Proposed On-site Document Sharing System using FIDO

Mingyu Lee
Soonchunhyang University
Department of Information Security Engineering
South Korea
cpssmgyu@gmail.com

Hanbyeol Kang
Soonchunhyang University
Department of Information Security Engineering
South Korea
khbfighting@gmail.com

Junbeom Kwak
Soonchunhyang University
Department of Information Security Engineering
South Korea
junbeomkwak@gmail.com

Donghyun Kim
Soonchunhyang University
Department of Information Security Engineering
South Korea
kimdong7596@naver.com

Hyeonho Jeong
Soonchunhyang University
Department of Information Security Engineering
South Korea
bestonbeat@gmail.com

Jung taek Seo*
Soonchunhyang University
Department of Information Security Engineering
South Korea
seojt@sch.ac.kr

ABSTRACT

With the advancement of information and communication technology, online file exchange such as document delivery, business reporting, and submission of reports within or between companies has been widely used. With the increase in convenience, however, security threats have also increased due to the high risk of external leak of files exchanged online. Although companies employ network separation by building an internal corporate network, this has a drawback, i.e., files cannot be used outside such as a work-from-home environment. To solve this problem, this paper proposes a secure file sharing system using the Fast IDentify Online (FIDO) technique. This service is expected to contribute to the improvement of file sharing security because it can encrypt and decrypt files through FIDO authentication when viewing internal files from outside.

CCS CONCEPTS

• Security and privacy → Security services → Authentication → Biometrics

KEYWORDS

File Encryption/Decryption, FIDO, Security

1 INTRODUCTION

With the advancement of information and communication technology, files are exchanged for business purposes through the Internet. As a result, convenience has increased, but so have security threats. Problems of the currently used document sharing include sharing of key matched with documents in advance, exposure of the key to an open place, or use of easily guessed passwords. These problems may give rise to vulnerability such as social engineering technique and man-in-the-middle attack, which lead to document leak. If documents are leaked, contents are exposed, which can wield a grave impact on companies. Although not only important files but also personal information or files containing personal information should be encrypted and stored according to the Personal Information Protection Act, the survey conducted by EST Security in 2018 showed that nearly 70% of employees experienced internal document leak (Fig. 1), suggesting that secure internal document measures are not in place in work environments.

To solve these problems, this paper proposes a file sharing system based on FIDO2 and file encryption and decryption. A file encrypted through the system is transmitted online. Even if the file is leaked, the content in the document is not exposed because the file can be decrypted only after biometric authentication through FIDO 2. The proposed system is divided into internal and external programs. The internal system provides member sign-up, login, encryption, and decryption functions. On the other hand, the external system provides only limited functions such as login, decryption, and viewer to increase security against file leak. This service is highly simple and secure because it does not need direct key generation, sharing, and memorization due to the use of the designated program and biometric authentication.

*Corresponding Author

ICEA 2020, December 12-15, 2020, Gangwon, Republic of Korea
© 2020 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-6843-8/20/10. . . $15.00
https://doi.org/10.1145/123_4

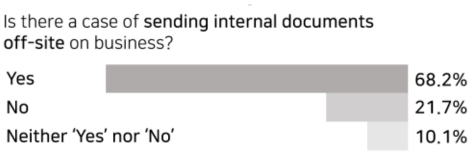

Figure 1. Experience of internal document leak in workplaces

The rest of this paper is organized as follows: Section 2 presents an overview of FIDO, which is used to authenticate the proposed program; Section 3 describes related studies on system development using FIDO; Section 4 explains the design of the proposed internal document sharing system using FIDO, components of the system, and scenarios of program uses; Finally, Section 5 presents the conclusions and suggests future studies.

2 FIDO OVERVIEW

2.1 FIDO

Developed by the FIDO Alliance, FIDO is one of the next-generation authentication techniques for biometric information such as fingerprints, iris, face, and vein of users in place of existing password methods to build highly secure, convenient authentication systems in online environments and devices such as smartphones and notebooks. The FIDO technology is divided into two: Universal Authentication Framework (UAF) and Universal 2nd Factor (U2F).

