receiver $\mathcal{R}$
- Set sizes n_{x} and n_{y}

Functionality:

- Wait for input $X = \{x_1, \ldots, x_{n_{\mathsf{x}}}\} \subset \{0,1\}^*$ from the receiver $\mathcal{S}$.
- Wait for input $Y = \{y_1, \ldots, y_{n_{\mathsf{y}}}\} \subset \{0,1\}^*$ from the sender $\mathcal{R}$.
- Give output $X \cup Y$ to the receiver $\mathcal{R}$.

Figure 1: Private Set Union Functionality $\mathcal{F}_{\mathsf{psu}}$

2 Commutative Pseudorandom Functions

Let F be a family of pseudorandom functions from D to R. The standard security requirement for PRFs is pseudorandomness.

Pseudorandomness. Let $\mathcal{A}$ be an adversary against PRFs and define its advantage as:

$$\mathsf{Adv}_{\mathcal{A}}(\lambda) = \Pr\left[b = b' : \begin{array}{l} pp \leftarrow \mathsf{Setup}(1^\lambda); \\ k \leftarrow \mathsf{KeyGen}(pp); \\ b \leftarrow \{0,1\}; \\ b' \leftarrow \mathcal{A}^{\mathcal{O}_{\mathsf{ror}}(b,\cdot)}(\lambda); \end{array} \right] - \frac{1}{2},$$

where $\mathcal{O}_{\mathsf{ror}}(0, x) = F_k(x)$, $\mathcal{O}_{\mathsf{ror}}(1, x) = \mathsf{H}(x)$ (here H is chosen uniformly at random from all the functions from D to R[1]). Note that $\mathcal{A}$ can adaptively access the oracle $\mathcal{O}_{\mathsf{ror}}(b, \cdot)$ polynomial many times. We say

[1]To efficiently simulate access to a uniformly random function H from D to R, one may think of a process in which the adversary's queries to $\mathcal{O}_{\mathsf{ror}}(1, \cdot)$ are "lazily" answered with independently and randomly chosen elements in R, while keeping track of the answers so that queries made repeatedly are answered consistently.

that F is pseudorandom if for any PPT adversary its advantage function $\mathsf{Adv}_{\mathcal{A}}(\lambda)$ is negligible in λ. We refer to such security as full PRF security.

Sometimes the full PRF security is not needed and it is sufficient if the function cannot be distinguished from a uniform random one when challenged on random inputs. The formalization of such relaxed requirement is *weak pseudorandomness*, which is defined the same way as pseudorandomness except that the inputs of oracle $\mathcal{O}_{\mathsf{ror}}(b, \cdot)$ are uniformly chosen from D by the challenger instead of adversarially chosen by $\mathcal{A}$. PRFs that satisfy weak pseudorandomness are referred to as *weak PRFs*.

Composable. For a family of keyed function F, F is 2-composable if $R \subseteq D$, namely, for any $k_1, k_2 \in K$, the function $F_{k_1}(F_{k_2}(\cdot))$ is well-defined. From now on, we will assume $R = D$ for simplicity.

Commutative. For a family of composable keyed function, we say it is commutative if:

$$\forall k_1, k_2 \in K, \forall x \in X : F_{k_1}(F_{k_2}(x)) = F_{k_2}(F_{k_1}(x))$$

It is easy to see that the standard pseudorandomness denies commutative property, but weak pseudorandomness and commutative property may co-exist.

2.1 Commutative Weak PRFs from the DDH Assumption

We build a concrete commutative weak PRF from the DDH assumption.

- $\mathsf{Setup}(1^\lambda)$: runs $\mathsf{GroupGen}(1^\lambda) \to (\mathbb{G}, g, p)$, outputs $pp = (\mathbb{G}, g, p)$.
- $\mathsf{KeyGen}(pp)$: outputs $k \xleftarrow{\text{R}} \mathbb{Z}_p$. Each k defines a function from $\mathbb{G}$ to $\mathbb{G}$, which takes $x \in \mathbb{G}$ as input and outputs x^k.

It is straightforward to verify that F is commutative. By the random self-reducibility of the DDH assumption, the weak pseudorandomness can be tightly reduced to the DDH assumption. We omit the details for triviality.

