Information & Security

P. Jurgała, T. Kurek & M. Niemiec
vol. 53, no. 1 (2022): 47-64
https://doi.org/10.11610/isij.5304

Published since 1998

ISSN 0861-5160 (print), ISSN 1314-2119 (online)

Research Article

Preserving Privacy of Security Services in the SecaaS Model

Piotr Jurgała, Tytus Kurek, and Marcin Niemiec (✉)

AGH University of Science and Technology, Mickiewicza 30, 30-059 Krakow, Poland, https://www.agh.edu.pl/en/

ABSTRACT:

Outsourcing security services to the cloud allows companies to minimize IT infrastructure costs, use services faster, improve manageability, and reduce their own maintenance effort. However, the security policy, which identifies the rules and procedures, also contains information about security architecture, threats, and vulnerabilities. Therefore, the privacy of security policies applied in a cloud environment is needed. This article describes a structure of security policies for selected network services that will ensure privacy protection and protect against the analysis of network traffic by an unauthorized person. The developed solution is based on the UNIPRIV model. This architecture for authentication and access control services was implemented and verified, taking into account safety and performance.

ARTICLE INFO:

RECEIVED: 06 JUNE 2022

REVISED: 28 AUG 2022

ONLINE: 20 SEP 2022

KEYWORDS:

cybersecurity, privacy, cloud environment, authentication, access control, secure services

Introduction

Nowadays, almost no one can imagine life without access to the Internet. The Internet network is considered a crucial technology for private users and business organizations of each size. Due to the fact that the COVID-19 pandemic was officially declared on March 11, 2020, the network traffic rapidly grew. The reason for this phenomenon was the worldwide restrictions by which people were confined to their homes for a certain period of time. The crisis also forced the transition of the work model of most organizations to remote work-from-home.

✉ Corresponding Author: Tel.: +48 12617 4803; niemiec@agh.edu.pl

Due to the growing number of internet users, data traffic, and new internet services, cybersecurity plays a crucial role. Cybersecurity is a set of methods and issues that serves to protect internet-connected systems as hardware, software, and data from digital attacks.[1] It is also responsible for ensuring data confidentiality, integrity, and availability. According to the 2021 Report: Cyberwarfare [2] in the C-suite published by Cybersecurity Ventures, the global cybercrime costs will grow by 15 percent per year over the next five years. Cybersecurity Ventures expects that this costs will reach $10.5 trillion USD annually by 2025. There are many solutions that increase the level of security and thus make cyberattacks carried out by threat actors difficult. In addition to implementing the security solution, proper management is also required.

Over the past few years, an upward trend in transition from traditional solutions to cloud has been noticed, what has been named a cloud shift. The transition of services to the cloud computing model has a number of advantages. The most important benefits of cloud computing include:

- reduced IT costs – moving to the cloud may reduce the cost of managing and maintaining IT systems,
- scalability – it is easy to scale up or scale down IT operations depending on the current situation without purchasing and installing expensive devices,
- business continuity – the data stored in cloud are protected from a natural disaster, power failure or other crisis what causes minimising of downtime for business.

However, the cloud shift can force the companies to disclose their confidential information to *Cloud Service Provider* (CSP).

With increasing of IT systems allocation in the cloud, it was proposed the cloud computing model of outsourcing a security service to the cloud called *Security as a Service* (SecaaS). The numbers of CSP, offers many cloud-based security services. From the outsourced security services to the cloud point of view, it is crucial to preserve the confidentiality of security policies. The security policy identifies the rules and procedures that apply to all IT assets and resources within organization. It also contains information about security architecture, threats and vulnerabilities. Exposing such confidential information by an unauthorized person can cause an unimaginable effects for the organization or the business.

Related Work

Numerous research papers relating to outsourcing security services to the cloud have been studied over the past several years.[3] The authors have the greatest concerns about exposing of customer security policies to the CSP. In 2014 V. Varadharajan and U. Tupakala[4] studied SecaaS model for cloud environment and identified two main situations where privacy of security policies may be violated by CSP: direct insights into security policies implemented in the cloud and eavesdropping of network traffic

In 2016, T. Kurek, M. Niemiec and A. Lason[5] introduced a novel framework for preserving cloud-based firewall policy confidentiality known as the *Ladon Hybrid Cloud* (LHC). The authors improved the firstly proposed Ladon framework by A. Khakpour and A. Liu[6] in terms of privacy preserving. To improve level of privacy preserving, the authors introduced in LHC framework hybrid cloud architecture. *Bloom Filter Firewall Decision Diagram* (BFFDD) presented in Ladon framework was placed in public cloud and the parameters of a Bloom filters were intentionally modified to increase a false positive rate. Thanks to such action, the decision taken in BFFDD on packet can be ambiguous and the ability to get to know a firewall security policy by the CSP is significantly reduced. However, LHC framework requires performing additional task by *Firewall Decision Diagram* (FDD) in the private cloud to take final decision relating to packets. The impact of Bloom filters parameters on security and efficiency in LHC framework was studied by M. Mencner and M. Niemiec.[7] Likewise, in 2015, the authors of the LHC framework, published a research article where the proposed and presented three solutions for preserving privacy of signature-based Intrusion Detection System.[8] In 2017, the authors of the LHC framework introduced the *Intrusion Prevention System Decision Diagram* (IPSDD) as a new representation of signature-based *Intrusion Prevention System* (IPS) security policies.[9] The IPSDD was introduced as a decision tree structure based on IPS rules.

