mPlane

an Intelligent Measurement Plane for Future Network and Application Management

ICT FP7-318627

Specification of mPlane Access Control and Data Protection Mechanisms

Author(s): SSB G. De Rosa
SSB S. Pentassuglia

Document Number:	D1.2
Revision:	1.0
Revision Date:	31 Jul 2013
Deliverable Type:	RTD
Due Date of Delivery:	31 Jul 2013
Actual Date of Delivery:	05 Aug 2013
Nature of the Deliverable:	(R)eport
Dissemination Level:	Public

Abstract:

This document primarily defines security specifications for the mPlane architecture (in terms of authentication, access control and safe communications), on the basis of what specified in the D1.1. Also, it provides a description of the measures that can be adopted in order to guarantee the privacy of the data gathered through the probes. This aspect of the mPlane infrastructure must not be neglected, since from a legal point of view the users' right to privacy must be protected in any case. The techniques to be adopted are anonymization and aggregation, but utility of data decreases as the level of privacy increases, hence it is necessary to find a good trade-off.

Two protocols are proposed for secure communications among components: TLS and SSH, which adopt respectively X.509 certificates and RSA keys for identity management. As the access control policy that will be adopted depends mostly on the mPlane administrators' choices, this document provides a survey of several approaches.

The cross-domain and the mobile scenarios are also analyzed, providing solutions that can guarantee access control, security and privacy.

Keywords: privacy, anonymisation, security, access control, authentication

Disclaimer

The information, documentation and figures available in this deliverable are written by the mPlane Consortium partners under EC co-financing (project FP7-ICT-318627) and does not necessarily reflect the view of the European Commission.
The information in this document is provided ``as is'', and no guarantee or warranty is given that the information is fit for any particular purpose. The user uses the information at its sole risk and liability.

Contents

Document change record

Version	Date	Author(s)	Description
0.1	08 Jul 2013	G. De Rosa (SSB)	initial draft
0.2	21 Jul 2013	G. De Rosa (SSB)	second draft
1.0	31 Jul 2013	G. De Rosa (SSB)	final draft

1 Introduction

The collection and analysis of measurement data over the Internet presents obvious risks in terms of privacy of End-Users and ISP owners.

Data collection and protection are subject to European Union and national-level regulations, whose principles are described in the next chapter. Every sensitive information (e.g. personal information about the data subjects, or potentially business-sensitive information about ISP networks or AP infrastructures) owned by the data controllers must not be exposed to unauthorized people, neither accidentally nor as a result of an attack. For this reason, the data controllers must provide strict security controls on data, limiting the access only to trusted and authorized subjects, and trying to mantain the integrity of the data subjects' privacy. In order to enforce the privacy on the data, several anonymization principles and techniques (such as perturbation, pseudonymization or aggregation) can be adopted, although increased anonymity implies reduced utility of data.

The distributed nature of mPlane implies the danger of data circulation across different administrative domains, especially network operators, as well as the potential of indirect data re-identification through linking and combination. In addition, as mPlane capabilities will include also active measurements, it is especially important to control the access to the probes to ensure they are not misused, e.g., in denial of service attacks. More in general, in order to provide a flexible access control method, not only at the probes but also at the other mPlane components, several access control models are taken into account.

Most of the communications between mPlane components must be secured, and must guarantee authenticity, confidentiality and integrity. Secure channel communication protocols such as HTTPS or SSH can provide a good level of security, exploiting respectively X.509 certificates and RSA keys.

This document defines the techniques and mechanisms that will be implemented in the mPlane components in order to ensure access control and data protection, according to the architecure specifications defined in D1.1, and specifies the requirements to fulfill the constraints on personal data privacy.

Each chapter of this deliverable is composed mainly of two sections: in the former one, the threat model or the requirements for a determined aspect of mPlane are analyzed and explained, while in the latter we propose a security model to solve those threats and requirements.

2 Data Privacy

2.1 Data Privacy EU Legislation

Article 8 of the Charter of Fundamental Rights of the European Union [22] specifically recognises the fundamental right to the protection of personal data, that can be processed only after the consent of the concerned subject, and must be treated fairly and only for the specified purposes.
Actually, all online users' activities may be closely monitored. Even where users are not required to provide personal data when accessing services on the Internet, individuals can be identified through the Internet Protocol (IP) address of their computer or smartphone, but also using digital ‘cookies’ stored in their browser cache memory visiting web sites.
The overall Internet communication traffic tends to leave user's footprints of Web pages visited, emails and instant message senders and recipients, voice over IP callers, advertisements viewed, web searches, commercial products examined and purchased, etc.
Moreover, this personal information leakage is not restricted only to the Internet activities. Mobile phones sending location information to the network providers enables fine-grained user tracking. Debit and credit card payment systems record amounts spent and stores visited that can be used for profiling. Massive use of online shopping and of social networking sites to share information about themselves and their family, friends and colleagues are exploited using data mining technologies, that find patterns in those large collections of personal data, attempting to predict individual interests and preferences.

2.1.1 State of the art

Nowadays the main relevant european legal framework on protection of personal data is the Directive 95/46/EC [16], that has been integrated by the e-privacy directive 2002/58/EC [17], as revised by 2009/136/EC [18], dealing with the regulation of a number of important issues such as confidentiality of information, treatment of traffic data, spam and cookies.
The main concepts that play a fundamental role in the data privacy regulation are the *Data Subject*, the *Personal Data*, and the *Data Controller*.
Personal Data is any information relating to an identified or identifiable natural person or *Data Subject*, whether it relates to his or her private, professional or public life. Depending on the identification process, there are several types of data that can reveal the data subject identity, such as a name, a photo, an email address, bank details, posts on social networks, medical informations, or its IP address. More generally, each information that can be used in any way to re-identify the subject of the data can be considered as Personal Data and is subject to the above EU regulations.
Data Controllers are those individuals or entities which collect and process personal data (e.g. an Internet Service Provider is the controller of all the traffic collected in its own network). Data controllers determine 'the purposes and the means of the processing of personal data' and must respect the privacy and data protection rights of those subjects whose personal data is entrusted to them.
Therefore, it is also useful to point out that, according to Article 2 of 2002/58/EC Directive, when we refer to *Traffic data* that means 'any data processed for the purpose of the conveyance of a *communication* on an electronic communications network or for the billing thereof', where *Communication* is 'any information exchanged or conveyed between a finite number of parties by means of a publicly available electronic communications service'. This does not include any information conveyed as part of a broadcasting service to the public over an electronic communications network except

to the extent that the information can be related to the identifiable subscriber or user receiving the information.

However, even if the Data Protection Act is mainly concerned with the disclosure of personal data outside the data controller's own boundaries, anyone who processes Personal Identifiable Information (PII) must comply with all the following principles as defined in the Data Protection Act:

- PII must be fairly and lawfully processed for specific, explicit and legitimate purposes;
- PII must be adequate, relevant and not excessive in relation to the purposes for which they are collected and/or further processed;
- PII must be accurate and up to date: every reasonable step must be taken to ensure that data which are inaccurate or incomplete, having regard to the purposes for which they were collected or for which they are further processed, are erased or rectified. Individuals have the right to access, rectify or erase PII related to them;
- PII must not be kept for longer than it is necessary for the purposes for which the data were collected or for which they are further processed;
- PII must be secured: appropriate technical and organizational measures must be taken to protect PII against accidental or unlawful destruction or accidental loss, alteration, unauthorized disclosure or access, in particular where the processing involves the transmission of data over a network, and against all other unlawful forms of processing;
- PII should not be shared between applications for different purposes;
- PII should not transferred to other countries without adequate protection;
- Traffic data relating to subscribers and users processed and stored by the provider of a public communications network or publicly available electronic communications service must be erased or made anonymous when it is no longer needed for the purpose of the transmission of a communication;

2.1.2 A way forward

Since the adoption of the Data Protection Directive in 1995, broad technological changes have taken place. The ability of organisations to collect, store and process personal data has incredibly increased, as the overall usage of online informations that are personally related.
Furthermore, another significant problem consists in the fact that the EU Member States have implemented the 1995 rules differently, resulting in divergences in enforcement, hence the definition of what personal data is and how this data must be protected can change depending on local jurisdiction. Discussions held at a panel on Legal Requirements and Issues in Network Traffic Data Protection [3] among U.S., European, and Japanese lawyers found that more work is needed to find appropriate inter-jurisdictional solutions for data sharing.
On January 2012 the European Commission has proposed a comprehensive reform [19] of the EU's 1995 data protection rules to strengthen online privacy rights and boost Europe's digital economy. A single law will get rid of the current fragmentation and costly administrative burdens.
Key changes in the reform include:

- *Rules uniformity*: a single set of rules on data protection, valid across the whole EU.
- *Administrative simplification*: unnecessary administrative requirements, such as notification requirements for companies, will be removed. Instead of the current obligation of all companies to notify all data protection activities to data protection supervisors the Regulation provides for increased responsibility and accountability for those processing personal data. For example, companies and organisations must notify the national supervisory authority of serious data breaches as soon as possible (if feasible within 24 hours).
- *Transparency*: wherever consent is required for data to be processed, it is clarified that it has to be given explicitly, rather than assumed.
- *Right to data portability*: people will have easier access to their own data and will be able to transfer personal data from one service provider to another more easily. EU rules must apply if personal data is handled abroad by companies that are active in the EU market and offer their services to EU citizens.
- *Right to be forgotten*: people will be able to delete their data if there are no legitimate grounds for retaining it.
- *Federation of authorities*: organisations will only have to deal with a single national data protection authority in the EU country where they have their main establishment. Likewise, people can refer to the data protection authority in their country, even when their data is processed by a company based outside the EU. Independent national data protection authorities will be strengthened so they can better enforce the EU rules at home. They will be empowered to fine companies that violate EU data protection rules.

The Commission's proposals will now be passed on to the European Parliament and EU Member States (meeting in the Council of Ministers) for discussion. They will take effect two years after they have been approved.

2.2 Network Monitoring Privacy Issues

The EU data protection framework is technologically neutral and it does not regulate specific monitoring and/or inspection techniques. More precisely, the ePrivacy Directive regulates privacy in the provision of electronic communication services in public networks (typically Internet access and telephony) and the Data Protection Directive regulates data processing in general.

Data sharing among international partners brings up the additional complication of heterogeneity in international data protection legislations.

Security policies might deny sharing because of a high risk of information disclosure. Even though collaboration might be useful, organizations have to carefully balance benefits with risks of potential damage. Even anonymized data may contain topological information, hint at particular services deployed, or reveal policies in place. In a competitive setting, overall statistics might reveal information about a participant's customer base. In summary, the situation is complex, so usually organizations refuse to exchange their collected data for fear of privacy breaches.

In fact, network traffic data contains very sensitive information about users, servers, apparatuses and networks. Among all the data collected from network monitoring, a simple information contained in the packet payload such as the IP addresses might be used to identify the sender or the receiver directly. Anyway, not all the IP addresses are necessarily linked to a person, so only in a limited amount of cases they can be used for direct identification.

Another important aspect to take into account is that all the attributes in a record (such as values of packet headers) can contribute to indirect identification, through e.g. usage patterns. For example timestamps can be used to identify the sender or the receiver in injection or fingerprinting attacks, whereas all invariant fields can be useful for linking or frequency attacks. See Sec. 2.2.2.1 for a deeper explanation about the main families of attacks.

A special attention should be payed for mobile data collection that can involve geolocation of users, long term identifiers such as IMEI (International Mobile Equipment Identity) and IMSI (International Mobile Subscriber Identity) [2], IP addresses, and in general fine-grained information about their behaviour and their applications usage, as described in the Opinion 13/2011 on these kind of services [20]. Since the application of network traffic inspection techniques can be either based on IP headers, which constitute traffic data, or based on deep packet inspection which also entails IP payloads and constitute communication data, then, in principle, the application of such techniques for purposes other than the conveyance of the service or security would be forbidden, unless a legitimate ground allows for the processing, such as consent (Article 5(1)). However, even if specific and informed consent is obtained from individuals to engage in content monitoring, it is particularly important to recall that the proportionality principle continues to apply.

In fact, in order to be compliant to data privacy in network monitoring, the law restricts the processing allowed on personal data and mandates anonymization for subsequent storage or before further processing.