3 PSU from Commutative Weak PRFs

Parameters:

- Common input: $F : K \times D \to D$, hash function $\mathsf{H} : \{0,1\}^* \to D$.
- Input of sender $\mathcal{S}$: $X = \{x_1, \dots, x_n\}$.
- Input of receiver $\mathcal{R}$: $Y = \{y_1, \dots, y_n\}$.

Protocol:

1. $\mathcal{R}$ picks $k_2 \xleftarrow{\text{R}} K$, then sends $\{F_{k_2}(\mathsf{H}(y_1)), \dots, F_{k_2}(\mathsf{H}(y_n))\}$ to $\mathcal{S}$.
2. $\mathcal{S}$ picks $k_1 \xleftarrow{\text{R}} K$, computes and sends $\{F_{k_1}(\mathsf{H}(x_1)), \dots, F_{k_1}(\mathsf{H}(x_n))\}$ to $\mathcal{R}$; then computes $\{F_{k_1}(F_{k_2}(\mathsf{H}(y_1)), \dots, F_{k_1}(F_{k_2}(\mathsf{H}(y_n))\}$, sends its permutation Γ to $\mathcal{R}$;
3. $\mathcal{R}$ computes $\{F_{k_2}(F_{k_1}(\mathsf{H}(x_1)), \dots, F_{k_2}(F_{k_1}(\mathsf{H}(x_n))\}$, then sets $v_i = 0$ iff the value is not in Γ.
4. $\mathcal{R}$ with select vector $(v_1, \dots, v_n)$ and $\mathcal{S}$ with input $\{(x_i, \perp)\}_{i \in [n]}$ engage in one-sided OT.
5. $\mathcal{R}$ obtains $X - X \cap Y$, and thus obtains the union $X \cup Y$.

Figure 2: PSU from Commutative weak PRFs

Correctness. The above protocol is correct except the case E that $F_{k_1}(F_{k_2}(\mathsf{H}(x))) = F_{k_1}(F_{k_2}(\mathsf{H}(y)))$ for some $x \neq y$ occurs. We further divide E to E_0 and E_1. E_0 denotes the case that $\mathsf{H}(x) = \mathsf{H}(y)$. E_1 denotes the case that $\mathsf{H}(x) \neq \mathsf{H}(y)$ but $F_{k_1}(F_{k_2}(\mathsf{H}(x))) = F_{k_1}(F_{k_2}(\mathsf{H}(y)))$, which can further be divided into sub-cases E_{10} — $F_{k_2}(\mathsf{H}(x)) = F_{k_2}(\mathsf{H}(y))$ and E_{11} — $F_{k_2}(\mathsf{H}(x)) \neq F_{k_2}(\mathsf{H}(y))$ but $F_{k_1}(F_{k_2}(\mathsf{H}(x))) = F_{k_1}(F_{k_2}(\mathsf{H}(y)))$. By the collision resistance of H, we have $\Pr[E_0] = 2^{-\sigma}$. By the weak pseudorandomness of F, we have $\Pr[E_{10}] = \Pr[E_{11}] = 2^{-\ell}$. Therefore, we have $\Pr[E] \leq \Pr[E_0] + \Pr[E_{10}] + \Pr[E_{11}] = 2^{-\sigma} + 2^{-\ell+1}$.

Theorem 3.1. *The above PSU protocol is secure in the semi-honest model assuming* H *is a random oracle and* F *is a family of commutative weak PRFs.*

Proof. We exhibit simulators $\mathsf{Sim}_{\mathcal{R}}$ and $\mathsf{Sim}_{\mathcal{S}}$ for simulating corrupt $\mathcal{R}$ and $\mathcal{S}$ respectively, and argue the indistinguishability of the produced transcript from the real execution. Let $|X \cap Y| = m$.

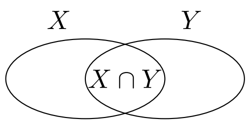
Corrupt sender: $\mathsf{Sim}_{\mathcal{S}}$ simulates the view of corrupt $\mathcal{S}$, which consists of $\mathcal{S}$'s randomness, input, output and received messages.

We argue the output of $\mathsf{Sim}_{\mathcal{S}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{S}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{S}}$.

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{S}}$ simulates with the knowledge of Y.

