Efficient Constructions of Non-interactive Secure Multiparty Computation from Pairwise Independent Hashing

Satoshi Obana\textsuperscript{a}@\textsuperscript{b} and Maki Yoshida\textsuperscript{a}\textsuperscript{b}

\textsuperscript{a}Hosei University, Tokyo, Japan
\textsuperscript{b}NICT, Tokyo, Japan

Keywords: Secure Multiparty Computation, Non-interactive, Information Theoretical Security, Communication Complexity, Pairwise Independent Hash Functions.

Abstract: An important issue of secure multi-party computation (MPC) is to improve the efficiency of communication. Non-interactive MPC (NIMPC) introduced by Beimel et al. in Crypto 2014 completely avoids interaction in the information theoretical setting by allowing a correlated randomness setup where the parties get correlated random strings beforehand and locally compute their messages sent to an external output server. Existing studies have been devoted to constructing NIMPC with small communication complexity, and many NIMPC have been presented so far. In this paper, we present a new generic construction of NIMPC for arbitrary functions from a class of functions called indicator functions. We employ pairwise independent hash functions to construct the proposed NIMPC, which results in smallest communication complexity compared to the existing generic constructions. We further present a concrete construction of NIMPC for the set of indicator functions with smallest communication complexity known so far. The construction also employs pairwise independent hash functions. It will be of independent interest to see how pairwise independent hash functions helps in constructing NIMPC.

1 INTRODUCTION

Since the seminal paper by Yao (Yao, 1982), secure multiparty computation (MPC for short) have been a central topic in the area of cryptographic research. The work is followed by a large number of literatures (Ben-Or et al., 1988; Chaum et al., 1988; Data et al., 2014; Hirt and Tschudi, 2013), and some of efficient implementations even possess a potential to deal with real-world application. Though, such efficient implementations are attractive, they demand high speed network connection (i.e., 10Gbps network) among parties for achieving high-throughput computation, and do not work well in poor network environment.

Beimel et al. have introduced a novel type of MPC called non-interactive multiparty computation (NIMPC for short). In NIMPC for a function $f : X_1 \times \cdots \times X_n \rightarrow \{0, 1\}^L$, each party $P_i$ receives correlated randomness $r_i$ and outputs $m_i$ computed from $r_i$ and a private input $x_i$, so that $f(x_1, \ldots, x_n)$ is computed only from $m_1, m_2, \ldots, m_n$. The notable feature of NIMPC is that it completely gets rid of interaction among parties since the message $m_i$ is locally computed by $P_i$. The security model presented by Beimel et al. guarantees information-theoretic security against honest-but-curious adversaries. More precisely, it guarantees any set of corrupted parties learns nothing about inputs of uncorrupted parties and the function they aim to evaluate other than the information inferred from their inputs and output. Beimel et al. also showed NIMPC for various classes of functions. In particular, they showed that NIMPC for arbitrary functions is possible by showing an exact construction of an NIMPC for arbitrary functions. Though, since the communication complexity of their NIMPC is very large (exponential in the input length), their construction is valuable only in the sense it shows the possibility of realizing NIMPC for arbitrary functions.

Since the seminal work by Beimel et al., the theory of NIMPC has been further developed by literatures (Yoshida and Obana, 2016; Obana and Yoshida, 2016; Halevi et al., 2016; Halevi et al., 2017; Agarwal et al., 2019). In Eurocrypt 2019, Agarwal et al. present elegant construction of NIMPC for arbitrary functions (Agarwal et al., 2019). In their con-
Table 1: The communication complexity of $n$-player NIMPC protocols for arbitrary functions $h : X \rightarrow \{0,1\}^L$ where $d \leq |X_i|$, and $\delta_{\text{ind}}$ is the communication complexity of NIMPC for the set of indicator functions.

<table>
<thead>
<tr>
<th>Construction</th>
<th>The communication complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction in (Agarwal et al., 2019)</td>
<td>$\log_2 d + \max(2L, \log_2 d) \cdot</td>
</tr>
<tr>
<td>Construction in (Beimel et al., 2014)</td>
<td>$\delta_{\text{ind}} \cdot L \cdot</td>
</tr>
<tr>
<td>Construction in (Obana and Yoshida, 2016)</td>
<td>$(\delta_{\text{ind}} + L \cdot \log_2 (d+1)) \cdot</td>
</tr>
<tr>
<td>Our construction (generic)</td>
<td>$(\delta_{\text{ind}} + \max(2L, L + \log_2 d)) \cdot</td>
</tr>
<tr>
<td>Our construction (concrete)</td>
<td>$(4 \cdot \log_2 d \cdot n + \max(2L, L + \log_2 d)) \cdot</td>
</tr>
</tbody>
</table>

Table 2: The communication complexity of $n$-player NIMPC protocols for the set of indicator functions.

