Sharing is Caring: A Collaborative Framework for Sharing Security Alerts

Muhammed Ajmal Azad[a,*], Samiran Bag[b], Farhan Ahmad[a], Feng Hao[b]

[a]*Department of Computer Science and Mathematics, University of Derby, Derby, United Kingdom*
[b]*Department of Computer Science, University of Warwick, Coventry, United Kingdom*

Abstract

Collaboration is a keystone of defense in the field of cybersecurity. A collaborative detection system allows multiple collaborators or service providers to share their security-incident-response data, in order to effectively identify and isolate stealthy malicious actors who hide their traffic under the umbrella of legitimate Internet data transmissions. The fundamental challenge in the design of a collaborative system is ensuring the privacy of collaborators in a decentralized setting without incurring substantial computation and communication overheads. In this paper, we use healthcare as a case study and present Sharing Is Caring (SIC), a framework that allows multiple healthcare organizations to share their security defense and attack data with other organizations for the collaborative defense against common attackers without compromising the privacy of their system configurations and user data. The SIC framework ensures two essential properties: 1) it ensures that no party should learn how a particular healthcare organization has reacted to suspected IP addresses, attacks or security incidents; and 2) it performs operations in a decentralized setting, without relying on a trusted third party. We provide an analysis of the privacy and security properties of our framework against honest-but-curious as well as malicious players. We prototype the proposed system and evaluate its performance in terms of computation time and communication bandwidth. The reasonable computation cost and bandwidth overhead make the SIC framework a feasible choice for the privacy-preserving exchange of security information among the collaborating healthcare organizations.

Keywords: Collaborative Security, Privacy, Secure Computation, Privacy-preserving Alert Sharing

1. Introduction

Today, healthcare providers are relying on the advancement of an Internet-based system to provide easy access to patients, provide on-demand services, and improve healthcare

[*] Corresponding Author

Email addresses: m.azad@derby.ac.uk (Muhammed Ajmal Azad), Samiran.bag@warwick.ac.uk (Samiran Bag), f.ahmad@derby.ac.uk (Farhan Ahmad), feng.hao@warwick.ac.uk (Feng Hao)

outcomes. The advancement of technology has enabled patients not to physically visit the hospital for their routine checkups. Instead, patients can benefit from the use of Interconnected Internet of Medical things (IoMT) for reporting their symptoms and getting medical advice. The use of IoMT devices in healthcare is rapidly increasing and is expected to reach a value of 52 billion US dollars by 2022 [1]. These devices generate massive amount of data related to the health of patients which could be used for personalized recommendations for the treatment and timely identification of any health problem. This health data is normally stored in the organization's centralized data center. The use of health data and Internet-connected Medical devices have not only brought benefits in terms of better health services, personalized advice but have also created serious cybersecurity and data privacy risks.

Hackers take advantage of software vulnerabilities in IoMT, open access points, flaws in the design of the Internet of Medical devices, and a lack of training in using the advanced technology systems to exploit data breach and security attacks. Cybercriminals not only attack the system for compromising the integrity and confidentiality of the data but also make system resources unavailable to legitimate users i.e., they disrupt the facilities of patient care through distributed denial of service (DDoS) attacks or ransomware attack. These types of attacks on the services of healthcare providers can bring catastrophic consequences to healthcare providers. A recent survey by IBM indicates that healthcare providers suffer a loss of nearly 6.5 million US dollars from these data breaches [2]. During 2019 in the UK, 67% of the healthcare organizations have reported a cybersecurity incident [3], 48% of these attacks were attributed to viruses or malware spread from the IoT devices e.g. Marai botnet [4]. In May 2017, WannaCry [5] exploited vulnerabilities in the unpatched Windows-based systems and launched a ransomware attack on many hospitals across the world. The medical devices running on the Windows operating system were infected with the virus, which resulted in the cancellation of patients' appointments and locking of patient-records. The virus also affected the working of Internet-enabled medical devices. Specifically, National Health Service (NHS) in the UK has reportedly paid around £92 million as the result of a ransomware attack to their system [6].

Cyber threats (Dos, DDos, worms, Malware, Trojan, viruses, spam, etc.) are continuously evolving because of the deployment of a large number of IoT devices [7, 8, 9]Cyber attacks bring serious problems to users and service providers, including disruptions of operations of organizations and the financial loss in terms of ransomware and regulator's fines. Large organizations (public and private) and Internet Service Providers (ISPs) have deployed standalone intrusion detection solutions (IDS) for protecting their customers, network resources, and data repositories from cyber attacks. A standalone IDS creates the detection model based on the data and traffic logs collected at a single source (e.g. healthcare organization or ISP). Sophisticated malicious actors (stealthy and coordinated attackers) could intelligently manage to circumvent the standalone IDS for a relatively long time, by exhibiting traffic patterns similar to legitimate traffic patterns and a simultaneous low rate attack traffic in a large number of autonomous networks [10, 11]. Under this attack pattern, the standalone IDS shows a slow reaction because of a lack of enough information to be used for efficient and timely decisions.

The convergence of computer networks into the cyberspace, including the use of computer systems for various sensitive areas like cyber-physical systems in the healthcare, requires timely identification of malicious actors because of the severe consequences of the cyber attack. The collaboration among standalone IDS is the obvious choice to create a collective defense model against the sophisticated and stealthy cyber attackers. For the effective and improved cyber defense, the public organizations (CERT-UK and CERT-US [12, 13], European Union Agency for Network and Information Security (ENISA) [14]) and the private organizations (Internet Storm Center or ISC [15]) have already been encouraging organizations to collaborate to achieve effective cyber defense. The exchange of cyber incident information could have the challenge of privacy-preservation that restrains organizations from implementing collaborative information exchange.

A number of proposals have been proposed for the information sharing among the collaborators [16, 17, 18, 19]. These approaches can be grouped into two architectures: a centralized system – where collaborating organizations submit their data to the centralized system and a distributed system – where information from the multiple collaborating organizations is aggregated in a distributed way. The centralized system can be a single point of failure. Furthermore, the collaborator has to trust the centralized system for the protection of the shared data. The distributed system has challenges of scalability and the trade-off between privacy and computational overheads.

Several works have been proposed to ensure the privacy of collaborators. These systems have adopted semantics of data anonymization [20, 21], data sanitization [22] and cryptographic techniques [23, 24] to address the challenges of privacy in the centralized and decentralized settings. The data anonymization-based systems do not provide any meaningful information to the collaborating healthcare organizations and are prone to de-anonymization and de-identification attacks [25]. For example, in a collaboration setup like *Dshield*, the active adversary can easily learn the blacklisted rules of a particular organization by making a controlled active attack towards the targeted organization. The cryptography-based systems ensure the privacy of collaborators with the use of a trusted set of privacy peers or semi-trusted systems [23, 24]. Freudier et al. [24] use the secure private set intersection technique for protecting the exchange of blacklist database information between two collaborators, without revealing the size of a blacklist and non-similar data points. The system requires a semi-trusted third party for finding similar collaborators. However, the system is not scalable to a large number of collaborators. Locasto et al. [26] use bloom filters to protect the privacy of collaborators, but it is prone to the simple guessing attack. In our work, the privacy of the collaborators is ensured in a completely decentralized way, without relying on the trusted central system for the cryptographic operations and data handling.

In this paper, we propose the SIC (Sharing is Caring) framework where the collaborators, organizations, and health providers can share their views about the cyber incident, source host, IP address, or firewall alert to identify the stealthy and zero-day attacks. We achieve the benefits of collaboration with the inherent properties of privacy preservation and minimal overheads. The building block of the SIC framework consists of three major players: the collaborators (i.e., healthcare provider, ISP or any entity using information systems), the tally server, and the analyst. The collaborator generates a decision about the entity and reports this to the tally server in an encrypted form. The analyst processes the data from the tally server and ranks the severity of each host, IP address, or alert. The adversary or any party in the system cannot learn anything about the collaborators from the tally server data. We analyzed the security properties of the SIC framework for the honest-but-curious and malicious adversarial models. We have developed a prototype to extensively analyze the computational and bandwidth overheads for the varying number of responses. The results show that the SIC framework reports encrypted responses with small communication and computation overheads. Our system has several core properties: privacy-preservation of collaborators, decentralized operations without the trusted third party, verified result aggregation, correct operations even under an active adversary model, and small overheads. We believe that the proposed privacy-preserving collaboration system will bring significant benefits to collaborators in collaboratively identifying stealthy and zero-day attacks.

The rest of the paper is organized as follows. We first review related work in Section 2. Then, we present motivation and formalize the problem in Section 3. Section 4 provides an overview of the designed system. In Section 5, we give security analysis of our framework. Section 6 provides the prototype implementation and micro-benchmarks for the computation and bandwidth overheads. Section 7 concludes the paper.

2. Related Works

A great number of research works have been proposed for collaboration in order to improve intrusion detection and alert correlation. A multistage blacklisting ranking system is proposed by Zhang et al. [27] that measures how closely related an attack source is to a blacklisted entity. In the first stage, collaborating organizations provide network logs to the DShield database [15] to filter false positive and noise. In the second stage, the filtered data is sent to the two parallel analysis engines, for the relevance ranking of attack sources per contributor, and to estimate the maliciousness score of the entity. Finally, relevance rankings and maliciousness scores are combined to generate a final blacklist for each collaborator. The system has not taken any measures to protect the privacy of participating collaborators.