In 2017, T. Kurek et al.[10] undertook to define general principles of protecting the confidentiality of security policies for services outsourced to the cloud. They proposed *Universal privacy-preserving platform for SecaaS services* called the UNIPRIV. The platform also works in hybrid cloud model where majority of computationally-expensive operations are performed in public cloud. The small number of operations responsible for calculating hashes to take the final decision are performed in the private cloud. The most important activity of the platform is creating anonymized decision diagram based on security policies of security service which is placed in the public cloud. Thanks to the use of Bloom filters to anonymize decision diagram as well as encryption and hashing decision, the CSP is deprived of direct insight into the original structure of the security policy of security services. The authors shown that the UNIPRIV platform can be applied to all security services for which security policies can be represented as a decision tree. The presented UNIPRIV has been implemented for IDS service and experiments were carried out in the field of platform performance. However, the platform has not been validated for other security services. In this paper, the preserving privacy of selected security services based on the concept of UNIPRIV platform was realized and verified.

Bloom Filters

The concept of the UNIPRIV platform supports privacy preserving in cloud environment using Bloom filters[11] (BF) – a mathematical data probabilistic structures. It is a space-efficient structure that is used to test whether an element is a member of a set in a time-efficient manner. From the mathematical point of view, BF is m-sized bit array which is generated by calculating k hash functions for each of the *n* element from a member set. To check whether the element *x* is included in original set, the *k* hash functions are calculated from *x* element, and the results are compared with a corresponding indexes of BF. If at least one of calculated indexes is set to 0, the *x* element is definitely not a member of the original set. If all of the calculated indexes are set to 1, the *x* element may be a member of the original set. This indicates that BF never generate false negative results, but the false positive results can occur. However, BF allows for a relatively quick check if an element has chance existing in original data set or if it does not exist.

Example operation of adding elements to a BF is shown in Figure 1. It was assumed that the original data set has two elements: x_1 and x_2. The bit array size of BF is 10 and there are 3 independent hash functions. In the first step, hash functions are calculated for x_1. The results indicates which bits of BF need to be set to 1. The calculated index based on hash function can be out of scope BF indices, so an extra modulo operation is performed. For a x_1 element, BF_1, BF_3 and BF_6 are set to 1. The same operations are performed for element x_2. Only the value of BF_0 and BF_8 are set to 1, because a BF_3 has already been set before (caused by a hash of x_1 element).

Figure 2 shows example of usage previously created BF. Checking if an element is included in a BF starts from calculating appropriate hash function for each element. The results of calculating hash function for item x_1 indicates that BF_1, BF_3 and BF_6 bits hash to be checked to confirm that the element may occur in the set. All bits are set to 1, so element x_1 might be included in a original set with an error specified by BF false-positive rate. The same procedure is repeated for item x_3. Even if BF_6 and BF_8 bits are set to 1, the zeroed BF_4 tells that x_3 element is definitely not a member of original set.

The BF are widely used in IT companies because of their advantages: time- and space-efficiency. Many big companies such a Facebook, Instagram and Google use this technology in the registration process to check whether a username is already taken or not.[12] This saves disk space, and what's more, checking if a username occurs in a database is much faster than known search algorithms.

Cybersecurity and Cloud Services

Cloud computing services can be deployed in some types of services models where each of them is different in terms of services provided by CSP. One of them is SecaaS – a service-oriented approach to IT security architecture as a cloud delivered model outsourcing cybersecurity services. In this model, CSP deliver security services on a subscription basis which allows for cost savings, being up to date with the latest patches and fast provisioning of service. SecaaS has gained popularity in many organizations as a way to simplify the responsibilities of an internal security and scaling security needs as the business grows.

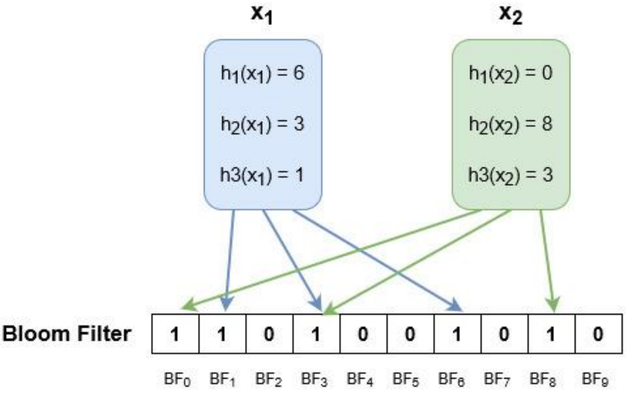


Figure 1: Adding elements to a BF.

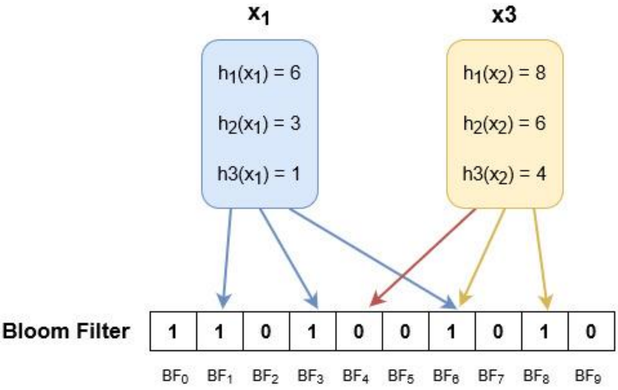


Figure 2: Usage of a BF.