- $\mathsf{Sim}_{\mathcal{S}}$ chooses the randomness for $\mathcal{R}$, i.e., picks $k_2 \xleftarrow{R} K$.
- RO queries: for random oracle query $\langle z \rangle$, picks $h \xleftarrow{R} D$ and set $\mathsf{H}(z) := h$.
- Let $h_i = \mathsf{H}(y_i)$ for each $y_i \in Y$. $\mathsf{Sim}_{\mathcal{S}}$ outputs $(F_{k_2}(h_1), \ldots, F_{k_2}(h_n))$.

Clearly, $\mathsf{Sim}_{\mathcal{S}}$'s simulation is identical to the real view.



$\mathsf{H}(z_i) := h_i \xleftarrow{R} D$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{S}}$ simulates without the knowledge of Y, and changes the simulation in the final input.

- $\mathsf{Sim}_{\mathcal{S}}$ outputs $(s_1, \ldots, s_n)$ where $s_i \xleftarrow{R} D$.

We argue that the view in hybrid 0 and hybrid 1 are computationally indistinguishable. More precisely, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) against commutative weak PRF (with secret key k) is given n tuples (h_i, s_i) where $h_i \xleftarrow{R} D$, and is asked to distinguish if $s_i = F_k(h_i)$ or s_i are random values. $\mathcal{A}$ implicitly sets $\mathcal{R}$'s randomness $k_2 := k$.

- RO queries: for random oracle query $\langle z \rangle$ where $z \notin Y$, picks $h \xleftarrow{R} D$ and set $\mathsf{H}(z) := h$; for random oracle query $\langle y_i \rangle$ where $y_i \in Y$, sets $\mathsf{H}(y_i) := h_i$.
- $\mathcal{A}$ outputs $(s_1, \ldots, s_n)$.

If $s_i = F_k(h_i)$ for $i \in [n]$, then $\mathcal{A}$'s simulation is identical to hybrid 0. If s_i are random values, then $\mathcal{A}$'s simulation is identical to hybrid 1.

Corrupt receiver: $\mathsf{Sim}_{\mathcal{R}}$ simulates the view of corrupt $\mathcal{R}$, which consists of $\mathcal{R}$'s randomness, input, output and received messages.

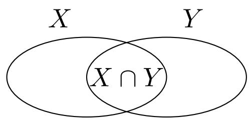
We argue the output of $\mathsf{Sim}_{\mathcal{R}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{R}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{R}}$.

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{R}}$ simulates with the knowledge of X.

- $\mathsf{Sim}_{\mathcal{R}}$ chooses the randomness for $\mathcal{S}$, i.e., picks $k_1 \xleftarrow{R} K$.

- RO queries: for random oracle query $\langle z \rangle$, picks $h \xleftarrow{R} D$ and sets $\mathsf{H}(z) := h$.
- $\mathsf{Sim}_{\mathcal{R}}$ computes and outputs $\{F_{k_1}(\mathsf{H}(x_i))\}_{x_i \in X}$, computes $\{F_{k_1}(F_{k_2}(\mathsf{H}(y_i))\}_{y_i \in Y}$, outputs its permutation Γ.

Clearly, $\mathsf{Sim}_{\mathcal{R}}$'s simulation is identical to the real view.



$\mathsf{H}(z_i) := h_i \xleftarrow{R} D$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{R}}$ simulates without the knowledge of $X \cap Y$, but with the knowledge of $X - X \cap Y$.

- $\mathsf{Sim}_{\mathcal{R}}$ picks $s_j \xleftarrow{R} D$ for $j \in [n-m]$ (associated with PRF values for elements in $X - X \cap Y$), picks $s_k \xleftarrow{R} D$ for $k \in [m]$ (implicitly associated with PRF values for elements in $X \cap Y$, though the exact elememts are unknown), outputs $(\{s_j\}_{j \in [n-m]}, \{s_k\}_{k \in m})$ according to the order of the final selection vector; picks $s_\ell \xleftarrow{R} D$ for $\ell \in [n-m]$ (associated with PRF values for elements in $Y - X \cap Y$), outputs the permutation of $(\{F_{k_2}(s_k)\}_{\{}k \in [m]\}, \{F_{k_2}(s_\ell)\}_{\ell \in [n-m]})$.

We argue that the view in hybrid 1 and hybrid 2 are computationally indistinguishable. More precisely, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) against commutative weak PRFs are given $n+m$ tuples (h_i, s_i) where $h_i \xleftarrow{R} D$, and is asked to determine if $s_i = F_k(h_i)$ or random values. $\mathcal{A}$ implicitly sets $\mathcal{S}$'s randomness $k_1 := k$, picks $k_2 \xleftarrow{R} K$.

