Zarezadeh M, Mala H, Khajeh H. Preserving privacy of software-defined networking policies by secure multi-party computation. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 35(4): 863–874 July 2020. DOI 10.1007/s11390-020-9247-5

Preserving Privacy of Software-Defined Networking Policies by Secure Multi-Party Computation

Maryam Zarezadeh[1], Hamid Mala[1,*], and Homa Khajeh[2]

[1] *Faculty of Computer Engineering, University of Isfahan, Isfahan 8174673441, Iran*
[2] *Independent Researcher, Isfahan 8183913851, Iran*

E-mail: {m.zarezadeh, h.mala}@eng.ui.ac.ir; khajeh121@yahoo.com

Received December 3, 2018; revised May 2, 2020.

Abstract In software-defined networking (SDN), controllers are sinks of information such as network topology collected from switches. Organizations often like to protect their internal network topology and keep their network policies private. We borrow techniques from secure multi-party computation (SMC) to preserve the privacy of policies of SDN controllers about status of routers. On the other hand, the number of controllers is one of the most important concerns in scalability of SMC application in SDNs. To address this issue, we formulate an optimization problem to minimize the number of SDN controllers while considering their reliability in SMC operations. We use Non-Dominated Sorting Genetic Algorithm II (NSGA-II) to determine the optimal number of controllers, and simulate SMC for typical SDNs with this number of controllers. Simulation results show that applying the SMC technique to preserve the privacy of organization policies causes only a little delay in SDNs, which is completely justifiable by the privacy obtained.

Keywords software-defined networking (SDN), privacy, secure multi-party computation (SMC), structure function, multi-objective optimization

1 Introduction

Computer networks form a critical infrastructure for enterprises. Data center networks carry a great amount of confidential traffic from the government, users and companies. Therefore, privacy and security are critical requirements for these networks. Software-defined networking (SDN) is an architectural approach that simplifies and optimizes network operations by more closely binding the interactions such as messaging between applications and devices which can be real or virtualized [1]. It is achieved by employing a controller which is a point of logically centralized network software. The separation of control plane and data plane facilitates the communication between applications needing to interact with network elements which convey information [2].

In SDN, the control and management planes are separated from the data plane on the networking device via the well-defined application programming interface. This enables an SDN controller to manage the flows in the forwarding network devices [1]. OpenFlow [3], the de facto standard for the interaction between the controller and the forwarding devices (e.g., SDN switches), follows a match-action paradigm. The SDN controller determines and installs rules of flow on the router consisting of a match and an action part. If a packet (flow) matches a rule, then the corresponding action such as dropping the packet or forwarding it will be applied [2]. However, SDN, and in particular OpenFlow, an SDN standard, introduces a secure network, but the paradigm depends on correctness of the underlying hardware including routers and switches. In recent years, many attackers have compromised routers or switches [4]. The attacker may compromise a router and convert it to an adversary which behaves arbitrarily, i.e., it may reroute, mirror or modify packets on its choice. Consequently, the adversarial router completely ignores the OpenFlow match-action rules which are installed by the SDN controller.

Regular Paper

*Corresponding Author

Protecting policies applied on SDN routers makes it more difficult for an attacker to compromise routers. For instance, if the attacker cannot determine which of routers forwards a message, then he/she cannot easily disturb routing operations. We use a state-of-the-art secure multi-party computation (SMC) framework on structure function to preserve the policies of SDN controllers about routers. In SMC, multiple parties can carry out a joint computation of any function on their respective inputs, but it does not reveal any information about the inputs[5]. SMC was first introduced in 1982 by Yao[6] who gave solutions for the so-called "millionaires" problem in which two millionaires want to know which one is richer without revealing their wealth to each other and without any third party. After that, many papers in the literatures of SMC have investigated the maximal number of dishonest parties out of n parties, which is tolerable by the secure n-party computation of any arbitrary function. The first multi-party solution is proposed by Goldreich *et al.* called GMW protocol[7], who proved that general SMC is possible if and only if $t < n/2$ players are actively corrupted and in the case of passive corruption if and only if $t < n$ players are corrupted.

The structure function is a model that determines the status of the system given the status of its components[8]. We use the structure function to evaluate the performance of routing operation of routers. In this work, we use SMC in an SDN scenario where many controllers secretly share their private inputs, i.e., their policies, and run a secure computation of structure function on these private inputs. The structure function can be considered as a Boolean circuit that consists of AND gates and OR gates. We use the GMW protocol[7] to implement the structure function which uses a Boolean circuit and it is based on Boolean sharing. In order to make the implementation of the GMW protocol in SDNs as efficient as possible, the number of SDN controllers involved in secure computation should be optimal. For this purpose, a heuristic algorithm called Non-Dominated Sorting Genetic Algorithm II (NSGA-II)[9] is used to solve a multi-objective problem for determining the optimal number of SDN controllers. We minimize the number of SDN controllers while reducing the cost and latency between controllers and routers and increasing the reliability.

The rest of this paper is organized as follows. Section 2 summarizes related work about applications of SMC in networks. Section 3 considers the preliminaries of our method. In Section 4, the proposed approach for modeling of structure function is described and the security and the privacy of our approach are discussed. Section 5 determines the optimal number of SDN controllers and Section 6 evaluates the proposed multi-party computation of structure function. Finally, we conclude the paper in Section 7.

2 Related Work

To the best of our knowledge, our paper is the first work that proposes a privacy-preserving method for protecting router policies of SDN based on SMC. However, SMC has many applications and it is applied for preserving privacy in networks. In this section, we investigate some researches that have investigated using SMC in networks as follows.

Border gateway protocol (BGP) is a standardized exterior gateway protocol designed to exchange routing information among autonomous systems on the Internet[10]. Unfortunately, BGP suffers from the leakage of sensitive information about the routing preferences of domains. To overcome this problem, some research papers such as [11–13] apply the idea of centralizing and using SMC for interdomain routing. They use the SMC approach to provide more privacy than BGP while deploying new policies. Asharov *et al.*[13] used a secure two-party computation (2PC) and they outsourced the route computation to two computational parties who supposed that they do not collude. To preserve the privacy of business relations of autonomous systems (ASes), the ASes shared their routing preferences with these two parties such that no party gets any information about the routing preferences between them. Then, these two parties run a secure interdomain routing computation protocol to determine the routes for any autonomous system. The deployment of SDN on the Internet has raised concerns about the correctness of the inter-domain data-plane. If operators can deflect traffic from default BGP routes, then SDN policies create permanent forwarding loops invisible to the control-plane. Dethise *et al.*[14] used SMC techniques to present a system for detecting SDN induced forwarding loops among software-defined eXchanges with a high accuracy without the leakage of private routing information of network operators.