Services in the SecaaS model are offered by most CSP and their scope is very wide, which offers protection at the most granular level. The most important services provided in the SecaaS model include: Data Loss Prevention, Email Security, Antivirus, Firewall, Intrusion Protection System, Authentication, Access Control and Security Information and Event Management.

For the purpose of implementation and verification of the solution, services such as authentication and access control were selected. These cloud-based security services play a key role in the entire IT world. Authentication is a security service that almost each of us uses on a daily basis. It is a process of confirming someone identity to another individual. Internet users use this process to login to electronic mail, favorite websites, social networks, bank accounts and all other services or systems that require validation of their identity. To verify the identity, it is required to provide credentials consisting of user ID and other information depending on the authentication type. In general, there are four types of authentication[13] used for validation of the identity:

- Type I: what someone knows – cognitive information like password, security code, PIN, passphrase;
- Type II: what someone has – items which are in possession such as photo ID, swipe card, security card, security dongles;
- Type III: what someone is – physical and behavioral attributes, for example, voice recognition, keystroke recognition, fingerprint recognition, retina scan;
- Type IV: where someone is – location information.

Information provided by one or more authentication types can be used in authentication process. A process where more than one authentication technique is required is called *Multi-Factor Authentication* (MFA). MFA combined with two or more type of authentication provides higher level of security of authenticating user.

Access control is another fundamental component of data security, which protects against unauthorized access to resources. Access control process decides who is allowed to access and use information or resources. There are four main types of access control[14] and an appropriate type is chosen based on security and usability requirements.

- Discretionary Access Control (DAC) is a type of access control where the way of restricting access to object is realized based on identity of subjects and the owner of resource defines access control policies.
- Mandatory Access Control (MAC) is a type of access control where a central authority regulates access rights to resources based on security levels.
- Roles-based Access Control (RBAC) is a type of access control where permissions are assigned to users based on their role in organization.
- Attribute-based Access Control (ABAC) is a type of access control where the access is granted based on attributes such as time of a day or location.

The mentioned cloud-based security services usually takes decisions based on the content of one or several first network packets. It is decision-making process. What's more, security policies of chosen security services can be presented as a list of rules. Therefore, they can be implemented using UNIPRIV model.

The main requirement of the UNIPRIV is that the security policies of network services must be represented by a decision tree. Decision tree is a structure consisting of a root node, branches (edges) and leaf nodes. The root node has no incoming edges. Each node is labelled with an attribute to test. Each edge describes the outcome of a test. The nodes that have outgoing edges, except

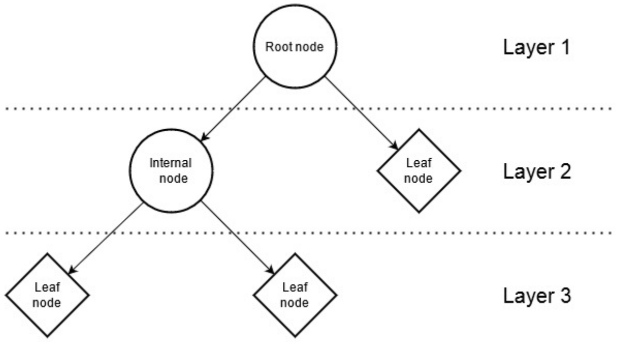


Figure 3: A concept of a decision tree with layers.

root node, are called internal nodes. Nodes without outgoing edges are called leaf nodes and have assigned decisions. A decision tree has layers with nodes.

The layer of a given node is determined by the distance from the root node. Example of a decision tree concept with layers is shown in Figure 3. The presented decision tree has three layers with specific nodes.

Security Policy Representation

The privacy-preserving solution based on UNIPRIV model supports selected security services. The following cloud-based security services were taking into account: authentication (type I), where user has to provide username and password and access control (RBAC). The structure of authentication policy rules set is shown in Table 1.

Table 1. Structure of authentication policy rules set.

Username	Password	Action	Role
USER_1	PASS_1	Authenticate	ROLE_1
USER_N	PASS_N	Authenticate	ROLE_N
ANY	ANY	Reject	NONE

The first row of the table describes column labels. Each rule consist of unique username for each user, password, action and role. There are two types of action:

- *Authenticate* for users who provide correct username and password,
- *Reject* for users who provide incorrect credentials.

The *Role* field is responsible for the role that the user will get if he/she gives correct credentials. The authentication policy rules set must have rules containing user credentials and one rule, which is responsible for rejecting all invalid credentials.

Table 2 shows the structure of access control policy rules set. The first row of the table consist of column labels. Each rule consist of:

- *Resource* which corresponds to asset IP address,
- *Activity* describes an activity that can be performed on specific asset,
- *Role* of the authenticated user,
- *Action* that describes if the activity is allowed or denied.

Table 2. Structure of access control policy rules set.

Resource	**Activity**	**Role**	**Action**
ASSET_1	PASS_1	Authenticate	ROLE_1
ASSET_N	PASS_N	Authenticate	ROLE_N
ANY_ASSET	ANY	Reject	NONE

Just like in the authentication policy rules set, the access control policy rules set must have rules containing information about activities which can be performed on resource by authenticated user with assigned role and one rule which deny all other actions.

