True Demand-Driven Semiconductor Supply Chains for Europe

Project Acronym:

SC4EU

Grant agreement no: 101139949

Deliverable no. and title	**D6.1 – Technical specifications of the Multi-Party Computation Survey Framework**	
Work package	WP6	Anonymous Survey and Multi-Party Computation
Task	T6.1	Technical implementation of the Multi-Party Computation survey framework
Subtasks involved		
Lead contractor	Infineon Technologies AG Thomas Gutt, mailto:thomas.gutt@infineon.com	
Deliverable responsible	TACEO GmbH Roman Markus Holler, holler@taceo.io	
Version number	V1.0	
Date	21/12/2024	
Status	Final	
Dissemination level	PU	

Authors

Partici-pant no.	Part. short name	Author name	Chapter(s)
1	TACEO	Roman Holler	
2	TACEO	Roman Walch	2

Document History

Version	Date	Author name	Reason
v0.1	06.04.2024	Alfred Hoess	Initial Template
V0.2	14.11.2024	Roman Holler	Initial document structure & content
V0.3	18.11.2024	Roman Walch	Section 2
V0.4	25.11.2024	Roman Holler	Incorporate feedback & content
V1.0	21.12.2024	Roman Holler	Final version & final editing. Deliverable submission.

Publishable Executive Summary

The SC4EU project aims to dampen the bullwhip effect by identifying problematic market signals early via collecting information from players across the supply chain and using market data. Suppliers in the semiconductors industry benefit from receiving early signals so they can adapt their strategy to effectively reduce the negative consequences of the bullwhip effect.

To be able to provide information that has an impact, the SC4EU project needs to collect the latest information possible from suppliers in the semiconductors industry. The data required from these players contains sensitive information that potentially harms the competitiveness of the respective player if that data were disclosed.

To bridge the gap between these contradictive requirements, Multi-Party Computation (MPC) is used to collect sensitive answers, to transform them into aggregated results, and to provide these results to a specific set of parties who can then use the results to react early to changes in the market. The aggregated results are created in a way that the competitiveness of parties providing their sensitive data is not harmed or threatened in any way, as the sensitive data is kept private to them. Instead, participants strengthen their competitiveness by providing their sensitive data as they retrieve information in return, giving them an advantage over other players who do not have access to that data.

This deliverable report goes into detail on the work that has been carried out as part of WP6 and its task T6.1 to implement Framework to be used within the SC4EU project. Additionally, this report provides insights into the progress of Tasks T6.2 and T6.3 that are concerned with the deployment of the MPC Survey Framework and the requirements from WP5 towards the MPC Survey Framework. An outlook highlights the topics that need to be tackled in the future and what inputs are required from other WPs to successfully complete the tasks of WP6.

A basic version of the MPC Survey Framework was implemented within the first four months of the project, followed by four end-to-end test runs of the MPC Survey Framework that also includes a deployment of the MPC Survey Framework at TACEO, TIB and the affiliated partner OPAIX. The deployment and the survey test runs were successful, meaning that survey participants provided their answers to the system, the system computed aggregated results, and that results were distributed back to the survey participants. The individual answers remain private due to the utilization of MPC.

The initial test runs have proven that the MPC Survey Framework runs smoothly and is ready to be used. The remaining effort of WP6 concerns the adaption of the MPC Survey Framework according to the requirements of WP5, where the requirements will be defined as part of the report D5.1 which will be released in six months from now. Furthermore, the MPC Survey Framework in its current state does not implement user authentication & identification. WP2 will provide a project-wide solution to authenticate and identify users and WP6 will integrate the MPC Survey Framework with the WP2 user authentication & identification solution.

It is important to note that this report is delivered in M12 of the project, whereas the task T6.1 concerning the implementation of the MPC Survey Framework is due M18 of the project. Therefore, all details mentioned in this report are subject to change in regard to the implementation of the MPC Survey Framework and this report reflects the current state (as of M12) of the system.

Table of contents

1. Introduction 6
1.1 Objective and scope of the document 6
1.2 Structure of the deliverable report 6
2. Multi-Party Computation (MPC) Protocol Description 7
3. Architecture Summary 8
3.1 Actors/Data Flow Overview 8
3.2 Technical Details 10
4. Frontend Client Interfaces 11
4.1 Admin Interface 11
4.2 Survey Participant Interface 13
4.3 Analyst Interface 14
5. Survey Definition Interfaces 15
5.1 Computation Security Considerations 17
6. User Authentication and Identification 19
7. Information Flow 20
8. Current Implementation Status 21
8.1 Management Server 21
8.2 MPC Nodes 21
8.3 Deployment 21
8.4 Test Runs 21
9. Future Work/Outlook 23
10. Conclusions 24
11. References 25
12. Appendices 26
12.1 Appendix A - Abbreviations 26

List of figures

Figure 1: System Architecture Overview.....9
Figure 2: Admin Interface: Survey Definitions screen. Administrators are presented with a list of survey definitions, where they can upload new survey definitions, delete existing survey definitions, or create a survey instance from a survey definition..... 12
Figure 3: Admin Interface: Survey Runs screen. A list of survey instances indicates the current status of each survey and its creation date and time. 12
Figure 4: Survey Participant Interface: Survey input and submission screen. Users can provide their answers and use the Submit button to split their answers into secret shares, encrypt them locally on their device, and send the secret shared answers to the Management Server..... 13
Figure 5: Analyst Interface: survey results screen. Analysts can view the results of a survey instance of state “Finished” and can also download the results as a file using the button on the bottom right. 14
Figure 6: Survey Participant Interface: Screen of the uploaded file "Example-Survey-Definition.json" . 17
Figure 7: Information Flow and involved Actors 20

List of tables

Table 1: Actors and involved parties within Chapter 3. 8
Table 2: Expected hardware requirements..... 10
Table 3: Terms used throughout Chapter 4. 11
Table 4: Temporary JSON object structure for an input, i.e. a survey question including answers. This structure (including the use of JSON) is temporary and subject to change once the requirements from WP5 towards WP6 are known (see D5.1)..... 15
Table 5: Abbreviations 26

List of listings

Listing 1: Content of the file “Example-Survey-Definition.json”..... 16

1. Introduction

Implementing a secure Multi-Party Computation (MPC) solution is crucial for the success of the SC4EU project to ensure that no sensitive information in terms of participants' answers is leaked, as well as that no information is leaked through the aggregation of results. This document highlights the implementation and design of a secure MPC Survey Framework.