A social network with sensitive information such as trust or hatred relationships between users has privacy concerns. This issue restricts users to access these networks. Kukkala *et al.*[15] proposed a multi-party computation protocol for securely constructing a graph. This graph is used to model a social network such that nodes represent users and edges capture the relationship between users. An unlabeled random isomorphic version of the graph is securely constructed and the resultant graph is distributedly held by n parties. The proposed protocol can be used to investigate the behavioral aspects of network users while preserving the privacy of their sensitive data. Furthermore, the authors[16] discussed that although in social network users can communicate personal information with others, there are privacy concerns when the personal information of the users collected by the provider of social networks is used for advertising purposes. To address this problem, they proposed a privacy-preserving group-based advertising system for secure social networks. Their design is run by n servers where each of them is provided by an independent authority. They utilized the group-based advertising notion for preserving user privacy and hid the identity of the exact target customers.

In ad hoc networks, mobile devices hold users information such as photos, passwords and banking data. Hence, they can provide an environment for secure computation. Also, the sensors in special smartphones collect a lot of sensitive information about users. Therefore, it is important to protect the privacy of data handled in the mobile domain. For this purpose, Demmler *et al.*[17] suggested a scheme for token-aided ad hoc SMC on mobile devices based on the GMW protocol. The Internet of Things (IoT) collects the information about the activities of people and private information such as travel routes or daily activities; therefore the user privacy preservation should be provided. Oleshchuk[18] used the idea of multi-party computation within the context of IoT and investigated which level of protection can be satisfied by applying SMC in ubiquitous applications. Also, in [19] a vision of privacy-preserving data processing is provided. The authors suggested an architecture which employs SMC at the core of the architecture to realize data processing systems incorporating support for preserving privacy. The architecture can be utilized in dynamic environments such as IoT where nodes are constrained devices and communication is done via unreliable connections.

Organizations offering Internet-based services such as Internet service providers and content delivery networks join the Internet exchange points (IXPs) for exchanging Internet traffic between their networks (ASes). IXPs offer a useful service, called route server (RS) which allows any member connected to an IXP to exchange traffic with other members. However, RS services have been deployed at IXPs to facilitate managing of the burden of BGP sessions for the operators, and the usage of such services is along with the privacy concerns and the members' routing policies disclosure to external commercial parties such as the IXPs. Chiesa *et al.*[20] suggested an approach for preserving the privacy of routing policies at IXPs where the RS service allows redistributing the information of BGP routing according to the policies which are specified by the IXP members while this approach reduces the risk of information leakage to the semi-honest or malicious entities.

Also, as mentioned in [21], with an increasing number of IXPs as the emerging physical convergence points for Internet traffic, privacy concerns have arisen. IXPs offer centralized RS services for ranking, selecting, and dispatching BGP routes to their member networks. But, for using these centralized services, IXP members should disclose private information including peering relationships and route-export policies to the IXP or to other IXP members. Such information can reflect sensitive operational and commercial information. Hence, Chiesa *et al.*[21] designed an IXP RS design by using SMC where routes are dispatched according to highly expressive routing policies of members and performance-related information of IXP. Furthermore, Cho *et al.*[22] presented a cloud-based system for secure navigation operations between multiple parties while privacy is preserved. In this system, public data is stored on a cloud computing infrastructure. The system performs a 2PC between the input data corresponding to the current location of the first party and the public data. Also, each party performs a 2PC on the public data and its input data. Then, the system performs SMC between multiple parties and the cloud computing infrastructure. The multiple parties privately update the public data with an obtained result of the 2PC. Finally, for the first party, a privacy-preserving navigation from the first party's current location to a desired location is generated using results obtained from the 2PC and the SMC.

In this paper, we suggest the use of SMC to compute the structure function in SDNs. This method preserves the privacy of policies of SDN controllers for routers. In this scenario, the privacy means that no SDN controller should learn anything more than the output of structure function. Also, we formulate an optimization problem to determine the optimal number of SDN controllers participating in SMC operations. Simulation results show that applying SMC in SDNs creates only a little delay in routing operation.

3 Preliminaries

This section introduces the preliminaries of this work.

3.1 Software-Defined Networking

The term SDN (software-defined networking) refers to a network architecture where the forwarding state in the data plane is managed by a remote control plane. An SDN can be defied as follows[23].

- The control and data planes are decoupled. Control functionality is removed from network devices which become the simple packet forwarding devices.
- Forwarding decisions are flow-based. A flow is defined by a set of packet field values and acts as a match/filter and a set of actions/instructions. In the SDN context, a flow is defined as a sequence of packets between a source and a destination and the packets of a flow receive same service policies at the forwarding devices. Using the flow, the controller unifies the behavior of different types of network devices including routers and switches.
- Control logic is moved to an external entity which is called SDN controller (or network operating system).
- The network is programmable through software applications which run on top of the network operating system interacting with the underlying data plane devices.

In computer networks, routers compute routes using routing protocols to decide which interfaces should forward packets. In SDNs, controllers support route computations and routers become just forwarding devices. SDN controllers determine and apply the flow-based forwarding rules instead of destination-based rules in order to provide a better control of the network traffic. Hence, the routing policy on defining the rule (forwarding or dropping the received packet) for routers is important, since attackers need to detect the active routers to compromise them and make interference in routing. Consequently, routing policies are considered as sensitive information which must be kept private to any entity except the SDN controller who determined this policy. In this paper, we provide a solution to this challenge. Hereafter, the privacy of routing policy means preserving the defined rule for any router about whether it forwards the received packet or not.