Platform Components

The platform to preserve the privacy of selected security services in SecaaS model has been implemented in hybrid cloud model. The key assumption of the hybrid cloud model is that some requiring computing power operations are outsourced to the CSP. This environment imitates a fairly frequently used architecture in many companies. The developed solution consists of three main components:

- Decision Diagram Generator (DDG),
- Anonymized Decision Taking Engine (ADTE),
- Decision Taking Engine (DTE).

Each of them was written in python programming language which was chosen for its brevity, clarity and universality. Figure 4 shows where are placed specific components in hybrid cloud model. Computing power operations are performed by ADTE in the public cloud delivered by CSP. The other two components in which confidential data about policies are processed, are placed in a private cloud.

DDG is the most important part of the solution. This module is responsible for generating *Unanonymized Decision Tree* (UDT) and *Anonymized Decision Tree* (ADT) based on security service policy rules set. This component is capable to create ADT for two previously described security services – type I authentication and RBAC. The main task of DDG is to create decision tree, replace edges with appropriate Bloom filters and encrypt a decision. The decision diagram generation process for authentication is shown in Figure 5. A sample of security policy for authentication shown in Figure 5A is transformed into UDT shown in Figure 5B. The decision tree has three layers, and in each of them are nodes labeled with appropriate column labels. Figure 5C shows ADT. Anonymization is achieved thanks to replacing edges values with Bloom filters and by replacing decisions with ciphered decisions. For each edge value is created BF

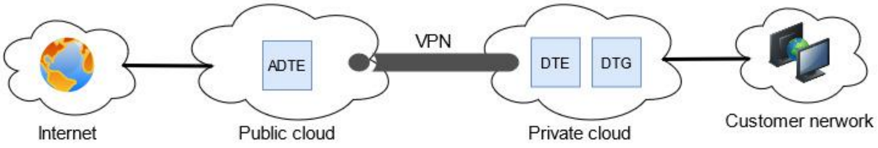


Figure 4: Arrangement of components in hybrid cloud model.

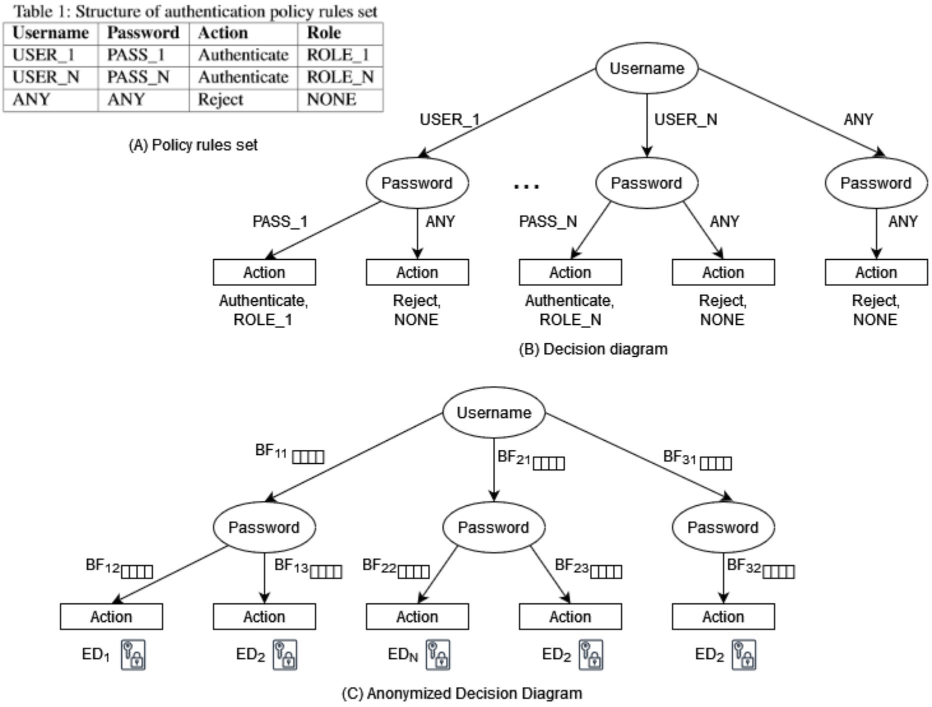


Table 1: Structure of authentication policy rules set

Username	Password	Action	Role
USER_1	PASS_1	Authenticate	ROLE_1
USER_N	PASS_N	Authenticate	ROLE_N
ANY	ANY	Reject	NONE

(A) Policy rules set

(B) Decision diagram

(C) Anonymized Decision Diagram

Figure 5: DDG process for authentication.

represented by BF$_{ij}$ where *i* and *j* are index values. Each unique decision is ciphered with secret *key* and is represented by ED$_m$ where *m* is the unique decision index. The *key* used for encryption is known only to the customer.

The UDT and ADT of access control are very similar to authentication. The main difference is the number of layers. There are four layers and each internal node can have more than two of outgoing edges. The decision trees generated in this way are fully anonymized. The BF placed on edges does not contain any information about tested values, and decisions are encrypted. For this reason, ADT does not contain any information about the original policy rules set. Thanks to the specification of BF, the decision can be taken based on data examined from packets. The functions performing the DDG tasks were created separately for each service due to differences in algorithms and specifications of chosen security services.

