COMPUTERS AND INFORMATICS
2757-8259 2023, Volume 3 Issue 2
Research Article

Implementation of smart saver, logged-in device protector

Gyudong Park*
Agency for Defense Development, ADS&TR Institute-C2 Systems PMO, Seoul, South Korea, iobject@add.re.kr
Hocheol Jeon
Agency for Defense Development, ADS&TR Institute-C2 Systems PMO, Seoul, South Korea, hcjeon71@add.re.kr
Kyungshik Yi
Telefield, Inc., Special Business Division, Seongnam, South Korea, baatar@telefield.com

Submitted: *22.02.2023*
Accepted: *08.06.2023*
Published: *31.12.2023*

** Corresponding Author*

Abstract:

Unauthorized access by malicious user could be very dangerous to all information systems. As a technical solution to prevent this, IAM (Identity and Access Management) is available. Many systems trust users who passed the system's authentication or log-in until log-out. However, IAM operates passively by traffic between devices and systems. Because IAM can't see the user of the device, it considers all traffic from the device after log-in is generated by the log-in user. Therefore, a logged-in and unattended device could be a security vulnerability of the system because it can be used by a malicious user nearby. Currently, many systems entirely rely on individual users to protect their devices. However, this study suggests an idea of technical solution called smart saver to protect the logged-in devices more securely. The smart saver triggers screen saver immediately upon detection of absence or change of the logged-in user using camera sensor of the device. For this, smart saver extracts and uses user's appearance features and tries not to violate the recent trend of strengthening identity information protection. And this study shows the feasibility of smart saver through experiments.

Keywords: *GDPR, IAM, Logged-in device protector, Smart saver, Zero trust*

© 2023 Published by peer-reviewed open access scientific journal, C&I at DergiPark (https://dergipark.org.tr/tr/pub/ci)

Cite this paper as: Park, G., Jeon, H. & Yi, K., Implementation of smart saver, logged-in device protector, *Computers and Informatics*, 2023, *3*(2), 57-66

1. INTRODUCTION

Unauthorized access by malicious user could be very dangerous to all information systems. Because some or all valuable information of the systems could be stolen, falsified, or destroyed by it. Many systems usually trust the users who passed the system's authentication or log-in until log-out. And if a user leaves the logged-in device unattended for a while, someone nearby can use it to make their own benefit or to damage the system.

Therefore, the logged-in devices must be very carefully protected. However, many systems place the responsibility of device protection entirely on individual users. Those system's users are educated or trained to lock, log-out, or at least activate screen saver if they have to leave their devices even for a moment. However, humans are not perfect, and they can make a mistake sometimes. Of course, a function which activates screen saver automatically if there is no action for a while, like 1 minute, also can be helpful. But it may be long enough time for someone to take control of the device. And the risk increases in proportion to the number of users or devices of the system.

To relieve the burden of individual users and increase the security of systems, we suggested an idea of smart saver in the previous study [1]. These days, most devices are equipped with one or more sensors such as camera. Smart saver extracts some approximate appearance information from the user using camera during log-in process without requiring additional actions from the user. And it continuously observes the user until logout. And it activates the screen saver instantly when the user's absence or change is detected.

User appearance information can be a kind of user identity information. And recently, the trend of strengthening privacy protection makes it difficult to collect and use user identity information than before. Smart saver also uses appearance information that can be personal private information, but it can be free from the related regulations by collecting it at very rough level just enough to detect the change of the user. And it uses the information until log-out and delete them all after that. And it doesn't exchange the information with remote server through network. In these ways, the risk of personal information leakage could be removed or reduced a lot.

This paper develops the idea of smart saver further and shows the feasibility of the idea through actual implementation and some experiments. For this, we present related works in chapter 2, and explain the procedure and structure of smart saver in chapter 3. And we show and explain the core functions' algorithms and the experiment results in chapter 4. And chapter 5 is conclusion.

2. RELATED WORKS

The main purpose of system security is to ensure all accesses by authorized users and to block all accesses by unauthorized users. In the past and even now, many systems adapted perimeter-based security approach to protect themselves from unauthorized users. Therefore, a user who had passed authentication could access all resources of the system without additional controls [2]. However, this approach was not sufficient because there was no way to prevent the further actions by an attacker who broke through the perimeter.