FIDO UAF facilitates the registration of users to online services through local authentication mechanisms such as fingerprints, face, and voice recognition of users. It also supports multi-factor authentication, which improves security as well as convenience without the need to enter a password in every authentication after registration.

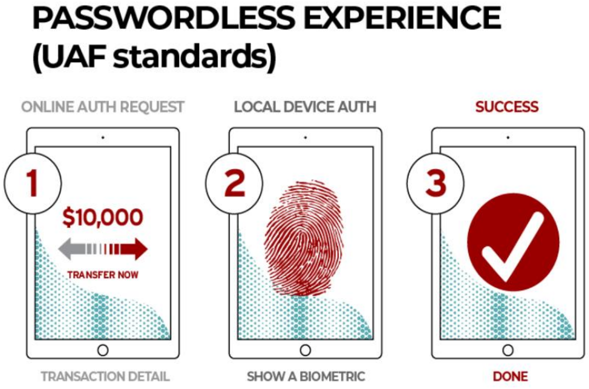

Figure 2. FIDO UAF

FIDO U2F provides two-factor authentication, which requires additional authentication with FIDO U2F devices such as FIDO security key after login through ID and password.

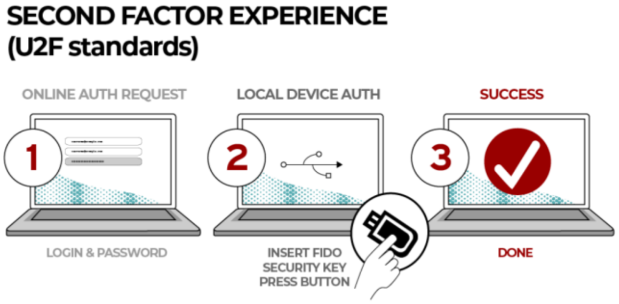

Figure 3. FIDO U2F

2.2 FIDO 2

FIDO 2 provides an authentication platform by expanding existing technologies. Unlike existing systems that supported biometric authentication in a mobile environment only, FIDO 2 expanded biometric authentication to a web browser environment. For personal computers, a biometric authentication login method called Windows Hello in the Windows operating system was adopted so that authentication is now possible in a web browser via a link with FIDO 2. FIDO 2 was also supported in Chrome, Firefox, and Edge by expanding its scope.

The system in this study also used FIDO 2 in the web part. When a security key is used in a personal computer, fingerprint authentication can be done through Windows Hello; based on the result, authentication is possible on a web page.

3 RELATED WORKS

In recent years, authentication has changed from user authentication such as password, public certificate, one-time password, and credit card to personal authentication such as fingerprint, iris, and face ID. User authentication cannot identify whether the user has the authorization or the right key if the key is shared. In contrast, personal authentication can identify a person accurately because authorization is based on the unique features of a person such as fingerprint and iris. Thus, several studies have been conducted on authentication technique such as FIDO.

[1] discussed privacy infringement caused by the collection of biometric information during biometric authentication. The knowledge-based technique has low security due to the easy exposure of authentication means, and the possession-based technique causes an increase in implementation cost for service uses and low user convenience. To solve these problems, FIDO, a biometric information-based technique, was used. Since FIDO employs biometric information on the user's devices, it can solve the problems above as the personal biometric information is not stored in a server.

In [2], the presence of vulnerability of access to existing military information systems due to forgery of user authentication data was identified, and access vulnerability was solved by performing user authentication to which inter-verification technique and blocking and obfuscation of application programming interface forgery and tampering were applied after building a FIDO 2-based secure repository of a web browser.

In [3], problems caused by changing the use of a password from an existing public authentication system to a user authentication method were solved by proposing a FIDO-based public authentication system.

Service development using FIDO has been carried out in various fields including [1], [2], and [3]. Note, however, that no studies have been conducted on the combination of FIDO with file encryption systems. Thus, this study proposes a service to solve the aforementioned problems.