1.1 Objective and scope of the document

This document serves as the technical specification of the Multi-Party Computation (MPC) Survey Framework which is currently under development as part of Task T6.1 [M01-M24]. At the time of writing this document (M12), the specification and implementation of basic functionality is finished and described in this document. The remaining part of T6.1 [M13-M24] concerns adapting the base functionality and specifications according to the requirements defined in Deliverable D5.1, the result of Task T5.1 (WP5) which is due M18.

The scope of this document includes a theoretical and technical description of the underlying MPC technology used, the system architecture design of the MPC Survey Framework, data flow, security & privacy considerations, user interaction (survey participants, administrators, data analysts) with the system, and future work within T6.1.

1.2 Structure of the deliverable report

This document is structured as follows. Chapter 2 provides a theoretical background on MPC and elaborates on how computations are performed within MPC, how these computations ensure confidentiality on the data being processed, and the security assumptions under which MPC provides privacy for the processed data.

Chapter 3 explains the system architecture that allows to securely use MPC within the MPC Survey Framework. Chapter 4 highlights the frontend client interfaces that allows survey participants, administrators, and data analysts to interact with the MPC Survey Framework, such as collecting inputs from participants and computing the results from the perspective of administrators and data analysts. Chapter 5 shows how to define surveys and upload them to the MPC Survey Framework. Chapter 6 expands on user authentication and identification. Chapter 7 demonstrates the data and usage flow on an end-to-end example considering all involved entities and related systems of the SC4EU project.

Chapter 8 reports on the current implementation status and Chapter 9 provides an outlook on future work within Task T6.1, as well as Tasks T6.2 and T6.3. Chapter 10 concludes this report.

2. Multi-Party Computation (MPC) Protocol Description

Secure multi-party computation (MPC) is a modern cryptographic technique that allows mutually untrusting parties to jointly compute functions on combined private inputs without leaking these inputs or intermediate results to each other. MPC was first introduced in 1982 by Yao [4] to solve the now famous *Millionaires Problem* where two millionaires want to find out who is richer without telling each other their net worth. While many different flavors of MPC protocols have been introduced since then, in this project we focus on *secret-sharing* based MPC protocols [1]. More concretely, we use ones based on so-called 3-party *replicated secret-sharing (REP3)* [3] [2].

In these protocols 3 parties, denoted as MPC Nodes, are responsible for executing the MPC computation privately. Whenever some data provider wants to input some private data x (for simplicity we let x be some integers modulo 2^k in this description) to the computation, it first has to split it into 3 additive shares: It first samples two random integers x_1 and x_2, and sets the third share to be $x_3 = x - x_1 - x_2$. Consequently, all 3 shares are required to reconstruct the original data x, and no pair of two shares leak anything about the original data. Then, each of the 3 MPC nodes gets two of these shares:

Node 1: (x_1, x_3),
Node 2: (x_2, x_1),
Node 3: (x_3, x_2).

Consequently, no MPC Node learns anything about the data x, but two out of three Nodes are required to reconstruct the data.

The MPC Nodes can use their shares of the input data to compute functions, which in turn produce shares of the output. Thereby, linear functions, such as additions, can be computed locally without party interactions: Let x_i and y_i be additive shares of x and y, then $z_i = x_i + y_i$ is a valid additive share of $z = x + y$. Unfortunately, non-linear functions (e.g., multiplications), require party interaction to be computed privately on shares. First, observe that MPC Nodes can use their replicated shares to produce arithmetic shares of the result,

Node 1: $z_1 = x_1 \cdot y_3 + x_3 \cdot y_1 + x_1 \cdot y_1 + r_1$,
Node 2: $z_2 = x_2 \cdot y_1 + x_1 \cdot y_2 + x_2 \cdot y_2 + r_2$,
Node 3: $z_3 = x_3 \cdot y_2 + x_2 \cdot y_3 + x_3 \cdot y_3 + r_3$,

where z_i is a valid additive share of $z = x \cdot y$, and r_i is a fresh additive share of 0 (which can be produced without interaction) which gets added for re-randomization. Thus, after each multiplication, the parties need to send their share to one other party to restore replication of the shares.

Furthermore, the ABY3 publication [2] describes efficient protocols to translate replicated shares of integers to shares of bits (and vice versa) which in turn allow to efficiently implement comparisons in MPC. Thus, using replicated sharing, one can compute additions, multiplications, comparison and simple branching programs (where branching follows a conditional multiplexer structure) privately in MPC without leaking inputs.

Security

The resulting MPC protocol is secure in the *semi-honest* security setting (i.e., the parties do not deviate from the protocol description) with *honest-majority* (i.e., no two parties collude to combine their shares). For a more comprehensive introduction to MPC see, e.g., [5].

3. Architecture Summary

The theoretical concepts of MPC explained in Chapter 2 will be used in this Chapter to craft an MPC Survey Framework that respects all (security) needs of the SC4EU project. The goal is a minimal system that allows

- **Admins** to define custom surveys, instantiate surveys, and to trigger survey result evaluation using MPC.
- **Survey Participants** to submit their data in a privacy-preserving manner.
- **Analysts** to retrieve the evaluated & aggregated survey results.

We first provide a high-level overview of the different actors and data flow in Section 3.1, whereas technical details are covered in Section 3.2. Table 1 serves as a glossary for the actors and involved parties within Chapter 3.

Table 1: Actors and involved parties within Chapter 3.