Fuentes et al. [28] use the format-preserving encryption technique for the information sharing where only the authorized participants can decrypt the aggregated ranking statistics. The system requires a centralized trusted system for generating the cryptographic parameters (public and private keys). A controlled data-sharing tool is proposed in [24] that allows collaborators to estimate the benefits of collaboration before taking part in the collaboration without disclosing their private data. The system is built on top of a secure private set intersection technique with the inherent properties of protecting the data set size and data points of the collaborators. The approach helps to minimize the false positive rate but is not scalable to support a large number of collaborators. Furthermore, the process of finding a closely related collaborator depends on the semi-trusted system.

The SEPIA (Security through Private Information Aggregation) [23] library allows participants to aggregate the data from multi-domains using the multiparty computation (MPC) technique. The library also has a function to correlate the events from multiple domains and compute multi-domain network statistics, i.e., entropy and distinct count. The library is dependent on a set of privacy peers for ensuring the privacy of data from multiple domains.

Bye et al. [20] proposed a decentralized intrusion detection system that anonymized IP-address of data contributors before sharing the data stream. However, anonymization does not guarantee privacy protection; it is subject to de-anonymization attacks by correlating information from multiple sources [29, 30, 25].

A number of collaborative distributed systems have been proposed to reduce the intrusion detection time. Janakiraman et al. [31] proposed a distributed scheme that enables a local IDS to share the incident report with trusted nodes through a peer-to-peer network. The DOMINO system [32] enables collaboration among multiple domains in a distributed way using the Distributed Hashing Table. The system consists of three participants: the overlay host, the collaborators providing alert data, and terrestrial contributors. The system improved the detection time and accuracy by correlating events from multiple sources, but it has major issues relating to the privacy protection of the exchanged data. A P2P-based distributed collaborative intrusion detection system is proposed in [26] that enables collaborators to exchange compressed information using bloom filters to the selected trusted peers. Vasilomanolakis et al. [33] present SkipMon, a distributed collaborative system that ensures the privacy of contributors through data compactness and the utilization of bloom filters. The system is not scalable to a large number of collaborators. Several privacy-preserving systems have also been proposed in other domains, for example protecting the network users from the spamming and worms [34, 35, 36, 37], optimizing the firewall rules [38], privacy-preserving anomaly detection, and data analytics [39, 40, 41].

A few standards have also been proposed for exchanging cyber incidents. Threat Information Expression (STIX) [42] is a machine-readable format that enables organizations to share their cybersecurity incident reports in order to empower an organization to react in response to incidents more effectively. Trusted Automated Exchange of Intelligence Information (TAXII) [43] is a RESTful API for exchanging the cyber incidents using HTTPS. These standards have not defined any mechanism to ensure the privacy of the participating organizations.

To the best of our knowledge, this work is the first attempt towards the design of a privacy-preserving alert sharing system among multiple collaborators. In our system, the participating collaborators can collaboratively uncover a certain type of malicious traffic from a particular source address and then establish the complete threat information without disclosing how this traffic has been affecting each individual organization. We achieve this by having the collaborators to exchange only the encrypted information to a decentralized system in such a manner that only the aggregate result is revealed while the individual input from each collaborating organization remains private.

3. Background and Motivation

The effectiveness of a security incident detection system can be substantially improved if a group of stakeholders (organizations, healthcare organizations, or Internet service providers) exchange their security-related incident for the collective defense against sophisticated attackers.

Information	**Accuracy**	**Scalability**	**Complexity**	**Bandwidth**	**Privacy**
Raw Data	High	low	High	High	No
Partial Processed	Medium	Medium	Medium	High	No
Processed Decision	Low	High	low	low	Partially
Encrypted Response	Low	Low	High	High	Yes

Table 1: Comparison in Characteristics among Different Information Sharing Methods.

The information used for the collaboration can be grouped into the following types: 1) exchanging raw data without filtering and anonymization, 2) exchanging partially processed data with anonymization (identity anonymization), 3) exchanging the final decision that the local collaborators have made against the source host or IP-address, and d) exchanging encrypted data. Each type of shared data provides different levels of privacy protection and different levels of accuracy. Table 1 presents the comparison between different information-sharing methods.

The exchanged data may contain information that could reveal sensitive information about the network users (e.g., IP addresses, the communication connection between IP addresses, interaction and browsing patterns of users, etc.) and the collaborators (firewall configurations, firewall rules, etc). The sharing of raw data could result in a threat to the privacy of the users, collaborators, and also have bandwidth and computation limitations. The privacy of the user can be protected by stripping off the payload part of the data packet and anonymizing the source and destination IP addresses. However, it does not guarantee privacy protection and is subject to deanonymization attacks [44].

Problem Formulation We consider a collaborative network that consists of N collaborating organizations. Each collaborating organization has deployed a standalone probe system that monitors the behavior of incoming and outgoing traffic. The standalone system triggers alert for the traffic from the malicious source IP address or the specific port. The collaborating organizations want to jointly compute the aggregation function on the generated alerts without disclosing their private data.

Let $N = \{N_1, N_2, ..., N_n\}$ be the set of organizations that could provide feedback about the particular query (a request sent out by the organization to other collaborating organizations for their view). The objective is to provide a platform that allows collaborating organizations to have an aggregate view of the result of a query in a privacy-preserving way. The problem is to compute the aggregate statistics for the query in such a way that the organization's response to the query should remain private throughout the aggregation process. The organization or the analyst should only learn the aggregate statistics, e.g., how many organizations have seen the same suspicious traffic that is being queried.

4. The SIC Framework

In this section, we describe the system architecture and the technical details behind the SIC framework.

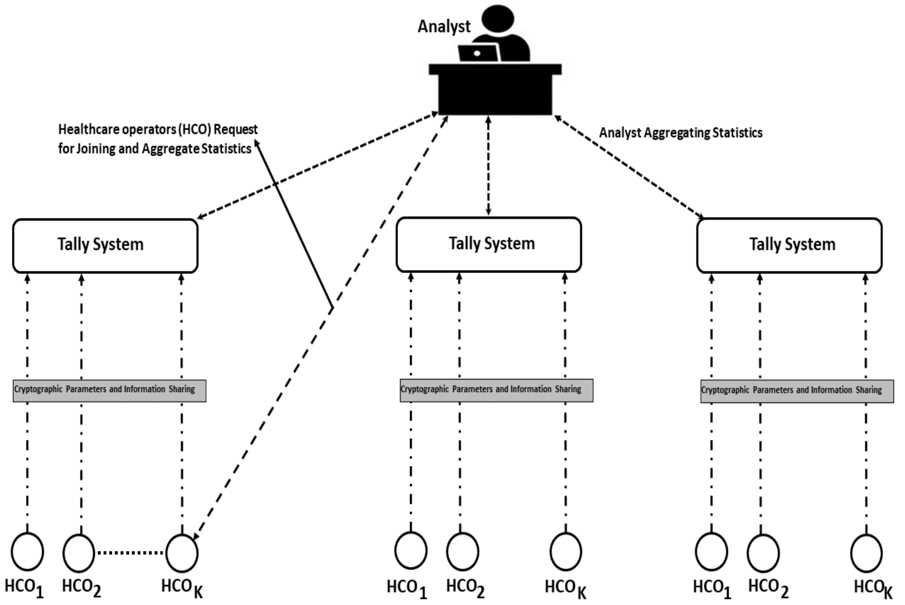
Analyst
Healthcare operators (HCO) Request for Joining and Aggregate Statistics
Analyst Aggregating Statistics
Tally System
Tally System
Tally System
Cryptographic Parameters and Information Sharing
Cryptographic Parameters and Information Sharing
Cryptographic Parameters and Information Sharing
HCO$_1$ HCO$_2$ HCO$_K$
HCO$_1$ HCO$_2$ HCO$_K$
HCO$_1$ HCO$_2$ HCO$_K$

Figure 1: The System Architecture of the Proposed System.

4.1. System Architecture

The system architecture of the SIC framework is shown in Figure 1, which consists of the following three main components.

The Collaborators: The collaborators monitor all incoming and outgoing traffic and classify it as malicious or non-malicious. For this purpose, each collaborator uses a standalone IDS for making any decision about the traffic patterns. The collaborators cooperate with each other by exchanging encrypted information about the query it received from other collaborators to compute an aggregate view.

The Tally Server (TS): The TS holds the public key of the collaborator, and the encrypted feedback provided by the collaborating entities. The TS is an append-only public database. It ensures the following: 1) all information published on the TS is publicly readable; 2) only the authenticated entities, namely collaborators, can write data to the system in an append-only manner. The TS essentially serves as a *public bulletin board*, which has been commonly used in e-voting, privacy-preserving recommendation and statistics aggregation systems [45, 46, 47].

The Analyst: The analyst serves as an anchor point between the collaborators and the TS to perform computation over the data from the TS. The analyst performs three responsibilities: first, it assigns collaborator an address of a suitable tally server; second, it broadcast a query to the collaborating organizations; and third, it performs aggregation of information from the TS for the collaborator. The analyst provides the final results for the query to the collaborating entities. The analyst does not need to be trusted, since all the computation tasks performed by the analyst are publicly verifiable.

4.2. Trust Model and Assumptions

We assume honest-but-curious (HBC) and malicious models. In HBC, collaborators perform operations honestly but try to learn the private information of other collaborators. In this model, we assume that collaborators provide the feedback honestly within the prescribed range. We consider this a reasonable assumption since collaborators are officially registered providers who care about their public reputation. We also assume that the TS is honest but curious i.e. the TS itself would not modify the previous information published on the tally server. In a malicious model, collaborators try to disrupt the operation of the system by providing out-of-range values. This can be addressed by using non-interactive zero-knowledge proofs to ensure a ciphertext is well-formed.