3.2 Secure Multi-Party Computation

SMC, first suggested by Yao[6], is a method that allows two or more parties to compute a function without trusting each other or one another. In the setting of SMC, two or more parties, $P_1, P_2, \ldots, P_n$, with the private inputs $x_1, x_2, \ldots, x_n$ wish to jointly compute some predetermined function of their inputs, $f(x_1, x_2, \ldots, x_n)$. The computation should be such that the parties received the correct output and the privacy of each party's input is preserved. No party P_i can obtain more information than what it can obtain from its input x_i and its output. SMC applications include any distributed computing task such as electronic voting, electronic auctions, electronic and anonymous transactions[24]. The security of SMC is preserved even in the presence of some adversarial entity who corrupts some of the parties and coordinates their behaviors. In SMC protocols, two types of adversaries are considered including a semi-honest adversary, also known as honest-but-curios or passive, and a malicious adversary or active. Semi-honest adversaries follow the protocol specification but they try to learn secret information about the private information of the honest parties from the exchanged messages while malicious adversaries deviate from the protocol specification and are allowed to follow any arbitrary behavior[25]. In our method, like [12, 13], we suppose that the computing parties, here SDN controllers, are semi-honest.

3.3 GMW Protocol

In distributed computing, a number of computing devices, or parties, carry out a joint computation of some function. The aim of SMC is to enable parties to compute any function on their private inputs without revealing anything but the result[24]. The GMW protocol[7] allows secure evaluation of a function which is represented as a Boolean circuit. In the GMW protocol, two parties interactively evaluate a Boolean circuit using secret shared values. Each party shares its input with the other party using a 2-out-of-2 secret sharing scheme[26]. For instance, for any value v, each party $P_i, i = 1, 2$, gains from secret sharing the share v_i such that $v = v_1 \oplus v_2$. Then the parties go through each gate of the circuit and compute results on their shares together.

In more details, for each gate the parties locally compute their shares of the output wire using their shares of the input wires. In the GMW protocol, each XOR gate is evaluated locally by XORing the shares. For the NOT gate, only one party complements its share of the bit v over the input wire of the gate and then all parties will hold shares of $\neg v$. Only the computation of the outputs of an AND gate requires the interaction between parties. For computing the outputs of each AND gate, it needs the interaction between parties to compute an oblivious transfer (OT) on inputs. OT is an important building block for secure computation [27]. In OT, a sender has two l-bit inputs (x_0, x_1) and a receiver has an input bit $s \in \{0, 1\}$. The receiver receives the message x_s while the sender has no output. OT guarantees that the receiver only learns x_s and nothing about x_{1-s} while the sender does not learn the value of s.

Due to the interaction between parties in evaluating AND gate, the round complexity of the GMW protocol depends on the depth of the circuit. Using multiplication triple [28] is an efficient method for securely evaluating an AND gate on inputs x and y, shared between two parties as x_0, x_1 and y_0, y_1, respectively. Multiplication triple consists of three random shares $a_i, b_i, c_i, i = 0, 1$, such that $(c_0 \oplus c_1) = (a_0 \oplus a_1) \wedge (b_0 \oplus b_1)$. The parties use these pre-generated multiplication triples to exchange $d_i = x_i \oplus a_i$ and $e_i = y_i \oplus b_i$. Then, by considering the values $d = d_0 \oplus d_1$ and $e = e_0 \oplus e_1$, the output shares are obtained as $z_1 = (d \wedge e) \oplus (b_1 \wedge d) \oplus (a_1 \wedge e) \oplus c_1$ and $z_2 = (b_2 \wedge d) \oplus (a_2 \wedge e) \oplus c_2$. The advantages of multiplication triples include sending only one message for each party and using a smaller size of the messages, which is 2 + 2 bits instead of 2 + 4 bits.

3.4 Structure Function

The focus of the theory of reliability is on describing the functioning of a system under the structure of this system and the functioning status of its components [29]. The structure function is a model that determines the status of the system given the status of its components [30] which plays an essential role in reliability assessment. For a system with m components, considering the status of components as the state vector $\boldsymbol{x} = (x_1, x_2, \cdots, x_m) \in \{0, 1\}^m$, $x_i = 1$ if the i-th component functions, and $x_i = 0$, otherwise. In definition of $\boldsymbol{x}$, the labelling of each component is arbitrary but must be fixed. The structure function φ: $\{0, 1\}^m \rightarrow \{0, 1\}$, is defined for all possible vectors $\boldsymbol{x} \in \{0, 1\}^m$. In more details, the structure function φ is a Boolean function that indicates the status of the system (success or failure). $\varphi(\boldsymbol{x})$ takes the value 1 if the system functions (success state) and 0 if the system does not function (failure state) for state vector $\boldsymbol{x}$ [31].

3.5 NSGA-II Algorithm

Multi-objective optimization models complex optimization problems in which objectives under consideration conflict with one another and optimizing a solution with respect to a specific objective can lead to an unacceptable result with respect to other objectives [32]. Researches have proposed methods using genetic algorithm (GA) to solve the multiple-objective optimization problem. NSGA-II is one of the most popular multiobjective optimization algorithms because of the specific characteristics including fast crowded distance estimation procedure, fast non-dominated sorting approach and simple crowded comparison operator. We briefly describe NSGA-II as following steps [33].

- *Population Initialization.* The NSGA-II algorithm initializes the population based on the problem range and constraints.
- *Non Dominated Sort.* The sorting process is done based on non domination criteria of the initialized population.
- *Crowding Distance.* After completing sorting, algorithm assigns the crowding distance value front wise and selects the individuals in population based on rank and crowding distance.
- *Selection.* In this step, the individuals are selected by a binary tournament selection with a crowded-comparison operator.
- *Genetic Operators.* The NSGA-II algorithm applies the single-point crossover and bitwise mutation for binary-coded GAs and the simulated binary crossover operator and polynomial mutation for real-coded GAs.
- *Recombination and Selection.* The current generation population and the offspring population are combined. Then, the individuals of the next generation are chosen by selection, crossover and mutation. Finally, the new generation is filled by each front subsequently until its size exceeds the size of current population.

4 Proposed Method

4.1 Motivation

In order to compromise a router and disrupt routing process, an attacker should identify the active routers. Therefore, if the router's status is unknown, the success of the attacker for its compromising will be reduced. Hence, the policies of SDN controllers about which routers forward packets in routing operations are important and the privacy of SDN policies must be preserved. In this problem, privacy means that the action of routers in packet forwarding according to defined rules should be kept private. On the other hand, the source node needs to be informed to ensure that its packet can reach the desired destination. One solution is that the SND controllers compute a Boolean function in which its output denotes the successful routing.