<table>
<thead>
<tr>
<th>Construction</th>
<th>The communication complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Construction in (Beimel et al., 2014)</td>
<td>$d^2 \cdot n$</td>
</tr>
<tr>
<td>Construction in (Yoshida and Obana, 2016)</td>
<td>$\log_2 (d+1)^2 \cdot n$</td>
</tr>
<tr>
<td>Our construction</td>
<td>$4 \cdot \log_2 d \cdot n$</td>
</tr>
</tbody>
</table>

construction, the correlated randomness $r_i$ consists of additively shared output table of the target function $f$ where input and output are masked with random values, and the message $m_i$ consists of masked output table of $f(x_1, \ldots, x_{i-1}, a_i, x_{i+1}, \ldots, x_n)$, together with the masked value of $a_i$. Such direct construction is very efficient in the sense that the communication complexity of the scheme is as small as $\log_2 d + L \cdot |X|$ where $d = \max_{i \in [n]} |X_i|$ and $X = X_1 \times \cdots \times X_n$. The communication complexity of their NIMPC is close to the lower bound on the communication complexity shown by Yoshida and Obana in (Yoshida and Obana, 2016), though, there is still a gap between the lower bound and the most efficient scheme known so far.

To deepen understanding of theory and practice of NIMPC, it is important to clarify to what extent we can construct a scheme with the communication complexity close to the lower bound. To answer the question, we must try various approaches to construct efficient NIMPCs. One of major and prominent approaches is generic construction. Generic construction of NIMPC is methodology to construct complex classes of function (e.g., arbitrary functions) based on simple classes of function. All the generic constructions known so far employ indicator function as a simple class of function, where indicator function $h_{\text{ind}}(x) : X \rightarrow \{0,1\}$ equals 1 if and only if the input $x$ is identical to $a$. There is line of research that tries to construct an efficient NIMPC with small communication complexity based on NIMPC for the set of indicator functions (Beimel et al., 2014; Yoshida and Obana, 2016; Obana and Yoshida, 2016).

The contribution of the paper is twofold. First, we presents an efficient generic construction of NIMPC for arbitrary functions based on any NIMPC for the set of indicator functions. Second, we presents an efficient construction of NIMPC for the set of indicator functions. Combining the first and the second contributions, we obtain a concrete construction of NIMPC for arbitrary functions with the smallest communication complexity compared to existing generic constructions of NIMPC for arbitrary functions. Tables 1 and 2 summarize the communication complexity of existing NIMPC for arbitrary functions with $L$-bit output, and that of existing NIMPC for the set of indicator functions, respectively.

We see that the proposed NIMPC for the set of indicator function is the most efficient one, and the proposed generic construction is most efficient among generic constructions based on NIMPC for the set of indicator functions. Let $\delta_{\text{ind}}$ be the communication complexity of underlying NIMPC for set of indicator functions, and let $\log_2 d = L$ for simplicity. Then the communication complexity of the proposed NIMPC for arbitrary functions is $(\delta_{\text{ind}} + 2L) \cdot |X|$ while that of (Obana and Yoshida, 2016) is $(\delta_{\text{ind}} + L^2) \cdot |X|$. Compared to the most efficient NIMPC presented in (Agarwal et al., 2019), proposed NIMPC is less efficient, though, the overhead is not so large. Again, let $\log_2 d = L$ for the sake of simplicity, then the communication complexity of the proposed NIMPC for arbitrary functions becomes $L \cdot (4n + 2) \cdot |X|$, which is about $4n + 2$ times larger than that of (Agarwal et al., 2019).

## 2 PRELIMINARIES

For an integer $n$, let $[n]$ be the set $\{1,2,\ldots,n\}$. For a set $X = X_1 \times \cdots \times X_n$ and $T \subseteq [n]$, we denote $X_T = \prod_{i \in T} X_i$. For $x \in X$, we denote by $x_T$ the restriction of $x$ to $X_T$, and for a function $h : X \rightarrow \Omega$, a subset $T \subseteq [n]$, its complement $T^c \subseteq [n]$, and $x_T \in X_T$, we denote by $h\mid_{T^c \times x_T} : X \rightarrow \Omega$ the function $h$ where the
inputs of \( T \) are fixed to \( x_T \). For a set \( S \), let \(|S|\) denote its size (i.e., cardinality of \( S \)).

An NIMPC protocol for a family of functions \( \mathcal{H} \) is defined by three algorithms: (1) a randomness generation function \( \text{GEN} \), which given a description of a function \( h \in \mathcal{H} \) generates \( n \) correlated random inputs \( R_1, \ldots, R_n \), (2) a local encoding function \( \text{ENC}_i \) \((1 \leq i \leq n)\), which takes an input \( x_i \) and a random input \( R_i \) and outputs a message, and (3) a decoding algorithm \( \text{DEC} \) that reconstructs \( h(x_1, \ldots, x_n) \) from the \( n \) messages. The formal definition given in (Beimel et al., 2014) is given as follows.