We make the following assumptions in the design of the SIC framework:

- we assume that each collaborating healthcare organization has a standalone IDS system for monitoring the behavior of the incoming traffic. The healthcare organizations wish to have an aggregated view about whether traffics with particular patterns and from particular sources are malicious.

- We assume there is an analyst that performs two functions: first, it assigns collaborator the address of a suitable tally server for posting the information, and second, it performs computation in a publicly verifiable manner on behalf of the collaborator using information from the TS.

- The availability of an append-only tally server (TS) is known to the collaborators and the analyst.

4.3. Example Scenario

We present an example of having three collaborating healthcare organizations $P1, P2, P3$. Each of these collaborators has a unique identity and has access to a tally server $TS1$ for sharing information. The collaborator locally generates private and public keys, keeps the private key to himself and posts the public key to the tally server. Suppose, $P1$ has classified a specific host or traffic pattern as malicious and wishes to know whether other hosts have also considered it malicious. For this purpose, it generates a query *(e.g "Is traffic from IP X.X.X.X, Port=80 malicious?")* to the analyst for the aggregate statistics. The analyst broadcasts the query to all collaborators. The collaborators in return submit responses to their TS. The collaborating healthcare organizations ($P1, P2$, and $P3$) first generate the encryption keys using public keys of P1, P2, and P3 from the TS, and secondly, report encrypted responses to the TS using the public keys as well as the private keys. Once all three collaborators have posted their responses, the aggregate statistics can be computed by any public observer. Finally, each collaborating healthcare organization updates its local blacklist and exchanges detailed information about the threat perception with all collaborators according to his policies.

4.4. The Cryptographic Protocol

The cryptographic operations of the SIC framework consist of the following steps.

4.4.1. Joining

To join the collaborative system, a collaborator first sends a joining request to the analyst. The analyst allocates the collaborator a suitable TS where the collaborator can submit its responses in an encrypted form. At this stage, it should be noted that the analyst does not know anything except the address of the TS to which collaborator should be required to report his encrypted data. The analyst can assign the same TS to the closely related organizations (e.g SME, hospitals, healthcare organizations within the same countries or cities).

4.4.2. Protocol Setup:

We assume that there exists a group G of prime order p, in which the Decisional Diffie-Hellman (DDH) assumption is intractable. Let g be a random generator of G. For providing the encrypted information for the query, each collaborating entity P_i within the same TS group has to generate the private key sk_i (a random number in $\mathbb{Z}_p$) and the public key $pk_i = g^{sk_i}$. The private key is kept secret to the collaborator itself, while the public key is made publicly available via TS to all other collaborating entities. The collaborator then computes the encryption key that it can use for encrypting the feedback about the query. The encryption key is generated as follows:

$$E_i = \prod_{j \in N, j<i} pk_j \Big/ \prod_{j \in N, j>i} pk_j \tag{1}$$

The computation of the encryption key E_i as above ensures that the following equation holds within the same TS group.

$$\prod_{i \in N} E_i^{sk_i} = 1. \tag{2}$$

4.4.3. Encrypting Responses

Once the collaborating entity has computed the encryption key E_i, it then generates the cryptogram of its response using E_i and the private key sk_i as follows.

$$c_i^{id_i} = E_i^{sk_i} * g^{v_i} \tag{3}$$

Where $v_i \in \{0, 1\}$ is the value of the response to the query, and id_i is the identifier of the collaborator providing the feedback value. The id may be the IP address of the collaborator. The collaborator reports the following information to the TS: a hash of the identity of the collaborator, the query to which the response has been submitted, and the encrypted response. Table 2 presents an example of the format of information posted on the tally server. In Table 2, the collaborator ID is the hash of the identity assigned by the analyst to the participating collaborator, the query is a particular firewall rule (an IP address, a port number, or combination of both), and feedback value is the encrypted response.

collaborator ID	**Src.IP**	**Src.Port**	**Des.Port**	**Encrypted Response**
ABCXYZ1296...	**192.168.1.1**	**200**	**80**	**Malicious**
ABCXYZ12196...	**192.168.1.2**	**10**	**22**	**Legitimate**

Table 2: Example of an entry submitted by Collaborators to the tally server.

4.4.4. Analyst Statistics Aggregation

The analyst or the collaborator can aggregate the encrypted responses for the query by utilizing information available at the TS as follows.

$$\theta_f = \prod_{k=1}^{N} c_k^{id_k} = g^{\sum_{i=1}^{N} v_i} \tag{4}$$

Here, f is the query. The decrypted value of the query T_f is then found out by performing brute-force search over the possible set of values for the T_f with respect to the relation $\theta_f = g^{T_f}$. A brute force search will be feasible since T_f is small. Once the aggregation for the query has been computed, the collaborator then uses this information to decide whether to share the complete information or not. The collaborator shares the complete security threat to all collaborators based on its threshold for the number of collaborators that have positively confirmed the answer to the query. Further, the collaborator can also use the aggregated score to decide the behavior of a particular query i.e., malicious or non-malicious based on the aggregated count.

4.5. Model for Malicious Collaborators

There are a set of $N = \sum_{i=1}^{t} n_i$ collaborators grouped into t classes, namely, $C_1, C_2, \ldots, C_t$, having $n_1, n_2, \ldots, n_t$ collaborators respectively. The jth SP in group i is designated as P_{ij}, where $i \in [1, t]$ and $j \in [1, n_t]$. Each SP P_{ij} holds a secret input $v_{ij} \in \{0, 1\}$. Each class $C_i : i \in [1, t]$ has got its own tally server TS_i. The tally server fetches encrypted scores from the collaborators within the same class and calculates the aggregate score $v_i \in \{0, 1\}$ as below.

$$v_i = \begin{cases} 0 \text{ if } \sum_{j=1}^{n_i} v_{ij} \leq \lfloor n_i/2 \rfloor \\ 1 \text{ else} \end{cases} \tag{5}$$

Then each tally server $TS_i : i \in \{0, 1\}$ encrypts v_i and sends it to the analyst. The analyst further aggregates the responses and computes

$$v = \sum_{i=1}^{t} w_i \cdot v_i$$

where $w_i \in [1, \Delta]$ for $i \in [1, t]$ and Δ is a small integer. w_i's are chosen by the analyst

We devise a protocol that allows the computation of v in a privacy-preserving manner. We provide two schemes, namely scheme I and scheme II. In the scheme I, we assume that

w_i's are publicly known, whereas, in scheme II, they are assumed to be secret inputs by the Analyst.

Here, we assume that the weights assigned to the tally servers by the analyst are all publicly known. Each of the tally servers initiates a protocol to compute the value of v_i, the local value of the aggregated score within the class C_i. The following 2-round protocol describes how the local value of the aggregated score is computed by the tally server.

Step I:. The tally server TS_i selects random $x_i \in_R \mathbb{Z}_p$ and publishes a public key $h_i = g^{x_i}$. Each SP $P_{ij} : j \in [1, n_i]$ within class $C_i : i \in [1, t]$ chooses a random secret key $x_{ij} \in_R \mathbb{Z}_p$ and posts on the tally server board of TS_i, the public key $X_{ij} = g^{x_{ij}}$. P_{ij} also posts on the same board a NIZK proof of knowledge $PW[x_{ij} : X_{ij} = g^{x_{ij}}]$. This NIZK proof proves that P_{ij} knows the value of x_{ij}.

Step II:. Each SP $P_{ij} : j \in [1, n_i]$ within class $C_i : i \in [1, t]$ computes a restructured key $Y_{ij} = g^{y_{ij}} = \prod_{k=1}^{j-1} X_{ik} / \prod_{k=j+1}^{n_i} X_{ik}$. P_{ij} also chooses a random $r_{ij} \in_R \mathbb{Z}_p$ and computes $R_{ij} = h_i^{r_{ij}}$. Then P_{ij} computes the encrypted feedback as $F_{ij} = (B_{ij}, R_{ij})$, where

$$B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}} g^{v_{ij}}.$$

P_{ij} also computes a NIZK proof of well-formedness of the feedback which is given by

$$PW\left[x_{ij}, r_{ij} : R_{ij} = h_i^{r_{ij}}, X_{ij} = g^{x_{ij}}, B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}} g^{v_{ij}}, v_{ij} \in \{0, 1\}\right].$$

This NIZK proof proves that given R_{ij}, X_{ij} and Y_{ij}, B_{ij} is either $g^{r_{ij}} Y_{ij}^{x_{ij}}$ or $g^{r_{ij}} Y_{ij}^{x_{ij}} g$. P_{ij} posts F_{ij} along with the NIZK proof to the tally server of TS_i.

Step III:. Each tally server $TS_i : i \in [1, t]$ looks into its own data on the bulletin board and checks the well-formedness of all feedbacks provided by the collaborators within its purview. If all the feedbacks are well-formed, TS_i computes $\tilde{B}_i = \prod_{j=1}^{n_i} B_{ij} = \prod_{j=1}^{n_i} g^{r_{ij}} Y_{ij}^{x_{ij}} g^{v_{ij}} = g^{\sum_{j=1}^{n_i} r_{ij}} g^{\sum_{j=1}^{n_i} v_{ij}} \prod_{j=1}^{n_i} Y_{ij}^{x_{ij}} = g^{\tilde{r}_i} g^{\tilde{v}_i}$. Here, $\tilde{r}_i = \sum_{j=1}^{n_i} r_{ij}$ and $\tilde{v}_i = \sum_{j=1}^{n_i} v_{ij}$. Now, TS_i computes $S_i = \tilde{B}_i / (\prod_{j=1}^{n_i} R_{ij})^{1/x_i} = g^{\tilde{v}_i}$. A simple brute force search on S_i will yield the value of $\tilde{v}_i$. Brute force search will be feasible since, $\tilde{v}_i = \sum_{j=1}^{n_i} v_{ij} < n_i$. Once, $\tilde{v}_i$ is calculated, v_i can be found using Equation 5.