- RO queries: for $z \notin X \cup Y$, picks $h \xleftarrow{R} D$ and returns $\mathsf{H}(z) := h$; for $z_i \in X \cup Y$, returns $\mathsf{H}(z_i) = h_i$.
- $\mathcal{A}$ prepares s_j for $z_i \in X - X \cap Y$, s_k for $z_k \in X \cap Y$, outputs $(\{s_j\}_{j \in [n-m]}, \{s_k\}_{k \in m})$ according to the order of the final selection vector; prepares $s_\ell \xleftarrow{R} D$ for $z_\ell \in Y - X \cap Y$, outputs the permutation of $(\{F_{k_2}(s_k)\}_{k \in [m]}, \{F_{k_2}(s_\ell)\}_{\ell \in [n-m]})$.

If s_i are function values, then the simulation is identical to hybrid 0, else it is identical to hybrid 1.

□

4 Instantiation Based on the DDH Assumption

We construct a linear complexity PSU protocol from the DDH assumption in Figure 3.

Theorem 4.1. *The above PSU protocol is secure in the semi-honest model assuming* H *is a random oracle and the DDH assumption.*

Proof. We exhibit simulators $\mathsf{Sim}_{\mathcal{R}}$ and $\mathsf{Sim}_{\mathcal{S}}$ for simulating corrupt $\mathcal{R}$ and $\mathcal{S}$ respectively, and argue the indistinguishability of the produced transcript from the real execution. Let $|X \cap Y| = m$.

Corrupt sender: $\mathsf{Sim}_{\mathcal{S}}$ simulates the view of corrupt $\mathcal{S}$, which consists of $\mathcal{S}$'s randomness, input, output and received messages.

Intuitively, the crux of the security proof is to simulate sender's view without the knowledge of Y, more precisely, the knowledge of $X \cap Y$. Looking ahead, we will use RO to program the hash outputs for elements in $Y - X \cap Y$ naively, and program the hash outputs for elements in X in a special way to ensure the associated messages sent from $\mathcal{R}$ are pseudorandom.

We argue the output of $\mathsf{Sim}_{\mathcal{S}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{S}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{S}}$.

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{S}}$ simulates with the knowledge of Y.

- $\mathsf{Sim}_{\mathcal{S}}$ chooses the randomness for $\mathcal{R}$, i.e., picks $b \xleftarrow{R} \mathbb{Z}_p$.
- RO queries: picks $h_i \xleftarrow{R} \mathbb{G}$ and sets $\mathsf{H}(z_i) := h_i$.

Parameters:

- Common input: $\mathbb{G} = \langle g \rangle$, hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$.
- Input of sender $\mathcal{S}$: $X = \{x_1, \dots, x_n\}$.
- Input of receiver $\mathcal{R}$: $Y = \{y_1, \dots, y_n\}$.

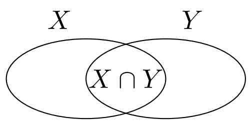
Protocol:

1. $\mathcal{R}$ picks $b \xleftarrow{\text{R}} \mathbb{Z}_p$, then sends $(\mathsf{H}(y_1)^b, \dots, \mathsf{H}(y_n)^b)$ to $\mathcal{S}$.
2. $\mathcal{S}$ picks $a \xleftarrow{\text{R}} \mathbb{Z}_p$, sends $(\mathsf{H}(x_1)^a, \dots, \mathsf{H}(x_n)^a)$ to $\mathcal{R}$; he also computes $(\mathsf{H}(y_1)^b)^a, \dots, \mathsf{H}(y_n)^b)^a)$ and sends its permutation Γ to $\mathcal{R}$.
3. $\mathcal{R}$ computes $(\mathsf{H}(x_1)^a)^b, \dots, \mathsf{H}(x_n)^a)^b)$, then sets $v_i = 0$ iff the value is not in Γ.
4. $\mathcal{R}$ with select vector $(v_1, \dots, v_n)$ and $\mathcal{S}$ with input $\{(x_i, \perp)\}_{i \in [n]}$ engage in one-sided OT.
5. $\mathcal{R}$ obtains $X - X \cap Y$, and thus obtains the union $X \cup Y$.