**Definition 1 (Syntax and Correctness).** Let \( X_1, \ldots, X_n, R_1, \ldots, R_n, M_1, \ldots, M_n \) and \( \Omega \) be finite domains. Let \( X \triangleq X_1 \times \cdots \times X_n \) and let \( \mathcal{H} \) be a family of functions \( h : X \to \Omega \). A non-interactive secure multi-party computation (NIMPC) protocol for \( h \) is a triper \( \Pi = (\text{GEN}, \text{ENC}, \text{DEC}) \) where

\[
\text{GEN} : \mathcal{H} \to R_1 \times \cdots \times R_n \text{ is a random function.}
\]

\( \text{ENC} \) is an \( n \)-tuple deterministic functions \((\text{ENC}_1, \ldots, \text{ENC}_n)\), where \( \text{ENC}_i : X_i \times R_i \to M_i \);

\( \text{DEC} : M_1 \times \cdots \times M_n \to \Omega \) is a deterministic function satisfying the following correctness requirement: for any \( x = (x_1, \ldots, x_n) \in X \) and \( h \in \mathcal{H} \),

\[
\Pr[R = (R_1, \ldots, R_n) \leftarrow \text{GEN}(h) : \text{DEC}(\text{ENC}(x, R)) = h(x)] = 1,
\]

where \( \text{ENC}(x, R) = (\text{ENC}_1(x_1, R_1), \ldots, \text{ENC}_n(x_n, R_n)) \).

The communication complexity of NIMPC \( \Pi \) is defined to be the maximum value of \( \log |X_1|, \ldots, \log |X_n|, \log |M_1|, \ldots, \log |M_n| \).

We next show the definition of robustness for NIMPC (Beimel et al., 2014), which states that a coalition can only learn the information they should. In the above setting, a coalition of players \( \tau \subseteq \{1, \ldots, n\} \) is \( \tau \)-robust if in \( \Pi \) the only information that is not known by all the players in \( \tau \) is \( h(x_\tau) \). Let \( \text{Sim}_T (h) \) be the simulator that generates a description of \( h \) according to the execution of the protocol.

**Definition 2 (Robustness).** For a subset \( \tau \subseteq \{1, \ldots, n\} \), we say that an NIMPC protocol \( \Pi \) for \( \mathcal{H} \) is \( T \)-robust if there exists a randomized function \( \text{Sim}_T \) (a “simulator”) such that, for every \( h \in \mathcal{H} \) and \( x_T \in X_T \), we have \( \text{Sim}_T (h|_{X_T \setminus \tau}) \equiv (M_T, R_T) \), where \( R \) and \( M \) are the joint randomness and messages defined by \( R \leftarrow \text{GEN}(h) \) and \( M_i \leftarrow \text{ENC}_i(x_i, R_i) \).

For an integer \( 0 \leq t \leq n \), we say that \( \Pi \) is \( t \)-robust if it is \( T \)-robust for every \( T \subseteq \{1, \ldots, n\} \) of size \( |T| \leq t \). We say that \( \Pi \) is fully robust (or simply refer to \( \Pi \) as an NIMPC for \( \mathcal{H} \)) if \( \Pi \) is \( n \)-robust. Finally, given a concrete function \( h : X \to \Omega \), we say that \( \Pi \) is \( (t, \delta) \)-NIMPC for \( h \) if it is \( (t, \delta) \)-NIMPC for \( \mathcal{H} = \{h\} \).

As the same simulator \( \text{Sim}_T \) is used for every \( h \in \mathcal{H} \) and the simulator has only access to \( h|_{X_T \setminus \tau} \). NIMPC hides both \( h \) and the inputs of \( T \). An NIMPC protocol is \( 0 \)-robust if it is \( \emptyset \)-robust. In this case, the only requirement is that the messages \( (M_1, \ldots, M_n) \) reveal \( h(x) \) and nothing else.

An NIMPC protocol is also described in the language of protocols in (Beimel et al., 2014). Such a protocol involves \( n \) players \( P_1, \ldots, P_n \), each holding an input \( x_i \in X_i \), and an external “output server,” a player \( P_0 \) with no input. The protocol may have an additional input, a function \( h \in \mathcal{H} \).

**Definition 3 (Protocol Description).** For an NIMPC protocol \( \Pi \) for \( \mathcal{H} \), let \( P(\Pi) \) denote the protocol that may have an additional input, a function \( h \in \mathcal{H} \), and proceeds as follows.

**Protocol \( P(\Pi)(h) \)**

**Offline Preprocessing.** Each player \( P_i \), \( 1 \leq i \leq n \), receives the random input \( R_i \leftarrow \text{GEN}(h) \in R \).

**Online Messages.** On input \( R_i \), each player \( P_i \), \( 1 \leq i \leq n \), sends the message \( M_i \leftarrow \text{ENC}_i(x_i, R_i) \in M_i \) to \( P_0 \).