The ADTE is another important component of the developed system. This module works on the basis of ADT generated by DDG. Its main task is to find decision in ADT based on examined content from packets. Finding the decision in anonymized structure requires computing power. Computing the path from the root to the node labelled with *Action* is not a trivial task. On the path in ADT are placed BF, so this module must compute appropriate BF for each field in examined content from packets. For this reason ADTE is placed in the public cloud to outsource computing operations to the CSP. The public cloud resources are considered as almost infinite from clients' point of view. This module is also responsible for attach the decision to the original packet and then send it to the next module. However, the decision is not attached in its encrypted form. The encrypted decision is transmitted in the hashed format. This operation is performed to prevent cryptanalysis. If someone could only get information in encrypted form by eavesdropping on traffic between public and private cloud, then someone would be able to crack the key in a finite period of time. The hash is based on encrypted decision, original packet and secret key.

The DTE is the final component of a solution. This module is responsible for taking the final decision. Its works based on ED and UDT structures generated by DDG. The DTE operations cannot be performed in public cloud because actions carry information about original structure of policy rules set of security service. Therefore, the DTE is placed in private cloud so it may contain information about original policy rules set, and the responsibility for performing all operations is on the customer side. Taking the final decision requires computing power but much less than ADTE operations.

The specification of BF's can lead to the appearance of false positive. So the BF parameters should be selected to minimize false positive rate. However for the crucial security services like authentication cannot be allowed to choose incorrect decision. Unauthorized authentication may poses an unimaginable threat. In authentication service in this way, an attacker can use brute force attack to test many different credentials and when the false positives will occur two times for username and for password, he/she will be authenticated.

On the other hand, BF specification ensures that if one of the calculated indices based on examined content is not present in array, this value is definitely not present. Thanks to this, for each negative decision like *Reject* there is one hundred percent certainty of its correctness. To prevent unauthorized authentication, for each positive decision like *Authenticate* chosen by algorithm described above, the decision is calculated again based on examined content from packet and UDT. In this way, there is no chance that wrong decision will appear.

The final decision chosen by DTE can be forwarded to the security service. Authentication service is able to authenticate or reject user credentials based on the taken decision. Access control service can allow or deny execution of activity on resource.

Verification

To confirm the correctness of operation, the integrated system consisting of the individual components presented in the previous section was verified. To meet the demands of the presented solution for preserving privacy of selected security services in SecaaS model, the testing environment has been implemented in hybrid cloud model. The testing environment architecture is shown in Figure 6.

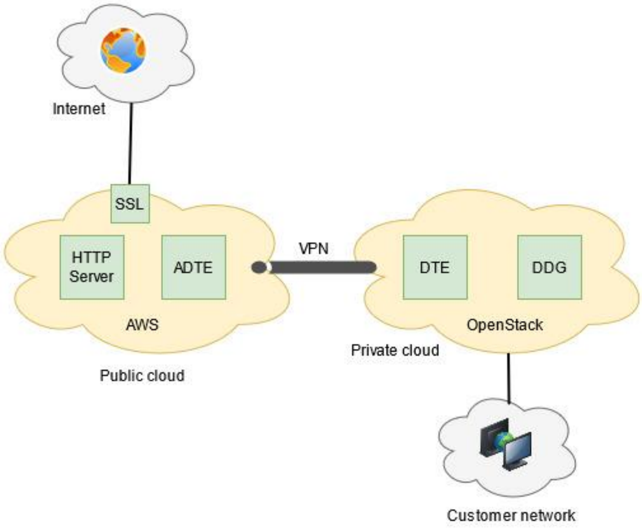


Figure 6: Architecture of the deployed system.

For the public cloud implementation, *Amazon Web Service* (AWS) was used as a cloud computing service. The private cloud was implemented in a OpenStack – the well-known open source cloud computing platform that allows to use of the concept of cloud computing on own infrastructure. The connection between both clouds was secured using *Virtual Private Network* (VPN) tunnel established with OpenVPN. Additionally, the customer network is connected to the private cloud and the public cloud has an internet gateway.

The main purpose of the presented platform is to preserve privacy of selected security services in SecaaS model. The platform should fulfill its purpose of preventing the possibility of policy leakage. Therefore, an example of security policy was implemented and verified. Figure 7 shows visualization of unanonymized decision diagram for a scenario supported access control functionality. Each node has an assigned appropriate label.

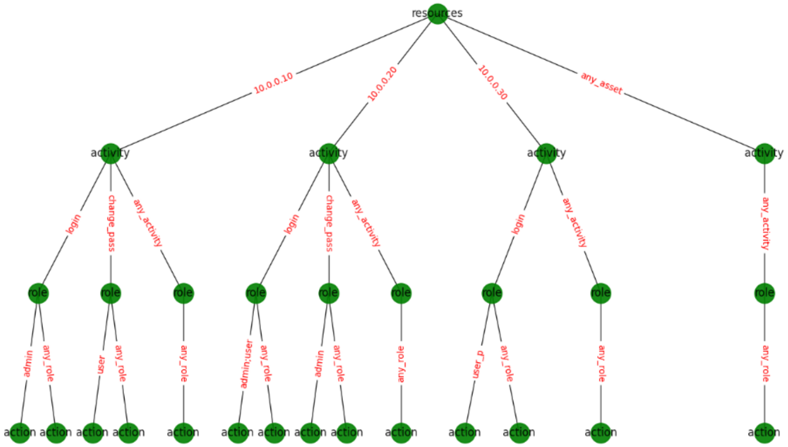
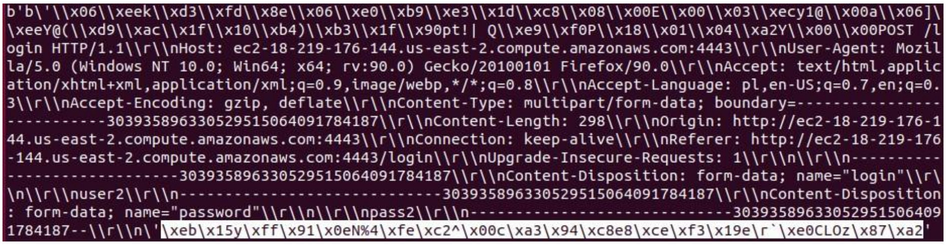