Actor/Party	Description
Management Server	A central server that orchestrates all communication (except direct connections between MPC Nodes) and computation efforts. Even though the Management Server rotates the data it has no cleartext access to the data and does not learn sensitive information. The Management Server provides web frontends for Survey Participants, Admins, and Analysts. MPC Nodes are registered at the Management Server and the Management Server assigns MPC Nodes once a computation needs to be carried out.
MPC Node	A standalone application that performs the actual Multi-Party Computation by engaging with two other MPC Nodes.
Admin	Manages the survey lifecycle using the Management Server. Has no access to sensitive data or Survey Results.
Survey Participant	An authenticated individual that represents a company who uses the frontend provided by the Management Server to contribute survey answers.
Analyst	Retrieves Survey Results and is responsible for distributing them to the consortium via the True Demand Framework.
Distributor	An entity that receives details about a Survey Instance from the Admin (e.g. the participation URL) and who has access to a list of to-be-invited participants to that survey. The distributor uses the WP2 user authentication & identification solution to send invitations for that Survey Instance to a list of survey participants that are known to the WP2 user authentication & identification system. The list of to-be-invited participants is defined by the SC4EU consortium

3.1 Actors/Data Flow Overview

The Management Server plays a central role in orchestrating the data flow and hosting the (web) frontends for Survey Participants, Admins, and Analysts. Three independent MPC Nodes (using the REP3 protocol [2]) are connected to the Management Server and are waiting for instructions to compute survey results. As mentioned in Chapter 2, these three MPC Nodes need to be operated by independent entities that do not collude.

Admins maintain the Management Server and are responsible for creating custom survey definitions according to the needs of the SC4EU consortium, initiating an answer collection phase, providing participation URLs to the Distributor, and invoking the survey result computations once the answer collection phase ends. The Admin uses a web interface provided by the management server and needs to authenticate as an administrator in order to perform actions. Custom survey creation is covered in Chapter 5.

Asymmetric cryptography allows two parties to exchange confidential information without anybody else having access to that information. For example, Alice may send an encrypted message to Bob by encrypting her message using Bob's public key. Only Bob's private key can decrypt Alice's message, and therefore only Bob can retrieve the message as long Bob keeps his private key secret to himself. [6], [7]

Every MPC Node possesses a distinct unique keypair (private and public keys) where the Management Server knows the public key of each MPC Node. The Management Server provides a survey questionnaire alongside the three public keys of the MPC nodes to Survey Participants via a web interface. Survey Participants enter their answers in the web interface. All answers never leave the web interface, which is run in the Survey Participant's web browser, i.e. the Survey Participants trusted device. The web interface splits the answers into three secret shares (according to the REP3 protocol [2]) and encrypts each secret share with one of the MPC nodes' public keys (more details on the Survey Participant frontend are given in Chapter 0). The shares are now protected and are ready to be sent back to the Management Server who cannot decrypt the shares, as only each MPC Node holds the respective private key. This process is illustrated in Figure 1, where every keypair is highlighted using the same color, where "priv" stands for private key and "pub" stands for public key.

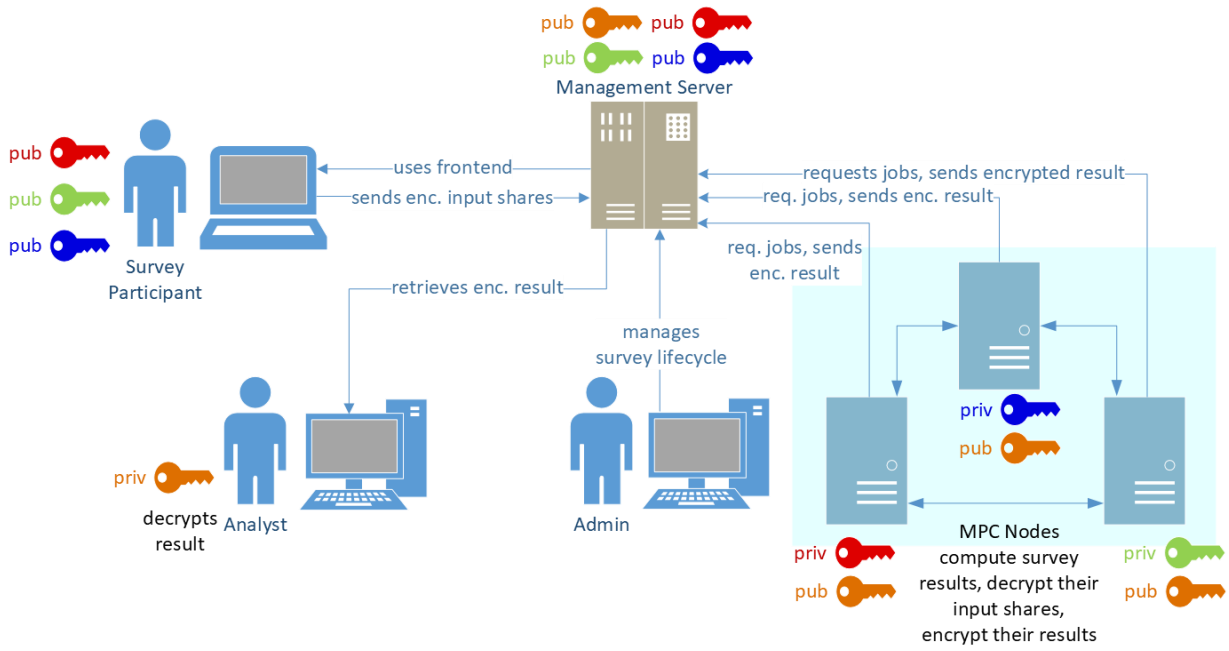

Figure 1: System Architecture Overview.

The Analyst owns a unique keypair where its public key is known by the Management Server. Once the Admin initiates the evaluation of a survey, the Management Server queues one survey evaluation job consisting of the survey definition (including aggregation computation instructions), the encrypted input shares, and the public key of the analyst. Every MPC Node fetches its required information for the job (their assigned input shares, survey definition,

and the Analyst's public key) from the Management Server, decrypts their input shares, and once all MPC Nodes are ready, the MPC Nodes engage in Multi-Party Computation to collaboratively compute the result. Every MPC Node encrypts their result using the Analyst's public key and sends the encrypted result back to the Management Server. Again, the Analyst's keypair is highlighted using the same color in Figure 1, where "priv" stands for private key and "pub" stands for public key.

3.2 Technical Details

The Management Server consists of a database (DB) instance (e.g. PostgreSQL) for storing the system state and a binary that contains the application logic and serves all requests. All web frontends are served via HTTPS and are based on HTML, JavaScript/Typescript, and WebAssembly. User interactions in the frontend communicate with the Management Server via REST and/or gRPC API calls. The Management Server listens on TCP sockets of configurable ports as the Management Server needs to serve incoming connections from clients (web frontend) and MPC Nodes.