**Output.** \( P_0 \) computes and outputs \( \text{DEC}(M_1, \ldots, M_n) \).

Informally, the relevant properties of protocol \( P(\Pi) \) are given as follows:

- For any \( h \in \mathcal{H} \) and \( x \in X \), the output server \( P_0 \) outputs, with probability 1, the value \( h(x_1, \ldots, x_n) \).

- Fix \( T \subseteq \{1, \ldots, n\} \), then \( \Pi \) is \( T \)-robust if in \( P(\Pi) \) the set of players \( \{P_i\}_{i \in T} \cup \{P_0\} \) can simulate their view of the protocol (i.e., the random inputs \( R_i \) of \( T \) and the messages \( M_i \) of \( T \)) given oracle access to the function \( h \) restricted by the other inputs (i.e., \( h|_{X_T \setminus \tau} \)).

- \( \Pi \) is \( 0 \)-robust if and only if in \( P(\Pi) \) the output server \( P_0 \) learns nothing but \( h(x_1, \ldots, x_n) \).

A lower bound on the communication complexity for any finite set of functions including the set of arbitrary functions was derived in (Yoshida and Obana, 2016). The result states that the communication complexity cannot be smaller than the logarithm of the size of the target class.

**Proposition 1 (Lower Bound).** Fix finite domains \( X_1, \ldots, X_n \) and \( \Omega \). Let \( X \triangleq X_1 \times \cdots \times X_n \) and \( \mathcal{H} \) a set of functions \( h : X \to \Omega \). Then, any fully robust NIMPC protocol \( \Pi \) for \( \mathcal{H} \) satisfies \( \sum_{i=1}^n \log |X_i| \geq \log |\mathcal{H}| \), and \( \sum_{i=1}^n \log |M_i| \geq \log |\Omega| \).
Proposition 2 (Lower Bound). Fix finite domains $X_1, \ldots, X_n$. Let $X = X_1 \times \cdots \times X_n$ and $H_{\text{ind}}$ be the set of all functions $h : X \to \{0, 1\}^L$. Any NIMPC protocol $\Pi$ for $H_{\text{ind}}$ satisfies $\sum_{i=1}^n \log |X_i| \geq L \cdot |X|$, and $\sum_{i=1}^n \log |H_i| \geq L$.

Here, we give definitions of indicator functions (Beimel et al., 2014), and generalized indicator functions (Obana and Yoshida, 2016) which are important classes of functions for our proposed construction.

Definition 4 (Indicator Functions). Let $X$ be a finite domain. For $n$-tuple $a = (a_1, \ldots, a_n) \in X$, let $h_{\text{ind}} : X \to \{0, 1\}$ be the function defined by $h_{\text{ind}}(a) = 1$, and $h_{\text{ind}}(x) = 0$ for all $a \neq x \in X$. Let $h_{\text{ind}} : X \to \{0, 1\}$ be the function that is identically zero on $X$. Let $H_{\text{ind}} = \{h_{\text{ind}} \mid \forall x \in \{0, 1\}^n \cup \{h_0\}\}$. In the next section, we will present a generic construction of NIMPC for arbitrary set of functions. We employ pairwise independent hash functions to construct NIMPC for the set of generalized indicator functions. We note that pairwise independent hash function plays an important role in constructing various cryptographic protocols.

Definition 6. A family of functions $G = \{g \mid g : X \to Y\}$ is pairwise independent if the following two conditions hold when $g \in G$ is a function chosen uniformly at random from $G$:

1. For any $x \in X$, the random variable $g(x)$ is uniformly distributed in $Y$.
2. For any distinct $x_1, x_2 \in X$, the random variables $g(x_1)$ and $g(x_2)$ are independent.

When the function $g$ is chosen uniformly at random from $G$, we can guarantee $g(x)$ does not reveal any information about $x$. Further, the value $g(x)$ does not reveal any information about the value $g(x')$ such that $x' \neq x$. These properties of pairwise independent hash family help us in constructing NIMPC.

The following proposition gives a well-known fact about pairwise independent hash functions (e.g., (Vadhan, 2012)).
complexity of underlying NIMPC for indicator function. Then the communication complexity of resulting NIMPC for arbitrary functions is $\delta_{\text{ind}} \cdot 2^L \cdot |X|$.

In (Obana and Yoshida, 2016), Obana and Yoshida present the second approach to construct NIMPC for arbitrary functions. While the first approach separately compute each output bit, the second approach simultaneously computes all output bits. The key idea of the second approach is to introduce generalized indicator functions $h_{a,v}(x)$ outputting $v \in \mathbb{0,1}^L$ if $x = a$ holds, and otherwise $0^L$. Their construction is based on the observation that arbitrary function $h : X \rightarrow \{0,1\}^L$ is represented by the sum of $h_{a,v} \in \mathcal{H}_{\text{ind}}^L$ (i.e., $h = \sum_{a \in X} h(a) \cdot \{0^L \cdot h_{a,v}(a)\}$), and use the fact to construct NIMPC for $\mathcal{H}_{\text{ind}}^L$. The generic construction of (Obana and Yoshida, 2016) reduces the communication complexity to $\delta_{\text{ind}} \cdot 2^L \cdot \log_2|X|^L$ times smaller than that of the first approach.