However, even if payload is stripped away or anonymized, the IP addresses still allow the identification of users and hosts. The associated connection information allows the creation of precise communication profiles, e.g., containing information about who is communicating with whom and when, or which websites a person visits.

Another problem is related to preserving privacy to data mining practises, where database belongs to different organizational domains, especially when referred to large amount of stored data.

In a formal opinion published on October 2011 [25], Peter Hustinx explained that some of the practices employed by ISPs to manage traffic on their networks may be contrary to EU data protection and privacy laws. The document is particulary focused on traffic filtering and selection policies, that are applied exploiting users' traffic monitoring and inspection, but they are distinctive and useful also for other purposes of network monitoring or all those activities that could lead to a generalised monitoring of Internet usage. Furthermore, The European Data Protection Supervisor (EDPS), an independent supervisory authority devoted to protecting personal data and privacy and promoting good practice in the EU institutions and bodies, stated that new EU laws on net neutrality may be necessary to stop internet service providers (ISPs) from infringing individuals' data protection and privacy rights.

Anyway, even if any technological approach could solve completely data privacy concerns, in order to implement current law restriction from a pratical prespective, the state-of-the-art methods that enable cross-organizational collaboration even on sensitive network data can be summarized with these different paradigms: secure multi-party computation (MPC), data reduction, data perturbation and anonymization.

2.2.1 Multi-domain data mining

In a distributed network scenario, i.e. a multi-domain context, measurements and collected data can reside outside a single domain, so one entity must usually know the inputs from one or more

network participants. However, if nobody can be trusted enough to know all the inputs, privacy will become a primary concern.
Different privacy-preserving data mining problems were proposed by Lindell and Agrawal, respectively. In Lindell's paper [32], the problem is defined as: two parties, each having a private database, want to jointly conduct a data mining operation on the union of their two databases. The problem is how could these two parties accomplish this without disclosing their database to the other party, or any third party. In Agrawal's paper [1], the privacy-preserving data mining problem is defined as: Alice is allowed to conduct data mining operation on a private database owned by Bob, how could Bob prevent Alice from accessing precise information in individual data records, while Alice is still able to conduct the data mining operations? The solutions to these two similar problems are quite different: Lindell and Pinkas use secure multi-party computation protocols (see Sec. 2.2.1.1), while Agrawal uses the data perturbation method (described in Sec. 2.2.1.3).
Sections from Sec. 2.2.1.1 to Sec. 2.2.2.3 describe common tecniques preserve privacy of data when multiple parties mine into them in a multi-domain scenario.

2.2.1.1 Secure Multiparty Computation

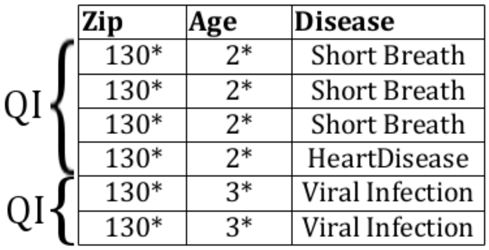
With Secure Multiparty Computation or MPC, sensitive data remains stored locally, e.g., in a local database. Using secret sharing techniques, random pieces of local data (shares) are distributed to a set of computation nodes. Together, they perform a distributed cryptographic protocol on the shares. In the end, only the final analysis result is made public and announced to input data providers.

2.2.1.2 Microdata Anonymization Models

Microdata refers to data (e.g., tables in a database) that contains unaggregated information about an individual, person, household, business or other entity. Publishing individual specific microdata has serious privacy implications. Privacy on microdata has been modeled with different generic statistical definitions mainly based on generalization and suppression of records. To counter linking attacks using quasi-identifiers (combinations of attributes within the data that can be used to identify individuals), Samarati and Sweeney [43, 41] proposed a model called k-anonymity.

k-Anonymity is a formal model of privacy created by L. Sweeney. The goal is to make each record indistiguishable from a defined number (k) of other records if attemps are made to identify the data. So in k-anonymity, each group of elements with similar non-sensitive *quasi-identifiers* must contain at least k elements, such that from learning values of quasi-identifiers, it is impossible to infer a specific sensitive attribute.
This model can be applied in a centralized mode, where existing k-anonymization algorithms assume a single party that has access to the entire original table, or in a distribuited mode, where we have N customers and a publisher (or miner) and each user/respondent/customer has her personal data, comprising a row of the table. The miner should not be able to associate sensitive information in the table with the corresponding customer.

Neverthless, the k-Anonymity model cannot protect against homogeneity and background knowledge attacks [33].



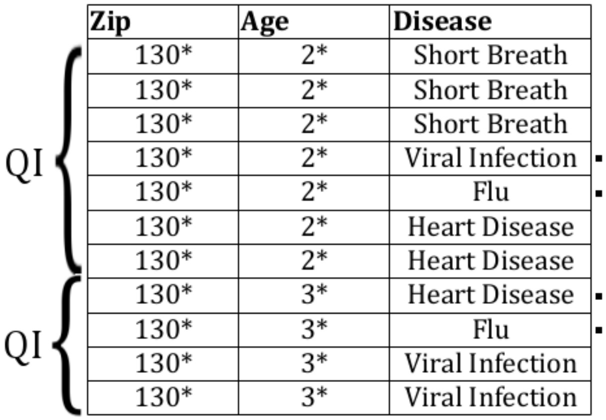
Zip	Age	Disease
130*	2*	Short Breath
130*	2*	Short Breath
130*	2*	Short Breath
130*	2*	HeartDisease
130*	3*	Viral Infection
130*	3*	Viral Infection

Figure 1: Example of 2-anonymous table

The notion of **l-Diversity** addressed these issues, assuring that each quasi-identifier has at least l well-presented values for each sensitive attribute. A popular interpretation of l-diversty is that, in each QI group, at most 1/l of the tuples should possess the same sensitive value.
A limitation of l-diversity consists in the fact that it doesn't prevent the probabilistic inference attacks (described in Section 2.2.2.1). For this reason, each equivalence class not only must have enough different sensitive values, but also the different sensitive values must be distributed evenly enough. Another important problem with l-diversity is that can be very difficult and unnecessary to achieve. Last but not least l-diversity does not consider semantic meanings of sensitive values, resulting insufficient to prevent attribute disclosure.

The **t-Closeness** model goes one step further and requires that the distribution of sensitive attributes within each equivalence class resembles the overall distribution of these attributes in the overall table. Privacy is measured by the information gain of an observer. An equivalence class is said to have t-closeness if the distance between the distribution of a sensitive attribute in this class and the distribution of the attribute in the whole table is no more than a threshold t.

With **m-uniqueness** a generalized table T*(j) is m-unique if each QI group contains at least m tuples, and all the tuples in the group have different sensitive values (l-diversity >= m-uniqueness, since some sensitive values could be not unique).



Zip	Age	Disease	
130*	2*	Short Breath	
130*	2*	Short Breath	
130*	2*	Short Breath	
130*	2*	Viral Infection	▪
130*	2*	Flu	▪
130*	2*	Heart Disease	
130*	2*	Heart Disease	
130*	3*	Heart Disease	▪
130*	3*	Flu	▪
130*	3*	Viral Infection	
130*	3*	Viral Infection	

Figure 2: Example of 3-diverse and 2-unique table. (Unique sensitive values marked with dots)

Further sophisticated approaches to microdata anonymization include the following models:

- **m-invariance**: a sequence of generalized tables T*(1),..., T*(n) (generalized tables generated from a time-variant dataset at instants 1,...,n) is m-invariant if the tables are m-unique, and if each individual has the same set of possible values for his sensitive fields, in every generalized table he is involved. That is that the key of m-invariance is that, if a tuple t (from the microdata) is published several times, all its generalized hosting groups must contain the same sensitive values.

- **p-safety**: anonymized data produced to meet privacy definition 'p' (e.g. k-anonymity, or k-anonymity + l-diversity) is considered safe if it has more then one potential original data that could have produced it.

2.2.1.3 Data Perturbation

Data perturbation techniques can be summarized as follows:

- data (and rank) swapping: swap the values of sensitive attributes among different rows;
- data sampling (or microaggregation), and rounding (useful for continuous variables), that avoids fine-grained characterization of data;
- global recoding: several categories are combined to form new less specific categories;
- additive random noise addition to values;
- multiplicative noise addition: additive noise with constant variance causes strong perturbation for small values, but weak perturbation for high values. A solution to this may be the multiplicative noise approach, which is independent from the size of the values;
- synthetic data generation through statistical models, that preserves the statistical properties of the original database;
- PRAM (Post-RAndomization Method): this method performs several perturbations. The values of certain records in the original file are changed to a different value according to a Markov matrix based probability mechanism. It performs noise addition, data suppression and data recoding
- MASSC: Micro Agglomeration, Substitution, Subsampling and Calibration. In this methodology, every record in the database is subject to modification or swapping. However, when applying this methodology, only a small random portion of the records are actually modified. [45].

Unfortunately, although all these techniques are valid privacy preserving approaches, they are mainly studied for microdata, e.g., medical records, whilst the complex semantics of network traffic make difficult a direct application of these methods to network data [12]. In fact, state-of-the-art approaches documented in various works [7, 11] raise severe concerns about the validity of anonymisation approaches in terms of privacy preservation. The ability to handle an enormous amount of anonymised data can be used to build linkability and inference-based attacks aimed at partially disclosing the protected information, hence it is impractical to apply these types of global constraints to voluminous streaming data. Therefore, privacy properties of network data anonymisation techniques have mainly been studied empirically by performing attacks against specific anonymisation schemes. Given this, it is better to avoid using anonymisation as a single or complete solution to the problem of privacy protection in network traces, and use it as part of an integrated approach.

2.2.2 Network traffic data anonymization and Pseudonymization

The most explicit reference to anonymisation in European data protection law is in Recital 26 of the European Data Protection Directive (95/46/EC) which:

- makes it clear that the principles of data protection shall not apply to data rendered anonymous in such a way that the data subject is no longer identifiable;
- recognises that a code of conduct, such as this one, can be a useful mean of guidance as to how personal data can be rendered anonymous; and
- is particularly important because it indicates clearly that the anonymisation of personal data is to be considered possible and that it can be used to provide important privacy safeguards for individuals.

Article 29 Data Protection Working Party defines the pseudonymisation as the "process of disguising identities. The aim of such a process is to be able to collect additional data relating to the same individual without having to know his identity. This is particularly relevant in the context of research and statistics.".

With anonymization, the process of collaboration is to first anonymize local data. Then, anonymized data are exchanged, either bilaterally or by using some sort of central (or distributed) repository. Data analyses are then run on the entirety of data instead of local data only.

On November 20, 2012, the UK's Information Commissioner's Office (ICO) issued the Code of Practice on data anonymization [36] trying to fill the gap between anonymization practise and references in the legislation.

The Code is developed within the framework of the Data Protection Act, 1998 (UK), and, therefore, should not be considered to be directly applicable outside the UK. However, the case studies and discussion of data anonymization techniques are useful reading for all organizations considering the conversion of PII to an anonymized form.

A specifical anonymization IETF Draft [4] has been proposed for IPFIX traffic data format and that can be definitely related to mPlane collected data. Anyway, whatever anonymisation technique will be adopted, it must be robust enough to protect users' data from malicious attacks. In general, those attacks can be grouped in four main families which are described in the following section.

2.2.2.1 Attack families

King et al. in [28] propose an extensive taxonomy of attacks against network flow sanitization methods; techniques fall into two main categories:

• *Fingerprinting*: with passive fingerprinting the attacker does not inject packets or influence the distribution of the trace in any way. He gains information about the victims strictly by inspecting the anonymized trace and from public information sources like search engines or web statistics. The general approach with passive fingerprinting is to try some sort of matching algorithm on the anonymized data set. Patterns that are either very similar to each other or well-known can be observed and then compared. Mostly the attacks work by means of behavioural profiling. Actions or attributes of hosts are analyzed and later used to recognize them. [44]

As an example each web site has a unique structure ("signature") and request/response pairs will

look similar in terms of size and response time (the time the web server takes to compute a dynamic script). Koukis et al. [31] obtained a signature for each web page to be identified and created a database of signatures. Those signatures were matched with information extracted from the trace and a similarity score for potential matches was computed. As a consequence they were able to reconstruct about 8% of the requests, which shows that matching can at least partially be successful. Once several web sites are discovered that way, it is possible to profile the web browsing behaviour of users through the anonymized logs.
Another example is the IP address truncation method: the attacker tries to map the IP addresses from the anonymized traces to the list of known original addresses. Such a list could, at least partially, be compiled by scanning of active hosts and using public information about well-known sites. Both in a worst case scenario where the attacker has exact information about the objects (e.g. hosts, servers [10]) and in a better scenario where the attacker has only the statistical ability to distinguish objects [6], this anonymization technique has failed.