4 PROGRAM DESIGN AND PLANNING

This study proposes a document sharing system that is secure and convenient by using FIDO. The structure of the proposed system has five components: web, app, server, FIDO server, and program (internal and external). The FIDO server was interlocked to add the FIDO function to existing file encryption systems.

The encryption algorithm in the encryption system employs the Advanced Encryption Standard (AES) algorithm, which is secure, fast, and recommended by the US Department of Defense. For the encryption key management, the Rivest–Shamir–Adleman algorithm is used through the Python cryptography package in this program because keys need to be exchanged between the program and server anytime, anywhere via user login and decryption. A user and the server have their public and private keys. When signing up for the service, a user exchanges his/her public key with the server. Since the encrypted file and encryption key are stored in the server by encrypting them with the server's public key during the use of the encryption function, they are secure even if they are leaked during the communication between the program and server because nobody can decrypt the file and key except for the server. The decryption function is also secure because the encrypted file and encryption key are sent to the user after encrypting them with the public key of the user, so no one can decrypt them except for the user.

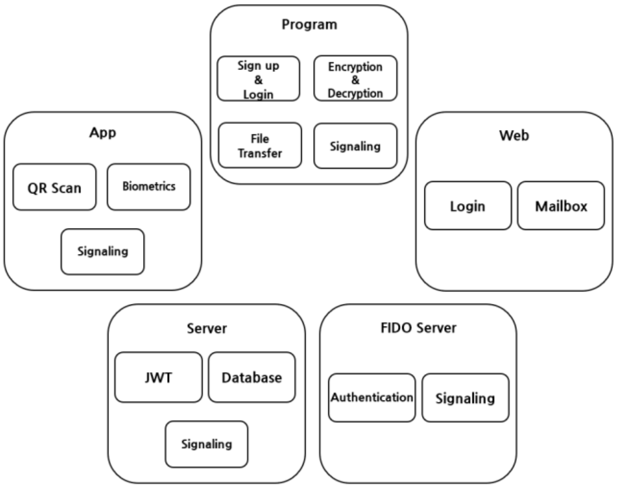

Figure 4. Structure of the proposed system

4.1 Web

As one of the components of the proposed system, the web provides login and mailbox functions. It is interlocked with the FIDO server for biometric authentication, and users who own the security key can log in using FIDO authentication after registering their fingerprints to Windows Hello. The mailbox is used to exchange encrypted documents. Although it can be used internally, it is designed to check document details through the viewer after downloading and decrypting encrypted files in the inbox when using the external program.

4.2 Android Application

Distributed by the company administrator, the app is used in sign-up, login, and FIDO authentication through biometrics during encryption and decryption in the program. The app is developed using the Android Studio development toolkit based on the Android operating system. During sign-up, information that can identify the user is only provided; the password is not used. It gives a convenient function for users through the alarm that requests biometric authentication when the user employs the program function such as encryption and decryption. Since biometric authentication through the app is performed using the FIDO protocol, the app can test whether the biometric information is matched with the user's biometric information stored in the device.

4.3 Server

The server plays the role of an intermediate using the signaling function that exchanges requests and replies during communications between the app and FIDO server, the program and FIDO server, and the web and FIDO server. The database of the server stores user information, *E_file*, and *E_key* and issues and manages Java Web Tokens (JWTs) whose expiry date is one day for user identification and secure information exchange. As a token used for authorization between client and server, JWT has all the information required including basic information of the token, information to be delivered, and signature that verifies the token. It is self-contained and easily exchangeable between two entities. JWT is issued to the user from the server during login to the program, thereby verifying the user authorization when using the encryption and decryption function and saving the server resource because only the token is verified rather than the session.

4.4 Internal Program

The internal program refers to a program used inside the company. Distributed by the company administrator like the app, this program is implemented by Python language. It includes sign-up, login, file encryption and decryption, file transfer, and biometric authentication. Its main modules are *qrcode* for Quick Response (QR) code generation, *requests* for information transmission such as signaling and key exchange, and *jwt* for JWT issuance.