Step IV:. Each tally server $TS_i : i \in [1, t]$ chooses random $\bar{x}_i \in \mathbb{Z}_p$ and computes a public key $\bar{X}_i = g^{\bar{x}_i}$. TS_i also computes a NIZK proof of knowledge of $\bar{x}_i$ given by $PW[\bar{x}_i : \bar{X}_i = g^{\bar{x}_i}]$. TS_i posts $\bar{X}_i$ and $PW[\bar{x}_i : \bar{X}_i = g^{\bar{x}_i}]$ on the tally server.

Step V:. Each tally server $TS_i : i \in [1, t]$ computes the restructured key $\bar{Y}_i = \prod_{j=1}^{i-1} \bar{X}_j / \prod_{j=i+1}^{t} \bar{X}_j$. Then she computes an encrypted feedback as $\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i \cdot v_i}$. She also computes a NIZK proof of well-formedness of $\bar{B}_i$, given by

$$PW\left[x_i, \bar{x}_i : \bar{X}_i = g^{\bar{x}_i}, h_i = g^{x_i}, \{F_{ij} : j \in [1, n_i]\}\right].$$

TS_i posts the feedback along with the NIZK proof on the tally server of the Analyst.

Once all the t participants have posted their feedbacks on the tally server, the Analyst can compute $\bar{B} = \prod_{i=1}^{t} \bar{B}_i = \prod_{i=1}^{t} \bar{Y}_i^{\bar{x}_i} g^{w_i v_i} = g^{\sum_{i=1}^{t} w_i v_i} = g^v$. A brute force search on $\bar{B}$ will yield the value of v. Brute force search will be feasible since the value of v is small.

5. Security Analysis

In this section, we show that the SIC framework securely computes the statistics and hides the secret inputs of collaborators. The framework ensures the privacy of collaborators under two conditions: a) the number of collaborators on TS should be greater than 2, and b) a maximum of (n-2) collaborators can collude to target the particular collaborator. Lemma 2 proves that the adversary who can corrupt all but two collaborators, cannot distinguish between two tally servers that report the same sum but have two different sets of inputs from at least two honest collaborators. This holds for the condition where the partial sum of the inputs from both honest collaborators is the same. Let us assume that there are two honest collaborators: P_j and P_k. The individual values of v_j and v_k will be known to the adversary only if $v_j + v_k \in \{0, 2\}$, that is, if both the inputs are either 0 or 1. If the inputs are distinct, then the adversary will learn nothing about the individual values of the inputs.

Assumption 1. (DDH Assumption) *Given $g, g^a, g^b \in G$ and a random challenge $\Omega \in \{g^{ab}, R\}$, where $R \overset{\$}{\leftarrow} G$, it is hard to find whether $\Omega = g^{ab}$ or $\Omega = R$.*

Assumption 2. *Given $g, g^a, g^b \in G$ and a challenge $\Omega \in \{g^{ab}, g^{ab}g\}$, it is hard to decide whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g$.*

Lemma 1. *Assumption 1 implies assumption 2.*

Proof 1. $(g, g^a, g^b, g^{ab}) \overset{c}{\approx} (g, g^a, g^b, R) \overset{c}{\approx} (g, g^a, g^b, R * g) \overset{c}{\approx} (g, g^a, g^b, g^{ab}g)$

Lemma 2. *Let $P_i : i \in [n]$ be the set of collaborators in a same group. Also let, $v_i \in \{0, 1\}$ be the input of P_i for each $i \in [n]$. Let, $j, k \in [n]$, $j < k$ and P_j, P_k are honest collaborators. As such, the adversary will not be able to distinguish between these two cases*

1. $v_j = 1, v_k = 0$
2. $v_j = 0, v_k = 1$

Proof 2. *We show that if there exists an adversary $\mathcal{A}$ that can distinguish between the two cases, she could be used to construct another adversary $\mathcal{B}$ that can break the assumption 2. $\mathcal{B}$ works as below:*
Let us assume the inputs to $\mathcal{B}$ are g^a, g^b and a challenge $\Omega \in \{g^{ab}, g^{ab}g\}$. $\mathcal{B}$ has to find whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g$. $\mathcal{B}$ lets $\mathcal{A}$ choose the secret keys Sk_i and the inputs v_i for all $i \in [1, n] \setminus \{j, k\}$. If there is any collaborator $P_i : i \in [1, n] \setminus \{j, k\}$, for which the secret key or the input is not chosen by $\mathcal{A}$, then $\mathcal{B}$ chooses the secret key or the input for that collaborator. $\mathcal{B}$ implicitly sets $Pk_j = g^{Sk_j} = g^a$ and $Pk_k = g^{Sk_k} = g^b$. Since, all public keys are now known, $\mathcal{B}$ can compute the restructured keys for all collaborators $P_i : i \in [1, n] \setminus \{j, k\}$. So, $\mathcal{B}$ can compute the encrypted feedback for all collaborators for whom she has selected the secret keys and the inputs. Similarly, $\mathcal{A}$ can compute the encrypted feedback for all collaborators for whom she has selected the secret keys and the inputs. $\mathcal{B}$

now sets $B_j = g * L_1/\Omega$ and $B_k = \Omega * L_2$. Here, $L_1 = (g^a)^{\sum_{i=1}^{j-1} Sk_i - \sum_{i=j+1}^{k-1} Sk_i - \sum_{i=k+1}^{n} Sk_i}$ and $L_2 = (g^b)^{\sum_{i=1}^{j-1} Sk_i + \sum_{i=j+1}^{k-1} Sk_i - \sum_{i=k+1}^{n} Sk_i}$. Note that, if $\Omega = g^{ab}$, we have $v_j = 1, v_k = 0$, and if $\Omega = g^{ab}g^w$, we will have $v_j = 0, v_k = 1$. If $\mathcal{A}$ can distinguish between these two cases, $\mathcal{B}$ can find whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g$. Hence, the lemma holds.

The result of Lemma 2 can be trivially extended to show that no adversary can learn any information about the secret inputs of uncompromised collaborators, other than their arithmetic sum. Note that, the adversary can learn the arithmetic sum of the inputs of honest collaborators by subtracting the sum of all inputs by the sum of all inputs of colluding collaborators. Hence, the adversary does not learn anything that the output of the protocol does not allow her to know. Thus, the transcript of the protocol does not provide any additional advantage to the adversary. We may conclude that as long as there are at least two honest collaborators that provided different inputs, the privacy of all uncompromised collaborators will remain preserved. Note that, if all the uncompromised collaborators provide identical inputs, the output of the protocol will trivially reveal the inputs of all the collaborators, as in that case the sum of the inputs of all the uncompromised collaborators will be either 0 or equal to the number of uncompromised collaborators.

5.1. Security Analysis for Malicious Model

In this section, we prove that our scheme is secure in the sense that it protects the privacy of the participating collaborators as well as the tally servers. We also show that if a particular tally server is corrupted, the adversary will only learn the sum of all inputs of all collaborators under the particular TS. Also, she will know the input coming from that particular TS.

Assumption 3. *Given g, g^a, and a challenge $\Omega \in \{g^{a^2}, R\}$, it is hard to decide whether $\Omega = g^{a^2}$ or $\Omega = R$.*

Assumption 4. *Given g, g^a, g^b, and a challenge $\Omega \in \{g^{ab}, R\}$, it is hard to decide whether $\Omega = g^{ab}$ or $\Omega = R$.*

Lemma 3. *Assumption 3 implies assumption 4.*

Proof 3. *We show that if there exists an adversary $\mathcal{A}$ against the assumption 4, it could be used to construct another adversary $\mathcal{A}'$ that can break assumption 3. $\mathcal{A}$ works as follows: it receives as input g, g^a and a challenge $\Omega \in \{g^{a^2}, R\}$. It chooses random $\alpha \in_R \mathbb{Z}_p$ and computes $g^b = (g^a)^\alpha$. It then computes $\omega = \Omega^\alpha$. Now if $\Omega = g^{a^2}$, then $\omega = g^{ab}$, else if $\Omega = R$, then $\omega = R' \in_R G$. Now, $\mathcal{A}'$ invokes $\mathcal{A}$ with inputs g, g^a, g^b and the challenge ω. If $\mathcal{A}$ can identify the correct value of ω, $\mathcal{A}'$ will be able to identify Ω correctly.*

Assumption 5. *Given g, g^a, g^b, and a challenge $\Omega \in \{g^{ab}, g^{ab}g\}$, it is hard to decide whether $\Omega = g^{ab}$ or $\Omega = g^{ab}g$.*

Lemma 4. *Assumption 4 implies assumption 5.*

Proof 4. *According to assumption 4,* $(g, g^a, g^b, g^{ab}) \overset{c}{\approx} (g, g^a, g^b, R) \overset{c}{\approx} (g, g^a, g^b, R * g) \overset{c}{\approx} (g, g^a, g^b, g^{ab}g)$.