• *Injection*: the adversary injects a sequence of flows in the network to be logged, that are easily recognized due to their specific characteristics; e.g., marked with uncommon TCP flags, or following particular patterns. This is analogous to a known-plaintext attack against a cryptosystem: by causing known traffic to be captured in the trace for subsequent anonymization and publication, the attacker has knowledge of some raw data within the trace [8].
For example, it has been demonstrated how an attacker capable of injecting traffic into a network can reverse any IP address anonymization technique based on permutation, including hashing, enumeration, random permutation, or (partial) prefix-preserving permutation. Permutations do not actually remove information from the data, but they just only transform it. Thus, in worst case, it is always possible to reverse the permutation and recover then the original information.
Another pratical case study can be represented by the well-known Crypto-PAn tool, which is currently incorporated within several network flow collector tools. Crypto-PAn is a sanitization tool for network flows that encrypts IP addresses in a prefix-preserving manner. A malicious user, which acts inside the monitored network, can inject bogus and easily detectable flows in order to understand how one IP address is mapped to its encrypted value inside the obfuscated flow set. Thanks to the fact that each IP address's octet (8 bit length integer value) is always mapped to the same encrypted value, an adversary can obtain the encrypted version of each one of the 255 possible octets values injecting a small number of bogus flows. Appropriate defense techniques adopt cryptographic primitives to hide real IP addresses, and obfuscation of flow fields' values. They provide strong confidentiality protection even when the adversary can reconstruct the mapping between an IP address and its encrypted value, possibly as a result of injection attacks.

Other typologies of attacks are the linking, inference and frequency attacks:

• *Linking*: the attacker tries to link publicly available information to partial informations acquired from anonymized traces, in order to re-identify the data subjects.

• *Probabilistic Inference*: with this technique, more sophisticated than the "trivial" linking attack, the adversary exploits data mining techniques that consist in correlating information from the anonymized traces and from publicly available data in order to infer additional information from the anonymized set, through the comparison of the feature distributions of the two data sets.

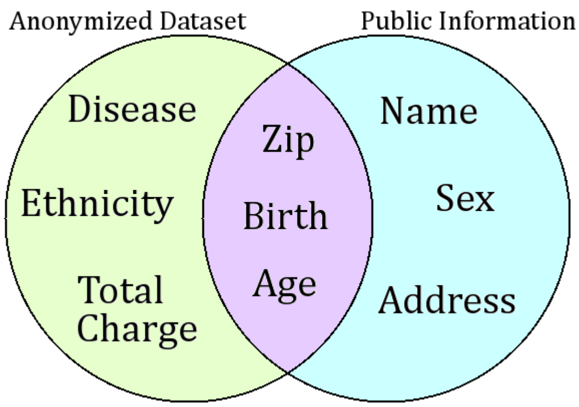


Figure 3: Example of Linking Attack

• *Frequency analysis*: cryptographic techniques that, if applied to anonymized traces, study the frequency of occurrence of determined sensitive values. For example, IP addresses of popular sites can be inferred from their high frequency of occurrence in an anonymized trace.

Sects. 2.2.2.2 and Sec. 2.2.2.3 respectively describe common tecniques to anonymize and de-anonymize network data.

2.2.2.2 Obfuscating Techniques

Obfuscating tecniques can be categorized for two main tipologies of network data: i) IP address and ii) secondary fields, as port numbers or timestamps.

IP Addresses

The most commonly employed IP address anonymization techniques are blackmarking, truncation, random permutation, prefix-preserving permutation, and partial prefix-preserving permutation.

- **Blackmarking** is the simplest of all studied anonymization techniques. It replaces all IP addresses in a trace with the same value.

- **Truncation** replaces a number of least significant bits of an IP address with 0. Thus, truncating 8 bits would replace an IP address with its corresponding class C network address

- **Random permutation** translates IP addresses using a random permutation that does not preserve the prefix structure. Since permutation creates a one-to-one mapping, the number of distinct IP addresses is the same. A special case of random permutation is the renumbering of IP addresses.

- **Partial prefix-preserving permutation** As proposed in [27], partial prefix-preserving permutation (PPP) permutes the host and network part of IP addresses independently. It preserves the prefix structure in a trace at one specific prefix length p, and at the level of IP addresses.

- **Prefix-preserving permutation** permutes IP addresses so that two addresses sharing a common real prefix of any length, also share an anonymized prefix of equal length.

Secondary Fields

In this section, we discuss techniques applied to fields other than IP addresses, such as port numbers or timestamps. Of course, generic methods like blackmarking or random permutation can be applied to many fields and also work in combination with other techniques. Some of the techniques described below are used for obscuring injected fingerprints. However, we do not evaluate these techniques in the same way as we do for IP address anonymization techniques.

- **Hashing** of values is a special form of random permutation. It is also applicable to larger fields, e.g., the payload. If seeded hash functions are used and the seed is kept secret, the adversary is kept from brute-forcing values. At the same time, it allows the owner of the data to retain mapping information without storing large mapping tables. Although usually rare, hashing can introduce collisions.
- **Classification** simply partitions the range of possible values into pre-defined classes. For instance, a popular way of anonymizing port numbers is to classify them into well-known (< 1024) and dynamic ports (≥ 1024).
- **Time Unit Annihilation** Timing fields can be broken down into year, month, day, minute, seconds, and milliseconds parts. The annihilation of time units corresponds to blackmarking of a specific part of timing, e.g., milliseconds and seconds can be annihilated to avoid the resolution of records below minute accuracy. Of course, this could be combined with classification, e.g., to bucketize all timestamps in 10 minute slots. Similarly, the year and month part could be annihilated to hide when a trace was collected.
- **Enumeration** sorts all occurring records by value and replaces the values by the rank of their record. It is usually applied to timestamp fields to preserve the logical order of events but remove the absolute timing information. In that case, the “rank” has to be mapped back to a valid timestamp. For instance, this could be done by choosing a random offset and for each rank increment the timestamp by one millisecond.
- **Random Shift** Sometimes it may be important to preserve the distances between values but hide their absolute values, for instance to assess how far apart two events are temporally. For this, a random shift is performed by adding a random offset to the values.

2.2.2.3 Re-identification or De-anonymization

Specifically, European law defines as personal data any data identifying a subject either directly or indirectly, i.e. through the use of additional information in possession of third parties. To this category belong, e.g., IP addresses and user profiles. The law restricts the processing allowed on these data and mandates anonymization for subsequent storage or before further processing (e.g., research). Note that anonymized data as defined by the law must prevent identification of subjects both directly and indirectly. Therefore, current anonymization techniques applied alone do not provide “anonymization” in the legal sense. Consequently, lawyers start arguing that legislation has to abandon the concept of PII (personally identifying information) and move on to more holistic definitions, considering a series of factors in context-specific solutions.

In fact, anonymization involves obfuscating sensitive PII by replacing it completely or partially with synthetic identifiers, or using aggregation or statistical techniques intended to break the connection

between the subjects and reference identifiers. Re-identification or de-anonymization, conversely, involves reversing data masks to link an obfuscated identifier with its associated subject. Shared anonymized data poses a misuse risk because it is vulnerable to reidentification attacks that make use of increasingly available public or private information beyond the knowledge or control of the original or intermediate data provider.
Besides anonymizing data locally maintained by a data holder, it is also important to anonymize data distributed through different interconnected parties. Reidentification combines datasets that were meant to be kept apart, and in doing so, gains power through accretion: every successful reidentification, even one that reveals seemingly nonsensitive data like movie ratings, breeds future successes. Second, regulators can protect privacy in the face of easy reidentification only at great cost. Because the utility and privacy of data are intrinsically connected, no regulation can increase data privacy without also decreasing data utility. No useful database can ever be perfectly anonymous, and as the utility of data increases, the privacy decreases [37].

Early techniques for network flow obfuscation were based on the anonymization of source and destination IP addresses. However, those techniques proved to be ineffective, since an adversary might be able to re-identify message source and destination by other values in a network flow, or in a sequence of flows (see, e.g., [5, 13, 46, 47]).

3 Data Protection in mPlane

In mPlane infrastructure the data collected by the probes are exposed to different threats (access control, data integrity, data privacy, etc.), so it is necessary to split the description of the security countermeasures and mechanisms in different separated chapters. How the privacy-by-design principle is applied in mPlane will be described in this chapter.

3.1 Threat Model

The most important and data protection threats and requirements in the mPlane infrastructure can be summarized as the following generic list:

- Privacy-sensitive information exceeds payload and spans across various protocol headers and other communication metadata
- Personal information can be inferred and extracted using advanced processing techniques (statistical analysis, fingerprinting, etc.)
- Specific regulations govern the underlying services and data
- Very high data rates and consequent performance requirements
- Distributed and cooperative nature of operations and infrastructures (intra-domain and inter-domain)

In more detail, from a privacy perspective, the network traffic data that can be considered PII are primarily end-users' IP addresses collected by probes, but can be also represented by geolocalization info and IMSI/IMEI codes for mobile users. This information can be aggregated in some way depending on the use case scenario and then sent to a supervisor or to a repository, such a database that exposes data that can be accessed by any mPlane authorized client. So basically there are two different privacy-aware threats: probe's data collection threats and data stored in the repository threats. Probe's data collection threats are the misuse of user related information or network profiles that reveals user behaviors, preferences or interests, knowledge that attackers or advertisers can then exploit. Another threat can be represented by the inference misuse risk, that involves synthesizing first or second-order PII to draw (possibly false and damaging) implications about a persons' behavior or identity. Furthermore, IP addresses may also represent servers or gateways of a company, so statistics about these important network infrastructure elements are likely to be protected by internal network security policies. But also statistics about entire subnets are sensitive, especially if these subnets match individual customers of an ISP.
Otherwise, threats against the data stored in the repository include re-identification using data linkage or data mining algorithms and data disclosure to unauthorized parties. *Linked information* is information about or related to an individual that is logically associated with other information about that individual. In contrast, *linkable information* is information about or related to an individual for which there is a possibility of logical association with other information about the individual. For example, if two databases contain different PII elements, then someone with access to both databases may be able to link the information from the two databases and identify individuals, as well as access additional information about or relating to the individuals. If the secondary information source is present on the same system or in a closely-related system and does not have

security controls that effectively segregate the information sources, then the data is considered as linked. If the secondary information source is maintained more remotely, such as in an unrelated system within the organization, available in public records, or otherwise readily obtainable (e.g., internet search engine), then the data is considered linkable.

3.2 Security Model

In order to be really effective, the network data collected in mPlane infrastructure should be protected through a combination of measures, including operational safeguards, privacy specific safeguards, and security controls as suggested by the NIST guide to protect PII [34]. In the specific mPlane context the main controls to be performed can be listed as:

- *Minimizing the Use, Collection, and Retention of PII*. The practice of minimizing the use, collection, and retention of PII is a basic privacy, but effective principle. By limiting PII collections to the least amount necessary to conduct the measurements, the organization may limit potential negative consequences in case of a data breach involving PII. In practice data reduction on the measurement device improves scalability and reduces sensitivity of collected data.

- *Creating Policies and Procedures*. Organizations should develop comprehensive policies and procedures to protect the confidentiality of PII.

- *Conducting Training*. Organizations should reduce the possibility that PII will be accessed, used, or disclosed inappropriately by requiring that all individuals receive appropriate training before being granted access to systems containing PII.

- *Using Access Control Enforcement*. Organizations can control physical and remote access to PII through access control policies and mechanisms (e.g., access control lists) as described in Chapter 5.