After that, zero trust was suggested as a new system security approach. Zero trust constantly doubts both inside and outside users of the system. Zero trust focuses on accessing resources in a secure manner, enforcing rigorous access controls, and continually inspecting, monitoring, and logging all traffic between users and systems [3].

These all-system security approaches rely entirely on IAM. As an essential solution for system security, IAM includes identity management, authentication, authorization, and access control functions [4]. And IAM determines the user's trust level and controls the user's access to system resources according to the user's authority. For this, IAM should collect and save the identity information of the users previously. So, identity information should consist of persistent properties.

IAM is already matured and proved solution [5]. However, IAM alone is not perfect to implement system security. IAM operates passively by traffic between users and systems. IAM can see only traffic, but can't see owner of the traffic. Therefore, IAM can't recognize the change of the user and can't protect logged-in devices from unauthorized users perfectly. Most systems depend on individual users to protect their devices.

Nevertheless, still many researches and developments are focusing on IAM to increase the system security, and some of them like [6] and [7] require more identity information to improve IAM. However, demands for protection of personal identity information are increasing recently. The European Union (EU)'s General Data Protection Regulation (GDPR) could be a typical example of that trend. GDPR is a data protection legislation about processing, storing, managing data of people within EU [8]. Therefore, it's more difficult to collect and use identity information than before. And there has been a lot of studies to improve legacy systems to keep pace with this trend or regulation. However, most studies are focusing on how to collect and use the user's private information more securely and carefully like [9] and [10].

In this study, we suggested a new idea and a technical solution called smart saver to cover the vulnerability which can't be covered by IAM. And we considered the current trend of strengthening the private information protection, so let smart saver use very lough information of user appearance and don't save the information after use. And smart saver can use variable appearance features of the users, and it can extend the concept of zero trust to include outside of the IAM, because it doubts users continuously besides traffic.

Meanwhile, there was another attempt to use the user appearance features extracted at login for subsequent authentications [11]. But it was just for the additional or temporal authentication required, so it's different from smart saver which consistently or periodically checks the users.

3. PROCEDURE AND STRUCTURE

In the previous study, we presented operation procedure and high-level structure of smart saver like shown in Figure 1 and Figure 2. During the log-in, smart saver takes some pictures of the user, and extracts some features of the user's appearance roughly, and saves them in the memory, not disk. And it continuously or periodically takes pictures and extracts features of the current user, and compares them with the log-in users until log-out. And if log-in user's absence or change is detected, smart saver activates screen saver immediately. And smart saver consists of three main functions such as feature extractor, feature saver, and feature comparator. Camera and screen saver can be provided by the device.