We utilize this concept of structure function which determines the status of a system based on the status of its components, and consider an SDN as the system and the routers as its components. Then, the structure function is computed, and the function output taking value 1 denotes that the network routers in path of source to destination forward packets. Also, we apply the SMC technique for secure computing of the structure function without disclosing the status of routers. Then, to deal with the scalability problem of the proposed method due to the number of controllers, in Section 5 we model the problem as a multi-objective problem and then compute the optimal number of SDN controllers for secure computing structure function by NSGA-II. The proposed method can be described in these steps, where their details are specified in the followings:

- modeling structure functions for the routing process in SDN;
- determining the inputs of SDN controllers for structure functions whenever a source node requests an SDN controller to check whether it can send its packets to a destination;
- constructing the circuit representing the resultant structure function;
- evaluating the circuit by running the GMW protocol;
- computing the optimal number of SDN controllers by modeling a multi-objective problem.

4.2 Modeling Structure Function

In this paper, the structure function $\varphi(x_1, x_2, \ldots, x_n)$ models the success or failure of the SDN in routing process from a source to a destination, given the policies defined for its route performance of routers in the routing process. We consider an SDN as a system and routers as its components. The network routes successfully if all routers in at least one path from the source to the destination successfully transmit the network packets.

First, we consider the following labeling for each router and the binary random variable x_i that indicates the status of router i.

- $x_i = 1$ denotes that the router i forwards a packet.
- $x_i = 0$ denotes that the router i does not forward a packet.

We define packet transmission as success or failure. In other words, two states are considered.

- *Success Status.* φ takes the value 1 if the SDN transmits data from the source to the destination for a given period of time.
- *Failure Status.* φ takes the value 0 if SDN fails to perform routing satisfactorily.

4.2.1 *Structure Function in Terms of Minimal Paths*

To describe the SDN reliability, we define the concepts of path, minimal path, cut set, and minimal cut set for the network according to [30]. A path in the network is a set of routers, such that if all the components in this set are successful, then the system will be successful. A minimal path is a set of routers that comprise a path, but the removal of any of these routers will cause what remains from this path not to be a network path. It is possible to determine the structure function in terms of the set of minimal paths[34]. Let P be a set of routers comprising a minimal path. Using x_i as an indicator of the status of router i, the event of a path to be successful is the Boolean function $\prod_{i\in P} x_i$. The event of the failed path is $1-\prod_{i\in P} x_i$.

Let $P_1, P_2, \ldots, P_n$ be the collection of all minimal paths of the network for a given source to a given destination. The network is successful if at least one of the minimal paths does not fail. Then the structure function can be determined by

$$\varphi(\boldsymbol{x}) = 1 - \left(1-\prod_{i\in P_1} x_i\right)\left(1-\prod_{i\in P_2} x_i\right)\ldots \left(1-\prod_{i\in P_i} x_i\right). \quad (1)$$

For example, the structure function for the bridge network of Fig.1 from (1) is

$$\varphi(\boldsymbol{x}) = 1-(1-x_1x_3x_5)(1-x_2x_3x_4)(1-x_1x_4)(1-x_2x_5).$$

4.2.2 *Input Data*

It is our goal to simulate the structure function for determining whether packets sent by the source are successfully received by the destination or not. We set $x_i = 1$ if the associated SDN controller defines a rule for forwarding the received packet in the routing table of the i-th router. If there is no information in the router table of the i-th router, then x_i will be set to 0.

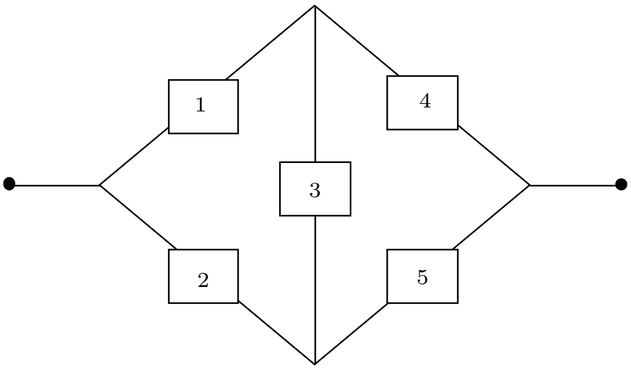

Fig.1. Bridge network graph.

4.2.3 Circuit Representation

For computing the structure function, any multiplication is represented by an AND gate. The addition or subtraction operation that is calculated between the sentences in structure function, can be computed with XOR gate.

4.2.4 Secure Computation of Structure Function

When a source node wants to send a packet to a destination node, it sends a request packet that contains the destination address to the corresponding SDN controller. Then the SDN controller sends a request to all SDN controllers. All SDN controllers participate in secure computation of the structure function over the status of their routers based on the GMW protocol. Suppose there are K SDN controllers and S_i denotes the i-th SDN controller who controls n_i routers. S_i shares the values of $x_{i,1},\dots,x_{i,n_i}$ with other SDN controllers using secret sharing scheme. Then all SDN controllers compute the structure function on inputs $x_{i,1},\dots,x_{i,n_i}, i=1,\dots,K$ by the GMW protocol, in which the shares of final gate's output are reconstructed by the requester SDN controller. If the result of the structure function is 1, then the requester SDN controller informs the source node that it can send its packet to the destination address.

4.3 Security and Privacy

In the proposed method, a circuit representing the structure function is public and the GMW protocol is applied for its secure evaluation. In other words, the GMW protocol securely evaluates a Boolean circuit that computes the structure function. Similar to [12, 13], we assume that the computing parties, i.e., SDN controllers, are semi-honest. The SDN controllers secretly share their inputs to one another. Then, the controllers apply the GMW protocol to evaluate the circuit gate-by-gate while they maintain the secret shard value on each wire invariant. The SDN controllers securely compute a secret sharing of the output wire of the gate using the secret shared values over the input wires, and XORing the shares and running OT protocol for XOR/NOT and AND gate, respectively. Finally, the SDN controllers send their corresponding shares of the output to the requester SDN controller to compute the structure function. Note that for running the GMW if only one party computes the output like our method, a secure channel is not required and in the case that multiple parties compute the output, only the final messages from the parties must be encrypted.