- *Providing Transmission Confidentiality*. Organizations can protect the confidentiality of transmitted PII using secured channels or encrypted messages as described in Chapter 4

- *Separation of Duties*. Organizations can enforce separation of duties for tasks involving access to PII. For example, the users of de-identified PII data would not also be in roles that allow them to access the information needed to re-identify the records.

- *Least Privilege*. Organizations can enforce the most restrictive set of rights/privileges or accesses needed by users (or processes acting on behalf of users) for the performance of specified tasks. Concerning PII, the organization can ensure that users who must access records containing PII only have access to the minimum amount of PII, along with only those privileges (e.g., read, write, execute) that are necessary to perform their job duties.

- *Identification and Authentication*. Organizational users should be uniquely identified and authenticated before accessing PII as described in Chapter 5.

- *Protecting Data Disclosure*. Data disclosure must be limited also on anonymized data collections. Exporters or publishers of anonymized data must take care that the applied anonymization technique is appropriate for the data source, the purpose, and the risk of deanonymization of a given application.

- *Protection of Information at Rest*. Organizations can protect the confidentiality of PII at rest, which refers to information stored on a secondary storage device, such as a hard drive or backup tape. This is usually accomplished by encrypting the stored information.
- *Information System Monitoring*. Organizations can employ automated tools to monitor PII internally or at network boundaries for unusual or suspicious transfers or events. An example is the use of data loss prevention technologies.
- *Auditing Events*. Organizations can monitor events that affect the confidentiality of PII, such as unauthorized access to PII. Organizations can regularly review and analyze information system audit records for indications of inappropriate or unusual activity affecting PII, investigate suspicious activity or suspected violations, report findings to appropriate authorities.

3.2.1 Data protection

To provide lawful compliance PII must be protected using at least one of these main countermeasures:

Aggregated data
Aggregated data are information averaged or summed over a adequately large number of clients, so it will not include individual's PII. For this reason then it can be used for any purpose (e.g., monitoring, marketing, etc) and has the great benefit that can be exported to authorized entities. Furthermore, the access to this aggregated data doesn't need strong encryption and authentication if indirect identification using cross-processing is prevented.

Timed data
Individual data can be collected and stored only for troubleshooting and for a very limited period, then the collected information must be completely erased after the troubleshooting process is finished. A repository containing detailed data must use encryption and access must occurr only with strong authentication (e.g, using smart-card, biometry, OTP, etc.). Each operation on this data must be authorized and traced in order to have a history log, that must be stored on an external system, in order to provide Separation of Duties (SoD).

Anonymization
The main information that must be anonymized is the source IP address of the user connection that is monitored. Anyway, if destination IP address belongs to the same ISP domain of the sender IP and it is related to an user (e.g. peer-to-peer), then both source and destination addresses must be anonymized. The anonymization process must be sufficiently fine-grained, in order to be able to implement white or black lists. For example if a field of a "record" (e.g. a log trace line, database record, etc.) requires anonymization, it should be possible to not anonymize the remaining values of the record. Identification of customers must be possible only for authorized people with a personal account and, as for aggregated data, strong authentication is not needed only if indirect identification is prevented. Although, because the main threat for anonymization is user re-identification using fingerprint or injection attacks, it must be possible to anonymize PII using tokens with different configurable lifetimes (e.g., short, medium and long period). For example the destination IP address could change anonymization token every week, whereas the source IP can use a different method every hour. There are many different techniques and tools for anonymization of network traffic data, so we introduce the most important ones with a special focus on IP address anonymiza-

tion.
Many simple tools like TCPdpriv [35], CryptoPAn [29], Ipsumdump [30] and Ip2anonip [38] come with predefined options for anonymizing certain fields in specific data formats, e.g., packet traces. Anonymization frameworks like and Anontool [21] provide a comprehensive collection of anonymization techniques that can be flexibly applied to various fields and allow the definition of fine-grained anonymization policies.

3.2.2 Repository Data Protection

If the data are not obtained in real-time, the activity is referred to as access to retained data (RD), so it can be considered any data stored in a mPlane repository.
The main privacy threats for the repository are represented by re-identification and data disclosure. To prevent this threat, data administrators may use several techniques — interactive techniques, aggregation, access controls, and audit trails — to share their data with a reduced risk of reidentification. Researchers have developed a few techniques that protect privacy much better than the traditional release-and-forget techniques. These work by relaxing either the release or the forget requirement. For example, some data administrators never release raw data, releasing only aggregated statistics instead.
Similarly, some researchers [15] favor interactive techniques. With these techniques, it may be possible to retrieve some information without ever releasing the underlying data. In most cases, reidentifiers will find much more difficult to link answers like these to identities than if they had access to the raw underlying data.
Finally, just as these techniques relax the requirement of release, other techniques work by monitoring what happens to data after release: they refuse to forget. These techniques involve the use of access controls and audit trails and borrow from computer security research. Using these techniques, data administrators release their data but only after first protecting it with software that limits access and tracks usage.

3.2.3 Inter-Domain Data Protection

mPlane will create an infrastructure spanning across different administrative domains, e.g., network operators, that need to collaborate while preserving the confidentiality of their business assets and the privacy of their users.
In this kind of infrastructure, a component - e.g., a reasoner - may want to access some information that can only be retrieved from another domain. This kind of interaction must be strictly regulated, since the disclosure of sensitive information raises severe concerns in terms of privacy and confidentiality.
User data privacy can be preserved using a domain data ownership separation: the mPlane architecture can be logically divided into layers, where each layer corresponds to an entity. Each layer is actually a domain, hence the data collected in a layer belong to that layer and are managed by relative owner. However, the data are not segregated in each layer, but can be disclosed to upper (and wider) layers, demanding the data ownership and management to a new layer administrator. For instance, in a mobile scenario in which the device is connected to a wifi network: the mobile device is the smallest layer, and data collected by the device belong to the device owner, that has complete control on them. If the owner gives his consent for the disclosure of data, all the collected information (or only a subset of it) is exposed to the control of the wifi administrator, that becomes

the new owner of the data. The administrator can in turn disclose his data to his internet provider, and the ISP can send it to a Service/CDN layer.

In this way, upper layers can only see what the lower layers decide to expose, based on the idea that the smaller is the layer, the more confidential and personal is the data, needing a stronger level of privacy. The level of privacy that each domain owner wants to achieve may be specified through the agreement, permitting fine-grained authorization, coming with a cost of low flexibility, beacuse of the static way in which the agreement is stipulated.

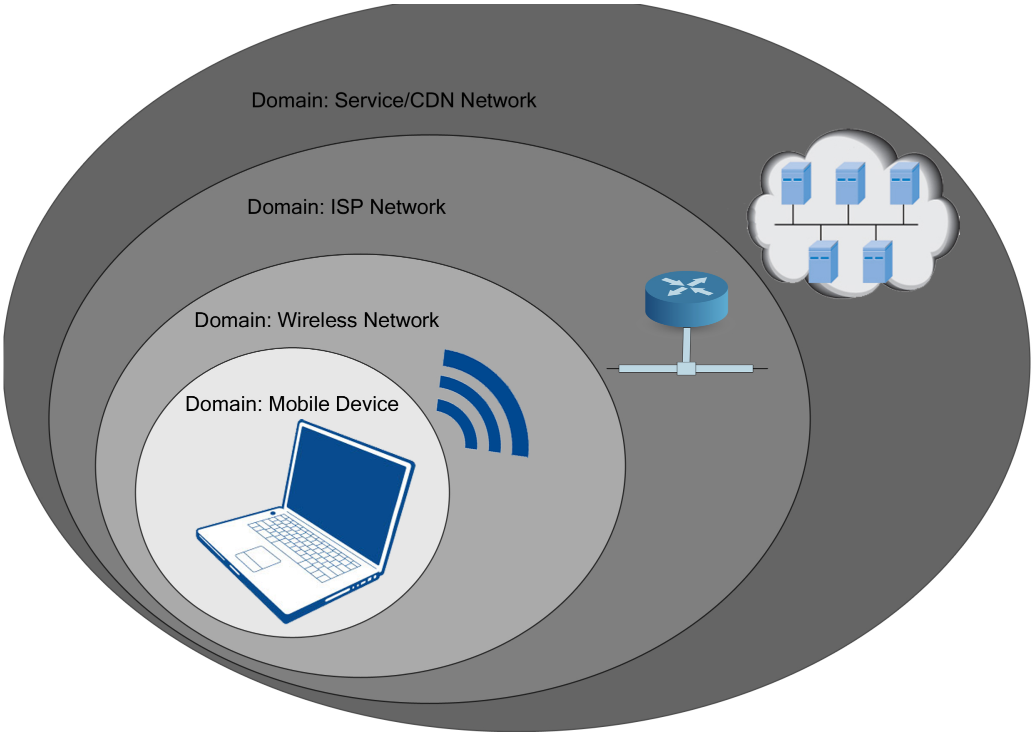


Figure 4: Cross-domain Data ownership

3.2.3.1 Business Confidentiality

In the context of the mPlane multi-domain infrastructure, approaches for anonymous communications can be used for ensuring the confidentiality of business information. The supervisor-to-supervisor approach allows the access from a domain to another as a whole mPlane component, without the need to disclosure any internal information. As a mPlane component the frontier supervisor will expose only the capabilities that are allowed in the specific inter-domain collaboration and will implement the necessary access controls for trusted domains.

4 mPlane Security Layer

4.1 Threat Model

The large-scale measurement architecture of mPlane has to prevent third parties' use of the probes in bot-nets or for other malicious purposes. For example a malicious third party could force a probe to initiate measurement traffic to victim hosts rather than mPlane data collection servers, whilst only successfully authenticated and authorized entities should issue control commands and query for traffic data.
Furthermore, an attacker (e.g. a malicious competitor) can also tamper with measured information or send large amount of bogus data to cause inefficiency or at least a full fledged Denial of Service (DoS) to the data collector server, that can be either a supervisor or a repository. Due to the strongly collaboration-oriented trait of the mPlane architecture, a DoS could represent a serious attack to the whole iterative measurement process.
Thus, the security requirements can be summarized with the following list:

- A mPlane component must be able to authenticate the correspondent that has to communicate to (e.g., the probe must authenticate the supervisor/repository to which is sending the measurement data)
- A mPlane component must be able to authenticate and authorize the corresponent that is requesting a specific measurement or export of data
- It must be possible to protect the integrity and confidentiality of the measurement data exchanged between two mPlane components (both for synchronous and asynchronous communication)
- It must be possible to protect against involuntary or malicious DoS attacks using authorization and/or quotas on number of concurrent connections and amount of data
- It must be possible to protect access between two different domain (intra and inter-domain)
- End-user probes must be authenticated and authorized for measurement

4.2 Security Model

4.2.1 Data transfer security

There are two basic communications levels at which security can be applied: end-to-end and link security.
With *end-to-end security* (sometimes called off-line security), a message is encrypted and/or signed when it is transmitted and is decrypted and/or verified when it is received.
The network may not even need to be aware that the message is encrypted. This type of security may sometimes be selected as an option by the client, depending on the correspondent or the confidentiality level of the message itself. Another specific property of this approach is that the message remains encrypted through the entire communications process, from start to finish. This has the

advantage of not depending on secure waypoints at every stage in the message path.
With *link security* (sometimes called online security), a message is encrypted when it is transmitted, but is decrypted and then encrypted again each time it passes through a network communications node. The message may therefore be encrypted, decrypted, and reencrypted a number of times during the communications process, and the message is exposed within each node, because with link encryption, the encryption is performed just before the message is physically transmitted. Anyway the huge benefit on this approach is that encryption is typically invisible to the client, because it is simply part of the transmission process.
In the mPlane infrastructure the connection between components is protocol agnostic, so any communication protocol can be used indifferently. Thus, the security between two mPlane components can be provided in different ways depending on the specific type of protocol that better fits to the operating environment requirements. Nevertheless, the protocols that will be provided as default implementations will be the Secure SHell (SSH) and the HTTP over SSL/TLS protocol (also known as HTTPS) [39]. In fact, for its wide support and popularity the HTTPS will be the default transfer protocol that will be implemented for communication, whereas the SSH protocol will be very helpful in test or small environments for its handiness. The HTTP protocol will be also supported for developement and testing purposes or for those scenarios where channel security is already provided.