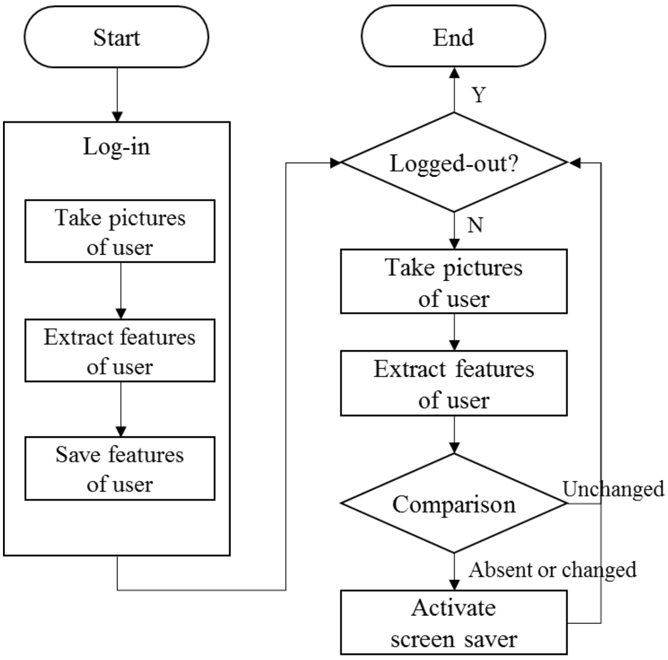

Figure 1. Smart saver operation procedure

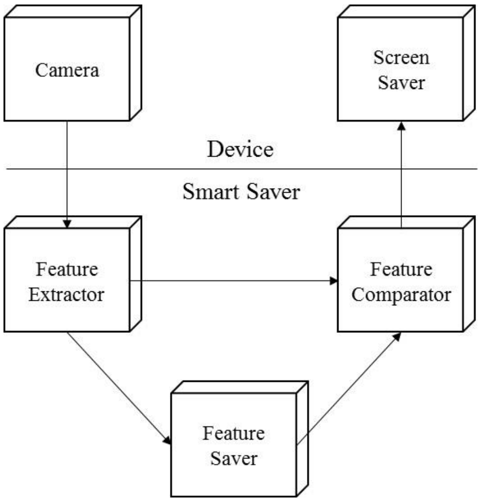

Figure 2. Smart saver high level structure

4. IMPLEMENTATION AND EXPERIMENTATION

This study implemented smart saver using rapid prototyping. And this study used python as a programming language and OpenCV as a vision function library. And we explain two most important functions of the smart saver, exception handling, and experiment results in this chapter.

4.1. Feature Extractor

One of the most important functions of smart saver is feature extractor. We selected face, skin, and glasses as features for smart saver implementation of this study. The algorithm of the feature extractor is summarized as follows.

Algorithm 1. Feature Extractor

```
1   Function ExtractFeatures(Image frames[])
2     FaceRecognizer fr ← createFaceRecognizer();
3    SkinRecognizer sr ← createSkinRecognizer();
4     GlassesDetector gd ← createGlassesDetector();
5     Image images[]; // image array to learn the user face
6     int i ← 0; // index for faces & images
7     boolean glasses = FALSE; // whether wearing glasses
8     for frames[i] <> NULL  do
9       Image frame ← frames[i];
10      Image faceImage ← extractFace(frame);
11      Image grayScale ← convertScale(faceImage);
12      images[i] ← grayScale;
13      i++;
14    end for
16    fr.learnFace(images);
17    sr.learnSkin(images);
18   if gd.detectGlasses(images) is TRUE then
19         glasses ← TRUE;
21   end if
22    saveFeatures(fr, sr, glasses);
23  end Function
```

Many other features also can be used by smart saver. For example, we considered face mask as a candidate because a lot of people wear it after covid-19. But we didn't use it because it couldn't and shouldn't be enduring trend. And because many people take off their masks momentarily to eat or drink something while using the device, and this let smart saver misunderstand right user as wrong user. However, clothes can be a very useful feature because many people change clothes every day, and variable features can be more secure than constant features as credentials. We will include these kinds of features more in next studies.

4.2. Feature Comparator

The other important function of smart saver is featuring comparator. The summarized algorithm is as follows.

Algorithm 2. Feature Comparator

```
1   Function CompareFeatures(LoginUserInfo info)
2    FaceRecognizer fr ← createFaceRecognizer();
3    SkinRecognizer sr ← createSkinRecognizer();
4     GlassesDetector gd ← createGlassesDetector();
5    while status is login do
6      Image frame ← captureCurrentFrame();
7     Image userFace ← fr.extractFace(frame);
8      if userFace is NULL then
9          callScreenSaver();
10     Else
11       Image grayScale ← convertScale(userFace);
12       SkinInfo skin ← sr.extractSkinInfo(grayScale);
13       boolean glasses ← gd.extractGlasses(userFace);
14     end if
15     if (info.getFace() <> userFace) || (info.getSkin() <> skin) || (info.getGlasses()
    <> glasses) then
16      callScreenSaver();
17     end if
18     Sleep(for a while);
19    end while
20  end Function
```

This function uses the login user's feature information as a parameter (line 1), which was saved before using the function of Algorithm 1 (line 22).

4.3. Exception Handling

Do not block the right users is important as much as block the wrong users. And there could be many exceptions that can lead to misunderstanding the right users for the wrong users. Therefore, we had to add some exception handling functions to smart saver. For example, glasses are accepted as one of the personal appearances features these days. They can be a part of face. So, it is very reasonable to judge that a person who wear glasses and a person who doesn't wear glasses are different. But there are many people who often take off glasses and put them on again, especially among old people. For these people, it would be desirable that smart saver allow them to choose whether to use the glasses as a feature or not.

4.4. Experimentation

Smart saver requires quite a lot of pictures to extract or learn features of the log-in user. This study takes 100 photos, and converts them to gray scale, and cut out the part outside ROI (Region of Interest) from each photo as shown in figure 3. And smart saver learns and processes these photos to extract features of log-in user and store them in its memory.