The program sign-up process is shown in Fig. 5. The name, ID, group code, and company code provided by the user are stored in their variables, thereby generating the QR code using the *qrcode* module. After installing the application in his/her device, the user scans the QR code generated through the QR code scan. The user information and sign-up signals are then transmitted to the server. The server stores the user information in the database, performs the FIDO authentication between the device app and FIDO server, and registers the user fingerprint. Upon successful authentication, the sign-up is complete, which is then notified to the program, and the public keys are exchanged for encryption.

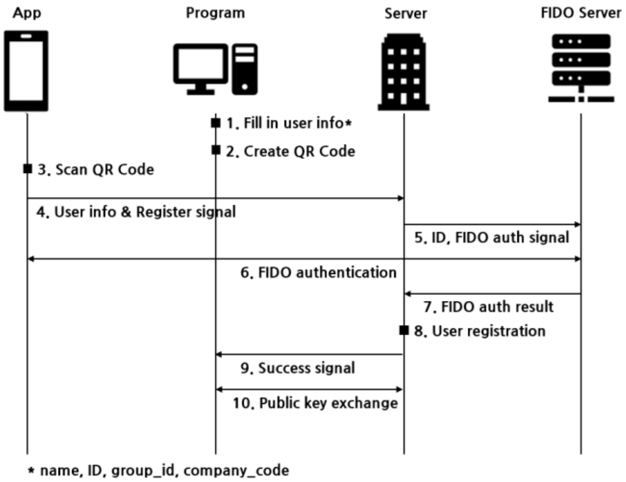

Figure 5. Internal program registration

In the program login function, the Python module *jwt* is imported and used for user identification and secure information exchange. The expiry date of JWT is set to one day, but it can be customized according to the intention of the administrator.

The program login process is shown in Fig. 6. After the sign-up process in Fig. 5, the user enters his/her ID and clicks the login button. Once the server receives the successful authentication signal after performing biometric authentication with the FIDO server through the notice from the registered device app, JWT is generated and sent to the program along with the successful signal.

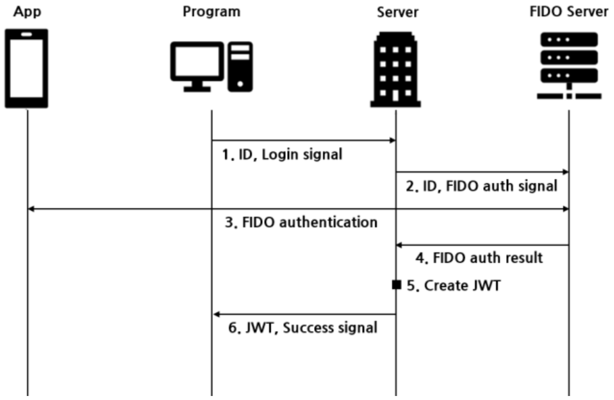

Figure 6. Internal program login

The program encryption function employs *Crypto* and *Cryptodome* which are the encryption modules of Python -- to use the AES encryption algorithm recommended by the US Department of Defense and stores *E_file* and *E_key* considering the decryption in the server. *E_file* refers to the re-encryption of files encrypted with the server's public key, whereas *E_key* pertains to the encryption of the key used to encrypt files with the server's public key. This is to prevent the leak of the key and files during the communication process. The key and files are decrypted using the server's private key on the server side. The exchange of public keys is conducted through the key exchange as explained in the section on the sign-up process.

The program encryption process is shown in Fig. 7. The same FIDO authentication as the login process in Fig. 6 is performed in the program. Upon successful FIDO authentication, the program generates a key to encrypt documents through the encryption module. After the program encrypts a file uploaded by the user, it transmits *E_file*, *JWT*, *E_key*, and sender and receiver information to the server. The server then checks the validity through JWT and stores the received file and information in the database, and a successful signal is sent.