All technologies listed in this report are subject to change, as the implementation is not finished yet. WP6 will start with the modifications for the final implementation once the requirements from WP5 towards WP6 are known (see D5.1).

One MPC Node consists of a binary that contains all application logic. The MPC Node communicates with the Management Server via gRPC API calls and establishes a connection to the Management Server (and requires no open port for Management Server communication). Computations require every MPC Node to listen on a TCP socket of a configurable port and need to be accessible by the other MPC Nodes via an IP address or hostname.

Once an MPC Node receives a job from the management server, it needs to connect to the two other MPC Nodes which happens by waiting for an incoming TCP connection from one MPC Node and opening a TCP connection to the other MPC Node. The hostnames/IP-addresses of the other nodes are provided by the Management Server as part of the job description.

MPC requires communication between the three MPC Nodes. To achieve viable execution times it is important to ensure good connectivity between the MPC Nodes in terms of high bandwidth and very low latency.

As highlighted in Chapters 1, 8, and 9 not all requirements for computations in MPC are known yet. We can therefore only give estimates for the hardware requirements in this deliverable based on our experiments. We expect that the hardware requirements listed in Table 2 will suffice.

Table 2: Expected hardware requirements

Type	# CPUs	RAM [GB]	Disk Space [GB]
Management Server	4	8	16
MPC Node	8	16	16

4. Frontend Client Interfaces

The MPC Survey Framework described in Chapter 3 provides user interfaces for humans to interact with the system. As mentioned in Section 3.2, all user interfaces are web apps hosted by the Management Server and accessible to humans by accessing a URL in their browser. This Chapter provides insights to all user interfaces provided by the MPC Survey Framework. Table 3 serves as a glossary for Chapter 4.

Table 3: Terms used throughout Chapter 4.

Term	Description
Survey Definition	Contains a definition of questions and a list of possible answers to each question. Furthermore, a set of computations defined on the answers to each question defines how the Survey Definition's results look like.
Survey Instance (Survey Run)	A Survey Instance uses the set of questions, answers, computations, and results defined by a Survey Definition to conduct one specific survey. Participants may provide their answers to a survey where the answers will be used to compute the results of the survey.
Survey Results	A list of aggregated values that have been computed from the answers of a Survey Instance where the computations are given by a Survey Definition.

All users of the system require authentication to access their services. Details on user authentication and identification are covered in Chapter 6. This Chapter focuses on the user interfaces for already authenticated users. Section 4.1 presents the Admin Interface for defining surveys, instantiating surveys, and closing surveys/computing survey results. Section 4.2 shows the Survey Participant Interface that allows participants to enter and submit their answers in a privacy preserving way. Section 4.3 goes into detail on the Analyst Interface that enables analysts to view survey results and to download the results as a file.

4.1 Admin Interface

Administrators of the system perform the following tasks:

- Creating custom Survey Definitions that are composed of questions, answers, and instructions for aggregating individual answers into a survey result.
- Creating Survey Instances from Survey Definitions. An instance accumulates answers from Survey Participants.
- Closing instances of custom surveys. Once a survey is closed, the individual answers get aggregated into Survey Results.

Creating custom Survey Definitions in terms of defining questions, answers, and result aggregation computations is covered in Chapter 5. In summary, a custom Survey Definition gets crafted by writing a Survey Definition using a domain specific language (DSL) as a single file. Once the survey definition is finished, the file can be uploaded to the Admin Interface via the "+ Upload new Survey" button depicted in Figure 2.

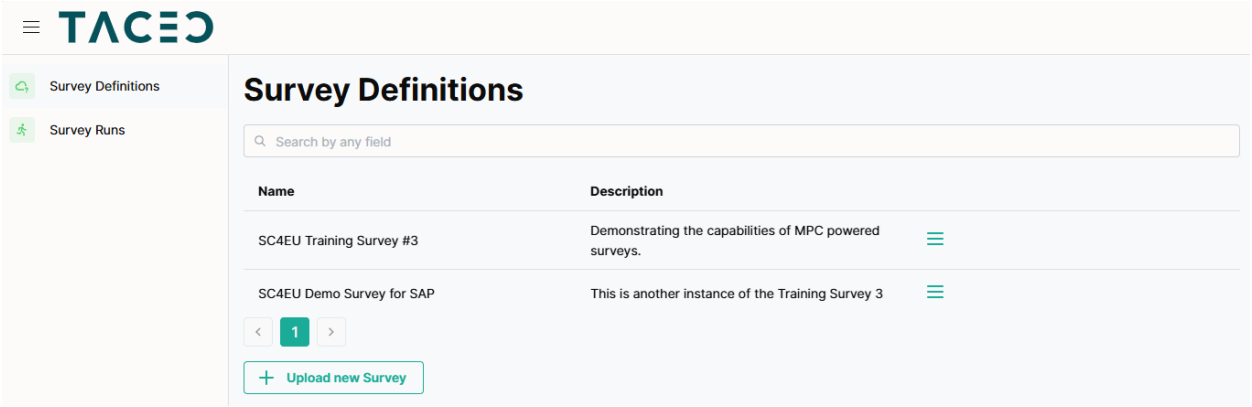

Figure 2: Admin Interface: Survey Definitions screen. Administrators are presented with a list of survey definitions, where they can upload new survey definitions, delete existing survey definitions, or create a survey instance from a survey definition.

The burger menu in the last column of each survey definition allows Admins to inspect the details of the Survey Definition, such as the questions, answers, and the results computation logic. Furthermore, Survey Definitions may be instantiated into a Survey Instance (Survey Run) for collecting answers from participants. One Survey Definition may have 0 to n Survey Instances. Additionally, Survey Definitions can be deleted via the burger menu.

Figure 3 shows the Survey Runs screen that allows to maintain Survey Instances. Every Survey Instance refers to its Survey Definition, has a status, and a creation date & time. The possible states of a Survey Instance are:

- Colleting Answers
- Computing Results
- Finished
- Error

Admins can retrieve a URL for Survey Participants from the burger menu in case the survey is in the state “Collecting Answers”. This URL can then be passed on a distributor who forwards the URLs to Survey Participants.