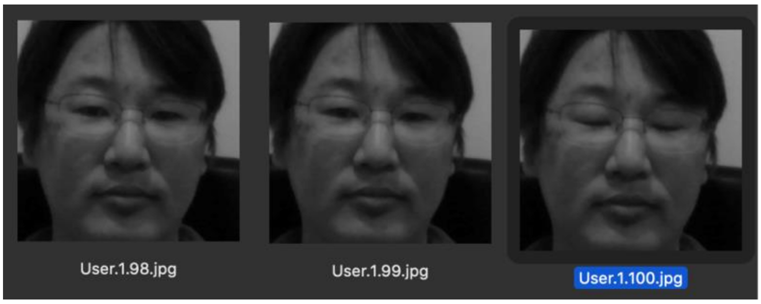

Figure 3. User photo taking and conversion example

In the subsequent experiment, when the user continued to face the front, smart saver could detect that he was a logged-in user well as shown in figure 4.

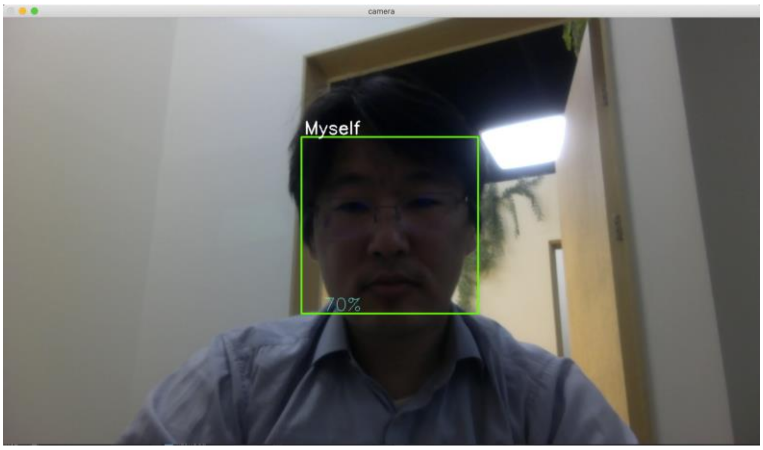

Figure 4. Frontal face user example

And in another experiment, even if the user turned gaze slightly, smart saver was able to judge that he was the same user. At this time, we asked the user to take off his glasses and smart saver detected it also very well. You can see the example of the result in Figure 5.

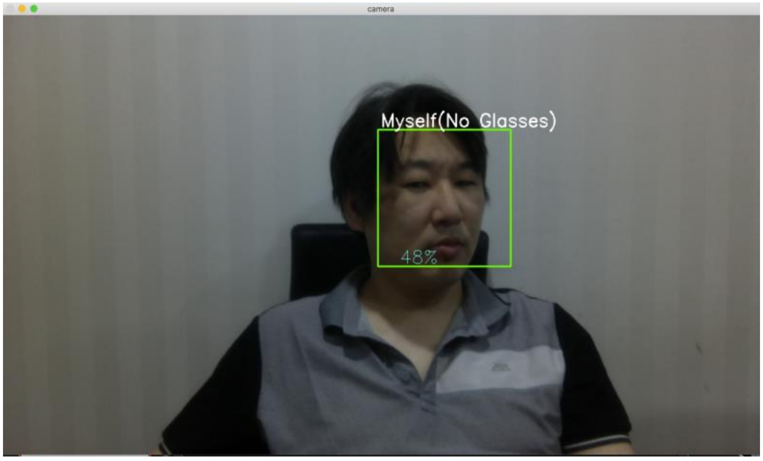

Figure 5. Slightly turned user example

And after that, when we replaced and removed the user, smart saver recognized those situations immediately and activated the screen saver successfully. You can see each example in Figure 6 and Figure 7.