Lemma 5. *Let,* P_{il} *and* P_{is} *be two honest collaborators such that* $i \in [1, t]$ *and* $1 \leq l < s \leq n_i$*. Also assume that* $v_{il} + v_{is} = 1$*. The adversary who has compromised* TS_i *and all the collaborators in the set* $\{P_{ij} : j \in [1, n_i] \setminus \{l, s\}\}$ *cannot distinguish between the two following cases:*

i) $v_{il} = 0, v_{is} = 1$
ii) $v_{il} = 1, v_{is} = 0$

Proof 5. *We show that if there exists an adversary* $\mathcal{A}$*, who with the help of the tally server and all other collaborators can distinguish between the two above cases, then* $\mathcal{A}$ *can be used to construct another adversary* $\mathcal{A}'$*, who can break the assumption 5.* $\mathcal{A}'$ *receives as input the following items:* g^a, g^b*, and a challenge* $\Omega \in \{g^{ab}, g^{ab}g\}$*. The goal of* $\mathcal{A}'$ *is to find whether* $\Omega = g^{ab}$ *or* $\Omega = g^{ab}g$*.* $\mathcal{A}'$ *works as follows: it chooses a random secret key* $x_i \in_R \mathbb{Z}_p$ *and generates the public key* $h_i = g^{x_i}$*. Then, it creates* $n_i - 2$ *other collaborators* $\Psi_i = \{P_{ij} : j \in [1, n_i] \setminus \{l, s\}\}$*.* $\mathcal{A}'$ *lets* $\mathcal{A}$ *choose all the secret keys* x_{ij} *and the inputs* v_{ij} *for all collaborators with indices in* Ψ_i*.* $\mathcal{A}'$ *implicitly sets* $X_{il} = g^{x_{il}} = g^a$ *and* $X_{is} = g^{x_{is}} = g^b$*. Now,* $\mathcal{A}'$ *can compute the encrypted feedbacks of all collaborators* $P_{ij} : j \in \Psi_i$ *as* $F_{ij} = (B_{ij}, R_{ij})$*, where* $B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}} g^{v_{ij}}$*, and* $R_{ij} = h_i^{r_{ij}}$ *for a randomly selected* $r_{ij} \in \mathbb{Z}_p$*. Now,* $\mathcal{A}'$ *selects random* $r_{il}, r_{is} \in \mathbb{Z}_p$ *and computes* $F_{il} = (B_{il}, R_{il}) = (L_1 * g^{r_{il}} g/\Omega, h_i^{r_{il}})$ *and* $F_{is} = (B_{is}, R_{is}) = (L_2 * g^{r_{is}}\Omega, h_i^{r_{is}})$*, where* $L_1 = \prod_{k=1}^{l-1} X_{ik}/(\prod_{k=l+1}^{s-1} X_{ik} * \prod_{k=s+1}^{n_i} X_{ik})$ *and* $L_2 = \prod_{k=1}^{l-1} X_{ik} * \prod_{k=l+1}^{s-1} X_{ik}/\prod_{k=s+1}^{n_i} X_{ik}$*. Note that, if* $\Omega = g^{ab}$*, then* $B_{il} = g^{r_{il}} Y_{il}^{x_{il}} g$*, and* $B_{is} = g^{r_{is}} Y_{is}^{x_{is}}$*. That is, if* $\Omega = g^{ab}$*, then* $v_{il} = 1$*, and* $v_{is} = 0$*. Alternatively, if* $\Omega = g^{ab}g$*, then* $v_{il} = 0$*, and* $v_{is} = 1$*. If* $\mathcal{A}$ *can distinguish between these two cases,* $\mathcal{A}'$ *can find whether* $\Omega = g^{ab}$ *or* $\Omega = g^{ab}g$*.*

Assumption 6. *Given* $g, S = \{g^{a_i} : i \in [1, n]\}$ *and a challenge* $\Omega \in \{U, V\}$*, it is hard to distinguish whether* $\Omega = U$ *or* $\Omega = V$*, where*

$$U = \{g^{a_i a_j} : 1 \leq i < j \leq n\}$$

$$V = \{R_{ij} : R_{ij} \in_R G, 1 \leq i < j \leq n\}$$

Lemma 6. *Assumption 3 implies assumption 6.*

Proof 6. *We show that if there exists an adversary* $\mathcal{A}$ *against assumption 6, the same can be used in the construction of another adversary* $\mathcal{A}'$ *that can break the assumption 3.* $\mathcal{A}'$ *works as under:*
it receives as input g, g^a*, and a challenge* $\Omega \in \{g^{a^2}, R\}$*, where* $R \in_R G$*.* $\mathcal{A}'$ *selects n integers* $\alpha_1, \alpha_2, \ldots, \alpha_n \in_R \mathbb{Z}_p$*, and computes* $g^{a_i} = (g^a)^{\alpha_i} : i \in [1, n]$*. Now,* $\mathcal{A}'$ *computes* $\omega_{ij} = \Omega^{\alpha_i * \alpha_j}$ *for all* $i, j \in [1, n], i < j$*. Let,* $\Psi = \{\omega_{ij} : 1 \leq i < j \leq n\}$*. Note that, if* $\Omega = g^{a^2}$*, then* $\omega_{ij} = g^{a_i a_j}$*, else if* $\Omega = R \in_R G$*, then* $\omega_{ij} = R' \in_R G$*. Now* $\mathcal{A}'$ *sends* $\{g^{a_i} : i \in [1, n]\}$ *and* Ψ *to* $\mathcal{A}$*. If* $\mathcal{A}$ *can identify the correct value of* Ψ*, then* $\mathcal{A}'$ *can identify* Ω *correctly. Hence, the result.*

Assumption 7. *Given $\beta \in \mathbb{Z}_p, X_i = g^{x_i} : i \in [1, n]$, $Y_i = \prod_{j=1}^{i-1} X_i / \prod_{j=i+1}^{n} X_i$, the following items are indistinguishable.*

$$U = (Y_1^{x_1}, Y_2^{x_2}, \ldots, Y_{n-1}^{x_{n-1}})$$
$$V = (R_1, R_2, \ldots, R_{n-1})$$

where $R_i \in_R G, \forall i \in [1, n-1]$.

Lemma 7. *Assumption 6 implies assumption 7.*

Proof 7. *We show, that if there exists an adversary $\mathcal{A}$ against assumption 7, then it could be used to construct another adversary $\mathcal{A}'$ against the assumption 6. $\mathcal{A}'$ works as below:*
It receives $X = \{g^{x_i} : i \in [1, n]\}$ and a challenge $\Omega = \{\Omega_{ij} : 1 \leq i, j \leq n\}$, such that either of the two following assertions holds:

i) $\Omega_{ij} = g^{x_i x_j}, \forall i, j \in [1, n], i < j$

ii) $\Omega_{ij} \in_R G, \forall i, j \in [1, n], i < j$

$\mathcal{A}'$ implicitly sets $X_i = g^{x_i}, \forall i \in [1, n]$. It then computes $\omega_i = \prod_{j<i} \Omega_{ij} / \prod_{j>i} \Omega_{ij}$. It is easy to see that if $\Omega_{ij} = g^{x_i x_j}, \forall i, j \in [1, n], i < j$, then $\omega_i = Y_i^{x_i}$, else $\omega_i \in_R G$ for all $i \in [1, n-1]$. Let, $\omega = (\omega_1, \omega_2, \ldots, \omega_{n-1})$. Thus, ω is either $(Y_1^{x_1}, Y_2^{x_2}, \ldots, Y_{n-1}^{x_{n-1}})$ or $(R_1, R_2, \ldots, R_{n-1})$. If $\mathcal{A}$ can identify the correct value of ω, then $\mathcal{A}'$ can identify the correct value of Ω. Hence, the result.

Assumption 8. *Given $x_i \in_R \mathbb{Z}_p, X_i = g^{x_i} : i \in [1, n]$, $Y_i = \prod_{j=1}^{i-1} X_i / \prod_{j=i+1}^{n} X_i$, the following items are indistinguishable.*

$$U = (Y_1^{x_1} g^{w_1 s_1}, Y_2^{x_2} g^{w_2 s_2}, \ldots, Y_n^{x_n} g^{w_n s_n})$$

$$V = (Y_1^{x_1} g^{w_1 s'_1}, Y_2^{x_2} g^{w_2 s'_2}, \ldots, Y_n^{x_n} g^{w_n s'_n})$$

where $\sum_{i=1}^{n} w_i s_i = \sum_{i=1}^{n} w_i s'_i$

Lemma 8. *Assumption 7 implies assumption 8.*

Proof 8. $U = (Y_1^{x_1} g^{w_1 s_1}, Y_2^{x_2} g^{w_2 s_2}, \ldots, Y_n^{x_n} g^{w_n s_n}) \overset{c}{\approx} (Y_1^{x_1} g^{w_1 s_1}, Y_2^{x_2} g^{w_2 s_2}, \ldots,$ $Y_{n-1}^{x_{n-1}} g^{w_{n-1} s_{n-1}}, \frac{g^{\sum_{j=1}^{n} w_i s_i}}{\prod_{i=1}^{n-1} Y_i^{x_i} g^{w_i s_i}}) \overset{c}{\approx} (R_1 g^{w_1 s_1}, R_2 g^{w_2 s_2}, \ldots, R_{n-1} g^{w_{n-1} s_{n-1}},$ $\frac{g^{\sum_{j=1}^{n} w_i s_i}}{\prod_{i=1}^{n-1} R_i g^{w_i s_i}}) \overset{c}{\approx} (R_1, R_2, \ldots, R_{n-1}, \frac{g^{\sum_{j=1}^{n} w_i s_i}}{\prod_{i=1}^{n-1} R_i}) \overset{c}{\approx} (R_1 g^{w_1 s'_1}, R_2 g^{w_2 s'_2}, \ldots,$ $R_{n-1} g^{w_{n-1} s'_{n-1}}, \frac{g^{\sum_{j=1}^{n} w_i s'_i}}{\prod_{i=1}^{n-1} R_i g^{w_i s'_i}}) \overset{c}{\approx} (Y_1^{x_1} g^{w_1 s'_1}, Y_2^{x_2} g^{w_2 s'_2}, \ldots, Y_{n-1}^{x_{n-1}} g^{w_{n-1} s'_{n-1}},$ $\frac{g^{\sum_{j=1}^{n} w_i s'_i}}{\prod_{i=1}^{n-1} Y_i^{x_i} g^{w_i s'_i}}) \overset{c}{\approx} (Y_1^{x_1} g^{w_1 s'_1}, Y_2^{x_2} g^{w_2 s'_2}, \ldots, Y_n^{x_n} g^{w_n s'_n}) = V.$

Metric/No.of Responses	**1000**	**10000**	**20000**	**40000**	**50000**
Computation Time (sec)	**6.03**	**63**	**130**	**260**	**304**
Bandwidth Overhead (kbs)	**14**	**1400**	**2800**	**5600**	**7000**
Aggregation Time (6 organization) (sec)	**2.4**	**25**	**49**	**97**	**120**

Table 3: Computational and Bandwidth Overheads.