Figure 7: Access control decision diagram.

A series of functional tests was then carried out to verify security of the policies and correct operation of the platform. Correct selection of the decision for authentication and access control was proved. ADTE module correctly calculated the decision and then transferred it in hashed form with an original packet to private cloud where the final decision has been chosen. The final decision was selected based on UDT.



```
b'b\'\\x06\\xeek\\xd3\\xfd\\x8e\\x06\\xe0\\xb9\\xe3\\x1d\\xc8\\x08\\x00E\\x00\\x03\\xecy1@\\x00a\\x06]\
\xeeY@(\\xd9\\xac\\x1f\\x10\\xb4)\\xb3\\x1f\\x90pt!| Q\\xe9\\xf0P\\x18\\x01\\x04\\xa2Y\\x00\\x00POST /l
ogin HTTP/1.1\\r\\nHost: ec2-18-219-176-144.us-east-2.compute.amazonaws.com:4443\\r\\nUser-Agent: Mozil
la/5.0 (Windows NT 10.0; Win64; x64; rv:90.0) Gecko/20100101 Firefox/90.0\\r\\nAccept: text/html,applic
ation/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8\\r\\nAccept-Language: pl,en-US;q=0.7,en;q=0.
3\\r\\nAccept-Encoding: gzip, deflate\\r\\nContent-Type: multipart/form-data; boundary=----------------
-----------303935896330529515064091784187\\r\\nContent-Length: 298\\r\\nOrigin: http://ec2-18-219-176-1
44.us-east-2.compute.amazonaws.com:4443\\r\\nConnection: keep-alive\\r\\nReferer: http://ec2-18-219-176
-144.us-east-2.compute.amazonaws.com:4443/login\\r\\nUpgrade-Insecure-Requests: 1\\r\\n\\r\\n----------
-------------------303935896330529515064091784187\\r\\nContent-Disposition: form-data; name="login"\\r\
\n\\r\\nuser2\\r\\n-----------------------------303935896330529515064091784187\\r\\nContent-Disposition
: form-data; name="password"\\r\\n\\r\\npass2\\r\\n-----------------------------30393589633052951506409
1784187--\\r\\n\'\xeb\x15y\xff\x91\x0eN%4\xfe\xc2^\x00c\xa3\x94\xc8e8\xce\xf3\x19e\r`\xe0CLOz\x87\xa2'
```

Figure 8: Captured packet for the authentication service.

Tests were also carried out to see if the platform protects privacy of selected security services. The packet was captured on exit from public cloud before VPN encryption. Figure 8 shows captured packet for authentication. The highlighted part of packet is a hashed encrypted decision and the rest is an original packet. Thus it was proved that a person who is able to eavesdrop on network traffic is not able to recognize the original policy of security services.

Improvements and Discussion

The authors have shown that the UNIPRIV platform concept can be successfully implemented for various security services (not only for IDS or IPS). The tool can be used for preserving the privacy of customers' security policies when outsourcing security services to CSP. However, from the security point of view for the authentication service of the presented solution is that the passwords are stored in plaintext. Someone who has access to the system configuration can easily read the passwords of any user – what may result in a system compromise. One of the better way to store password is to store them in the form of salted hashes. In such a solution, the user is authenticated based on the calculated hash from the password and comparing it with the hash from database. For the correct calculation of the hash, the salt that is assigned to the particular user is required. The implementation of such password storage in the presented solution could result in the disclosure of the usernames of users who have accounts in the system. The reason for this is that salt for usernames that are not in the database will not be obtained.

In order to meet the security requirements and to eliminate the found limitations the architecture model of the solution has been changed and security-related functionality has been added. First of all the function of storing passwords in the form of salted hashes has been added.

Another limitation is that the return traffic cannot be route through the public cloud where the operation of finding the ciphered decision in ADT is performed. This is because the return network traffic may contain information about original structure of security policies. For authentication service the return traffic should contain information whether the user has been properly authenticated and the obtained permissions. For access control return traffic includes information if the action was performed on the asset. If the CSP is able to eavesdrop return traffic and the original packet based on which the decision was made the privacy of security policies may be violated using the traffic analysis.

