Practical Single-Round Secure Wildcard Pattern Matching

Abstract

Here we provide a formal proof of security for our protocol.

Security. We are now ready to prove the security of our protocol Π_{WPM} in the presence of semi-honest adversaries.

Theorem 1 *Assume the 1-out-of-3 oblivious transfer is secure against semi-honest adversaries, and the secret sharing scheme satisfies that all shares are necessary and sufficient to reconstruct the secret. Then the protocol Π_{WPM} securely computes the functionality $\mathcal{F}_{WPM}$ in semi-honest setting.*

Proof We prove Theorem 1 in a hybrid model where a trusted party is used to compute the oblivious transfer functionality $\mathcal{F}_{OT}$. We separately prove the case that $\mathcal{S}$ is corrupted and the case that $\mathcal{R}$ is corrupted.

$\mathcal{S}$ **is corrupted.** Let $\mathcal{A}$ be an adversary controlling $\mathcal{S}$ in an execution of protocol in Figure where a trusted party is required to compute the oblivious transfer functionality $\mathcal{F}_{OT}$. We construct a simulator $\mathcal{S}^*$ that runs $\mathcal{A}$ upon its inputs. The simulator $\mathcal{S}^*$ acts the honest receiver $\mathcal{R}$ and interacts with $\mathcal{A}$ according to the protocol description. The simulator $\mathcal{S}^*$ for a corrupted sender $\mathcal{S}$ may proceed as follows:

- $\mathcal{S}^*$ firstly invokes $\mathcal{A}$ upon the input $\boldsymbol{T}$ of $\mathcal{A}$ and the auxiliary input $\boldsymbol{r}$. And then $\mathcal{S}^*$ receives the value tuples $(k^0_{i,j}, k^1_{i,j}, k^2_{i,j})$ from $\mathcal{A}$ and simulates the trusted party computing the oblivious transfer functionality with input a uniformly chosen random pattern $\boldsymbol{p^*}$.
- $\mathcal{S}^*$ sends the input $\boldsymbol{T}$ to the trusted party computing the functionality $\mathcal{F}_{WPM}$ and output whatever $\mathcal{S}$ outputs. Note that in this case, $\mathcal{A}$ who corrupts $\mathcal{S}$ outputs nothing.

It is remarkable that $\mathcal{S}$ outputs nothing in the real process. Therefore, only to prove that the distribution of the ideal execution between $\mathcal{S}^*$ and an honest $\mathcal{R}$ is identical to the output of a real execution of the protocol with $\mathcal{A}$ and $\mathcal{R}$ is sufficient. We now show that for every $\mathcal{A}$ corrupting $\mathcal{S}$ it holds that

$$\{\text{IDEAL}_{\mathcal{F}_{WPM},\mathcal{S}^*(1^k,r)},\mathcal{R}(\boldsymbol{T},\boldsymbol{p})\} \overset{c}{\equiv} \{\text{HYBRID}^{\text{OT}}_{\Pi_{WPM},\mathcal{A}(1^k,r)},\mathcal{R}(\boldsymbol{T},\boldsymbol{p})\}$$

During the ideal simulation, $\mathcal{S}^*$ chooses a random pattern $\boldsymbol{p^*}$ as the input of honest party $\mathcal{R}$. However, in the real protocol, the honest $\mathcal{R}$ uses its own input pattern $\boldsymbol{p}$. This is the only difference between the two above distributions. Given the semi-honest secure oblivious transfer protocol, it is impossible for an adversary to distinguish the $\mathcal{R}$'s input and obtain any private information of $\mathcal{R}$. Therefore, the distribution of the view of $\mathcal{A}$ in the ideal world is indistinguishable from the distribution of the view of $\mathcal{A}$ in a real protocol execution.

$\mathcal{R}$ **is corrupted.** Without loss of generality we assume that $\mathcal{R}$ is corrupted by the adversary $\mathcal{A}$. We construct a simulator $\mathcal{S}^*$ playing the role of $\mathcal{R}$ in the ideal world and using $\mathcal{S}^*$ as a subroutine as follows:

1. $\mathcal{S}^*$ invokes $\mathcal{A}$ upon the input $\boldsymbol{p}$ of $\mathcal{A}$ and the auxiliary input $\boldsymbol{r}$.

2. $\mathcal{S}^*$ sends $\boldsymbol{p}$ to the trusted party that computes the functionality $\mathcal{F}_{WPM}$ and receives the matching result. Without loss of generality again we assume the matching result is denoted by $I = \{i_1, i_2, \cdots, i_c\}$ where $1 \leq i_1 \leq \cdots \leq i_c \leq n'$ and i_c means the substring $\boldsymbol{t}_c$ of $\boldsymbol{T}$ starting the c-th location matches $\boldsymbol{p}$ successfully.

3. For each $i \in [I]$, $\mathcal{S}^*$ acts as $\mathcal{S}$ and shares the public string $\boldsymbol{r}$ into $\boldsymbol{r}_{i,j}$ using secret sharing scheme, where $j \in [m]$. At the same time, $\mathcal{S}^*$ chooses random shares $\boldsymbol{s}_{i',j}$ to represent the remaining substrings for $i' \in [n']/[I]$.

4. $\mathcal{S}^*$ prepares the input tuples for the OT phase according to the protocol description. For each $i \in [I]$, $\mathcal{S}^*$ generates the value tuple honestly but a value tuple of random order for $i' \in [n']/[I]$. This is due to $\mathcal{S}^*$ does not know the unmatched strings, and therefore it can not generates the tuples relevant to the i'-th substring of $\boldsymbol{T}$ in the right order.

5. $\mathcal{S}^*$ sends these tuples to the trusted party computing $\mathcal{F}_{OT}$ and returns the output share to the adversary $\mathcal{A}$. $\mathcal{S}^*$ then outputs what $\mathcal{A}$ outputs and halts.

Denoting the protocol in Figure by Π_{WPM}, we show that the joint distribution of the output of the honest party $\mathcal{S}$ and the view of $\mathcal{A}$ in the ideal world is computationally indistinguishable from the output of the honest party $\mathcal{S}$ and the view of $\mathcal{A}$ in the real world. We have:

$$\{\mathrm{IDEAL}_{\mathcal{F}_{WPM}, \mathcal{S}^*(1^k, r)}, \mathcal{S}(\boldsymbol{T}, \boldsymbol{p})\} \overset{c}{\equiv} \{\mathrm{HYBRID}^{\mathrm{OT}}_{\Pi_{WPM}, \mathcal{A}(1^k, r)}, \mathcal{S}(\boldsymbol{T}, \boldsymbol{p})\}$$

The only difference between the above two distributions is that the simulator $\mathcal{S}^*$ generates the shares $\boldsymbol{s}_{i',j}$ dishonestly for $i' \in [n']/[I]$. On the basis of these invalid shares, the public string $\boldsymbol{r}$ can not be reconstructed by $\mathcal{S}$ in the ideal world. And note that in the real process, $\mathcal{S}$ also can not reconstruct the string $\boldsymbol{r}$ for the substrings starting at the location of $i' \in [n']/[I]$. Therefore, we conclude that $\mathcal{S}$ cannot distinguish the party interacting with him, and the above two distributions are computationally indistinguishable.

To conclude the proof, we show that the real and the simulated view of the adversary is indistinguishable.