The correctness of the proposed approach is derived from correctness of the circuit representing the structure function and the correctness of the GMW protocol[7]. The security and the privacy of our method are derived from the security proof of the GMW protocol[7]. To privately evaluate a given circuit, GMW hides the intermediate values on internal wires of the circuit. These properties are formally descried in Theorem 1. In short, we say that the security of the semi-honest computing parties lies in the security of the OT protocol. Assuming the security of the OT protocol, the security of the GMW protocol is derived from the fact that parties only see random values until the end of the protocol and hence, they learn nothing beyond the output, as required. Note that the security of the GMW is information-theoretic if an ideal OT can be realized.

Theorem 1. *The GMW protocol privately evaluates the circuit representing a Boolean (for example the structure function) in the presence of a semi-honest adversary who corrupts at most $n-1$ out of n computing parties*[7].

Proof. We use the standard method for showing security of our scheme using the ideal/real world paradigm[24]. This proof is inspired from [35] and to put it simply, we prove and define the security of the protocol for semi-honest setting as follows.

Definition 1. *A protocol π is said to securely compute the functionality f in the semi-honest model if for every probabilistic polynomial-time real adversary $\mathcal{A}$ there exists a probabilistic polynomial-time ideal adversary or simulator $\mathcal{S}$ such that*

$$\{IDEAL_{f,\mathcal{S}(z)}(x,y)\}_{x,y,z} \overset{c}{\approx} \{REAL_{\pi,\mathcal{A}(z)}(x,y)\}_{x,y,z},$$

where $x,y,z \in \{0,1\}^*$ and $IDEAL_{f,\mathcal{S}(z)}(x,y)$ is defined as the output pair of the honest party and the adversary $\mathcal{S}$ from the ideal execution, and $REAL_{\pi,\mathcal{A}(z)}(x,y)$

is defined as the output pair of the honest party and the adversary $\mathcal{A}$ from the real execution. In other words, the input/output distributions of the adversary and the participating parties in the real and ideal executions are computationally indistinguishable.

Consider the parties $P_1, \ldots, P_n$ with inputs $x_1, \ldots, x_n$ and outputs $y_1, \ldots, y_n$. Suppose that the adversary $\mathcal{A}$ controls all parties but P_m, denoted by C. Then, the simulator $\mathcal{S}$ receives values of x_i and y_i, from all $P_i \in C$ and behaves as follows. For shares of inputs, $\mathcal{S}$ sends the random shares received from P_i's, denoted by $s_{i,m}$, to P_m. Moreover, it forwards the random shares generated by P_m, denoted by $s_{m,i}$, to all $P_i \in C$. For emulation of a multiplication gate, the simulator $\mathcal{S}$ considers the following values. $\forall i < m$, $\mathcal{S}$ selects a random bit as the value which is obtained in running the OT with P_m. Also, $\forall i > m$, $\mathcal{S}$ selects the required random shares and sets the inputs of the OT with P_m. Finally, for any output y_i of $P_i \in C$, $\mathcal{S}$ computes the message received from P_m by XORing of y_i and the shares of input owned by P_i.

According to the described simulation, the random shares $s_{i,m}$ and $s_{m,i}$ are identically distributed. Also, similar to the protocol in running of the OT, for $i < m$, OT's output is random and for $i > m$, the OT's input is specified as in real execution of the protocol. Furthermore, for outputs, messages received from P_m in ideal and real worlds are identically distributed. Consequently, the protocol is secure since the output distributions of the adversary in the real and ideal executions are identical and computationally indistinguishable. □

5 Determining the Optimal Number of SDN Controllers

As described in Subsection 4.2.4, the SDN controllers should participate to compute the structure function. We determine the optimal number of SDN controllers by considering this issue as a multi-objective optimization problem. Multi-objective optimization is concerned with optimization problems which include more than one objective function to be optimized simultaneously. Multi-objective optimization is applied in situations where optimal decisions must be taken in the presence of trade-offs between two or more conflicting objectives[36]. Our goal is determining the number of SDN controllers while minimizing their number in a way to reduce the cost and the latency between routers and their corresponding controllers and simultaneously increase the network reliability. We consider the system model in Fig.2.

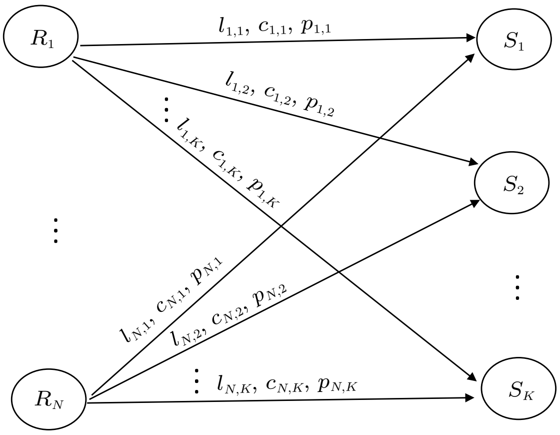
R_1
R_N
S_1
S_2
S_K
$l_{1,1},\ c_{1,1},\ p_{1,1}$
$l_{1,2},\ c_{1,2},\ p_{1,2}$
$l_{1,K},\ c_{1,K},\ p_{1,K}$
$l_{N,1},\ c_{N,1},\ p_{N,1}$
$l_{N,2},\ c_{N,2},\ p_{N,2}$
$l_{N,K},\ c_{N,K},\ p_{N,K}$

Fig.2. System model.

The parameters used in the system model are as follows.

- R_i represents a router i, for all integers i in the interval $[1, N]$.
- S_j represents an SDN controller j, for all integers j in the interval $[1, K]$.
- c_{ij} represents the cost of a connection of router R_i to the SDN controller S_j.
- a_{ij} is set to 1, if the router R_i is attached to controller S_j. Otherwise, it is set to 0.
- p_{ij} represents the probability of honest behavior of the controller S_j in the status announcement of the router R_i.
- l_{ij} represents the latency of the shortest path between router R_i and controller S_j.