4.2.2 Identities Lifecycle Management

4.2.2.1 SSH and RSA keys

SSH exploits public key cryptography using a one-to-one non hierarchical trust relationship. The public key must be distribuited to each trusted pair, that will add it as trusted in a explicit whitelist. Therefore, the weakest link in the SSH identities' lifecycle management is represented by the key distribution process, because each client's public key must be distribuited in each component that will act as a server for those clients. This means that is necessary to install in the supervisor and in the repository all authorized probes and administrative clients' public keys. Otherwise, if a user/component has to be de-provisioned from accessing the architecture, then the corrispondent public key should be removed from all the whitelist in which has been installed. However, the most important benefit of SSH is that can be very comfortable to manage in test and/or small environments, where the number of clients is pretty small and the management of a Public Key Infrastructure (PKI) can represent a useless and excessive effort.

4.2.2.2 TLS and Certificates

In cryptography, a public key certificate (or identity certificate) is a short document that can be used to verify that a public key belongs to an individual. The certificate uses a digital signature to bind together a public key with information such as the name of a person, an organization, or address information. In fact, a digital certificate usually follows the ITU-T X.509 standard as also specified in [24], which includes the following information:

- The public key being signed by a trusted authority.
- A name, which can refer to a person, a computer, or an organization.

- A validity period during which the certificate should be considered to be reliable.
- The Internet address (URL) of a revocation center that can be consulted to determine if the certificate has been declared to be invalid.
- The key usage purpose (e.g., for digital signature, key encipherment, etc.)

Certificates make it possible to use public-key cryptography on a large scale. To securely exchange secret keys between network users becomes impractical as the number of users increases beyond a few (the key distribution problem). The system used to exchange keys as networks scale in size is called the Public Key Infrastructure (PKI). If a user wants to establish a trusted connection using SSL or TLS secure channel, he needs only to install the root Certificate Authority (CA) digital certificate that issued the any correspondant's user certificate.

Thus, a key point for the whole security of SSL/TLS as used in practice is strictly bound with the security of the certificate hierarchy used to authenticate the servers, so it is absolutely crucial to ensure the correct distribution and installation of the root CA (or trust anchor) on any mPlane component that will use HTTPS.

For a correct use of PKI, another important aspect is the management and verification of certificate validity. X.509 states that it is a CA's responsibility to maintain "a time-stamped list of the certificates it issued which have been revoked.“ Two can be the reasons for a CA to revoke a certificate: suspected compromise of a private component (invalidating the corresponding public component) or change of user affiliation (invalidating the Distinguished Name). A revocation of a valid certificate should not be used as an authorization revocation process (e.g. to deny access to an employee that changed his/her role), but as a identity management de-provisioning task (e.g. a component has been dismissed, an employee left the company, etc.).

5 Access Control Layer

This chapter covers the most important concepts and mechanisms concerning the access control layer, whose description is useful for the mPlane access control as described in the next chapter. Access control generally suggests that there is an active user and/or application process, with a desire to read or modify a data object (file, database, etc). For simplicity, we will hereafter refer to an entity as a user and a data object as a file. Access control typically involves two steps: authentication and authorization. In order to authenticate an active user, the distributed system needs some way of determining that a user is in who he/she claims to be. A password is an example of a standard authentication method. On the other hand, authorization to access a file relies on a set of rules that are specified formally and are used to decide which users have the permissions required to access a resource.

5.1 Access Control Models

An important requirement of any information management system is to protect data and resources against unauthorized disclosure (confidentiality) and unauthorized or improper modifications (integrity), while at the same time ensuring their availability to legitimate users (no denials-of-service). Enforcing protection therefore requires an access control, so that every access to the system and its resources is controlled and that all and only authorized accesses can take place.
The development of an access control system requires the definition of the regulations according to which access is to be controlled. The development process is usually carried out with a multi-phase approach based on the following concepts:

- Security policy: it defines the (high-level) rules according to which access control must be regulated.
- Security model: it provides a formal representation of the access control security policy and its working. The formalization allows the proof of properties on the security provided by the access control system being designed.
- Security mechanism: it defines the low level (software and hardware) functions that implement the controls imposed by the policy and formally stated in the model.

The three concepts above correspond to a conceptual separation between different levels of abstraction of the design, and provides the advantages of multi-phase software development. In particular, the separation between policies and mechanisms introduces an independence between protection requirements to be enforced on the one side, and mechanisms enforcing them on the other. The access control mechanism should work as a reference monitor, that is, a trusted component intercepting each and every request to the system.
A lot of different security models are currently developed to implement authorization security policies, so this section will cover the most important or widely adopted ones, describing the main carateristichs and drawbacks.

5.1.1 DAC

The Discretionary Access Control (DAC) is a user-centric access control model in the sense that a file owner determines the permissions that are assigned to other users requiring access to the resource. There is no central control so this model is easy to implement in a distributed applications on the Web. Using a DAC mechanism allows users control over the access rights to their resources without the necessity of complying with a set of pre-specified rules. When these rights are managed correctly, only those users specified by the file owner may have some combination of read, write, execute, etc. permissions (privileges).
The most important limitation of DAC are:

- Global policy: DAC let users to decide the access control policies on their data, regardless of whether those policies are consistent with the global policies. Therefore, if there is a global policy, DAC has trouble to ensure consistency.
- Information flow: Discretionary policies do not enforce any control on the flow of information once this information is acquired by a process, makes it possible for processes to leak information to users not allowed to.
- Malicious software: DAC policies can be easily changed by owner, so a malicious program (e.g., a downloaded untrustworthy probe) running by the owner can change DAC policies on behalf of the owner.
- Flawed software: Similarly to the previous item, flawed software can be “instructed” by attackers to change its DAC policies.

5.1.2 MAC

The Mandatory Access Control (MAC) model counters control flow threats by controlling access centrally. A system-wide policy decrees who is allowed to have access and an individual user cannot alter that access. This procedure allows the system to use the concept of information flow control to provide additional security.
Information flow control allows the access control system to monitor the ways and types of information that are propagated from one user to another. A security system that implements information flow control typically classifies users into security classes and the resources are tagged with security labels that are used to restrict access to authorized users. All the valid channels along which information can flow between the classes are regulated by a central authority or security administrator.

5.1.3 RBAC

In the RBAC method, the role of a requester is the key determinant for access. The idea of role-based access control emerged as a sort of middle ground between mandatory and discretionary access control because on the one hand, discretionary access control was considered to be too flexible and on the other hand, mandatory access control to be too rigid for concrete implementation purposes.

In the role-based access control model, a role is typically a job function or authorization level that gives a user certain privileges with respect to a resource and these privileges can be formulated in high level or low level languages. RBAC models are more flexible than their discretionary and mandatory counterparts because users can be assigned several roles and a role can be associated with several users. Unlike the access control lists (ACLs) used in traditional DAC approaches to access control, RBAC assigns permissions to specific operations with a specific meaning within an organization, rather than to low level resources. For example, an ACL could be used to grant or deny a user modification access to a particular file, but it does not specify the ways in which the file could be modified. By contrast, with the RBAC approach, access privileges are handled by assigning permissions in a way that is meaningful, because every operation has a specific pre-defined meaning within the application.

In an RBAC model, a user's role is not mutually exclusive of other roles for which the user already possesses membership. The operations and roles can be subject to organizational policies or constraints and, when operations overlap, hierarchies of roles are established.

The challenge of RBAC is the contention between strong security and easier administration. On the one hand, for stronger security, it is better for each role to be more granular, thus having multiple roles per user. On the other hand, for easier administration, it is better to have fewer roles to manage. Organizations need to comply with privacy and other regulatory mandates and to improve enforcement of security policies while lowering overall risk and administrative costs.

The role-based approach has several advantages. Some of these are:

- **Authorization management** Role-based policies benefit from a logical independence in specifying user authorizations by breaking this task into two parts: i) assignement of roles to users, and ii) assignement of authorizations to access objects to roles. This greatly simplifies the management of the security policy: when a new user joins the organization, the administrator only needs to grant her the roles corresponding to her job; when a user's job changes, the administrator simply has to change the roles associated with that user; when a new application or task is added to the system, the administrator needs only to decide which roles are permitted to execute it.

- **Hierarchical roles** In many applications there is a natural hierarchy of roles, based on the familiar principles of generalization and specialization. The role hierarchy can be exploited for authorization implication. For instance, authorizations granted to roles can be propagated to their specializations (e.g., the probes' administration role can be allowed all accesses granted to administration staff). Anyway propagating all authorizations is contrary to the least privilege principle.

- **Least privilege** Roles allow a user to sign on with the least privilege required for the particular task he needs to perform. This minimizes the danger of damage due to inadvertent errors or intruders masquerading as legitimate users.

- **Separation of duties** Separation of duties refer to the principle that no user should be given enough privileges to misuse the system on their own. For instance, the person authorizing a measurement should not be the same person who can perform it. Separation of duties can be enforced either statically (by defining roles which cannot be executed by the same user) or dynamically (by enforcing the control at access time). An example of dynamic separation of duty restriction is the two-person rule. The user authorized to execute a delayed-measurement operation can be any authorized user, whereas the user that can use the measurement's receipt can be any authorized user equal to the previous one.

- **Constraints enforcement** Roles provide a basis for the specification and enforcement of further protection requirements that real world policies may need to express. For instance, cardinality constraints can be specified, that restrict the number of users allowed to activate a role or the number of roles allowed to exercise a given privilege. The constraints can also be dynamic, that is, be imposed on roles activation rather than on their assignment. For instance, while several users may be allowed to activate role chair, a further constraint can require that at most one user at a time can activate it.

5.1.4 RuBAC

In the commercial world, RBAC is the de facto access control implementation at the enterprise level because RBAC is what most solutions support. One obstacle to RBAC is the initial complexity involved in setting it up, a process known as role engineering—defining roles, user-role assignments, permission-role assignments, and role hierarchies. Rule-based approaches on user and resource attributes can be adopted as a way of avoiding this obstacle. Under Rules Based Access Control (RuBAC), access is allowed or denied to resource objects based on a set of rules defined by a system administrator. As with DAC, access properties are stored in ACLs associated with each resource object. When a particular account or group attempts to access a resource, the access control checks the rules contained in the ACL for that object. Examples of Rules Based Access Control include situations such as permitting access for an account or group to a network connection at certain hours of the day or days of the week. As with MAC, access control cannot be changed by users.

5.1.5 MLS

The multilevel security (MLS) model is essentially a special case of the MAC model implemented for different contexts or scenarios. In the MLS model, a security goal is set and information flow is regulated in a way that enforces the objectives determined by the security goal. Practical implementations of security schemes based on the MLS concept include the Bell-Lapadula (BLP), Biba Integrity Model, Chinese Wall, and Clark-Wilson models.

5.1.6 CAC

Hierarchical cryptographic access control (CAC) schemes emerged in an attempt to design MLS models that are more general and capable of providing security in different contexts without requiring extensive changes to the fundamental architecture. For instance, in situations that require data outsourcing CAC schemes are useful because the data can be double encrypted to prevent a service provider from viewing the information but yet be able to run queries or other operations on the data and return a result to a user who can decrypt the data using the keys in their possession. CAC schemes are typically modeled in the form of a partially ordered set (poset) of security classes that represent groups of users requesting access to a portion of the data on the system. Cryptographic keys for the various user groups requiring access to part of the shared data in the system are defined by classifying users into a number of disjoint security groups.

5.1.7 ABAC

ABAC denotes access control based on attributes and policies. Attributes are distinguishable characteristics of users or resources, conditions defined by an authority, or aspects of the environment, and policies specify how to use attributes to determine whether to grant or deny an access request. Whatever access control can be defined with DAC or RBAC can also be defined with ABAC. In addition, the ABAC method can provide more complex access control than can be accomplished with DAC or RBAC. This approach might be more flexible than RBAC because it does not require separate roles for relevant sets of subject attributes, and rules can be implemented quickly to accommodate changing needs. The trade-off for this flexibility is the complexity of cases that must be considered: for n boolean attributes or conditions using attributes, there are 2n̂ n possible combinations. Thus, the ability to define more sophisticated access control comes with the additional administrative and managerial burdens imposed by complexity.