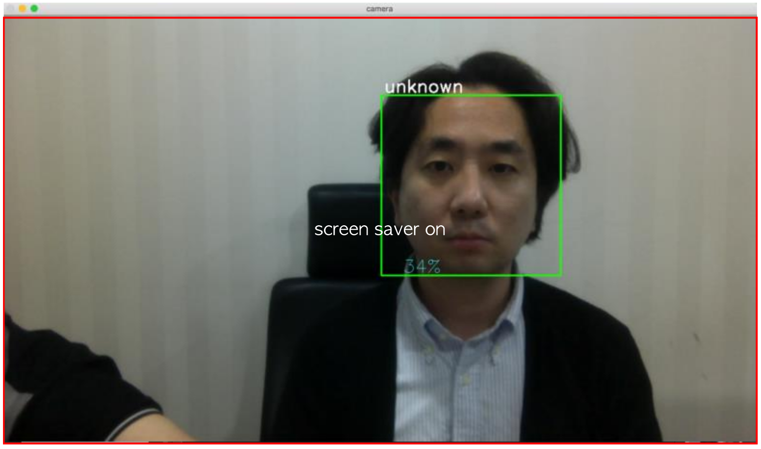

Figure 6. User replacement example

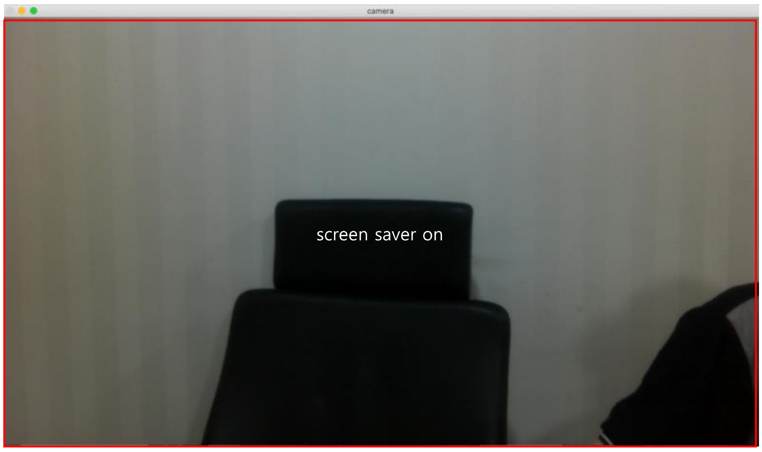

Figure 7. User removal example

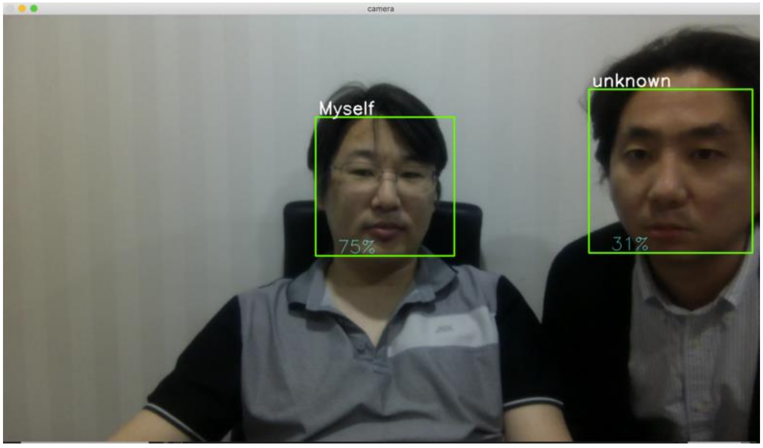

Figure 8. Two user’s examples

In a series of experiments, we considered some exceptions additionally. For example, sometimes two or more users can share a device for explanation or discussion. Therefore, there may be more than one person in a screen. In this case, Smart Saver could recognize the logged-in user and others successfully like in Figure. 8. Based on this capability, smart saver can implement and provide system specific policies for this situation.

Through the above experiments, we demonstrated all main functions of smart saver. Smart saver depends on OpenCV to extract and learn user appearance features and to measure the similarity between log-in user and current user. And this study determined the number of pictures and similarity threshold used by smart saver through the following experiments.

This study tested 25, 50, 75, 125, and 150 pictures to extract and learn log-in user’s features. And per each, this study measures the similarity 10 times with the same user [Table Ⅰ] and different user [Table Ⅱ]. As a result, we can get the best performance when we use 100 pictures. And we set the threshold to determine same user or different user to 45. It is a slightly tuned value to be lower than the average value between 60 and 37. Because we want to decrease the risk to determine the right person as a wrong person.