Lemma 9. *Let us assume $\phi \subseteq [1, t]$. If there exists a group of tally servers $\tau = \{TS_i : i \in \phi\}$, such that there are two distinct sets of inputs $V = \{\bar{v}_i : i \in \tau\}$ and $\tilde{V} = \{\tilde{v}_i : i \in \tau\}$ such that $\sum_{i\in\tau} \bar{v}_i w_i = \sum_{i\in\tau} \tilde{v}_i w_i$, then the adversary will not be able to distinguish between the two following cases:*

i) $v_i = \bar{v}_i; \forall i \in \phi$
ii) $v_i = \tilde{v}_i; \forall i \in \phi$

Proof 9. *Let us first assume that $|\phi| = n$. Also assume that $\Omega = (\Omega_1, \Omega_2, \ldots, \Omega_n)$. We show that if there exists an adversary $\mathcal{A}$, that can make the above distinction, then it can be used to construct another adversary $\mathcal{A}'$ that can break assumption 8. Without loss of generality, we may assume that the honest tally servers are $TS_1, TS_2, \ldots, TS_n$ and the colluding tally servers are $TS_{n+1}, TS_{n+2}, \ldots, TS_t$. The adversary $\mathcal{A}'$ receives as input $g, X_i = g^{x_i}, w_i : i \in [1, n]$ and a challenge $\Omega \in \{U, V\}$, where $\sum_{i=1}^{n} w_i s_i = \sum_{i=1}^{n} w_i s'_i, s_i, s_i \in \{0, 1\}, \forall i \in [1, n]$, and*

$$U = (Y_1^{x_1} g^{w_1 s_1}, Y_2^{x_2} g^{w_2 s_2}, \ldots, Y_n^{x_n} g^{w_n s_n})$$
$$V = (Y_1^{x_1} g^{w_1 s'_1}, Y_2^{x_2} g^{w_2 s'_2}, \ldots, Y_n^{x_n} g^{w_n s'_n})$$

$\mathcal{A}'$ lets $\mathcal{A}$ choose the secret keys $\bar{x}_i$ for all tally servers $TS_i : i \in [n + 1, t]$. Then $\mathcal{A}'$ sets $\bar{X}_i = g^{\bar{x}_i}$ for $i \in [n + 1, t]$. $\mathcal{A}'$ also sets $\bar{X}_i = X_i$, for $i \in [1, n]$. Now, $\mathcal{A}'$ can compute the restructured key $\bar{Y}_i$ for all $i \in [1, t]$. $\mathcal{A}'$ can compute the encrypted feedbacks for all $i \in [n+1, t]$ with the help of $\mathcal{A}$. Then $\mathcal{A}'$ computes the encrypted feedbacks for all the honest tally servers $TS_i : i \in [1, n]$ as follows:

$$\bar{B}_i = \Omega_i / \prod_{j=i+1}^{t} \bar{X}_i^{x_j} : i = 1, 2, \ldots, n$$

Note that, if $\Omega = U$, then $\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i s_i}$, else if $\Omega = U$, then $\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i s'_i}$. Now, $\mathcal{A}'$ can send all these feedbacks to $\mathcal{A}$. If $\mathcal{A}$ can distinguish between the two sets of feedbacks, $\mathcal{A}'$ can identify Ω correctly.

6. Evaluation and Deployment

In this section, we present the prototype implementation of the proposed SIC framework and analyze the system for computational and bandwidth overheads. Finally, we discuss how the system can be deployed in a real scenario.

6.1. Prototype Implementation

We evaluate the performance of the cryptographic operations of the SIC framework in terms of computational cost and bandwidth overheads. We programmed the functionalities of the collaborator, analyst, and TS in Java using the bouncy castle cryptographic library. We implemented the collaborator as a Java client, the TS as a web server, and the analysts as the aggregation system. For encrypting the feedback values we used the standard NIST Curve P-256 [48] for 128-bit security. We obtained the measurements over a single core on the Intel i-7 system (3.47GHz), with 8GB of RAM, and Windows 8 operating system. The system can be easily parallelized over multiple cores. In the proof-of-concept implementation, we focus on the honest-but-curious model in which participants provide encrypted responses within a prescribed range. This is realistic in a healthcare application where healthcare organizations are reputed entities and they want to collaboratively find out if a traffic pattern is malicious or not while preserving the privacy of their input. In Section 4, we presented solutions for the malicious adversarial model by using the non-interactive zero-knowledge proof (ZKP) to enforce that every participant follows the protocol honestly. The most expensive operations in ZKP are those involved to verify the proofs instead of to compute the proofs [36]. However, these verification operations will be performed by the analyst. Hence, the effect on each participant is limited. One possible way to implement the analyst in a malicious adversary model is to employ a blockchain, leveraging the underlying consensus mechanism in a blockchain to verify the ZKPs and compute the tally in a publicly verifiable manner. We leave this for future work.

6.2. Microbenchmarks

We present experimental results for three phases: 1) generating the public, private, and encryption keys, 2) generating the encrypted responses, and 3) performing the aggregation. The key generation process is not computationally expensive. Each collaborating collaborator generates public and private keys in around 100 msec and posts the public key to TS. The collaborating collaborator computes the encryption key from the public keys provided by 100 collaborators in less than 10 msec. The number of collaborators at the TS may vary, however, the size should be small for efficient computation. The expensive operation on the collaborator side is the creation of the cryptogram of the feedback values. Table 3 presents the computation time required for creating the encrypted response when the number of responses varies from 1000 to 50000. The results reveal that the computation time is not very high even for a high number of responses i.e. the collaborator can encrypt 50,000 responses in 304 sec on a single core. However, with the parallelization, this time would decrease to around 40 seconds (using 8 cores in parallel).

Table 3 presents the bandwidth overheads required for submitting feedback to the TS. A single encrypted response requires 140 bytes and the bandwidth usage increase linearly with the number of responses. The computation cost for aggregating the responses from 6 collaborators in a single TS is provided in table 3. The system takes around 50 seconds (using 8 cores) to process 1 million responses from the TS, and the cost increases linearly with the number of responses.

6.3. Practical Deployment

We now discuss the practical deployment of the proposed system i.e. the role of the analyst, how to set the size of the tally server, and how to set the duration of submitting the feedback to the tally server. The analyst can be any entity with the condition that it is able to performs computation. It could be either a private or government entity providing a platform for the collaboration. The analyst does not need to be trusted since its computational result is publicly verifiable. For the number of collaborators in an intermediate tally server, we used the statistics based on the study presented in [10]. In [10], Katti et al. perform the study on data collected from the 1,700 standalone intrusion detection systems (IDSes) and find out that collaboration with only a few IDSes would achieve the same detection accuracy as it achieves with the collaboration among all the IDSes. We recommend that the number of collaborators in the tally server should be at least 3 (as in the case of two the collaborators' privacy cannot been protected at all), and should not exceed 10 as a higher number of collaborators would increase the load on the analyst. The participating collaborators should neither present feedback frequently as it would increase the overheads, but it would detect the malicious actor in real-time. On the other hand, less frequent updates would minimize the overhead but would not detect the malicious actor in real-time. There should be a trade-off between the frequency of updates and the overheads. In [10], Kati et al. identify that more than 75% of the time, a common attacker sends malicious traffic to other IDSes within 10 minutes of its first attack. Based on these statistics, we suggest that collaborators should exchange information regularly with an interval not longer than 10 minutes.

7. Conclusions

Standalone intrusion detection systems have been widely deployed by healthcare organizations at the edge of their network in order to protect their resources and users from cybersecurity threats. A standalone detection system considers the traffic patterns from a single source to decide whether certain traffic is malicious or legitimate. A stealthy and sophisticated attacker can bypass a standalone IDS by making a coordinated attack against several organizations at the same time, which could bring catastrophic damage to the healthcare organizations in terms of financial loss and the public reputation. Collaboration among standalone systems with the exchange of information related to cyber incidents can enable healthcare organizations to collectively use the knowledge from the community for the early identification of malicious actors. In this paper, we proposed a privacy-preserving collaborative system, which enables a healthcare organization to have aggregate statistics without revealing their sensitive information to other collaborators. The protocol ensures the privacy and correctness of computation under honest-but-curious and malicious models. We implemented our protocol and conducted extensive experiments to evaluate the performance of the system in terms of computation time and bandwidth consumption. The lightweight cryptographic operations, inherent properties of decentralization, and privacy preservation along with reasonable overhead make SIC a suitable choice for privacy-preserving collaborations and alert sharing.