To meet the limitation that the return traffic cannot be routed via public cloud in which ADTE computing power operations are performed, an improved approach was proposed. It uses multiple public clouds from different vendors. Figure 9 shows the novel architecture with multiple CSP. For tool verification, *Microsoft Azure* was added as an another cloud computing service. The figure also shows flow operations in updated solution architecture for authentication. The user establishes a secure SSL connection to the HTTP server and then sends the network packet with credentials (step 1). The username is taken from packet and sent to the private cloud to check the salt for the username (step 2). In the private cloud, the salt is extracted from the hash for the given username based on UDT. If the username is not present in the UDT, the salt is generated randomly and then sent back to the public cloud (step 3). Thanks to this approach it is possible to implement password storage in the secure form based on hashes. Next, a hash is computed based on cost factor, salt, and password, on the basis of which the decision is made by ADTE. The original packet with attached hashed ciphered decision and computed password hash is sent to the DTE component (step 4). DTE makes the final decision and conducts operations

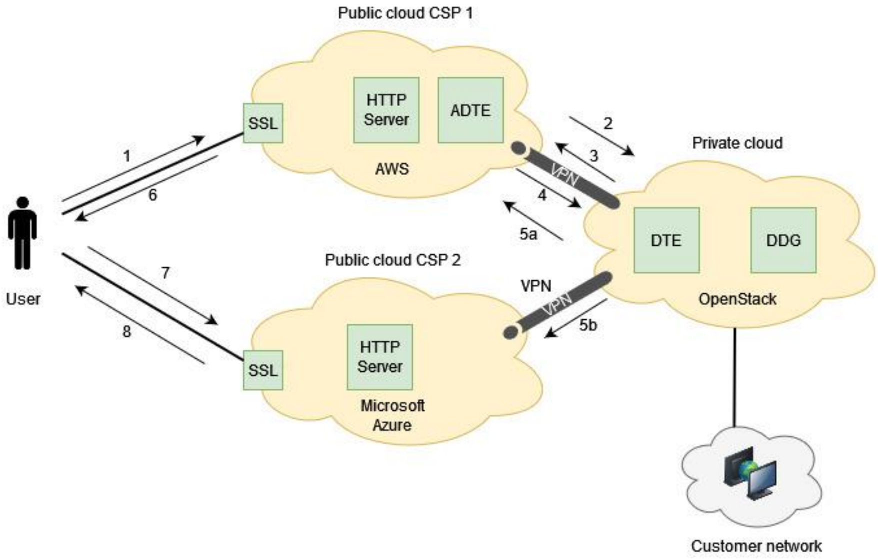


Figure 9: Flow operations in architecture of updated solution.

based on it. If the decision is to *Reject*, a 32-bytes token is generated and sent to public cloud CSP 1 (step 5a). If the authentication decision is positive, a 32-byte token is generated and sent to public cloud CSP 1 (step 5a) and the same token with the role of authenticated user is sent to public cloud CSP 2 (step 5b). In the AWS public cloud, a response is created with the generated token and a redirection to the HTTP server located in Microsoft Azure public cloud. The prepared response is sent back to the user (step 6). Thanks to redirection, the user establishes a secure SSL connection to the HTTP server placed in the second cloud and sends the request with token (step 7). Based on the token, a response is sent to the user whether he/she has been authenticated or not (step 8). Thanks to this approach, neither one nor the other CSP in able to recognize the security policies for authentication by analyzing the flowing network traffic through each public cloud. The new approach using multiple CSP and token generation could also be successfully implemented for access control service to allow bidirectional network traffic while maintaining confidentiality of policies.

The improved solution for preserving privacy of authentication in SecaaS model has been tested. The scope of the experiment included testing the correctness of the operation and measuring the time of performing calculations by DTE and ADTE, excluding the time associated with sending and receiving network packets. Time of operations performed by ADTE also includes the process of calculating the password hash. A set of 1000 credentials was prepared as input data for the experiment, where 5% were valid credentials and 95% were random usernames and passwords. To send this number of queries, a python script was prepared that uses the curl software for sending HTTP requests, which took credentials as an input argument. All responses and times have been saved to a file.

By analyzing the response outputs from the experiment, the solution was found to be working correctly. Thanks to the specification of BF and recalculation of decisions based on the UDT, the correctness of the authentication service was 100%. Table 3 presents average time it took for each of the component, to perform computing power operations. The mean time of the ADTE operation was 328 ms and was much more greater than the mean time of the DTE operation. Such a big difference is that the ADTE had to calculate the password hash. A reduction in the cost factor would certainly reduce the mean time differences.

Table 3. Average time of performing operations by ADTE and DTE.

Component	**Average time** [ms]
ADTE	328 ± 2
DTE	0.3 ± 0.7

It was also noticed that the presented tool not only preserves privacy of authentication policies, but it can also be resistant to brute force attacks while maintaining the stability of the system. This is because all computing power operations, especially related to password hash calculation are performed in public cloud where computing resources are almost infinite. However, this requires the design and implementation of an appropriate architecture inside the public cloud. For example, in the AWS cloud, this would require the implementation of an Elastic Load Balancing service [15] that distributes network traffic between multiple instances on which the HTTP server and the ADTE component would be launched. The next step would be to implement the AWS Auto Scaling mechanism,[16] which enables the dynamic creation of new instances based on CPU utilization.

Conclusions

This paper considers a form of security policies for selected network services – such as authentication and access control – to ensure privacy and protect against network traffic analysis by an unauthorized person. Thanks to this approach, network services can be successfully used in the SecaaS model without worrying about the confidentiality of security policies. The solution based on the proposed structure of security policies was implemented and tested.