We suppose that the number of SDN controllers is equal to or smaller than the number of routers, i.e., $K \leqslant N$. Also, any router is connected to only one controller. In other words, for all i:

$$\sum_{j=1}^{K} a_{ij} = 1. \tag{2}$$

We define the following reference object function to model the latency metric.

$$L\left(a_{11},\ \ldots, a_{NK}\right) = \sum_{i=1}^{N}\sum_{j=1}^{K} a_{ij} l_{ij}.$$

Also, the object function for cost metric is:

$$C(a_{11},\ \ldots, a_{NK}) = \sum_{i=1}^{N}\sum_{j=1}^{K} a_{ij} c_{ij}.$$

We suppose that SDN controllers may have a malicious behavior in the GMW protocol and share an incorrect value. Hence, we define the reliability of SDN controllers as the following.

$$R(a_{11},\ \ldots, a_{NK}) = \prod_{i=1}^{N} (\sum_{j=1}^{K} a_{ij} p_{ij}).$$

Our goal is determining the number of SDN controllers which decreases network latency, reduces cost, and increases the reliability. Therefore, the optimization problem is

$$\begin{aligned}&\text{minimize } L\left(a_{11},\ \ldots, a_{NK}\right),\ C\left(a_{11},\ \ldots, a_{NK}\right)\\&\text{maximize } R(a_{11},\ \ldots, a_{NK})\\&\text{subject to (2).}\end{aligned}$$

We use NSGA-II[9] to solve this multi-objective optimization problem as described in Section 6.

6 Evaluation

In this section, we analyze the performance of the proposed method. The experiments are run on a 64-bit machine with an Intel® Core™ i7-312QM CPU at 2.10 GHz and 6GB RAM, running Windows 7. We use Python version 3.6.2 and Intel® Distribution to support parallelism across Python. As previously stated, our work includes two parts. The objective of the first part is secure and fast computing of structural function in SDNs using the GMW protocol. The purpose of the second part is to determine the optimal number of SDN controllers that provides the least amount of cost and delay in the SDN and the most reliability of SDN controllers while assigning the routers to the controllers.

First, we evaluate the second part using NSGA-II[9]. The performance of NSGA-II is investigated on several samples including different numbers of routers. The parameters set are shown in Table 1. Also, the NSGA-II algorithm applies the values according to Table 2. We consider the cost of the SDN controllers in the interval [3 000\$, 200 000\$]. We use the data of research[37] for determining the values for the latency between a router and an SDN controller and it is set to values between 10 and 10 000 miles. Also, the probability of honest behavior of any controller in the router's status announcement is set to [0.4, 0.8].

Table 1. Parameters of Our Simulation

Parameter	Value
Latency	[10, 10 000]
Cost (\$)	[3 000, 200 000]
Probability of honest behavior	[0.4, 0.8]

Table 2. Parameters of NSGA-II

Parameter	Description	Value
Max_gen	Maximum of generation	[100, 1 000]
Pop_size	Size of population	[20, 50]
Mut_prob	Mutate probability	0.2
Mutate	Mutate	Random
Crossover	Crossover	Two_point

It should be mentioned that all simulations have been performed on random graphs. In this paper, six samples are considered based on the number of routers in the SDN, which covers a network with a number of routers from 25 to 250. This number also supports SDNs with millions of end-users. The results of simulation are described in Table 3. This table contains the minimum number of controllers required, K, and the fitnesses which are provided by NSGA-II. It should be noted that in genetic algorithms, a fitness function, known as an evaluation function, measures how close a given solution is to the optimum solution. It determines how a solution is fit for the desired problem. It is obvious that with the increase in the number of routers in SDN, the number of controllers required is also increased. This increase can be observed in Table 3. For 250 routers, at least 13 controllers are needed. Therefore, on average about 7.6% of routers are connected to the same controller. In this method, the exact number of routers and which router is connected to which controller are also determined by considering the probability of honest behaviors of the controllers, the latency and the cost.

Table 3. Results of Samples Used in NSGA-II

Number of Samples	N	K	Latency	Cost	Reliability of SDN Controllers
1	25	2	2 756.429 93	51 927.800 8	1.641 115 13e-09
2	50	3	3 132.788 82	62 044.261 7	1.507 452 60e-18
3	100	3	2 150.242 92	52 401.238 3	2.106 253 83e-16
4	150	4	4 073.672 12	76 685.242 2	5.241 838 23e-31
5	200	5	3 776.374 27	86 505.148 4	5.560 912 83e-41
6	250	13	3 939.700 44	84 648.031 2	1.987 775 04e-52

In Fig.3, the number and values of the fitness function are observed for different generations. Fig.3 represents the optimal performance and convergence of the selected response to the optimal value of cost, latency and reliability. This output is calculated for 1 000 generations and for 100 routers. As seen in Fig.3(a), the minimum and the best number of controllers is 3.