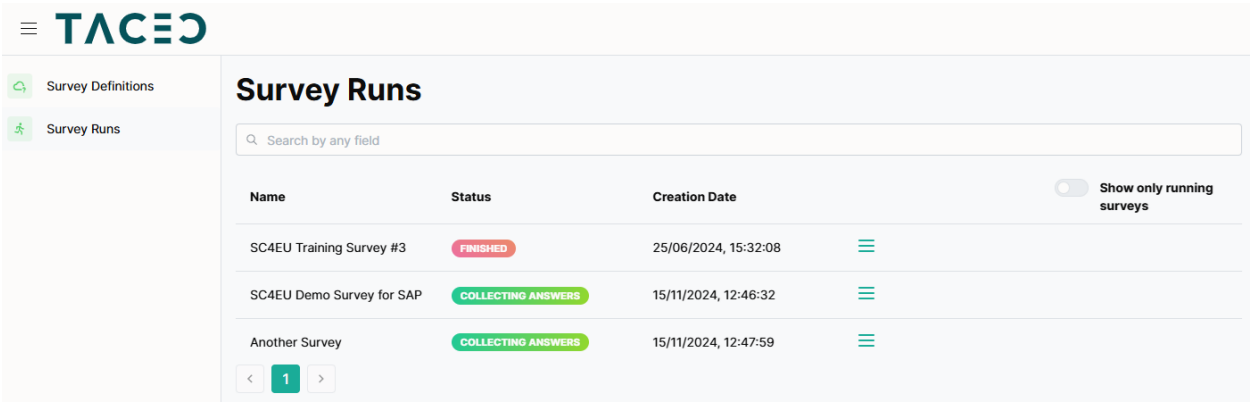

Figure 3: Admin Interface: Survey Runs screen. A list of survey instances indicates the current status of each survey and its creation date and time.

Once it is time to end a survey and to compute the results, the Admin can initiate the survey closing action from the burger menu of a Survey Instance that is in the state “Collecting Answers”. The Survey Instance will change its state to “Computing Results”, where the MPC Nodes start their work in the background. If the computation succeeds, the Survey Instance

will transition into the state “Finished”. In case the results computation using the MPC nodes runs into problems, the Survey Instance transitions into the state “Error”.

A Survey Instance of state “Finished” has its answers ready to be fetched for Analysts. Administrators cannot access the results of any survey. Section 4.3 shows how to retrieve the results of a survey instance as the Analyst role.

4.2 Survey Participant Interface

Survey Participants receive participation URLs from their distributor to access the input form for one specific Survey Instance. An exemplary survey input form is shown in Figure 4 where Survey Participants can provide answers and submit the form.

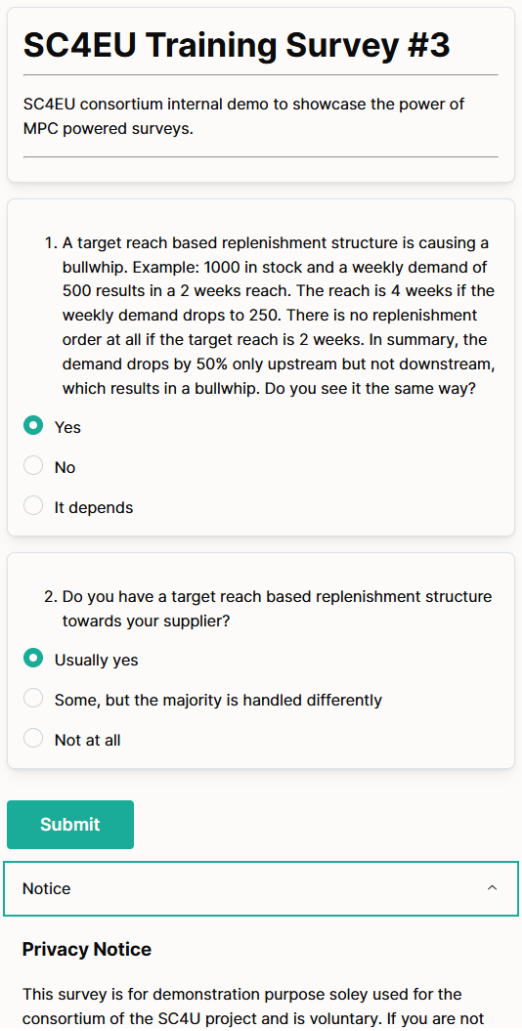

Figure 4: Survey Participant Interface: Survey input and submission screen. Users can provide their answers and use the Submit button to split their answers into secret shares, encrypt them locally on their device, and send the secret shared answers to the Management Server.

As described in Chapter 3, the answers in cleartext never leave the Survey Participant’s device. The web app uses JavaScript code (including WebAssembly) on the client device of the Survey Participant and performs two important operations when triggering the “Submit” button:

1. Splitting the cleartext answers into secret shares according to the REP3 protocol [2].

2. Secret sharing is not enough, as recombining the shares yields the cleartext answer of the Survey Participant. The secret shares are sent to the Management Server who then further distributes the shares to the MPC Nodes. Each share gets encrypted using one of the public keys of the MPC Nodes as depicted by Figure 1 so that relaying the secret shares via the Management Server does not leak information.

The encrypted secret shares of the Survey Participant's answers are then sent to the Management Server who cannot decrypt the shares, as only the MPC Nodes own the respective private keys needed for decryption.

4.3 Analyst Interface

Analysts may view aggregated Survey Results computed via MPC. The Analyst has access to the Survey Runs screen shown in Figure 3, but its actions and permissions are different from Admins. The Analyst may view survey results for survey instances of state "Finished" using an action from the burger menu in the Survey Runs screen. Figure 5 shows an example for a survey results screen where Analysts can view the results in the web interface and also have the possibility to download the results as a file via the download button on the bottom right of the screen.