In the proposed construction, we adopt the same approach as in (Obana and Yoshida, 2016), that is, starting from an NIMPC for the set of indicator function, we construct an NIMPC for the set of generalized indicator function, which is used to construct NIMPC for the set of arbitrary function. The main difference between our construction and that in (Obana and Yoshida, 2016) is in the building block to construct an NIMPC for the set of generalized indicator functions. The construction in (Obana and Yoshida, 2016) employs binary vectors to extend the range of indicator function. On the other hand, we employ pairwise independent hash functions to extend the range, which results in NIMPC for arbitrary functions with smaller communication complexity.

### 3.2 NIMPC $\mathcal{H}_{\text{ind}}^L \Rightarrow$ NIMPC $\mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$

Here, we will give a generic construction of NIMPC for $\mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$ from any NIMPC for $\mathcal{H}_{\text{ind}}^L$. The basic idea behind the proposed generic construction is as follows. We will use an NIMPC $\Pi_{\text{ind}} = (\text{GEN}, \text{ENC}', \text{DEC}')$ for $\mathcal{H}_{\text{ind}}^L$ to check whether the function $h \in \mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$ outputs non-zero value with the input $(x_1, \ldots, x_n) \in X$. To obtain the actual output value (i.e., $h(x_1, \ldots, x_n)$), we employ functions $g_i$ from pairwise independent hash family $G_i : X_i \rightarrow \mathbb{F}_2^L$ for $i \in [n]$. Functions $g_i \in G_i$ are chosen in such a way that $\sum_{i=1}^n g_i(x_i) = h(x_1, \ldots, x_n)$ holds if the input $(x_1, \ldots, x_n)$ is identical to the input with which $\text{DEC}'$ outputs 1.

Let $\Pi_{\text{ind}} = (\text{GEN}, \text{ENC}', \text{DEC}')$ be any NIMPC for $\mathcal{H}_{\text{ind}}^L$. Then the concrete description of the proposed construction of NIMPC for $\mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$ denoted by $\Pi_{\text{ind}_{\text{mnd}}} = (\text{GEN}, \text{ENC}, \text{DEC})$, is given as follows. For $i \in [n]$, let $g_i$ be an element of pairwise independent hash family $G_i : X_i \rightarrow \{0,1\}^L$.

Fix a function $h \in \mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$ that we want to compute.

**Offline Preprocessing.** First, define a function $h' \in \mathcal{H}_{\text{ind}}^L$ as follows,

$$h' = \begin{cases} h_0 & \text{if } h = h_0^L \\ h_a & \text{otherwise} \end{cases}$$

and let $R' = (R_1', \ldots, R_n') \in \text{GEN}(h')$. Next, if $h = h_0^L$ then choose $n$ random functions $g_i \in G_i$. If $h = h_a$ for some $a = (a_1, \ldots, a_n) \in X$ and $v \in \{0,1\}^L \setminus \{0^L\}$, choose $n - 1$ functions $g_i$ uniformly and randomly from $G_i$ for $i \in [n - 1]$ and choose a function $g_a \in G_a$ such that $\sum_{i=1}^n g_i(a_i) = v$ holds, which can be done by choosing $g_a$ from the function family $\{g_a | g_a \in G_a, g_a(a_i) = v - \sum_{i=1}^{n-1} g_i(a_i)\}$ uniformly and randomly. Define $\text{GEN}(h) \triangleq R = (R_1, \ldots, R_n)$ where $R_i = (R_i', \text{desc}(g_i))$.

**Online Messages.** For $R_i = (R_i', \text{desc})$, and an input $x_i$, we first evaluate $(M_1', \ldots, M_n') \leftarrow \text{ENC}(x_i, R_i')$. Next, we evaluate $v_i = g_i(x_i)$ where $g_i$ is an element of $G_i$ described by desc. Finally, let $\text{ENC}(x,R) \triangleq (M_1', \ldots, M_n')$ where $M_i = (M_i', v_i)$.

**Output** $h(x_1, \ldots, x_n)$. DEC$(M_1, \ldots, M_n) = \sum_{i=1}^n v_i$ if DEC$(M_1', \ldots, M_n') = 1$ holds. Otherwise DEC$(M_1, \ldots, M_n) = 0^L$.

**Theorem 1.** Fix finite domains $X_1, \ldots, X_n$, and let $X \subseteq X_1 \times \cdots \times X_n$. If there exists a robust NIMPC for $\mathcal{H}_{\text{ind}}: X \rightarrow \{0,1\}$ with communication complexity $\delta_{\text{ind}}$, then there is an NIMPC protocol for $\mathcal{H}_{\text{ind}}^{L_{\text{ind}}}$ with the communication complexity $\delta_{\text{ind}} + \max\{2L, L + \log_2 d\}$.