5.1.8 CBAC

In those scenarios in which the communication is established from clients that change in a very fast and unpredictable way (e.g. mobile or smartphone users), then a security administrator is not able to specify authorizations for all those users with respect to their identity. Thus, the traditional separation between authentication and authorization cannot be applied in this context. A possible solution to this problem is represented by Certificate-Based Access Control (CBAC) [42], with the use of digital certificates (or credentials), representing statements certified by given entities (e.g., certification authorities), which can be used to establish properties of their holder (such as identity, accreditation, or authorizations).

5.2 Access Control Constraints

There are some contraints that usually are specified to controlling access to resources in a context-based way. The most used constraints are:

- *Temporal* Authorizations could have associated a validity specified by a temporal expression identifying the instants in which the authorization applies. The temporal expression is formed by a periodic expression (e.g., 9 a.m. to 5 p.m. on working-days, identifying the periods from 9 a.m. to 5 p.m. in all days excluding weekends and vacations), and a temporal interval bounding the scope of the periodic expression (e.g., [2/2013,5/2013], restricting the specified periods to those between February and May 2013).

- *Spatial* Authorizations could have associated a spatial contraint specified by a IP range or prefix, or by any spatial-related information.

- *Cardinality* Authorizations could be associated with a maximum of total operations or concurrent users on a specific operation. This constraint is especially usefull to set quotas that prevents involountary or malicious DoS attacks.

Constraints can be used to enforce high-level security objectives such as the separation of duty principle or conflict of interest policy. For instance, constraints may prevent users from being assigned to two conflicting roles or activating them simultaneously.

5.3 Access Control Standards

The access control models discussed in the previous section have not a standardized representation form, even if the XACML language has been defined by the Organization for the Advancement of Structured Information Standards (OASIS) as a standard for specifying access control policies.

5.3.1 ISO 10181-3

ISO 10181-3 [26] defines an architecture for access control. The framework defines four roles for components participating in an access request:

- Initiators

- Targets

- Access Control Enforcement Functions (AEFs)

- Access Control Decision Functions (ADFs)

Initiators submit access requests. An access request specifies an operation to be performed on a Target.

Access Control Enforcement Functions (AEFs) mediate access requests. AEFs submit decision requests to Access Control Decision Functions (ADFs). A decision request asks whether a particular access request should be granted or denied.

ADFs decide whether access requests should be granted or denied.

5.3.2 aznAPI

This authorization API [23] defined by the Open Group as a Technical Standard, represents a programmatic interface through which system components that need to control access to resources can request an access control decision from a system's access control service. The Open Group intended it to be used within the architectural framework defined in ISO 10181-3.

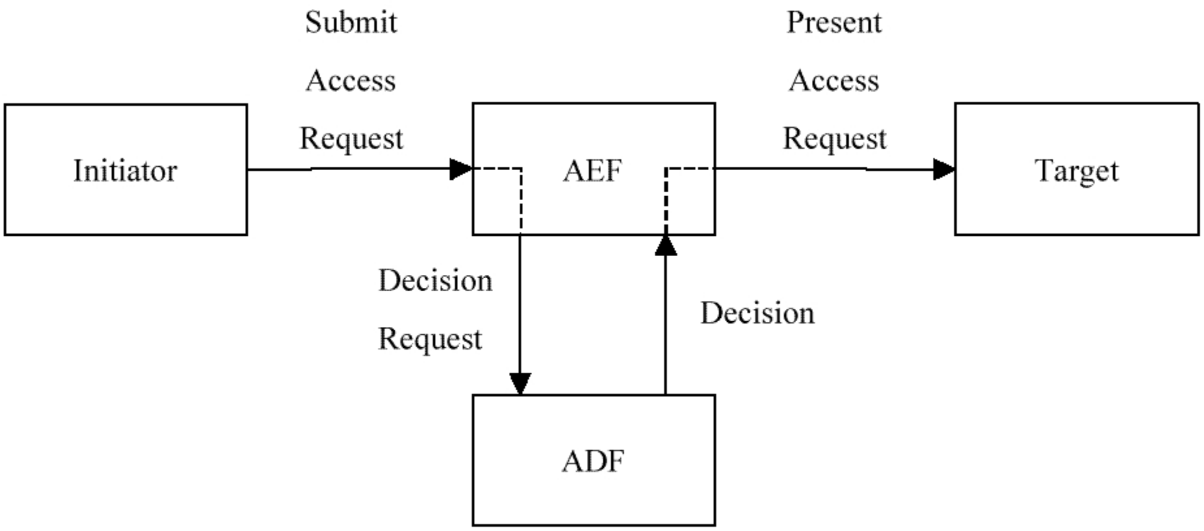


Figure 5: ISO 10181-3 Access Control Framework

5.3.3 XACML

The Extensible Access Control Markup Language (XACML) is a general-purpose language for specifying access control policies. In XML terms, it defines a core schema with a namespace that can be used to express access control and authorization policies for XML objects. Since it is based on XML, it is, as its name suggests, easily extensible. XACML supports a broad range of security policies and uses a standardized syntax for formatting requests so that any one of the following responses to an access request will be valid:

- Permit: action allowed
- Deny: action disallowed
- Indeterminate: error or incorrect/missing value prevents a decision
- Not Applicable: request cannot be processed.

The XACML's standardized architecture for this decision-making uses two primary components: the Policy Enforcement Point (PEP) and the Policy Decision Point (PDP). The PEP constructs the request based on the user's attributes, the resource requested, the action specified, and other situation-dependent information through Policy Information Point (PIP). The PDP receives the constructed request, compares it with the applicable policy and system state through the Policy Access Point (PAP), and then returns one of the four replies specified above to the PEP. The PEP then allows or denies access to the resource.

The PDP, PEP, PRP, and PIP components are not actually XACML-specific and are all defined in the AAA Authorization Framework [48]. In ISO 10181-3 [26] terms, XACML specifies an "Access Control Decision Function" (ADF), and defines its interactions with an "Access Control Enforcement Point" (AEF). In order to make the PEP and PDP work, XACML provides a policy set, which is a container that holds either a policy (or other policy sets), plus links to other policies. Each individual

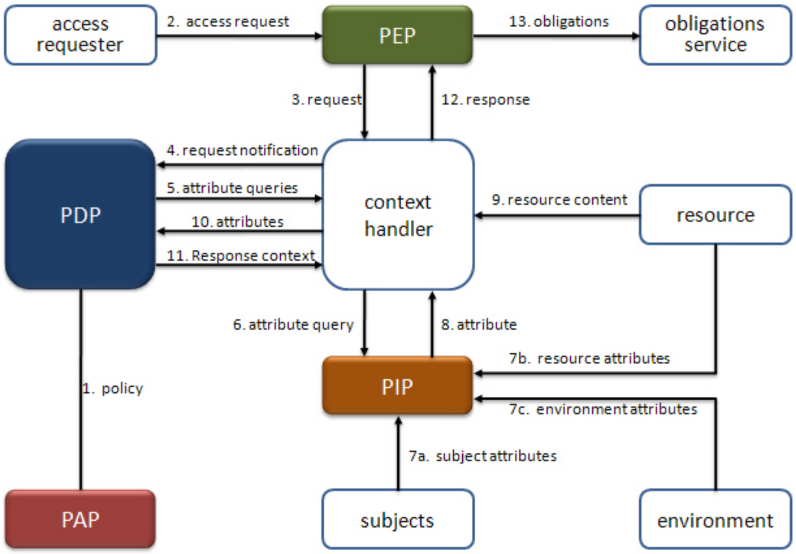


Figure 6: XACML Architecture

policy is stated using a set of rules. Conflicts are resolved through policy-combining algorithms. XACML also includes methods of combining these policies and policy sets, allowing some to override others. This is necessary because the policies may overlap or conflict.

Determining what policy to apply is accomplished using the *target* component. A target is a set of rules or conditions applied to each subject, object, and operation. When a rule's conditions are met for a user (subject), object, operation combination, its associated policy is applied using the process described above. The associated access control data for a given enterprise domain can then be encoded in an XML document, and the conformance of data to the enterprise access control model can be obtained by validating the XML document against the XML schema that represents the enterprise access control model using XML parsers.

Although, even if XML-based and other access control languages provide capabilities for composing policies from scratch, allowing users to specify access control policies, together with the authorizations through the programming of the language, they lack a formal specification language for access control constraints that prevent assigning overlapping privileges.

In addition the XACML language implements ABAC, whilst other models as e.g. RBAC, even if can be implemented as a specification of ABAC, cannot be completely supported. In fact, domain constraints are based on the semantic information pertaining to an enterprise context; a grammar-based language cannot deal with content-based constraints. So, an XML schema is insufficient for a complete specification of the RBAC model for an enterprise since the latter contains content-based domain constraints. An example is not allowing more than one user to be assigned to the role of "supervisor administrator" (role cardinality constraint) and not allowing the roles ""supervisor administrator" and "reasoner administrator" to be assigned to the same user (separation-of-duty constraint).

Here, we note that the specification languages assume a static environment where changes in access control policies are generally effected manually by a security administrator. So in essence, although XML-based access control languages provide features that enable them to specify a broad range of policies, a formal specification is still needed in order to define constraint rules adaptively.

6 mPlane Access Control Layer

This chapter describes the main mPlane access control layer specifications and requirements. In general the mPlane access control layer will develop a policy-based access control framework that will regulate the circulation of collected data throughout the mPlane systems. From an architectural point of view, the access control layer will represent the authorization controller between two different trust boundaries (a trust boundary can be intra-domain or inter-domain) and it will provide the right level of security and flexibilty for all mPlane components, even if they belong to different organizational or geographical domains.

6.1 Access Control Requirements

The mPlane authorization framework has the following key points:

- It is built using standardized technologies, thus providing support for extensions and enables interoperation between various platforms
- It allows extensions as to support the needs for a variety of environments.
- It allows inter-domain authorization, by enabling authorization upon examination of domain related policies

The inter-domain access control is necessary when users or components from one domain need to be assigned privileges to access data from other federated domains. In order to achieve this interconnection between different domains, several issues need to be taken under consideration:

- Access to data should be regulated by data privacy generic guidelines, applicable for all the cooperating domains.
- While the data access guidelines should be uniform, as it is in the purposes of the EU data privacy directive, enforcement points should be autonomous and have a large degree of freedom in managing their IT infrastructure.
- The coalition in the whole infrastructure is dynamic by design. The number of domains who participate in the mPlane cooperating architecture is not fixed. Domains and components can join or leave at any time, increasing thus the complexity of the overall management.
- Decentralized authorization architecture. Security policies can be defined locally without the necessity for a hierarchical central management. This approach will permit each domain to be completely autonomous, avoiding the introduction of single points of failure.
- Transparency to the users. The data traffic measurements, whether retrieved locally or from a remote domain should be of no difference to the end-user.

6.2 Access Control Model

The basic operational principles of the access control can be divided in two major categories: authentication-related and authorization-specific. Authentication is performed by implementing a mechanism that allows authenticating credentials with various methods depending on different protocols (e.g., digital certificates for SSL channels or RSA keys for SSH). Thus, the first task for a user or component is to provide appropriate credentials that will allow him/her identification within the domain he/she belongs to.

The authentication layer will provide the mPlane control of the identity of a person or a mPlane component that attempts to access to provided capabilities. This control implementation will depend on the various implementation of the security layer (e.g, SSH, HTTPS, etc.), but the base behaviour is that it will provide the same information to the authorization layer. In fact, on a successfull login attempt, indipendently from the security protocol, the authentication layer will provide to the authorization layer the authenticated identity. For example that identity will consist in the provided login for SSH protocol or a username derived from Subject CommonName or subjectAlternativeName identifier for HTTPS as described in RFC6125 [40].

The layer must use mutual authentication of the peers; that is, both mPlane components acting as clients and mPlane components acting as servers must be identified by e.g. a X.509 certificate [9]. In particular for X.509 certificate authentication it is necessary the full path validation on each certificate, as defined in RFC5280 [9].