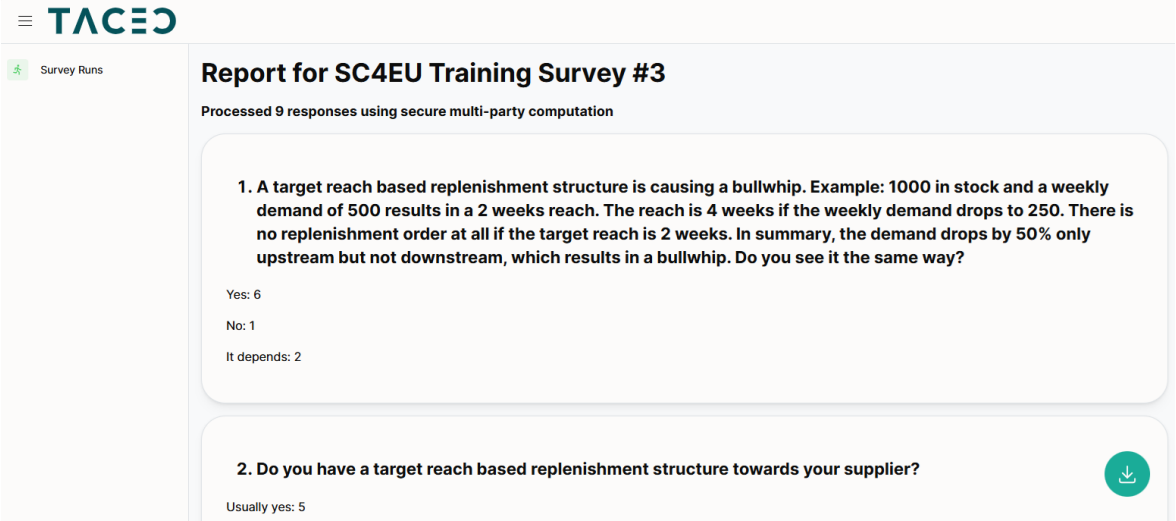

Figure 5: Analyst Interface: survey results screen. Analysts can view the results of a survey instance of state "Finished" and can also download the results as a file using the button on the bottom right.

5. Survey Definition Interfaces

User interaction with the web interface of the MPC Survey Framework is presented in Chapter 4, where Section 4.1 describes the interactions of Admins with the system. Creating a custom survey definition is not possible using the web interface and must be performed using a text editor of the user's choice. A survey definition is described in JSON format and needs to be uploaded to the Admin Interface as a single file, as described in Section 4.1.

The survey definition JSON document uses a pre-defined set of fields to describe questions, answers, datatypes, and computations for aggregating results. As mentioned in Chapters 1, 8, and 9 not all requirements towards the MPC Survey Framework are known at the time of writing this document [M12]. Therefore we present a simple and non-exhaustive JSON document structure which is subject to change until the release of Deliverable D5.1 [M18] of the SC4EU project. JSON might not be suitable for survey definitions that fulfill the upcoming requirements and a different domain specific language (DSL) may be introduced in the future (especially for expressing computations) that completely replaces the JSON document.

At the time of writing this document, the MPC Survey Framework supports only basic aggregation functions, like summing up the input votes, computing an average or aggregating the number of votes for different input ranges in buckets. This functionality will be extended to support the operations identified by D5.1 in the future as part of Task T6.3. At the moment, the following JSON document structure defines a survey: The root of the JSON document contains exactly one child with the label `"inputs"` and its datatype is a list of objects, where every object must follow the structure defined in Table 4.

Table 4: Temporary JSON object structure for an input, i.e. a survey question including answers. This structure (including the use of JSON) is temporary and subject to change once the requirements from WP5 towards WP6 are known (see D5.1).

Label	Datatype	Description
`"label"`	`string`	To be used by operations to identify data sources in the future.
`"description"`	`string`	Question text to be shown to Survey Participants.
`"shared"`	`boolean`	Input needs to be kept private (secret shared).
`"answers"`	`array of string`	List of possible answers.
`"operation"`	`string`	Frontend Interface/Answer Collection Method, where the following operations are supported now: • `"RadioU64"` … single choice question, counts the number of answers provided. • `"CheckboxU64"` … multiple choice question, counts the number of answers provided. • `"AverageU64"` … computes the average of all provided answers. • `"AverageStdDevI64"` … computes the average and standard deviation of all provided answers. Supported operations are subject to change once all requirements from WP5 towards WP6 are known (see D5.1)

Listing 1 shows the content of the file “Example-Survey-Definition.json” which adheres to the JSON structure definition given above. Admins can create Survey Definitions according to their needs and upload them to the Admin Interface. Figure 6 shows the Survey Participant Interface for a survey instance of the survey definition given in Listing 1.

```
{
  "inputs": [
    {
      "label": "first",
      "description": "1. What impact will chiplets have for the prosperity of the semiconductor industry?",
      "shared": true,
      "answers": [
        "They will be a game changer.",
        "They will significantly support the prosperity of the semiconductor industry.",
        "They will somehow support the prosperity of the semiconductor industry.",
        "No real impact.",
        "No impact at all."
      ],
      "operation": "RadioU64"
    },
    {
      "label": "second",
      "description": "2. Chiplets will be mainly used in the following applications (when one is domination to 50%, name only one, when two are dominating more than 70% name only two, don’t name those with shares < 10%):",
      "shared": true,
      "answers": [
        "Datacenters",
        "Automotive",
        "Sensors & edge applications",
        "Mobile devices",
        "Other"
      ],
      "operation": "CheckboxU64"
    }
  ]
}
```

Listing 1: Content of the file “Example-Survey-Definition.json”