**Proof:** First, we will show the correctness. Let $M_i = (M_i', v_i)$. It holds that $\sum_{i=1}^n v_i = \sum_{i=1}^n g_i(x_i)$. If $h = h_0^L$, then DEC$(M_1', \ldots, M_n') = 1$ holds if and only if $a = x$. In this case $\sum_{i=1}^n v_i = \sum_{i=1}^n g_i(a_i) = v$ holds. This means DEC$(M_1, \ldots, M_n) = v$ if and only if $x = a$. If $h = h_0$, then DEC$(M_1', \ldots, M_n') = 1$ never happens because of the correctness of the underlying NIMPC for $\mathcal{H}_{\text{ind}}$. This means DEC$(M_1, \ldots, M_n) = 0^L$ holds for any $x \in X$.

To prove robustness, fix a subset $T \subseteq [n]$ and $x_T \in X_T$. The encodings $M_T$ of $T$ consist of $(M_i', v_i)_{i \in T}$. The randomness $R_T$ consists of $(R_i, \text{desc}(g_i))_{i \in T}$. Now we will construct a simulator $\text{Sim}_T$ which queries $h_{T,x_T}$ on all possible inputs in $X_T$. First we will simulate $(R_T', M_T')$. Since $R' = \text{GEN}(h')$ and $M' = \text{ENC}(R', x_T)$ hold, and $\Pi_{\text{ind}} = (\text{GEN}', \text{ENC}', \text{DEC}')$ is robust, it is possible to simulate $(R_T', M_T')$ if we can answer to a query to $h_{T,x_T}'$ which is easily computed from $h_{T,x_T}$ as follows.
Next, we will simulate \( \text{desc}(g_i) \) for \( i \in T \) and \( v_i = g_i(x_i) \) for \( i \in \hat{T} \). If \( h'_{T,x_T}(x_T) = 0 \), there are two possible cases. The first case is \( h = h_0 \). In this case \( \text{desc}(g_i) \) for \( i \in T \) and \( v_i = g_i(x_i) \) are uniformly and independently distributed since all \( g_i \) are uniformly and independently distributed. The second case to consider is \( h = h_{a,v} \) for some \( a,v \) and \( \sigma_T \neq \sigma_T \). In this case, \( g_i \) (and therefore \( \text{desc}(g_i) \)) for \( i \in T \) and \( v_i = g_i(x_i) \) are uniformly and independently distributed. From the above argument, we conclude that the desc \( (g_i) \) for \( i \in T \) and \( v_i = g_i(x_i) \) are uniformly and independently distributed in both cases. Therefore, if \( h'_{T,x_T}(x_T) = 0 \) then desc \( (g_i) \) and \( v_i = g_i(x_i) \) are simulated simply by assigning uniformly distributed random strings to them. On the other hand, if \( h'_{T,x_T}(x_T) = 0 \) holds for some \( x_T \neq x_T \), then \( \sum_{i \in T} g_i(x_i) \) holds. Let \( \hat{i} \in \hat{T} \), then desc \( (g_i) \) and \( v_i = g_i(x_i) \) are simulated by assigning uniform random strings to desc \( (g_i) \) and \( v_i \) by assigning \( v + (\sum_{i \in T} g_i(a_i)) \) to \( v_i \.

Now, we will evaluate the communication complexity of the resulting NIMPC. Let \( \delta_{\text{ind}} \) be the communication complexity of the underlying NIMPC for \( H_{\text{ind}} \). The correlated randomness \( R_i \) is composed of \( R_i' \) and \( L + \log_2 d \) binary string, whereas the encoding \( M_i \) is composed of \( M_i' \) and \( L \)-bit binary string. Therefore, the communication complexity is at most \( \delta_{\text{ind}} + \max(2L, L + \log_2 d) \).

### 3.3 NIMPC \( H_{\text{ind}}^L \) \( \Rightarrow \) NIMPC \( H_{\text{all}}^L \)

In this section, we present a generic construction of NIMPC for all \( L \)-bit boolean functions \( H_{\text{all}}^L \) with input domain \( X = X_1 \times \cdots \times X_n \) from any NIMPC for \( H_{\text{ind}}^L \) with the same input domain. The idea is to express any \( h : X \rightarrow \{0,1\}^L \) as a sum of generalized indicator functions \( H_{\text{ind}}^L \) with \( L \)-bit output. The communication complexity of the resulting construction is much smaller than the existing constructions since a single invocation of the proposed NIMPC for \( H_{\text{all}}^L \) given in §3.2 is much more efficient than \( L \) invocation of the existing NIMPC for \( H_{\text{ind}}^L \) for most \( L \). The detailed description of the compiler to construct \( H_{\text{all}}^L \) is identical to that presented in (Obana and Yoshida, 2016). Let \( H_{\text{ind}}^L = (\text{GEN'}, \text{ENC'}, \text{DEC'}) \) be any NIMPC for \( H_{\text{ind}}^L \) and let \( h : X \rightarrow \{0,1\}^L \) that we want to compute. We construct a protocol \( P(\Pi)(h) \) for \( H_{\text{all}}^L \), whose algorithms are denoted by \( (\text{GEN}, \text{ENC}, \text{DEC}) \), as follows.