As output of the authentication process, the trusted subject, that is represented by a unique string, is passed to the authorization layer in a way that depends on the implementation (e.g. example setting the USER environment variables or passing it as an argument). Other context information as IP address could also provided if a much more complex policy has to be evaluated.

6.2.1 User probes

End-users probes introduce a special scenario in the mPlane security architecture, due to the volatility, access variability and mobility of the users.

6.2.1.1 Authentication

The most evident problem with end-users is that probes cannot be authenticated in a easy and static way. In fact, the main authentication is the Internet provider's one, so there are different types of access to internet (e.g., wireless, DSL, LAN, etc.), depending on the device type or on the best available connection. On the other hand, authentication using user digital certificates is not feasable in a real-world environment with thousands of users, but also a solution based upon a user/password authentication specific for mPlane measurement usage requires an extra management effort, that cannot be acceptable. Thus a good choice can be to exploit a authorization approach, where the user is authenticated by the ISP during the connection access and associating only users that are authorized to run measurements on a specific range of IP addresses. As result the communication to the ISP supervisor will be permitted only to those IPs that belong to the specified range.

6.2.1.2 User mobility

The user mobility is transparent for example in the case of Managed Measurement Service Provider (MMSP) supervisor, because it does not depend on which network the user is on. Otherwise, this mobility becomes a problem especially in a scenario in which there is one supervisor per ISP, so a user can change ISP connection using the same probe installed on his/her device (e.g., smartphone, laptop, etc.). Thus, the supervisor that the probe has to contact belongs to the ISP to which the user is connected, so the problem of a transparent selection of the right supervisor may arise in a such mobility scenario. In the mPlane architecture a solution is represented by a DNS implementation of a private Top Level Domain (TLD) that redirects to a local supervisor (like APN does in a mobile infrastructure). Furthermore, a network implementation (routing or firewall) that allows connections only from IP addresses that are internal to the ISP network will prevent any improper push of collected data from a misconfigured or malicious probe. On the other hand, the probe identifies the supervisor with its certificate's Common Name. In order to have an explicit user consent on privacy data treatment, the probe verifies that the certificate Common Nane is not changed since the last connection, otherwise the probe will ask the customer an explicit authorization to send traffic to the new provider that is currently active for measurements.

6.2.2 Cross-Domain

The model should be also a solution attempting to enable intra-domain communication, that should be characterized by its interoperability and scalability features. The approach in order to enable cooperation between different access policies, builds upon a delegation process through the supervisor to supervisor communication, which enables to control the visibility of the domain's internal specific components. The decentralization is achieved by implementing multiple autonomous domains each one of which is responsible for enforcing local access control policies. This issue may be solved limiting the inter-domain interactions only to the communications between supervisors.

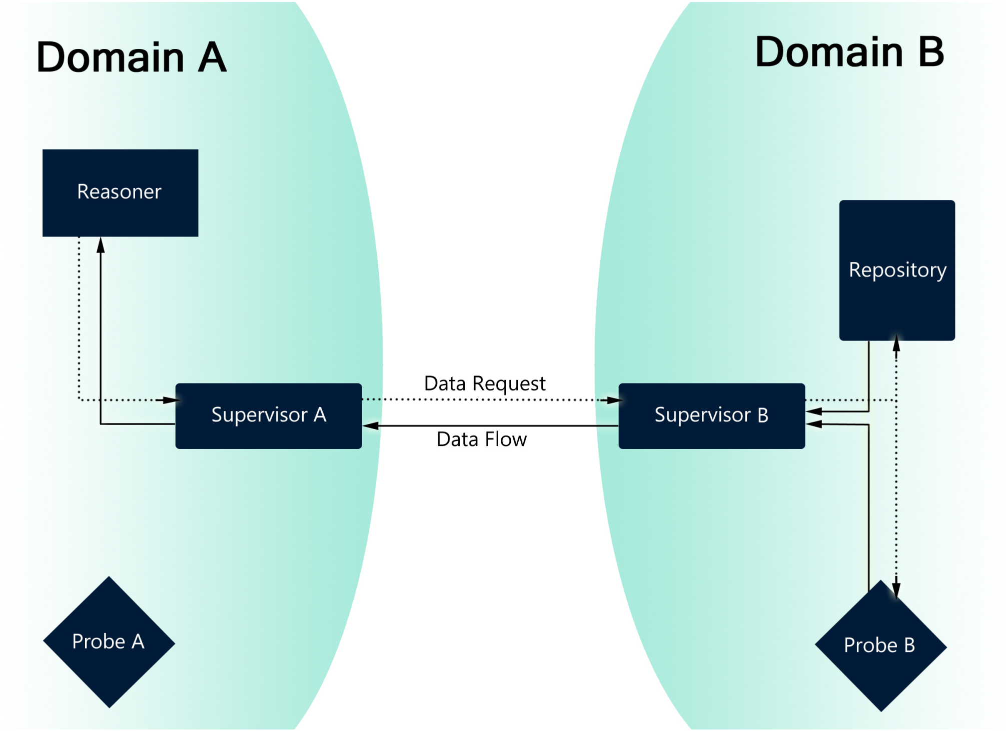


Figure 7: Cross-domain mPlane communication

Each supervisor will act as a proxy between the components belonging to its domain and the external world, exposing their capabilities and forwarding them requests coming from outside. The set of capabilities that can be exposed to an external domain is defined by an agreement stipulated in a static way between the administrators of the domains that want to cooperate. In this way, each supervisor will have an authorization layer that will expose to the external world only a subset of the capabilities of its own domain, which may change depending on the supervisor with whom is talking. Hence each supervisor, that controls the trust-boundary domain, then performs the authorization check for any successfully authenticated external supervisor, whilst the requests from any external mPlane component are controlled and "proxied" by the authorized supervisor (that acts as a delegated requester). Thus each domain has user/components associated with roles to its own jurisdiction, that can communicate to other domains with a many-to-one domain-level identity mapping, that permits access delegation of the supervisor. This approach results in a fully distributed implementation of the coalition, which only establish trust relationship on a supervisor-level basis. With the "supervisor-level" access control approach, the supervisor can act as an access controller also in a intra-domain scenario, where an access control needs to be partitioned in different organizational subdomains, e.g. for large networks and/or companies.

6.3 Implementation

6.3.1 Access control architecture

The main mPlane access control interface requirements to fulfill are the following:

- support for different authentication mechanisms, because of the protocol agnostic trait of mPlane infrastructure
- good separation of the authentication service from the authorization one. In fact, in order to support different authentication mechanisms, the authorization service has to be loosely coupled with the authentication service, that will represent the trusted module for the verification of the identity of the requester
- support for different authorization attribute types (e.g., identities, roles, context constraints, etc.)
- possibly support for variety of access control mechanisms as implementations for different deployment scenarios
- management of authorization policies must be independent of the application layer
- evaluation of policy rules must be completely transparent to the application layer

Thus, the access control interface that could accomplish it, can be implemented in a way that is very similar to the aznAPI architecture. Therefore, even if the mPlane interface implementation details will be described in another deliverable, the overall architecture will be as depicted in Figure 8.

The above architecture uses the ISO 10181-3 terminology, so we have the Initiators, that can be represented by any mPlane component, requesting for access to one of the target mPlane component's capabilities (Targets), and the Access Control Enforcement Functions (AEFs), that are the set

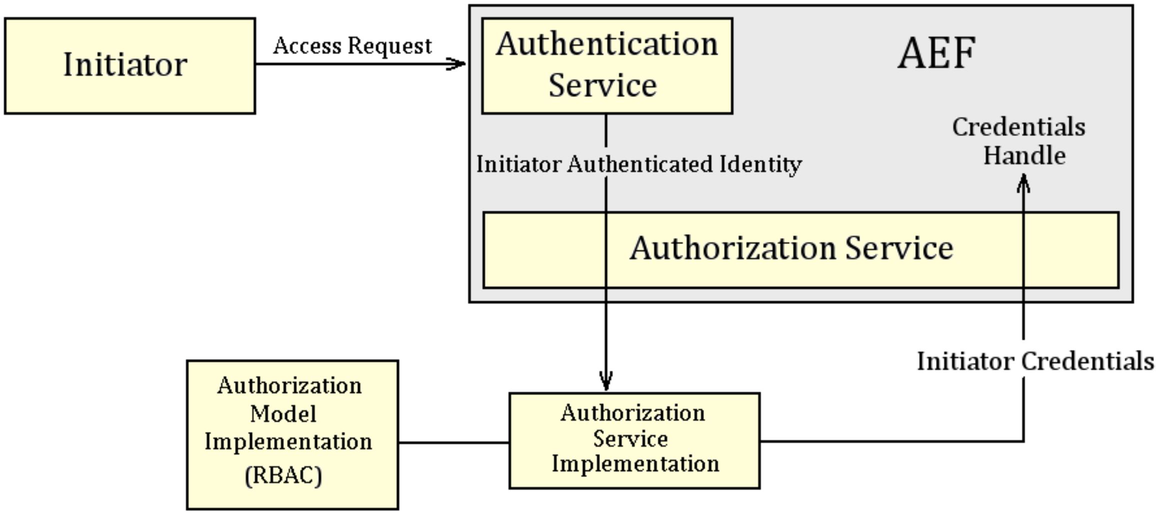

Initiator
Access Request
Authentication Service
AEF
Credentials Handle
Initiator Authenticated Identity
Authorization Service
Initiator Credentials
Authorization Model Implementation (RBAC)
Authorization Service Implementation

Figure 8: mPlane Access Control Architecture

of functions that manage the request and enforce the access control decisions.
Thus, the components of the access control architecture that will be implemented are:

- *authentication service*: it implements the authentication mechanism depending on the communication protocol (e.g. digital certificates for HTTPS) and it is responsible of the correct identity verification. It is also in charge of the creation of the credentials information that will be supplied to the authorization service (e.g. entracting the required information from the DN of the requester's certificate)
- *authorization service*: it implements the authorization API library and uses the authorization model implementation (e.g. SELinux) for authorization checks. If a successful result is returned then a credential handle will be created and returned to the application layer
- *authorization model*: it implements the choosen authorization model (e.g. RBAC) and performs all the necessary checks for the credentials of the requester and the requested target supplied by the authorization service. The result of all the required checks on permissions (e.g. membership to a role granted for the requested resource or operation, valid context constraints, etc.) will be returned to the authorization service

6.3.2 RBAC

Among all the access models discussed in the previous chapter, considering that DAC and MAC models don't fit the mPlane access control requirements, the most supported and widely used model that can be implemented is the RBAC one.

6.3.2.1 RBAC Components

There several components on the shelf that can be used in a Unix-like platform that provide a RBAC implementation. The most relevant open-source ones that run on a Linux platform are the following:

- **SELinux**
  The security policy implemented in Security-Enhanced Linux (SELinux) is Type Enforcement (TE) under a layer of role-based access control. TE is the most visible, and therefore the most well known, server because it enforces fine-grained permissions: when something breaks because of unexpected access denials, TE is most likely responsible. In TE, a process's security domain (its domain of influence over the system) is determined by the task's history and the currently executing program.

- **Grsecurity**
  grsecurity is a set of patches for the Linux kernel with an emphasis on enhancing security. It allows the system administrator to, among other things, define a least privilege policy for the system, in which every process and user have only the lowest privileges needed to function.

- **AppArmor**
  To achieve RBAC, AppArmor uses a combination of two Pluggable Authentication Module (PAM) security modules. The *pam_cap* module is used to raise a users privileges while the *pam_apparmor* module is used to further restrict the users processes from what would be possible with the granted capabilities.

6.3.2.2 RBAC using Digital Certificates

Distinguished name (DN) is a term that describes the identifying information in a certificate and is part of the certificate itself. A certificate contains DN information for both the owner or requestor of the certificate (called the Subject DN) and the CA that issues the certificate (called the Issuer DN). Depending on the identification policy of the CA that issues a certificate, the DN can include a variety of information. The DN information that you can provide for a certificate includes:

- Certificate owner's Common Name (CN)
- Organization
- Organizational unit
- Locality or city
- State or province
- Country or region

In X.500-based directory systems, including those accessed using the Lightweight Directory Access Protocol (LDAP) [49], DNs are used to unambiguously refer to directory entries. Therefore, exploiting the PKI hierarchy and the DN information it is possible to implement a RBAC model using digital certificates. On the other hand adding to certificate one or more certificate policies [24] as critical can be a valid method to hava a finer graned access control check.
The certificate policies extension contains a sequence of one or more policy information terms, each of which consists of an object identifier (OID) and optional qualifiers. These policy information terms indicate the policy under which the certificate has been issued and the purposes for

which the certificate may be used (e.g., troubleshooting, administration, etc.).
So, in conjunction with Certification Authority (CA) categorization this approach can also be exploited for inter-domain authorization based on different trust levels.