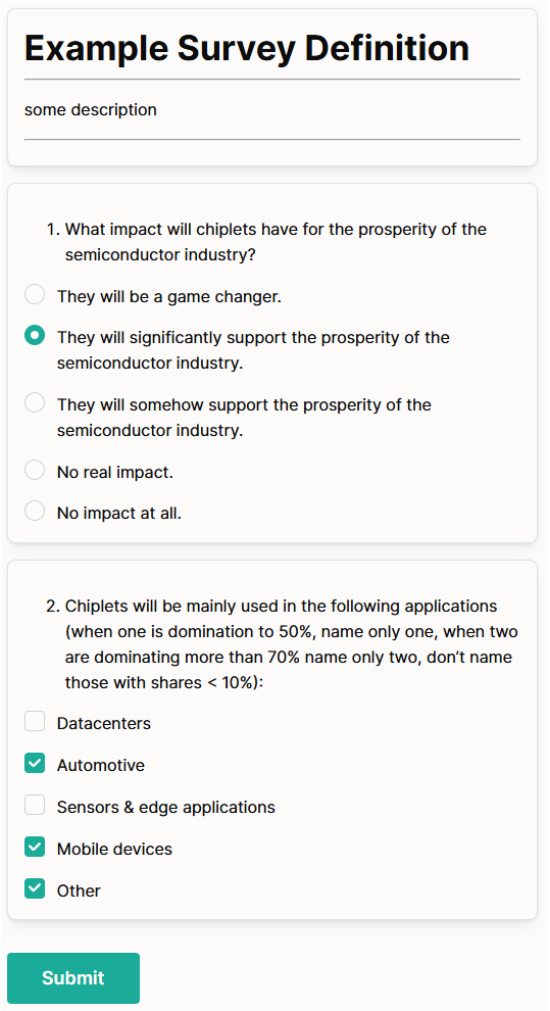

Figure 6: Survey Participant Interface: Screen of the uploaded file "Example-Survey-Definition.json"

5.1 Computation Security Considerations

The Admin is responsible for creating survey definitions according to the needs of the SC4EU consortium and defines the computations as they will be executed by the MPC Nodes. It is crucial to ensure that the computations of the survey do not leak any unwanted information. The final computations need to be reviewed by the SC4EU consortium, the Analyst, and Survey Participants to avoid problematic configurations that lead to undesired disclosure of sensitive information.

WP6 creates the infrastructure for running secure Multi-Party Computation surveys, but even if the technology (i.e. MPC) is secure, the computation itself may partially or fully disclose secrets. Consider the simple example of a survey that computes the average of a value provided as input from all survey participants. Assume that there are two participants where each participant remembers its own submitted value. Releasing the result of the survey (the average of all inputs) to the Analyst and all Survey Participants leads in the Analyst knowing only the average, but the two survey participants have learned all individual inputs (since they each know one input and the average, they can recover the second input), even though MPC did not reveal these values.

This simple example from the previous paragraph demonstrates an obvious flaw, but as computations become (slightly) more complex, the implications of potential data disclosure are more difficult to spot. Properly auditing any planned computations in MPC is a necessity to ensure anonymity in the SC4EU project and must be enforced by the Admin.

6. User Authentication and Identification

While user identification for Admins and Analysts is an obvious requirement to ensure the security of the MPC Survey Framework, Survey Participant authentication and identification is a relevant topic as well. Survey Participants shall be able to submit a single vote and submitted answers need to be the untampered answers of the Survey Participant. Furthermore, it might be necessary to use information about the participant during aggregating survey results (e.g. the participant's weight or market segment, …).

WP2 implements Platform Security and User Administration of the True Demand Framework within Task T2.4 [M01-M30] and describes the system in the reports of Deliverables D2.4 [M18] and D2.5 [M30]. The system will provide user authentication and identification for all components of the True Demand Platform, including the MPC Survey Framework.

The MPC Survey Framework will use the user authentication and identification system to uniquely identify Admins and Analysts and grant adequate permissions to use the system according to the requirements of the SC4EU consortium. The list of survey participants is maintained by the SC4EU consortium and authorized participants are marked with metadata in the user authentication and identification system so that the MPC Survey Framework can recognize authorized Survey Participants.

7. Information Flow

This chapter elaborates on the information flow and the actors involved analogous to Chapter 3 which defines the MPC Survey Framework architecture.

TACEO will act as the Admin and the Analyst throughout the SC4EU project, where Table 1 provides a definition thereof. The Admin is responsible for maintaining the lifecycle of surveys and performs actions according to the needs of the SC4EU consortium. The SC4EU consortium defines surveys consisting of questions and answers in a written form and clearly defines the computations that lead to the aggregated survey results. TACEO creates a Survey Definition based on this information and takes care for auditing the computations defined by the consortium as described in Section 5.1. A definition of the terms Survey Definition, Survey Instance, and Survey Results is given in Table 3.

The SC4EU consortium further requests the creation of a Survey Instance based on a Survey Definition (including a deadline on when to collect Survey Results). The Admin (TACEO) performs all necessary tasks to create the instance and to provide participation information to the Distributor who forwards the information to Survey Participants, where Survey Participants provide answers to the survey questions.

Once the deadline has passed, the Admin will close the survey and trigger the computation of the Survey Results. The Survey Results will then be forwarded by the Analyst (TACEO) to the True Demand Framework which further provides access to the Survey Results to users of the True Demand Framework. The information flow is visualized in Figure 7.

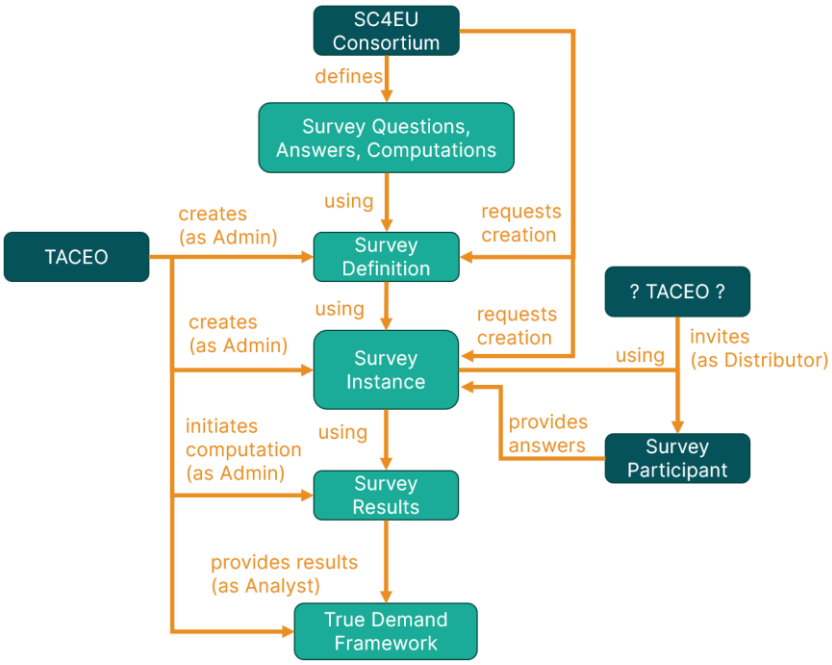

Figure 7: Information Flow and involved Actors

8. Current Implementation Status

The architecture of the MPC Survey Framework as described in Chapter 3 is fully implemented and needs to be adapted to the requirements defined in Deliverable D5.2 [M18]. This Chapter highlights the current implementation status, lists open tasks, and shows some .