Figure 3: DH-PSU

- $\mathsf{Sim}_{\mathcal{S}}$ outputs h_i^b for $y_i \in Y$.

Clearly, $\mathsf{Sim}_{\mathcal{S}}$'s simulation is identical to the real view.

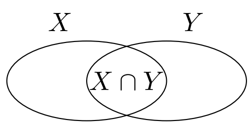


$\mathsf{H}(z_i) := h_i \xleftarrow{\text{R}} \mathbb{G}$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{S}}$ still simulates with the knowledge of Y, but slightly change the simulation of random oracle

- RO queries: picks $d_i, e_i \xleftarrow{\text{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$. Here, h is a fixed random group element.
- $\mathsf{Sim}_{\mathcal{S}}$ outputs h_i^b for $y_i \in Y$.

The modification in RO simulation does not alter the view.



$\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{d_i}$

Hybrid$_2$: $\mathsf{Sim}_{\mathcal{S}}$ simulates without the knowledge of Y, and changes the simulation in the final input.

- $\mathsf{Sim}_{\mathcal{S}}$ outputs random f_i for $i \in [n]$.

We argue that the view in hybrid 2 and hybrid 3 are computationally indistinguishable. More precisely, given $(g, g^\alpha, g^\beta, g^\gamma)$, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) implicitly sets $\mathcal{R}$'s randomness $b := \alpha$.

- RO queries: sets $h := g^\beta$; picks $d_i, e_i \xleftarrow{\text{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h^{d_i} \cdot g^{e_i}$.
- $\mathcal{A}$ outputs $g^{\gamma d_i + \alpha e_i}$ for $i \in [n]$.

If $(g, g^\alpha, g^\beta, g^\gamma)$ is a DDH tuple, then the simulation is identical to hybrid 1, else it is identical to hybrid 2.

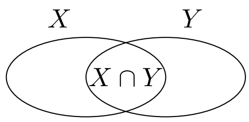
Corrupt receiver: $\mathsf{Sim}_{\mathcal{R}}$ simulates the view of corrupt $\mathcal{R}$, which consists of $\mathcal{R}$'s randomness, input, output and received messages.

We argue the output of $\mathsf{Sim}_{\mathcal{R}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{R}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{R}}$.

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{R}}$ simulates with the knowledge of X.

- $\mathsf{Sim}_{\mathcal{R}}$ chooses the randomness for $\mathcal{S}$, i.e., picks $a \xleftarrow{\text{R}} \mathbb{Z}_p$.
- RO queries: picks $h_i \xleftarrow{\text{R}} \mathbb{G}$ and sets $\mathsf{H}(z_i) := h_i$.
- $\mathsf{Sim}_{\mathcal{R}}$ computes and outputs $\{h_i^a\}_{x_i \in X}$; computes $\{(h_i^b)^a\}_{y_i \in Y}$, outputs its permutation.

Clearly, $\mathsf{Sim}_{\mathcal{R}}$'s simulation is identical to the real view.

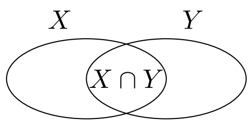


$$\mathsf{H}(z_i) := h_i \xleftarrow{\text{R}} \mathbb{G}$$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{R}}$ still simulates with the knowledge of X, but slightly changes the simulation of random oracle

- RO queries: for $z_i \notin Y$, picks $r_i \xleftarrow{\text{R}} \mathbb{Z}_p$ and sets $\mathsf{H}(z_i) := h_i = g^{r_i}$; for $z_i \in Y$, picks $d_i, e_i \xleftarrow{\text{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$. Here, h is a fixed random group element.
- $\mathsf{Sim}_{\mathcal{R}}$ computes and outputs $\{h_i^a\}_{x_i \in X}$; computes $\{(h_i^b)^a\}_{y_i \in Y}$, outputs its permutation.

The modification in RO simulation does not alter the view.



$$\mathsf{H}(z_i) = \begin{cases} g^{r_i} & \text{if } z_i \notin Y \\ h^{d_i} \cdot g^{d_i} & \text{if } z_i \in Y \end{cases}$$

Hybrid$_2$: $\mathsf{Sim}_{\mathcal{R}}$ simulates without the knowledge of X, and changes the simulation in the final input.