Acknowledgement

Feng Hao and Samiran Bag would like to acknowledge the support by the Royal Society grant, ICA/R1/180226.

References

[1] "Security in smart healthcare must make a fast recovery," https://www.kaspersky.com/blog/secure-futures-magazine/smart-healthcare-iot/35642/, 2020.

[2] "Ibm study shows data breach costs on the rise; financial impact felt for years," http://tiny.cc/m3a3rz, 2018.

[3] "Cyber attacks hit more than half of healthcare orgs in last year," http://bit.ly/33gRH9u, 2018.

[4] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran, Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, D. Kumar, C. Lever, Z. Ma, J. Mason, D. Menscher, C. Seaman, N. Sullivan, K. Thomas, and Y. Zhou, "Understanding the mirai botnet," in *26th USENIX Security Symposium (USENIX Security 17)*. Vancouver, BC: USENIX Association, Aug. 2017, pp. 1093–1110. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis

[5] "What is wannacry/wanacrypt0r?" http://tiny.cc/e782rz, 2018.

[6] "Wannacry cyber attack cost the nhs 92m," http://bit.ly/2QdzKDd, 2018.

[7] S. Deep, X. Zheng, A. Jolfaei, D. Yu, P. Ostovari, and A. Kashif Bashir, "A survey of security and privacy issues in the internet of things from the layered context," *Transactions on Emerging Telecommunications Technologies*, vol. n/a, no. n/a, p. e3935. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/ett.3935

[8] A. K. Bashir, S. H. Chauhdary, S. C. Shah, and M. Park, "Mobile rfid and its design security issues," *IEEE Potentials*, vol. 30, no. 4, pp. 34–38, July 2011.

[9] Y. Mehmood, F. Ahmad, I. Yaqoob, A. Adnane, M. Imran, and S. Guizani, "Internet-of-Things Based Smart Cities: Recent Advances and Challenges," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 16–24, 2017, doi: 10.1109/MCOM.2017.1600514.

[10] S. Katti, B. Krishnamurthy, and D. Katabi, "Collaborating against common enemies," in *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, ser. IMC '05. Berkeley, CA, USA: USENIX Association, 2005, pp. 34–34. [Online]. Available: http://dl.acm.org/citation.cfm?id=1251086.1251120

[11] A. K. Bashir, A. H. Akbar, S. A. Chaudhary, C. S. Hussain, and Ki-Hyung Kim, "Collaborative detection and agreement protocol for routing malfunctioning in wireless sensor networks," in *2006 8th International Conference Advanced Communication Technology*, vol. 1, Feb 2006, pp. 327–332.

[12] "United states computer emergency readiness team," https://www.us-cert.gov/, 2018.

[13] "United kingdom community emergency response," http://www.ukcert.org.uk/, 2018.

[14] "European union agency for network and information security," https://www.enisa.europa.eu/, 2018.

[15] (2018) Internet storm center. https://www.dshield.org/.

[16] C. Zhou, C.Leckie, and S.Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers & Security*, vol. 29, no. 1, pp. 124 – 140, 2010.

[17] G. Meng, Y. Liu, J. Zhang, A. Pokluda, and R. Boutaba, "Collaborative security: A survey and taxonomy," *ACM Computing Survey*, vol. 48, no. 1, pp. 1:1–1:42, Jul. 2015.

[18] E. Vasilomanolakis, S. Karuppayah, M. Mühlhäuser, and M. Fischer, "Taxonomy and survey of collaborative intrusion detection," *ACM Computing Survey*, vol. 47, no. 4, pp. 55:1–55:33, May 2015.

[19] E. Toch, C. Bettini, E. Shmueli, L. Radaelli, A. Lanzi, D. Riboni, and B. Lepri, "The privacy implications of cyber security systems: A technological survey," *ACM Comput. Surv.*, vol. 51, no. 2, pp. 36:1–36:27, Feb. 2018. [Online]. Available: http://doi.acm.org/10.1145/3172869

[20] R. Bye, S. Camtepe, and S. Albayrak, "Collaborative intrusion detection framework: Characteristics, adversarial opportunities and countermeasures," in *Proceedings of the 2010 International Conference*

on Collaborative Methods for Security and Privacy, ser. CollSec'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–1.

[21] P. Porras and V. Shmatikov, "Large-scale collection and sanitization of network security data: Risks and challenges," in *Proceedings of the 2006 Workshop on New Security Paradigms*, ser. NSPW '06. New York, NY, USA: ACM, 2007, pp. 57–64.

[22] P. Lincoln and P. Porras, "Privacy-preserving sharing and correlation of security alerts," in *In USENIX Security Symposium*, 2004, pp. 239–254.

[23] M. Burkhart, M. Strasser, D. Many, and X. Dimitropoulos, "Sepia: Privacy-preserving aggregation of multi-domain network events and statistics," in *Proceedings of the 19th USENIX Conference on Security*, ser. USENIX Security'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 15–15. [Online]. Available: http://dl.acm.org/citation.cfm?id=1929820.1929840

[24] J. Freudiger, E. Cristofaro, and A. Brito, "Controlled data sharing for collaborative predictive blacklisting," in *Proceedings of the 12th International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment - Volume 9148*, ser. DIMVA 2015. New York, NY, USA: Springer-Verlag New York, Inc., 2015, pp. 327–349.

[25] S.E.Coull, C.V.Wright, F. Monrose, M.P.Collins, and M.K.Reiter, "Playing devil's advocate: Inferring sensitive information from anonymized network traces," in *Proceedings of the 4th Conference on Networked Systems Design & Implementation*, 2007.

[26] M. E. Locasto, J. J. Parekh, A. D. Keromytis, and S. J. Stolfo, "Towards collaborative security and p2p intrusion detection," in *In Proceedings of the IEEE Information Assurance Workshop (IAW*, 2005, pp. 333–339.

[27] J. Zhang, P. Porras, and J. Ullrich, "Highly predictive blacklisting," in *Proceedings of the 17th Conference on Security Symposium*, ser. SS'08, 2008, pp. 107–122.

[28] J. M. de. Fuentes, L. G.M, J.Tapiador, and P.Peris-Lopez, "Pracis: Privacy-preserving and aggregatable cybersecurity information sharing," *Computers & Security*, vol. 69, pp. 127 – 141, 2017, security Data Science and Cyber Threat Management.

[29] A. Narayanan and V. Shmatikov, "De-anonymizing social networks," in *Proceedings of the 2009 30th IEEE Symposium on Security and Privacy*, ser. SP '09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 173–187. [Online]. Available: http://dx.doi.org/10.1109/SP.2009.22

[30] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *2008 IEEE Symposium on Security and Privacy (sp 2008)*, 2008, pp. 111–125.

[31] R. Janakiraman, M. Waldvogel, and Q. Zhang, "Indra: a peer-to-peer approach to network intrusion detection and prevention," in *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, June 2003, pp. 226–231.

[32] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the domino overlay system," in *In Proceedings of Network and Distributed System Security Symposium (NDSS)*, 2004.

[33] E. Vasilomanolakis, M. Krügl, C. G. Cordero, M. Möuhlhöauser, and M. Fischer, "Skipmon: A locality-aware collaborative intrusion detection system," in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, 2015, pp. 1–8.

[34] M.Sirivianos, K.Kim, and X.Yang, "Socialfilter: Introducing social trust to collaborative spam mitigation," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2300–2308.

[35] M. Azad and R.Morla, "Rapid detection of spammers through collaborative information sharing across multiple service providers," *Future Generation Computer Systems*, pp. –, 2018.

[36] MA.Azad, S.Bag, S.Tabassum, and F.Hao, "privy: Privacy preserving collaboration across multiple service providers to combat telecoms spam," *IEEE Transactions on Emerging Topics in Computing*, 2017.

[37] J. J. Parekh, K. Wang, and S. J. Stolfo, "Privacy-preserving payload-based correlation for accurate malicious traffic detection," in *Proceedings of the 2006 SIGCOMM Workshop on Large-scale Attack Defense*, ser. LSAD '06. New York, NY, USA: ACM, 2006, pp. 99–106. [Online]. Available: http://doi.acm.org/10.1145/1162666.1162667

[38] F. Chen, B. Bruhadeshwar, and A. X. Liu, "Cross-domain privacy-preserving cooperative firewall op-

timization," *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 857–868, June 2013.

[39] B. Applebaum, H. Ringberg, M. J. Freedman, M. Caesar, and J. Rexford, "Collaborative, privacy-preserving data aggregation at scale," in *Privacy Enhancing Technologies*, M. J. Atallah and N. J. Hopper, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 56–74.

[40] U. Erlingsson, V. Pihur, and A. Korolova, "Rappor: Randomized aggregatable privacy-preserving ordinal response," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '14. New York, NY, USA: ACM, 2014, pp. 1054–1067. [Online]. Available: http://doi.acm.org/10.1145/2660267.2660348

[41] Y. Lee, W. Hsiao, Y. Lin, and S. T. Chou, "Privacy-preserving data analytics in cloud-based smart home with community hierarchy," *IEEE Transactions on Consumer Electronics*, vol. 63, no. 2, pp. 200–207, May 2017.

[42] "Introduction to STIX," https://oasis-open.github.io/cti-documentation/taxii/intro, 2018.

[43] "Introduction to TAXII," https://oasis-open.github.io/cti-documentation/taxii/intro, 2018.

[44] P. Ohm, "Broken promises of privacy: Responding to the surprising failure of anonymization," *UCLA Law Review, Vol. 57, p. 1701, 2010*, 2009.