**Offline Preprocessing.** Let \( I \subseteq X \) be the set of inputs \( x \in X \) such that \( h(x) \neq 0 \). For each \( a \in I \), let \( R_i = (R_i^1, \ldots, R_i^n) \leftarrow \text{GEN}'(h_{a,v}) \). For \( a \in X \setminus I \), let \( R_i \leftarrow \text{GEN}(h_0) \). Then, choose random permutation \( \pi \) of \( X \) and let \( R_{i,b} = R_i^{\pi(b)} \) for \( i \in [n], b \in X \). Define \( \text{GEN}(h) \triangleq R = (R_1, \ldots, R_n) \), where \( R_i = \{R_{i,b}\}_{b \in X} \).

**Online Messages.** For an input \( x_i \), \( P_i \) computes \( M_{i,b} \leftarrow \text{ENC}'(x_i, R_{i,b}) \) for every \( b \in X \). Define \( \text{ENC}(x, R) \triangleq (M_1, \ldots, M_n) \) where \( M_i = \{M_{i,b}\}_{b \in X} \).

**Output** \( h(x_1, \ldots, x_n) \), \( \text{DEC}(M_1, \ldots, M_n) = v \) if and only if there exists \( b \in X \) such that \( \text{DEC}(M_{1,b}, \ldots, M_{n,b}) = v \). Otherwise \( \text{DEC}(M_1, \ldots, M_n) = 0 \).

**Theorem 2.** Fix finite domains \( X_1, \ldots, X_n \), and let \( X = X_1 \times \cdots \times X_n \). Let \( H_{\text{all}}^L \) be the set of all functions \( h : X \rightarrow \{0,1\}^L \). If there exists a robust NIMPC for \( H_{\text{ind}}^L : X \rightarrow \{0,1\}^L \) with communication complexity \( \delta_{\text{ind}} \), then there is an NIMPC protocol for \( H_{\text{all}}^L \) with the communication complexity \( \delta_{\text{all}} \cdot |X| \).

The proof is almost identical to that of Theorem 2 of (Obana and Yoshida, 2016), and is omitted here. By combining Theorem 1 and Theorem 2, we obtain the following corollary.

**Corollary 1** Fix finite domains \( X_1, \ldots, X_n \), and let \( X = X_1 \times \cdots \times X_n \). Let \( H_{\text{all}}^L \) be the set of all functions \( h : X \rightarrow \{0,1\}^L \). If there exists a robust NIMPC for \( H_{\text{ind}}^L : X \rightarrow \{0,1\}^L \) with communication complexity \( \delta_{\text{ind}} \), then there is an NIMPC protocol for \( H_{\text{all}}^L \) with the communication complexity \( (\delta_{\text{ind}} + \max(2L, L + \log_2 d)) \cdot |X| \).

### 4 EFFICIENT NIMPC for \( H_{\text{ind}}^L \)

In this section, we present a construction of NIMPC for \( H_{\text{ind}}^L \), which results in \( H_{\text{all}}^L \) via generic construction given in the previous section. As the generic construction to construct \( H_{\text{ind}}^L \), we also employ pairwise independent hash family to construct \( H_{\text{ind}}^L \). It should be noted that, if \( d \geq 4 \) (i.e., if the maximum bit length of input is larger than 1), the proposed construction of NIMPC for \( H_{\text{ind}}^L \) offers smallest communication complexity known so far. Namely, the communication complexity of the proposed construction is \( 4 \cdot \left\lceil \log_2 d \right\rceil \cdot n \), whereas that of the best known construction (i.e., the construction in (Yoshida and Obana, 2016)) is \( \left( \left\lceil \log_2 (d+1) \right\rceil \right)^2 \cdot n \).
The detailed description of the protocol is as follows. For $i \in [n]$, let $\phi_i$ be a one-to-one mapping from $X_i$ to a finite field $\mathbb{F}$ with the order larger than $\max |X_i|$. Fix a function $h \in \mathcal{H}_{\text{ind}}$ that we want to compute.

**The proposed NIMPC $\Pi_{\text{ind}}(h)$**

**Offline Preprocessing.** If $h = h_0$, then choose $2n$ linearly independent random vectors $\{v_i, v'_i\}_{i \in [n]}$ in $\mathbb{F}^{2n}$. If $h = h_a$ for some $a = (a_1, \ldots, a_n) \in X$, then choose $2n$ random vectors $\{v_i, v'_i\}_{i \in [n]}$ in $\mathbb{F}^{2n}$ such that $\sum_{i=1}^n (v_i + \phi(a_i)v'_i) = 0$, and there are no other linear relations other than $\sum_{i=1}^n c \cdot (v_i + \phi(a_i)v'_i) = 0$ for $c \in \mathbb{F}$. Let $\text{GEN}(h) = R = (R_1, \ldots, R_n)$, where $R_i = \{v_i, v'_i\}$.