- $\mathsf{Sim}_{\mathcal{R}}$ computes $f_j = h_j^a$ for $x_j \in X - X \cap Y$ (note that the information of such x_j can be inferred from $\mathcal{R}$'s output), picks random f_k for $k \in [m]$ (implicitly assigned for elements in $X \cap Y$), outputs $(f_1, \ldots, f_n)$; then picks random f_ℓ for $\ell \in [n-m]$ (implicitly assigned to elements in $Y - X \cap Y$), outputs the permutation of $(\{f_k^b\}_{k \in [m]}, \{f_\ell^b\}_{\ell \in [n-m]})$.

We argue that the view in hybrid 2 and hybrid 3 are computationally indistinguishable. More precisely, given $(g, g^\alpha, g^\beta, g^\gamma)$, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) implicitly sets $\mathcal{S}$'s randomness $a := \alpha$, picks $b \xleftarrow{\text{R}} \mathbb{Z}_p$.

- RO queries: sets $h := g^\beta$; for $z_i \notin Y$, picks $r_i \xleftarrow{\text{R}} \mathbb{Z}_p$ and sets $\mathsf{H}(z_i) := h_i = g^{r_i}$; for $z_i \in Y$, picks $d_i, e_i \xleftarrow{\text{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$.
- $\mathcal{A}$ computes $f_j = (g^\alpha)^{r_i}$ for $x_j \in X - X \cap Y$, computes $f_k = g^{\gamma d_k + e_k}$ for $k \in [m]$, outputs $(f_1, \ldots, f_n)$; then prepares $f_\ell = g^{\gamma d_\ell + e_\ell}$ for $\ell \in [n-m]$, outputs the permutation of $(\{f_k^b\}_{k \in [m]}, \{f_\ell^b\}_{\ell \in [n-m]})$.

If $(g, g^\alpha, g^\beta, g^\gamma)$ is a DDH tuple, then the simulation is identical to hybrid 1, else it is identical to hybrid 2.

□

References

A Diffie-Hellman Based PSI

We recall the ideal PSI functionality in Figure 4.

Parameters:

- Sender $\mathcal{S}$, Receiver $\mathcal{R}$
- Set sizes n_{x} and n_{y}

Functionality:

- Wait for input $X = \{x_1, \ldots, x_{n_{\mathsf{x}}}\} \subset \{0,1\}^*$ from the receiver $\mathcal{S}$.
- Wait for input $Y = \{y_1, \ldots, y_{n_{\mathsf{y}}}\} \subset \{0,1\}^*$ from the sender $\mathcal{R}$.
- Give output $X \cap Y$ to the receiver $\mathcal{R}$.

Figure 4: Private Set Intersection Functionality $\mathcal{F}_{\mathsf{psi}}$

We first recall the DH-PSI in Figure 5.

Parameters:

- Common input: $\mathbb{G} = \langle g \rangle$ with prime order p, hash function $\mathsf{H} : \{0,1\}^* \to \mathbb{G}$.
- Input of sender $\mathcal{S}$: $X = \{x_1, \ldots, x_n\}$.
- Input of receiver $\mathcal{R}$: $Y = \{y_1, \ldots, y_n\}$.

Protocol:

1. $\mathcal{S}$ picks $a \xleftarrow{\text{R}} \mathbb{Z}_p$, then sends random permutation of $(\mathsf{H}(x_1)^a, \ldots, \mathsf{H}(x_n)^a)$ to $\mathcal{R}$.
2. $\mathcal{R}$ picks $b \xleftarrow{\text{R}} \mathbb{Z}_p$, then sends $(\mathsf{H}(y_1)^b, \ldots, \mathsf{H}(y_n)^b)$ to $\mathcal{S}$.
3. $\mathcal{S}$ sends $(\mathsf{H}(y_1)^{ab}, \ldots, \mathsf{H}(y_n)^{ab})$ to $\mathcal{R}$.
4. $\mathcal{R}$ sets $\Omega = \{\mathsf{H}(x_i)^{ab}\}_{i \in n}$, and outputs $\{y \mid \mathsf{H}(y)^{ab} \in \Omega\}$.