Table 1. Similarity between same users

Number of Photos	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Avg.
25	50	58	57	43	60	54	51	52	50	53	53
50	54	54	58	56	53	57	57	55	52	55	55
75	56	58	62	59	51	56	53	60	59	55	57
100	58	62	59	58	64	60	61	56	61	56	60
125	54	54	60	55	50	51	55	61	58	55	55
150	58	61	54	59	54	55	60	61	62	59	58

Table 2. Similarity between different users

Number of Photos	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7	Test 8	Test 9	Test 10	Avg.
25	37	37	32	42	43	35	43	45	40	42	40
50	43	43	39	42	41	42	40	38	39	40	41
75	38	41	41	38	31	40	37	41	39	40	39
100	35	37	34	40	37	33	42	43	39	32	37
125	43	42	40	44	29	41	43	40	32	38	39
150	39	30	32	29	34	38	41	35	33	31	34

5. CONCLUSION

Building perfect system security is not easy challenge. There are many known or unknown vulnerabilities in the system. And attackers and defenders always compete fiercely with each other to discover, use, or eliminate the more vulnerabilities of the system before their opponents. Some of the vulnerabilities are very difficult to eliminate technically. The logged-in and unattended device presented in this study could be a representative example of such vulnerabilities. Until now, many systems have completely relied on the individual user to protect it. But this study further developed and validated the idea of smart saver that could technically eliminate or reduce the possible risks from the vulnerability.

Generally, security imposes extra burden and inconvenience on users, and requires some sensitive private information of the users. Therefore, the introduction of new security measures is always faced with complaints and resistances from users. Currently, many IAM-based security solutions require more credentials and burdens to the users to build multi-factor authentication to increase the security of the system. However, smart saver doesn’t collect any detailed identity information and additional actions from the users. Therefore, smart saver can be applied more easily than others. And we summarize the contributions of this study as follows.

- *Identify a blind spot of IAM, a logged-in and unattended device*
- *Propose a technical solution with a low application barrier to eliminate the blind spot*
- *Propose some exception handling functions to improve the solution*

The smart saver presented in this study is currently at the prototype level and needs a lot of improvement for practical application. Therefore, if there are needs, we will develop smart saver further through follow-up researches. We are considering adding new and more features, sensors, exception handling, and continuous learning to improve the function and performance of smart saver.

Acknowledgment

This work was supported by the Agency for Defense Development Grant funded by the Korean Government (911194202).

REFERENCES

[1] Park, G., A Proposal to apply smart saver to prevent identity theft, 2022 Spring Conf. Korean Society for Internet Information, Apr. 2022

[2] Collier, Z. A., Sarkis, J., The zero-trust supply chain: Managing supply chain risk in the absence of trust, *International Journal of Production Research*, Vol. 59, No. 11, 2021, pp. 3430-3445

[3] Kerman, A., Borchert, O., Rose, S., Implementing a zero-trust architecture, Draft, National Cybersecurity Center of Excellence (NCCOE), NIST, Mar. 2020

[4] Pol, V. J., Identity and access management tools, *International Journal of Trend in Scientific Research and Development (IJTSRD)*, Vol. 3, Issue 4, May-Jun 2019, pp. 796-798

[5] Burhop, D., Greenberg, M., Maxwell, J., Identity and Access Management, I AM Who I Say I AM (WHITE PAPER), Virginia’s Council on Technology Services Identity and Access Workgroup, Jun. 20, 2007

[6] Mohammed, I. A., Identity Management Capability Powered by Artificial Intelligence to Transform the Way User Access Privileges Are Managed, Monitored and Controlled, *2021 International Journal of Creative Research Thoughts (IJCRT)*, Vol. 9, Issue 1, Jan. 2021

[7] Kunza, M., Puchta, A., Groll, S., Fuchs, L., Pernul, G., Attribute Quality Management for Dynamic Identity and Access Management, *Journal of Information Security and Applications*, Nov. 2018

[8] Zaeem, R. N., Barber, K. S., The effect of the GDPR on privacy policies: recent progress and future promise, *ACM Transactions on Management Information Systems*, Vol. 12, No. 1, Article 2. Dec. 2020

[9] Haque, A. B., Islam, A. K. M. N., Hyrynsalmi, S., Naqvi, B., Smolander, K., GDPR compliant blockchains-A systematic literature review, *IEEE Access*, Vol. 9, Apr. 2021

[10] Li, H., Yu, L., He, W., The impact of GDPR on global technology development, *Journal of Global Information Technology Management*, Vol. 22, No. 1, 2019.

[11] Park, S., Yoon, S., Jung, E., Yang, J., Method and apparatus for controlling authentication state of electronic device, US Patent, No. US 2015/0288681 A1, Oct. 8, 2015.