[45] F. Kerschbaum, "A verifiable, centralized, coercion-free reputation system," in *Proceedings of the 8th ACM Workshop on Privacy in the Electronic Society*, 2009, pp. 61–70.

[46] L. Melis, G. Danezis, and E. Cristofaro, "Efficient private statistics with succinct sketches," in *Proceedings of 23rd Network and Distributed System Security Symposium (NDSS)*, 2015.

[47] J. Canny, "Collaborative filtering with privacy," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, 2002, pp. 45–57.

[48] U. D. o. C. I. o. S. Digital Signature Standard (DSS) and Technology., 2017. [Online]. Available: FIPS-186-4, 2013. http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf.

Appendix

NIZK proof systems used in this paper

$PW[x_{ij} : X_{ij} = g^{x_{ij}}]$**:** This NIZK proof proves that the prover knows the value of x_{ij}, given g and $X_{ij} = g^{x_{ij}}$. This proof is generated by P_{ij} for all $i \in [1, t]$ and $j \in [1, n_t]$. The prover can construct this NIZK proof as follows:
The prover chooses a random $t \in_R \mathbb{Z}_p$, and computes a commitment $com = g^t$. Let the challenge be ch. The prover computes a response $res = t - x_{ij}ch$. The verification equation is as follows:

$$g^{res} \overset{?}{=} \frac{com}{X_{ij}^{ch}}$$

The prover needs to do one exponentiation for generating the proof, whereas the verifier needs to do two exponentiations. Also, the communication cost of the proof is 3.

$PW\left[x_{ij}, r_{ij} : R_{ij} = h_i^{r_{ij}}, X_{ij} = g^{x_{ij}}, B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}} g^{v_{ij}}, v_{ij} \in \{0, 1\}\right]$: This NIZK proof proves the well-formedness of the encrypted feedback B_{ij} submitted by P_{ij}, given h_i, $R_{ij} = h_i^{r_{ij}}, X_{ij} = g^{x_{ij}}$ and for $v_{ij} \in \{0, 1\}$. That is, this NIZK proof shows that B_{ij} is either $g^{r_{ij}} Y_{ij}^{x_{ij}} g$ or $g^{r_{ij}} Y_{ij}^{x_{ij}}$. The prover (P_{ij}) can construct NIZK arguments for this statement as follows:
Let us assume that $B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}} g$. The prover chooses random $t_1, t_2 \in_R \mathbb{Z}_p$ and computes $com_1 = h_i^{t_1}, com_2 = g^{t_2}, com_3 = g^{t_1} Y_{ij}^{t_2}$. The prover also chooses random $res'_1, res'_2, res'_3, ch_2 \in_R \mathbb{Z}_p$ and computes three commitments:

$com'_1 = h_i^{res'_1} R_{ij}^{ch_2}, com'_2 = g^{res'_2} X_{ij}^{ch_2}, com'_3 = g^{res'_1} Y_{ij}^{res'_2} B_{ij}^{ch_2}$. Let, the grand challenge of the NIZK proof be ch. The prover computes $ch_1 = ch - ch_2$. The prover calculates two responses as follows: $res_1 = t_1 - r_{ij} * ch_1, res_2 = t_2 - x_{ij} * ch_1$. The verification equations are as follows:

1. $h_i^{res_1} \stackrel{?}{=} \frac{com_1}{R_{ij}^{ch_1}}$
2. $g^{res_2} \stackrel{?}{=} \frac{com_2}{X_{ij}^{ch_1}}$
3. $g^{res_1} Y_{ij}^{res_2} \stackrel{?}{=} \frac{com_3}{(B_{ij}/g)^{ch_1}}$
4. $h_i^{res'_1} \stackrel{?}{=} \frac{com'_1}{R_{ij}^{ch_2}}$
5. $g^{res'_2} \stackrel{?}{=} \frac{com'_2}{X_{ij}^{ch_2}}$
6. $g^{res'_1} Y_{ij}^{res'_2} \stackrel{?}{=} \frac{com'_3}{B_{ij}^{ch_2}}$

If the 6 above equations are satisfied, the NIZK proof is correct. The prover needs to perform 11 exponentiations for generating the proof, whereas the verifier needs to perform 15 exponentiations for verifying the same. The NIZK proof consists of 6 commitments, 2 challenges and 4 responses, hence, the communication cost of this NIZK proof is 12. Similarly, the prover can generate a NIZK proof when $B_{ij} = g^{r_{ij}} Y_{ij}^{x_{ij}}$, i.e. when $v_{ij} = 0$.

$PW[\bar{x}_i : \bar{X}_i = g^{\bar{x}_i}]$**:** The construction of this NIZK proof is same as the proof $PW[x_{ij} : X_{ij} = g^{x_{ij}}]$, described above.

$PW\left[x_i, \bar{x}_i : \bar{X}_i = g^{\bar{x}_i}, h_i = g^{x_i}, \{X_{ij}, F_{ij} : j \in [1, n_i]\}\right]$**:** This NIZK proof proves that the encrypted feedback submitted by the Tally Server TS_i is well-formed in the sense that it represents the correct cryptogram which reflects the correct feedback computed from the SPs within the purview of TS_i. The Tally Server TS_i needs to show that the encrypted feedback $\bar{B}_i$ correctly reflects the feedbacks submitted to TS_i by the SPs. The statement the Tally Server needs to show is given below:

$\sigma \equiv (\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i v_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i} g^{\tilde{v}_i}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})$ Here, $\tilde{R}_i = \prod_{j=1}^{n_i} R_{ij} = \prod_{j=1}^{n_i} h_i^{r_{ij}} = h_i^{\sum_{j=1}^{n_i} r_{ij}} = h_i^{\tilde{r}_i} = (g^{x_i})^{r_i}$.

So, the above statement can be rewritten as $\sigma \equiv (\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i v_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i} g^{\tilde{v}_i}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})$ or

$$\sigma \equiv \Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i} g^{\lceil \frac{n}{2} \rceil}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i}) \Big) \vee$$
$$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i} g^{\lceil \frac{n}{2} \rceil + 1}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i}) \Big) \vee$$

. . . .

. . . .

. . . .

$$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i} g^{w_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i} g^{n}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i}) \Big) \vee$$

$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})\Big) \vee$
$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i}g) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})\Big) \vee$
$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$
$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$
$\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots$
$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i}g^{\lceil \frac{n}{2} \rceil - 1}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})\Big)$

This is a 1-out-of-n OR statement. Exactly one of the constituent sub-statements is true. The prover needs to provide a real proof for the correct sub-statement. This can be done as follows:
Let us assume the first sub-statement is true, i.e. the following statement is true:

$$\Big((\bar{B}_i = \bar{Y}_i^{\bar{x}_i}g^{w_i}) \wedge (\bar{X}_i = g^{\bar{x}_i}) \wedge (\tilde{B}_i = g^{\tilde{r}_i}g^{\lceil \frac{n}{2} \rceil}) \wedge (h_i = g^{x_i}) \wedge (\tilde{R}_i = h_i^{\tilde{r}_i})\Big)$$

The prover selects random $r_{11}, r_{12} \in_R \mathbb{Z}_p$, and compute commitments $com_{11} = g^{r_{11}}, com'_{11} = Y_i^{r_{11}}, com_{12} = g^{r_{12}}, com'_{12} = h_i^{r_{12}}$

The prover also selects $res_{j1}, res_{j2}, ch_j \in_R \mathbb{Z}_p$, for $j \in [2, n]$ and computes $com_{j1} = g^{res_{j1}}\bar{X}_i^{ch_j}, com'_{j1} = \bar{Y}_i^{res_{j1}}(\bar{B}_i/g^{w_i I_j})^{ch_j}, com_{j2} = g^{res_{j2}}h_i^{ch_j}, com'_{j2} = (\tilde{B}_i/g^{\overrightarrow{v}_j})^{res_{j2}}\tilde{R}_i^{ch_j}$. Here $I_j = 1$ for $j \in [1, n - \lceil \frac{n}{2} \rceil + 1]$ and $I_j = 0$ for $j \geq n - \lceil \frac{n}{2} \rceil + 1$. Also, $\overrightarrow{v}_j = \lceil \frac{n}{2} \rceil + j - 1 \mod (n+1)$ for all $j \in [1, n+1]$.

Let the grand challenge of the NIZK proof be ch. The prover computes $ch_1 = ch - \sum_{j=2}^{n+1} ch_j$. The prover generates two responses as follows: $res_{11} = r_{11} - \bar{x}_i * ch_1, res_{12} = r_{12} - x_i * ch_1$.
$\forall j \in [1, n+1]$, the verification equations are as below:

1. $g^{res_{j1}} \stackrel{?}{=} \frac{com_{j1}}{\bar{X}_i^{ch_j}}$
2. $\bar{Y}_i^{res_{j1}} \stackrel{?}{=} \frac{com'_{j1}}{(\bar{B}_i/g^{w_i I_j})^{ch_j}}$
3. $g^{res_{j2}} \stackrel{?}{=} \frac{com_{j2}}{h_i^{ch_j}}$
4. $(\tilde{B}_i/g^{\overrightarrow{v}_j})^{res_{j2}} \stackrel{?}{=} \frac{com'_{j2}}{\tilde{R}_i^{ch_j}}$

The prover needs to perform $8n+4$ exponentiations for generating the entire NIZK proof. The verifier needs to do $8n+8$ exponentiations for verifying all the NIZK arguments. This NIZK proof consists of $4n+4$ commitments, $n+1$ challenges, and $2n+2$ responses. Thus, the communication overhead of the NIZK proof is $7n+7$, which is $O(n)$.