Figure 5: DH-PSI

Theorem A.1. *The above PSI protocol is secure in the semi-honest model assuming* H *is a random oracle and the DDH assumption.*

Proof. We exhibit simulators $\mathsf{Sim}_{\mathcal{R}}$ and $\mathsf{Sim}_{\mathcal{S}}$ for simulating corrupt $\mathcal{R}$ and $\mathcal{S}$ respectively, and argue the indistinguishability of the produced transcript from the real execution. Let $|X \cap Y| = m$.

Corrupt sender: $\mathsf{Sim}_{\mathcal{S}}$ simulates the view of corrupt $\mathcal{S}$, which consists of $\mathcal{S}$'s randomness, input, output and received messages. $\mathsf{Sim}_{\mathcal{S}}$ proceeds as follows.

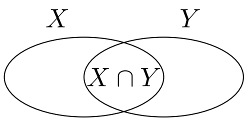
We now argue the output of $\mathsf{Sim}_{\mathcal{S}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{S}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{S}}$.

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{S}}$ simulates with the knowledge of Y.

- $\mathsf{Sim}_{\mathcal{S}}$ chooses the randomness for $\mathcal{R}$, i.e., picks $b \xleftarrow{\text{R}} \mathbb{Z}_p$

- RO queries: picks $h_i \xleftarrow{R} \mathbb{G}$ and sets $\mathsf{H}(z_i) := h_i$.
- $\mathsf{Sim}_{\mathcal{S}}$ outputs h_i^b for $y_i \in Y$ (message in step 2).

Clearly, $\mathsf{Sim}_{\mathcal{S}}$'s simulation is identical to the real view.

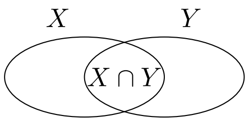


$\mathsf{H}(z_i) := h_i \xleftarrow{R} \mathbb{G}$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{S}}$ still simulates with the knowledge of Y, but slightly change the simulation of random oracle

- RO queries: picks $d_i, e_i \xleftarrow{R} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$. Here, h is a random group element fixed at beginning.
- $\mathsf{Sim}_{\mathcal{S}}$ outputs h_i^b for $y_i \in Y$.

The modification in RO simulation does not alter the view.



$\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$

Hybrid$_2$: $\mathsf{Sim}_{\mathcal{S}}$ simulates without the knowledge of Y, and changes the simulation in the final input.

- $\mathsf{Sim}_{\mathcal{S}}$ outputs random f_i for $i \in [n]$.

We argue that the view in hybrid 2 and hybrid 3 are computationally indistinguishable based on the DDH assumption. More precisely, given $(g, g^\alpha, g^\beta, g^\gamma)$, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) implicitly sets $\mathcal{R}$'s random tape $b := \alpha$:

- RO queries: sets $h = g^\beta$; picks $d_i, e_i \xleftarrow{R} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$.
- $\mathcal{A}$ outputs $g^{\gamma d_i + \alpha e_i}$ for $i \in [n]$.

If $(g, g^\alpha, g^\beta, g^\gamma)$ is a DDH tuple, then the simulation is identical to hybrid 1 because $g^{\gamma d_i + \alpha e_i} = g^{\alpha\beta d_i + \alpha e_i} = h_i^b$, else it is identical to hybrid 2.

Remark A.1. From hybrid 1, we program $\mathsf{H}(z_i) = h^{d_i} \cdot g^{e_i}$ to obtain a tight reduction by leveraging the random self-reducible property of the DDH assumption. Alternatively, we can simply program $\mathsf{H}(z_i) = h^{d_i}$, then argue the indistinguishability using hybrid argument. As a result, the security reduction comes with a loose factor n.

Corrupt receiver: $\mathsf{Sim}_{\mathcal{R}}$ simulates the view of corrupt $\mathcal{R}$, which consists of $\mathcal{R}$'s randomness, input, output and received messages. $\mathsf{Sim}_{\mathcal{R}}$ proceeds as follows.

We now argue the output of $\mathsf{Sim}_{\mathcal{R}}$ is indistinguishable from the real execution. For this, we formally show the simulation by proceeding the sequence of hybrid transcripts, where T_0 is the real view of $\mathcal{R}$, and T_2 is the output of $\mathsf{Sim}_{\mathcal{R}}$.