The proposed solution for preserving privacy of authentication and access control in SecaaS model consists of three components. Each of them has its own functionality. The most important element is DDG which is responsible for correct creation of a decision tree based on security policies. On the basis of UDT, ADT is created by replacing the edges with BF and encrypted decisions. The ADT created in this way is sent to ADTE where decision is made based on data examined from network packets. Then, the hashed and encrypted decision along with the original packet is sent to the DTE, where the final decision is made. To work properly, the tool requires the use of a hybrid cloud model consisting of public and private clouds.

The verification of the functionality of the solution was done. However some security issues/limitations were identified in the original idea – such as incorrect password storage for authentication and return network traffic restriction. An attempt based on multiple public clouds from different providers was made to improve the tool by eliminating security flaws and limitations for authentication. The improved tool has been tested taking into account security and performance. The new approach with the use of multiple CSP and token generation can also be successfully implemented for access control service.

Future work of this solution can be focused on examining performance depending on the number of security rules, used hash function, and parameters used to create BF. The functionality of the solution for other type of authentication, in particular MFA and other types of access control, can also be investigated.

Acknowledgements

This work has been funded by the European Union's Horizon 2020 Research and Innovation Programme, under Grant Agreement no. 830943, project ECHO (European network of Cybersecurity centres and competence Hub for innovation and Operations).

References

[1] Atle Refsdal, Bjørnar Solhaug, and Ketil Stølen, "Cybersecurity," *Cyber-Risk Management,* Chap. 4 (Springer, 2015), 29–32.

[2] Steve Morgan, "2021 Report: Cyberwarfare in the C-suite," 2021, https://cybersecurityventures.com/wp-content/uploads/2021/01/Cyberwarfare-2021-Report.pdf.

[3] Mahesh Shankarwar and Ambika Pawar, "Security and Privacy in Cloud Computing: A Survey," *Advances in Intelligent Systems and Computing* 328 (2015).

[4] Vijay Varadharajan and Udaya Tupakula, "Security as a Service Model for Cloud Environment," *IEEE Transactions on Network and Service Management* 11, no. 1 (2014): 60-75.

[5] Tytus Kurek, Marcin Niemiec, Artur Lason, "Taking back control of privacy: a novel framework for preserving cloud-based firewall policy confidentiality," *International Journal of Information Security* 15, no. 3 (2016): 235–25.

[6] Amir R. Khakpour and Alex X. Liu, "First Step toward Cloud-Based Firewalling," *2012 IEEE 31st Symposium on Reliable Distributed Systems*, *Irvine, CA, USA*, 2012.

[7] Maciej Mencner and Marcin Niemiec, "Impact of Bloom Filters on Security and Efficiency of SecaaS Services," *Multimedia Communications, Services and Security,*

MCSS 2020, Communications in Computer and Information Science, vol. 1284, 2020, pp. 154–167.

[8] Tytus Kurek, Marcin Niemiec, and Artur Lason, "First step towards preserving the privacy of cloudbased IDS security policies," *Security and Communication Networks* 8, no. 18 (2015).

[9] Tytus Kurek, Marcin Niemiec, and Artur Lason, "Intrusion Prevention System Decision Diagram in Security-as-a-Service Solutions," *International Conference on Multimedia Communications, Services and Security,* 2017.

[10] Tytus Kurek, Marcin Niemiec, Artur Lason, and Andrzej R. Pach, "Universal privacy-preserving platform for SecaaS services," *International Journal of Network Management* 27, no. 5 (2017).

[11] Burton H. Bloom, "Space/time trade-offs in hash coding with allowable errors," *Communications of the ACM* 13, no. 7 (1970).

[12] Arpit Mishra, "How do giant sites like Facebook and Google check Username or Domain availability so fast?" *Hackerearth*, 2017, https://www.hackerearth.com/blog/developers/how-websites-check-username-availability-quickly.

[13] Dipankar Dasgupta, Arunava Roy, and Abhijit Nag, *Advances in User Authantication* (Springer, 2017).

[14] "What is access control?" C*itrix*, 2022, https://www.citrix.com/pl-pl/solutions/secure-access/what-is-access-control.html.

[15] "Elastic Load Balancing," *AWS*, 2022, https://aws.amazon.com/elasticloadbalancing.

[16] "AWS Auto Scaling," *AWS*, 2022, https://aws.amazon.com/autoscaling.

About the Authors

Piotr Jurgała received the MSc degree in ICT from the AGH University of Science and Technology, Poland in 2021. Currently he works as an IT Security Specialist, where he deals in particular with the implementation and operation of the SIEM platform.

Tytus Kurek received the MSc degree and the PhD degree in Telecommunications from the AGH University of Science and Technology, Krakow, Poland, in 2011 and 2018 respectively. Currently he works as a Product Manager at Canonical Ltd., the company behind Ubuntu Operating System. His research interests focus on Network Functions Virtualization, 5G Networks, Container Network Functions and Unikernels.

Marcin Niemiec is working as a university professor at the Institute of Telecommunications, AGH University of Science and Technology. His research interests focus on cybersecurity and data protection, especially security services, symmetric ciphers, network security, intrusion detection, and quantum cryptography. He has actively participated in 6th and 7th FP European programs (ePhoton/ONE+, BONE, SmoothIT, INDECT), Horizon 2020 Framework Programme (SCISSOR, ECHO), Eureka-Celtic (DESYME), and many national research projects. He co-authored over 90 publications and reports.