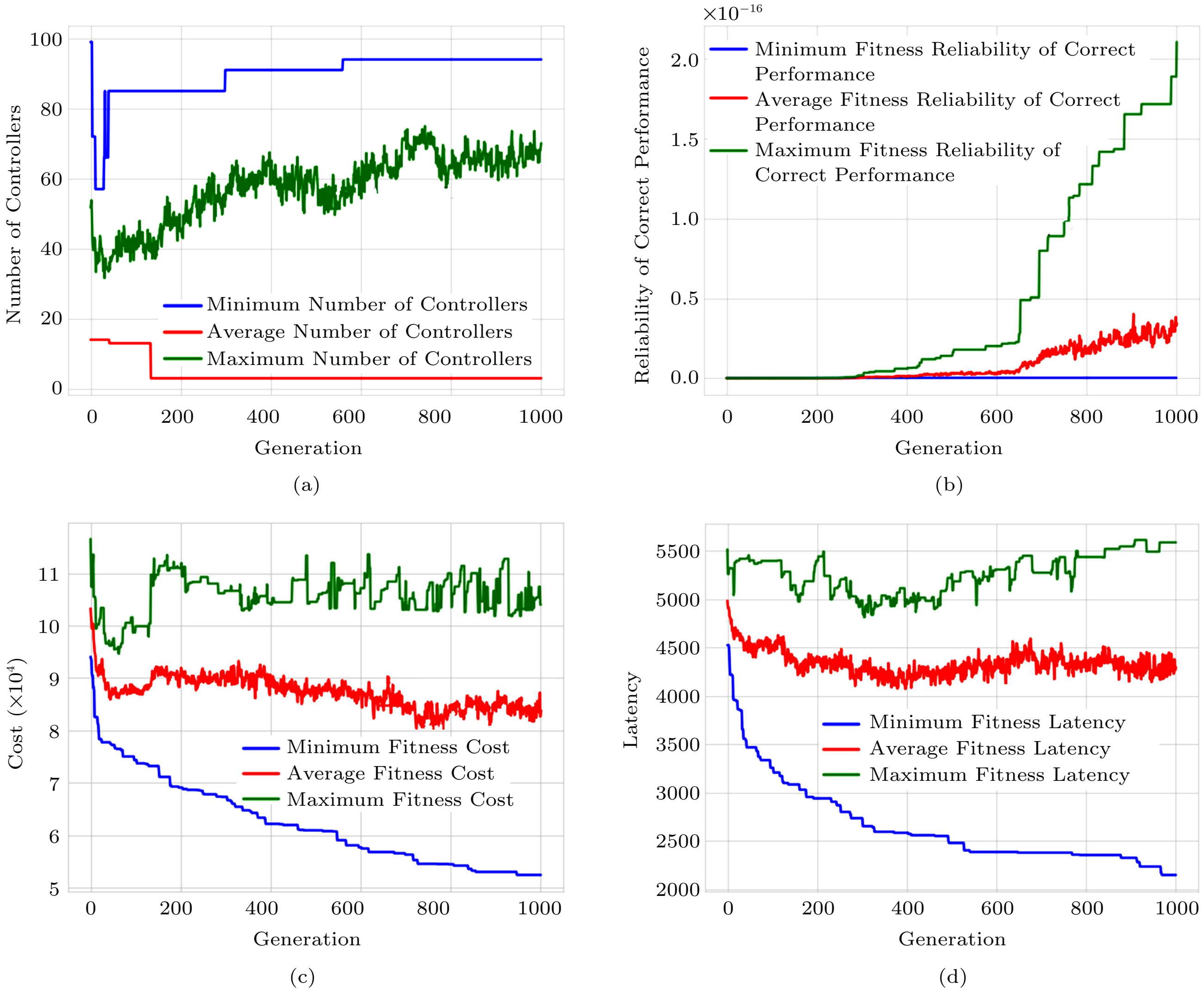

Fig.3. Values of multi-objective fitness in NSGA-II. (a) Number of controllers. (b) Maximum reliability. (c) Minimum cost. (d) Minimum latency.

This value converges to 3, after the 160th generation. The maximum reliability of SDN controllers shown in Fig.3(b) indicates that this method is successful in selection of honest controllers. On the other hand, the tendency to the lowest cost is clearly visible in Fig.3(c). This figure indicates that the selection of controllers and the allocation of routers to the controllers are such that will lead to the least cost. Also, in Fig.3(d) the minimum latency is selected.

We compute the optimal number of SDN controllers according to NSGA-II and then evaluate the proposed method for secure computing of structural function using the GMW protocol. According to Table 4, the run time for almost all samples is less than 0.50 seconds that indicates the efficiency of our method. As the number of routers increases, the number of paths between them increases and the length of relationships increases. As a result, the number of gates increases at different levels and the runtime increases. It should be mentioned that the calculations of the gates are done in parallel at each level, and the increase of the number of gates at one level has no effect at run time. According to Table 4, the growth rate of the number of SDN controllers is low, which indicates the efficiency of the first proposed method for secure computing of structural function in networks with a large number of routers.

The accuracy of the results is also evaluated by an algorithm that directly correlates the values in the different equations of structural functions and the accuracy of 100% of the results is numerically verified. Regarding other relationships for other random graphs, the correctness of relationships has also been obtained. All the materials presented indicate the applicability and the efficiency of the proposed method.

Table 4. Computation Time of the Structure Function in GMW Protocol

Number of	Number of	Number of	Time (s)
Samples	Routers	Controllers	
1	25	2	⩽1.0e-19
2	50	3	⩽1.0e-19
3	100	3	0.010 000 228 881 835 938
4	150	4	0.029 999 971 389 770 508
5	200	5	0.050 000 190 734 863 280
6	250	13	0.480 000 734 329 223 630

7 Conclusions

In this paper, we employed secure multi-party computation protocol for preserving the privacy of policies of SDN controllers about the routers for forwarding a received packet. Our framework is based on structure function evaluation by the GMW protocol. We suggested a method for determining the optimal number of SDN controllers for the number of routers by NSGA-II. The proposed solution seeks to minimize the number of controllers in the network while reducing the cost and latency and increasing the reliability that leads to a multi-objective optimization problem. Then, the optimal number of controllers is used in SDN to compute the structure function. The simulation results showed that the proposed number of SDN controllers results in a short time for computing the structure function. Therefore, the proposed method can be used in secure SDN to preserve the privacy of policies of network administrators.

References

[1] Nadeau T D, Gray K. SDN: Software Defined Networks: An Authoritative Review of Network Programmability Technologies (1st edition). O'Reilly Media, 2013.

[2] Feldmann A, Heyder P, Kreutzer M *et al.* NetCo: Reliable routing with unreliable routers. In *Proc. the 46th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshop*, June 2016, pp.128-135.

[3] McKeown N, Anderson T, Balakrishnan H *et al.* OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review*, 2008, 38(2): 69-74.

[4] Sezer S, Scott-Hayward S, Chouhan P K *et al.* Are we ready for SDN? Implementation challenges for software-defined networks. *IEEE Communications Magazine*, 2013, 51(7): 36-43.

[5] Cramer R, Damgård I, Nielsen J B. Secure Multiparty Computation and Secret Sharing (1st edition). Cambridge University Press, 2013.

[6] Yao A C. Protocols for secure computations. In *Proc. the 23rd Annual Symposium on Foundations of Computer Science*, November 1982, pp.160-164.

[7] Goldreich O, Micali S, Wigderson A. How to play any mental game or a completeness theorem for protocols with honest majority. In *Proc. the 19th Annual ACM Symposium on Theory of Computing*, January 1987, pp.218-229.

[8] Aven T, Jensen U. Stochastic Models in Reliability (2nd edition). Springer, 2013.

[9] Deb K, Pratap A, Agarwal S, Meyarivan T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 2002, 6(2): 182-197.

