Technische Universität München, Department of Informatics

Chair for Network Architectures and Services

Prof. Dr.-Ing. Georg Carle

Thesis M.Sc.

Secure and Controlled Querying of Sensor Data

Motivation

In a former thesis, we developed a solution for privacy-preserving collection and processing of sensor data in distributed environment. In the environment, small and locally distributed nodes sense their environment and collect information about it. They store these information only locally and keep a history of it. A central node is able to gather collected data and perform computations on it (e.g. statistics) using Secure Multiparty Computation; a technology which allows distributed computation with input data, which has to be kept private and may not be shared itself. This central node also takes care of all connection management and computation orchestration, so that these privacy-preserving computations can be performed in an efficient, stable and robust manner.
In this context, the proposed thesis shall provide SMC as a service. This means, the whole SMC part and the nodes management shall be concealed and a “normal” API shall be presented to clients which want to query data from the distributed system.

Approach

Data sources and data sinks (= clients) are separated by a gateway which performs a first layer of access control: It authorizes clients to query individually predefined types of data and forbids other requests. When a request is performed by a client, it has to authenticate and its authorization is checked by the gateway.
The second layer is authorization by the data sources themselves. They receive the request and should be allowed to veto against request if they object the purpose of the request.

Your Task

The goal of this thesis is to build an API and the corresponding gateway logic which enables the controlled querying of the data available in these contexts. The API should provide a sophisticated mechanism of access control and enable the actual data sources to stay in control of their data.

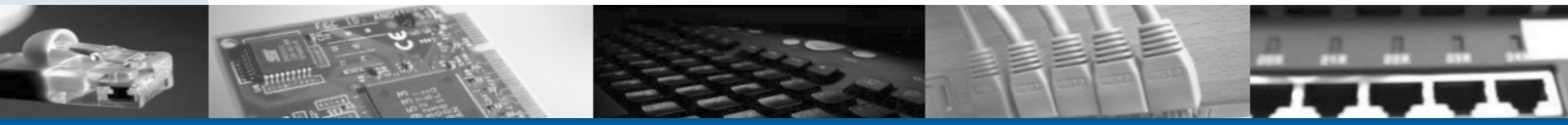
Requirements

Python or Java, Linux, basic network security knowledge, self-dependence.

Contact

Dr. Holger Kinkelin kinkelin@net.in.tum.de
Marcel von Maltitz vonmaltitz@net.in.tum.de

http://go.tum.de/039305



Secure and Controlled Querying of Sensor Data

June 30, 2017

1 Use Cases

This thesis is motivated by two use cases:

SMC as a Service In a former thesis, we developed a solution for privacy-preserving collection and processing of sensor data in distributed environment. In the environment, small and locally distributed nodes sense their environment and collect information about it. They store these information only locally and keep a history of it. A central node is able to gather collected data and perform computations on it (e.g. statistics) using Secure Multiparty Computation; a technology which allows distributed computation with input data, which has to be kept private and may not be shared itself. This central node also takes care of all connection management and computation orchestration, so that these privacy-preserving computations can be performed in an efficient, stable and robust manner.

In this context, the proposed thesis shall provide SMC as a service. This means, the whole SMC part and the nodes management shall be shadowed and a "normal" API shall be presented to clients which want to query data from the distributed system. Further requirements are elaborated after the second use case.

Hierarchical distributed Anomaly Detection System The second use case is based on the DecADe project we are currently working on. In a distributed embedded network (e.g. in Smart Cars and Airplanes), sensor devices and components collect specific data from their proximity. They are interconnected and data flow between them is generally possible. A hierarchic Anomaly Detection System (ADS) will be incorporated in these networks in order to detect suspicious and/or faulty behaviour. The first level of the ADS consists of the measurement nodes itself which share certain data with their neighbours. Higher levels of the ADS are realized by dedicated components, called "Forensics

Centers", which use the information of a predefined set of sensor nodes (or other Forensics Centers of lower hierarchy levels). The data privacy should not be endangered by allowing all ADS components to collect and process all information available.

Similar to the previous context, data possessing components should provide an API which enables specific querying of data currently relevant by an ADS component.

2 Abstraction

An abstraction to both use cases is as follows: A distributed set of *nodes* are data sources which collect and hold specific data and its history. These information is privacy-critical and should be protected (which happens differently in the use cases). The purpose of a *client* is to query information and to process it for a given goal. There exists a *gateway* (which can be a dedicated node or simply a software component on a data source) which should accept queries, perform access control and be able to obtain the queried information.

3 Goal

The goal of this thesis is to build an API and the corresponding gateway logic which enables the controlled querying of the data available in these contexts. The API should provide a sophisticated mechanism of access control and enable the actual data sources to stay in control of their data.

3.1 Requirements

Directory Service When nodes add themselves to the proximity of a gateway, they should announce the data they offer. The gateway should be able to create a directory which data it can obtain. The gateway should provide a Directory API which enables clients to dynamically find out, which data the gateway can obtain.

Stateless Access Control The gateway should provide an API for permission requests. A client can request the permission to obtain a specific type of data. If the request is legitimate, the gateway sends a signed permission grant. This has to be stored by the client and must be provided when querying for actual data. The gateway should provide an API for actual data queries. The matching permission grant must be provided by the client in order to legitimate its request. The grant must be verified by the gateway.

Request Translation In the current cases, the gateway often does not possess the data itself but has to obtain it by other means. Hence, it must be able to interpret the request, perform necessary steps and translate the given request into a target language for the actual data sources.

Transparency and Intervenability for Data Sources In order to achieve the privacy goals of Transparency and Intervenability for the data sources, they should be able to obtain the original request, understand the purpose of it and validate its legitimacy. Furthermore, the gateway has to provide a proof that the request actually stems from a client who requested it at the given timestamp (no fake requests by other services or the gateway itself). If some of these checks fail, data sources must be able to reject the request.

Efficient Retry The actions of the gateway and the client are validated by the nodes. If a data gathering action fails, a retry should be efficient and not all checks should be necessary to be made again. This could be achieved by giving a request a unique ID which is then stored as "permitted" on the nodes, even when the request fails.