**Online Messages.** For an input $x_i$, let $\text{ENC}(x, R) = (M_1, \ldots, M_n)$ where $M_i = v_i + \phi(x_i)v'_i$.

**Output** $h(x_1, \ldots, x_n)$. $\text{DEC}(M_1, \ldots, M_n) = 1$ if $\sum_{i=1}^n M_i = 0$.

**Theorem 3.** Fix finite domains $X_1, \ldots, X_n$. Then, there is an NIMPC protocol $\Pi_{\text{ind}}$ for $\mathcal{H}_{\text{ind}}$ with the communication complexity $4 \cdot \lfloor \log_2 d \rfloor \cdot n$.

**Proof:** The correctness is obvious from the description of Offline preprocessing. Namely, $\sum_{i=1}^n (v_i + x_i v'_i) = 0$ never happen with $(x_1, \ldots, x_n) \neq (a_1, \ldots, a_n)$. In fact, $\sum_{i=1}^n (v_i + x_i v'_i) = 0$ is the only possible solution since $v_i$ is fixed to 1. Moreover, $\sum_{i=1}^n (v_i + x_i v'_i) = 0$ never happen when $h = h_0$ since all $v_i, v'_i$ are linearly independent in this case.

To prove the robustness, we describe a simulator $\text{Sim}_\text{y}$: the simulator queries $h_{\mathcal{T}, \sigma^m}$ on all possible inputs in $X_\mathcal{T}$. If all answers are zero, this simulator generates random independent vectors $v_i, v'_i$ (for $i \in \mathcal{T}$) and $m_i$ (for $i \in \mathcal{T}$). Otherwise, there is an $x_\mathcal{T} \in X_\mathcal{T}$ such that $h_{\mathcal{T}, \sigma^m}(x_\mathcal{T}) = 1$, and the simulator outputs random vectors such that $\sum_{i \in \mathcal{T}} v_i + \sum_{i \in \mathcal{T}} v'_i = 0$, and there are no other linear relations other than $\sum_{i=1}^n c \cdot (v_i + \phi(a_i)v'_i) = 0$ for $c \in \mathbb{F}$.

The communication complexity of the resulting protocol is $4 \cdot \lfloor \log_2 d \rfloor \cdot n$ since $R_i$ consists of $2 \cdot 2n$ elements of finite field $\mathbb{F}$ with $|\mathbb{F}| \leq d$. By combining Theorem 3 and Corollary 2, we obtain the following corollary.

**Corollary 2** Fix finite domains $X_1, \ldots, X_n$ with $|X_i| \leq d$ for all $1 \leq i \leq n$ and let $X = X_1 \times \cdots \times X_n$. Then, there is an NIMPC protocol for $\mathcal{H}_{\text{full}} : X \rightarrow \{0, 1\}^L$ with communication complexity at most $4 \cdot \lfloor \log_2 d \rfloor \cdot n + \max(2L, L + \lfloor \log_2 d \rfloor) \cdot |X|$.

Let $\delta_{\text{ind}}$ be the communication complexity of underlying NIMPC for $\mathcal{H}_{\text{ind}}$, and suppose, for the sake of simplicity, $|X_i| = 2^L$ for any $i \in [n]$. Then the communication complexity of the proposed NIMPC for $\mathcal{H}_{\text{full}}$ becomes $(\delta_{\text{ind}} + 2L)|X|$, which is the most efficient construction among existing NIMPCs for arbitrary functions constructed based on NIMPC for the set of indicator functions since the best known communication complexity of such NIMPC is $(\delta_{\text{ind}} + L^2)|X|$.

**5 CONCLUSION**

In this paper, we have presented a novel generic construction of NIMPC for the set of arbitrary functions $\mathcal{H}_{\text{full}}$ from NIMPC for the set of indicator functions $\mathcal{H}_{\text{ind}}$. The communication complexity of the resulting scheme is the most efficient compared to that of NIMPC for arbitrary functions constructed based on NIMPC for the set of indicator functions. Further, we have presented an NIMPC for the set of indicator functions with the smallest communication complexity known so far. By combining the proposed generic construction and the proposed NIMPC for $\mathcal{H}_{\text{ind}}$, we have obtained a concrete NIMPC for arbitrary functions with the communication complexity $(4 \cdot \lfloor \log_2 d \rfloor \cdot n + \max(2L, L + \lfloor \log_2 d \rfloor)) \cdot |X|$. Compared to the most efficient NIMPC known so far (i.e., NIMPC presented in (Agarwal et al., 2019), the proposed NIMPC is less efficient, though, the gap is as small as $4n + 2$.

Though the proposed construction is pretty efficient with respect to the communication complexity, there still remains a gap between the lower bound in (Yoshida and Obana, 2016) and our upper bound. Therefore, reducing the gap will be a challenging future work.

**REFERENCES**