Issuer DN mapping
In a inter-domain scenario with a high level of trust the authorization can be performed on the certificate's issuer DN rathen than the subject's one, because the identification scope is less granular. So, for example a supervisor can be authorized to perform operation in a another domain (e.g O=University of Zurich, OU= Measurement Group), because for example the Organization Unit level of the DN (e.g. O=University of Turin, OU=Measurement Group) of the issuer is completely trusted and mapped to a specific role (e.g. Researcher), so that any further identity specification (e.g. the Locality or the Common Name) is needed to the authorization process. This kind of mapping permits also a more relaxed and stable role mapping management, because of the much more durable lifecycle of a CA digital certificate. Anyway, this approach cannot be exploited if the CA is not well categorized and cannot be used to uniquely identify a specific domain.

Subject DN mapping
For a much more fine-grained authorization check the mapping can be performed on the whole subject DN (identity to role mapping) or part of it (group to role mapping). In fact, for example an external supervisor's identity is trusted because its certificate has been issued by a trusted CA, but a DN (e.g. OU = DataMining) to which is not permitted to perform a specific set of measurement on probes, so it is needed to map it to a different role. Obviously, the main drawback of this approach is that relies on digital certificate issue management and their DN policy, so the mapping could not be confortable in a complex Public Key Infrastructure (PKI) scenario.

6.4 Access Control Policy Roles

Usually access control policies are implemented using two classical approaches:

- **Closed policy** allows an access if there exists a positive authorization for it, and denies it otherwise. Thus authorizations specify permissions for an access.

- **Open Policy** denies an access if there exists a negative authorization for it, and allows it otherwise. So in this case authorizations specify denials for an access.

In general, except in a test or developement scenario, the defaul policy in mPlane should be a closed one, whereas nothing is permitted to none, except for the administration tasks, that are all permitted to administrators. In fact an open policy is usually a good choice only in those scenarios where the need for protection is not strong and by default access is to be granted, whilst closed policy, which, denying access by default, ensures better protection. The combined use of positive and negative authorizations is discouraged because it brings to the problem of how the two specifications should be treated in the case no authorization is specified (*incompleteness*) or both a negative and a positive authorization (*inconsistency*) has been defined for a specific access. While completness can be achieved by defining a policy that have priority, it is much more difficult to solve inconsistency policy conficts [14].

6.4.1 Functional Role

Implementing the RBAC model, roles and attached rights should be defined for the system to clearly separate different activities (development, operation, management). In this case, an identity may access an operation if it is active in the role to which the permission has been assigned.
In this case an example of access control matrix (role/permissions) can be this:

	Administrator	Developer	Researcher	Operator
measurement	-	-	-	x
configuration	x	-	-	-
export data	-	-	x	x

6.4.2 Level Of Trust

Access decision can also be based on the trust level of a requestor. The level of trust implements domain trust boundaries access control for example as described in "RBAC using Digital Certificates" of section 5.1. The main benefit of this approach (that implements in same way the CBAC model) is that can be used to have a lesser access control granularity, that is that any identity belonging to a trusted domain can perform all the operations permitted to that domain, even if that identity is not previously known. In fact, in an inter-domain scenario the identity of components involved in the domain-to-domain interaction can change without any communication between the two collaborationg parties. Because the level of trust doesn't change if the components continue to belong to the same domain, then authorization mapping doesn't need to be changed.

6.4.3 Context constraints

Access can be denied if requestors who have asked for a measurement belong to the rigth role, but do not have an adequate trust level to access the operation depending on e.g. a spatial (IP address) or time contraint. For example in a mobile scenario as described in "User probes" of section 5.1 the user's probe cannot be authenticated and authorized in a unique and static way. So an authorization that relies on the context in which the probe is operating (e.g. checking if it belongs to a specific IP range) can represent an excellent way to control the access to the mPlane architecture from all those components that operates on the edge and that can change very dinamically.

References

[1] R. Agrawal and R. Srikant. Privacy-preserving data mining. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, SIGMOD '00, pages 439--450, New York, NY, USA, 2000. ACM.

[2] M. Arapinis, L. Mancini, E. Ritter, M. Ryan, N. Golde, K. Redon, and R. Borgaonkar. New privacy issues in mobile telephony: Fix and verii¬cation. In *Proceedings of the 2012 ACM conference on Computer and communications security*, pages 205--206, 2012.

[3] E. Boschi. Legal requirements and issues in network traffic data protection. In *Proceedings of the 1st ACM workshop on Network data anonymization*, NDA '08, pages 19--20, New York, NY, USA, 2008. ACM.

[4] E. Boschi and B. Trammell. Ip flow anonymisation support., May 2011.

[5] T. Brekne and A. Arnes. Circumventing ip-address pseudonymization. *Proc. of International Conference on Computer Communications and Networks*, pages 43--48, 2005.

[6] M. Burkhart. Enabling collaborative network security with privacy-preserving data aggregation, 2011.

[7] M. Burkhart, D. Brauckhoff, M. May, and E. Boschi. The risk-utility tradeoff for ip address truncation. In *Proceedings of the 1st ACM workshop on Network data anonymization*, NDA '08, pages 23--30, New York, NY, USA, 2008. ACM.

[8] M. Burkhart, D. Schatzmann, B. Trammell, E. Boschi, and B. Plattner. The role of network trace anonymization under attack, January 2010.

[9] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk. Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile. RFC 5280, May 2008.

[10] S. Coull, C. Wright, A. Keromytis, F. Monrose, and M. Reiter. Taming the devil: Techniques for evaluating anonymized network data. *Network and Distributed System Security Symposium (NDSS)*, 2008.

[11] S. E. Coull, M. P. Collins, C. V. Wright, F. Monrose, and M. K. Reiter. On web browsing privacy in anonymized netflows. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, SS'07, pages 23:1--23:14, Berkeley, CA, USA, 2007. USENIX Association.

[12] S. E. Coull, F. Monrose, M. K. Reiter, and M. Bailey. The Challenges of Effectively Anonymizing Network Data. *Conference For Homeland Security, Cybersecurity Applications &amp; Technology*, 0:230--236, 2009.

[13] 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 *in Proceedings of the Network and Distributed System Security Symposium*, pages 35--47, 2007.

[14] S. D. C. di Vimercati, P. Samarati, and S. Jajodia. Policies, models, and languages for access control. In S. Bhalla, editor, *DNIS*, volume 3433 of *Lecture Notes in Computer Science*, pages 225--237. Springer, 2005.

[15] C. Dwork, F. McSherry, K. Nissim, B.-G. University, and A. Smith. Calibrating noise to sensitivity in private data analysis. In *TCC Proceedings of the Third conference on Theory of Cryptography*, pages 265--284, 2006.

[16] Directive 95/46/ec of the european parliament and of the council. OJ L 281, 23.11.1995, October 1995.

[17] Directive 2002/58/ec, european parliament and of the council concerning the processing of personal data and the protection of privacy in the electronic communications sector (directive on privacy and electronic communications). OJ L 201, 31.7.2002, July 2002.

[18] Directive 2009/136/ec of the european parliament and of the council of 25 november 2009 amending directive 2002/22/ec on universal service and users' rights relating to electronic communications networks and services, directive 2002/58/ec concerning the processing of personal data and the protection of privacy in the electronic communications sector and regulation (ec) no 2006/2004 on cooperation between national authorities responsible for the enforcement of consumer protection laws. OJ L 337, 18.12.2009, December 2009.

[19] Proposal for a regulation of the parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (general data protection regulation). COM(2012) 11 final, 25.1.2012, January 2012.

[20] Opinion 13/2011 on geolocation services on smart mobile devices. WP 185, 16.05.2011, May 2011.

[21] M. Foukarakis, D. Antoniades, S. Antonatos, and E. P. Markatos. Flexible and high-performance anonymization of netflow records using anontool.

[22] Charter of fundamental rights of the european union. OJ 2000/C 364/01, 18.12.2000, December 2000.

[23] T. O. Group. Authorization (azn) api, January 2000.

[24] R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure certificate and crl profile. RFC 2459, January 1999.

[25] Opinion of the european data protection supervisor on net neutrality, traffic management and the protection of privacy and personal data, October 2011.

[26] Iso. Iso/iec 10181-3:1996 - information technology -- open systems interconnection -- security frameworks for open systems: Access control framework.

[27] S. J. S. J. J. Parekh, K. Wang. Privacy-preserving payload-based correlation for accurate malicious traffic detection. *ACM Workshop on Large-scale Attack Defense (LSAD)*, 2006.

[28] K. L. J. King and A. J. Slagell. A taxonomy and adversarial model for attacks against network log anonymization. In *Proc. of ACM SAC.*, pages 1286--1293. ACM, 2009.

[29] M. H. A. Jinliang Fan, Jun Xu and S. Moon. Crypto-pan.

[30] E. Kohler. ipsumdump.

[31] D. Koukis, S. Antonatos, and K. G. Anagnostakis. On the privacy risks of publishing anonymized ip network traces. In *Communications and Multimedia Security, volume 4237 of Lecture Notes in Computer Science*, pages 22--32. Springer, 2006.

[32] Y. Lindell and B. Pinkas. Privacy preserving data mining. *Advances in Cryptology—CRYPTO 2000*, pages 36--54, 2000.

[33] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data*, 1(1), Mar. 2007.

[34] E. McCallister, T. Grance, and K. A. Scarfone. *Guide to Protecting the Confidentiality of Personally Identifiable Information (PII)*, volume NIST SP - 800-122. NIST, 2010.

[35] G. Minshall. Tcpdpriv.

[36] I. C. Office. Anonymisation: managing data protection risk code of practice.

[37] P. Ohm. Broken promises of privacy: responding to the surprising failure of anonymization. *UCLA Law Review*, 57, 2009.

[38] D. Plonka. ip2anonip.

[39] E. Rescorla. Http over tls. RFC 2818, May 2000.

[40] P. Saint-Andre and J. Hodges. Verification of domain-based application service identity within internet public key infrastructure using x.509 (pkix) certificates in the context of transport layer security (tls). RFC 6125, March 2011.

[41] P. Samarati. Protecting respondents' identities in microdata release. *IEEE Trans. on Knowl. and Data Eng.*, 13(6):1010--1027, Nov. 2001.

[42] P. Samarati and S. de Capitani. *Foundations of Security Analysis and Design: Tutorial Lectures, Volume 1*, chapter Access control: Policies, Models and Mechanisms. Springer, 2001.

[43] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. In *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1998.

[44] D. Sauter. Invasion of privacy using fingerprinting attacks, 2009.

[45] W. D. Singh A., F. Yu. Measuring disclosure risk and information loss for massc-treated micro-data. *Proceedings of the American Statistical Association, Toronto, Canada*, pages 4374--4381, 2004.

[46] A. A. T. Brekne and A. Øslebø. Anonymization of ip traffic monitoring data: Attacks on two prefix-preserving anonymization schemes and some proposed remedies. *5th Workshop on Privacy Enhancing Technologies*, 3856, 2006.

[47] F. M. T.-F. Yen, X. Huang and M. K. Reiter. Browser finger-printing from coarse traffic summaries: Techniques and implications. *Proc. of Detection of Intrusions and Malware and Vulnerability Assessment*, 3856:157--175, 2009.

[48] J. Vollbrecht, P. Calhoun, S. Farrell, L. Gommans, G. Gross, B. de Bruijn, C. de Laat, M. Holdrege, and D. Spence. Aaa authorization framework. RFC 2904 (Informational), August 2000.

[49] K. Zeilenga. Lightweight Directory Access Protocol (LDAP): Directory Information Models. RFC 4512 (Proposed Standard), June 2006.