The simulator has to emulate two messages:

- message in step 1: a permutation of $(\mathsf{H}(x_1)^a, \ldots, \mathsf{H}(x_n)^a)$
- message in step 3: $(\mathsf{H}(y_1)^{ab}, \ldots, \mathsf{H}(y_n)^{ab})$

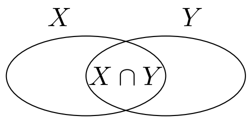
Since the elements in $X - X \cap Y$ are unknown,

Hybrid$_0$: $\mathsf{Sim}_{\mathcal{R}}$ simulates with the knowledge of X.

- $\mathsf{Sim}_{\mathcal{R}}$ chooses the randomness for $\mathcal{S}$, i.e., picks $a \xleftarrow{R} \mathbb{Z}_p$
- RO queries: picks $h_i \xleftarrow{R} \mathbb{G}$ and sets $\mathsf{H}(z_i) := h_i$.

- $\mathsf{Sim}_{\mathcal{R}}$ outputs a random permutation of $(\mathsf{H}(x_1)^a, \ldots, \mathsf{H}(x_n)^a)$ and $(\mathsf{H}(y_1)^{ab}, \ldots, \mathsf{H}(y_n)^{ab})$.

Clearly, $\mathsf{Sim}_{\mathcal{R}}$'s simulation is identical to the real view.

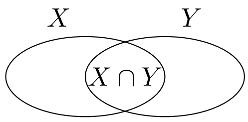


$$\mathsf{H}(z_i) := h_i \xleftarrow{\mathrm{R}} \mathbb{G}$$

Hybrid$_1$: $\mathsf{Sim}_{\mathcal{R}}$ still simulates with the knowledge of X, but slightly change the simulation of random oracle

- RO queries: for $z_i \in Y$, picks $r_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ and sets $\mathsf{H}(z_i) := h_i = g^{r_i}$; for $z_i \notin Y$, picks $d_i, e_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$. Here, h is a random group element fixed at beginning.

The modification in RO simulation does not alter the view.



$$\mathsf{H}(z_i) = \begin{cases} g^{r_i} & \text{if } z_i \in Y \\ h^{d_i} \cdot g^{e_i} & \text{if } z_i \notin Y \end{cases}$$

Hybrid$_2$: $\mathsf{Sim}_{\mathcal{R}}$ simulates without the knowledge of X, and changes the simulation in the output.

- $\mathsf{Sim}_{\mathcal{R}}$ picks random f_j for $j \in [n-m]$ (associated with elements in $X - X \cap Y$), prepares $f_k = h_k^a$ for $k \in [m]$ (associated with elements in $X \cap Y$), then outputs a random permutation of $\{f_i\}_{i \in [n]}$, and $(\mathsf{H}(y_1)^{ab}, \ldots, \mathsf{H}(y_n)^{ab})$.

We argue that the view in hybrid 2 and hybrid 3 are computationally indistinguishable based on the DDH assumption. More precisely, given $(g, g^\alpha, g^\beta, g^\gamma)$, a PPT adversary $\mathcal{A}$ (with knowledge of X and Y) implicitly sets $\mathcal{S}$'s randomness $a := \alpha$, picks $\mathcal{R}$'s randomness $b \xleftarrow{\mathrm{R}} \mathbb{Z}_p$:

- RO queries: sets $h = g^\beta$; for $z_i \in Y$, picks $r_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$ and sets $\mathsf{H}(z_i) := h_i = g^{r_i}$; for $z_i \notin Y$, picks $d_i, e_i \xleftarrow{\mathrm{R}} \mathbb{Z}_p$, sets $\mathsf{H}(z_i) := h_i = h^{d_i} \cdot g^{e_i}$.
- $\mathcal{A}$ computes $f_j = g^{\gamma d_j + \alpha e_j}$ for $j \in [n-m]$, $f_k = (g^\alpha)^{r_k}$ for $k \in [m]$, then outputs a random permutation of $\{f_i\}_{i \in n}$ and $\{\mathsf{H}(y_1)^{ab}, \ldots, \mathsf{H}(y_n)^{ab}\}$, where $\mathsf{H}(y_i)^{ab} = (g^\alpha)^{r_i b}$.

If $(g, g^\alpha, g^\beta, g^\gamma)$ is a DDH tuple, then the simulation is identical to hybrid 1 because $g^{\gamma d_j + \alpha e_j} = g^{\alpha\beta d_i + \alpha e_i} = h_j^a$, else it is identical to hybrid 2.

$\square$