[10] Rekhter Y, Li T. A border gateway protocol 4 (BGP-4). https://www.rfc-editor.org/rfc/pdfrfc/rfc1771.txt.pdf, May 2020.

[11] Zhao M, Zhou W, Gurney A J, Haeberlen A, Sherr M, Loo B T. Private and verifiable interdomain routing decisions. *IEEE/ACM Transactions on Networking*, 2016, 24(2): 1011-1024.

[12] Gupta D, Segal A, Panda A *et al.* A new approach to interdomain routing based on secure multi-party computation. In *Proc. the 11th ACM Workshop on Hot Topics in Networks*, October 2012, pp.37-42.

[13] Asharov G, Demmler D, Schapira M, Schneider T, Segev G, Shenker S, Zohner M. Privacy-preserving interdomain routing at Internet scale. *Proceedings on Privacy Enhancing Technologies*, 2017, 2017(3): 147-167.

[14] Dethise A, Chiesa M, Canini M. Prelude: Ensuring interdomain loop-freedom in SDN-enabled networks. In *Proc. the 2nd Asia-Pacific Workshop on Networking*, August 2018, pp.50-56.

[15] Kukkala V B, Saini J S, Iyengar S. Secure multiparty computation of a social network. https://eprint.iacr.org/2015/817.pdf, May 2020.

[16] Boshrooyeh S T, Küpçü A, Özkasap Ö. Privado: Privacy-preserving group-based advertising using multiple independent social network providers. https://eprint.iacr.org/2019/372.pdf, May 2020.

[17] Demmler D, Schneider T, Zohner M. Ad-hoc secure two-party computation on mobile devices using hardware tokens. In *Proc. the 23rd USENIX Security Symposium*, August 2014, pp.893-908.

[18] Oleshchuk V. Internet of Things and privacy preserving technologies. In *Proc. the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology*, May 2009, pp.336-340.

[19] von Maltitz M, Carle G. Leveraging secure multiparty computation in the Internet of Things. In *Proc. the 16th Annual International Conference on Mobile Systems, Applications, and Services*, June 2018, pp.508-510.

[20] Chiesa M, di Lallo R, Lospoto G, Mostafaei H, Rimondini M, di Battista G. PrIXP: Preserving the privacy of routing policies at Internet eXchange points. In *Proc. the 2017 IFIP/IEEE Symposium on Integrated Network and Service Management*, May 2017, pp.435-441.

[21] Chiesa M, Demmler D, Canini M, Schapira M, Schneider T. SIXPACK: Securing internet exchange points against curious onlookers. In *Proc. the 13th International Conference on Emerging Networking Experiments and Technologies*, December 2017, pp.120-133.

[22] Cho C, El Defrawy K, Kim H T J, Lampkins J D. Privacy-preserving multi-client and cloud computation with application to secure navigation. U.S. Patent, 2019. http://www.freepatentsonline.com/20190042788.pdf, May 2020.

[23] Kreutz D, Ramos F M, Veríssimo P, Rothenberg C E, Azodolmolky S, Uhlig S. Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, 2015, 103(1): 14-76.

[24] Hazay C, Lindell Y. Efficient Secure Two-Party Protocols: Techniques and Constructions. Springer-Verlag Berlin Heidelberg, 2010.

[25] Schneider T. Engineering Secure Two-Party Computation Protocols: Design, Optimization, and Applications of Efficient Secure Function Evaluation. Springer-Verlag Berlin Heidelberg, 2012.

[26] Shamir A. How to share a secret. *Communications of the ACM*, 1979, 22(11): 612-613.

[27] Rabin M O. How to exchange secrets with oblivious transfer. https://eprint.iacr.org/2005/187.pdf, May 2020.

[28] Beaver D. Efficient multiparty protocols using circuit randomization. In *Proc. the 11th Annual International Cryptology Conference*, August 1991, pp.420-432.

[29] Coolen F P, Coolen-Maturi T. The structure function for system reliability as predictive (imprecise) probability. *Reliability Engineering & System Safety*, 2016, 154: 180-187.

[30] Rausand M, Høyland A. System Reliability Theory: Models, Statistical Methods and Applications (2nd edition). Wiley-Interscience, 2003.

[31] Gertsbakh I, Shpungin Y. Network Reliability and Resilience. Springer, 2011.

[32] Konak A, Coit D W, Smith A E. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 2006, 91(9): 992-1007.

[33] Yusoff Y, Ngadiman M S, Zain A M. Overview of NSGA-II for optimizing machining process parameters. *Procedia Engineering*, 2011, 15: 3978-3983.

[34] Marichal J L. Structure functions and minimal path sets. *IEEE Transactions on Reliability*, 2016, 65(2): 763-768.

[35] Goldreich O. Foundations of Cryptography: Volume 2, Basic Applications (1st edition). Cambridge University Press, 2009.

[36] Deb K. Multi-objective optimization using evolutionary algorithms (1st edition). Wiley, 2001.

[37] Heller B, Sherwood R, McKeown N. The controller placement problem. In *Proc. the 1st Workshop on Hot Topics in Software Defined Networks*, August 2012, pp.7-12.

Maryam Zarezadeh received her B.Sc. degree in information technology (IT) engineering from University of Isfahan, Isfahan, in 2010, and her M.Sc. degree in IT engineering (information security) from Shahed University, Tehran, in 2013. She is currently a Ph.D. student in IT engineering (information security) at University of Isfahan, Isfahan. Her research interests are security protocols, secure multiparty computation, and network security.

Hamid Mala received his B.S., M.S., and Ph.D. degrees in electrical engineering from Isfahan University of Technology (IUT), Isfahan, in 2003, 2006, and 2011, respectively. He joined the Department of Information Technology Engineering, University of Isfahan (UI), Isfahan, in September 2011, as an assistant professor. He is currently with the Faculty of Computer Engineering, UI, as an associate professor. His research interests include design and cryptanalysis of block ciphers, cryptographic protocols, and secure multiparty computation.

Homa Khajeh received her B.Sc. degree in software engineering from Islamic Azad University, Najafabad Branch (IAUN), Isfahan, in 2009, and her M.S. degree in software engineering from Science and Art University, Yazd, in 2014. Her research interests are mainly in the field of information retrieval, search engine, machine learning, and big data.