8.1 Management Server

The Management Server provides logic to persist any state (e.g. Survey Definitions, Survey Instances, Survey Participant inputs, aggregated results, …) in a database and is fully implemented in regards to the base implementation defined in Task T6.1. Communication with MPC Nodes takes place via gRPC API calls, where the API is fully implemented and functional. The web frontend for Analysts, Admins, and Survey Participants is fully implemented and functional, except a clear distinction between Analysts and Admins, where both roles are assumed once authenticated.

Missing topics at the time of writing this document is the implementation of requirements collected in WP5 D5.1, where the implementation is part of Task T6.3. Furthermore, the integration of the WP2 user authentication and identification system into the MPC Survey Framework is not implemented yet. WP6 is waiting for WP2 to provide details on how to connect the MPC Survey Framework to the WP2 user authentication and identification solution.

8.2 MPC Nodes

One MPC Node is a standalone process that communicates with the Management Server via gRPC API calls and via TCP connections to other MPC Nodes, where all logic and communication is fully implemented in regards to the base implementation defined in Task T6.1.

Further adaptions to the MPC Node might be necessary in the future as part of Task T6.3 to accommodate the requirements of WP5 towards the MPC Survey Framework. The architecture itself is subject to change as well, depending on the structural changes of the Management Server.

8.3 Deployment

The applications (Management Server, Database, and MPC Node) are bundled into one Docker image each and may be deployed individually on any system that is able to run containerized applications on x86 or ARM architectures. The applications are configured via (environment) variables upon container creation.

8.4 Test Runs

The SC4EU consortium carried out multiple end-to-end test runs of the MPC Survey Framework (without authentication), demonstrating a ready to be used system where the first test run took place in March 2024 [M04]. The first test run fully utilized the MPC Survey

Framework, but all three MPC Nodes were hosted by TACEO, which is insecure. Efforts have been made in deploying two nodes at members of the SC4EU consortium, where the partners TIB and OPAIX volunteered to host one MPC Node each. The third SC4EU internal “training survey” in June 2024 [M07] uses the full MPC Survey Framework in a secure way, as all three MPC Nodes are hosted by trusted, individual parties.

Another demo survey was conducted at the Hannover Messe 2024 in April using three MPC Nodes hosted by TACEO (insecure). Infineon gave a presentation that included a demonstration of MPC powered surveys to a broad audience who were able to participate in the demo survey run.

All survey test runs were successful in collecting inputs, aggregating and providing results to the SC4EU consortium. TACEO took the role of the Admin, Analyst, and the Distributor, while Survey Participants were unauthenticated.

9. Future Work/Outlook

The current implementation will be adapted once all requirements towards the MPC Survey Framework are defined by WP5 in D5.1. In particular, the following topics are future work:

- Survey Definition Language: The possibility to perform computations aside from summing up individual single/multiple choice votes.
- Adaption of the underlying MPC engine to perform computations according to the updated survey definition language.
- Implementation of collecting more complex survey inputs, for example inputs of structured format, such as tables.
- Integration of the WP2 user authentication and identification system into the MPC Survey Framework.

All of these features will be implemented in close collaboration with other work packages, especially WP2 and WP5. As concrete requirements and design decisions for other components are not available at the time of writing this document, the future work may undergo further changes.

The role of the Distributor (as defined in Chapter 3) is not clearly defined yet. Its main purpose is to send invitations to a list of authenticated survey participants, where the list is defined by the SC4EU consortium for each survey. The Distributor needs to interact with the MPC Survey Framework, the WP2 user authentication & identification system, and the SC4EU consortium. TACEO could assume the role of the Distributor throughout the project.

10. Conclusions

This report outlined the current state of the implementation of the MPC Survey Framework. The base functionality as defined in Task T6.1 was fully implemented, and four end-to-end tests were conducted withing the SC4EU consortium demonstrating a working and stable system. WP6 deployed two MPC Nodes at TIB and the affiliated partner OPAIX, the third MPC Node and the Management server are hosted by TACEO which marks progress on Task T6.2. WP5 is in the progress of defining requirements towards the MPC Survey Framework at the time of writing this document (Deliverable D5.1). These requirements will be incorporated into the existing MPC Survey Framework as part of Task T6.3 in the future. Additionally, the MPC Survey Framework will be incorporated into the WP2 user authentication & identification solution.

11. References

[1] A. Shamir. How to Share a Secret. Communications of the ACM, 22(11):612-613, 1979.
[2] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. In CCS, pages 35–52. ACM, 2018.
[3] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara. High-throughput semi-honest secure three-party computation with an honest majority. In E. R. Weippl, S. Katzenbeisser, C. Kruegel, A. C. Myers, and S. Halevi, editors, Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016, pages 805–817. ACM, 2016.
[4] Andrew Yao. Protocols for Secure Computations (Extended Abstract). FOCS. IEEE Computer Society, 1982, pp. 160–164.
[5] David Evans, Vladimir Kolesnikov, Mike Rosulek. A Pragmatic Introduction to Secure Multi-Party Computation, 2018, https://securecomputation.org/docs/pragmaticmpc.pdf
[6] Diffie, W., and Hellman, M. New directions in cryptography. IEEE Trans. Inform. Theory IT-22, (Nov. 1976), 644-654.
[7] R.L. Rivest, A. Shamir, and L.M. Adleman, “A Method for Obtaining Digital Signature and Public-Key Cryptosystems,” Communications of the ACM, Vol. 21, No. 2, pp. 120–126, 1978.

12. Appendices

12.1 Appendix A - Abbreviations

Table 5: Abbreviations

Abbreviation	Meaning
SC4EU	True Demand-Driven Semiconductor Supply Chains for Europe
MPC	Multi-Party Computation
DSL	Domain Specific Language