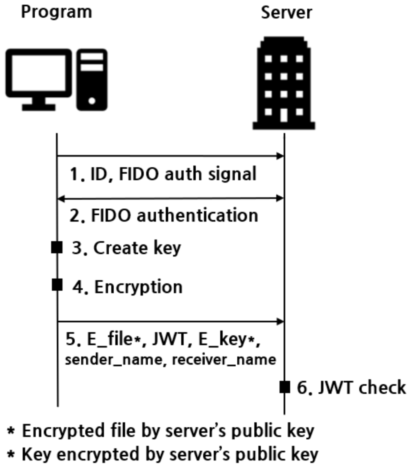

Figure 7. Internal program encryption

The program decryption function employs *jwt*, *Crypto*, and *Cryptodome* modules, similar to the encryption process. *E_file* and *E_key* in the decryption process are encrypted with the public key of the user, which is different from those of the aforementioned encryption process. This is to prevent the leak of the key and files during the communication process as well. The key and files are decrypted using the user's private key on the program side.

The decryption process is shown in Fig. 8. After performing the same FIDO authentication in Fig. 6, the program sends *JWT*, *E_file* name, and decryption request signal to the server. The server then verifies the validity of JWT and decrypts the *E_file* and *E_key* stored in Fig. 7 with its private key followed by encrypting the file with the public key of the user and sending it to the program. The user decrypts *E_key* and *E_file* using his/her private key in the program and downloads the original file.

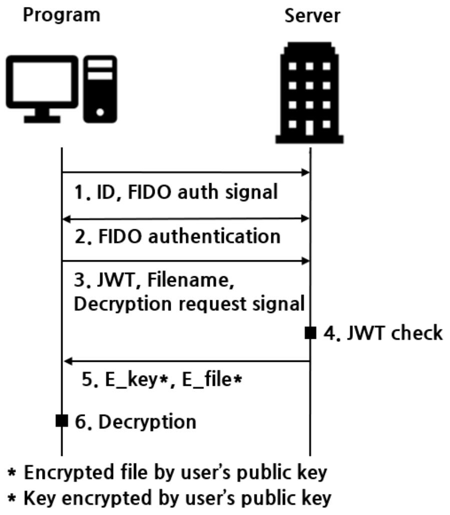

Figure 8. Internal program decryption

4.5 External Program

The external program refers to a program used outside the company. The difference from the internal program is that it can only use the viewer after decryption. By doing this, leak of the original file can be prevented. The decryption process of the external program is shown in Fig. 9. To use the decryption function, the same FIDO authentication as the login process in Fig. 6 is performed. The program sends *JWT*, *E_file* name, and decryption request signal issued from the server to the server, which then transmits the *E_file* and matched *E_key* to the program after verifying *JWT*. The user decrypts *E_file* and *E_key* using his/her private key. The user checks the contents of the original file through the viewer provided by the program.

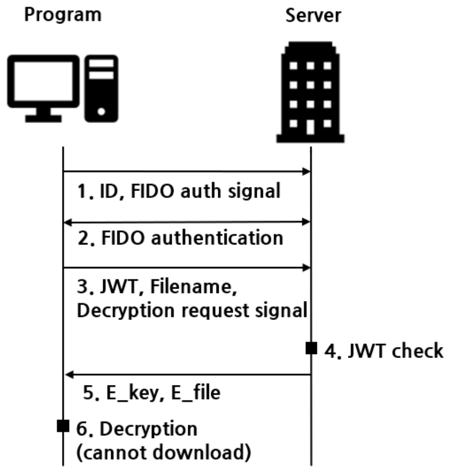

Figure 9. External program decryption

4.6 Usage Scenario

In Sections 4.1 to 4.5, the functions by the proposed system components were described. In this section, a scenario of the system is presented assuming use inside the company.

Suppose there are new employee *A* and senior *B*. *A* completes sign-up and logs in to submit a document to *B* after encryption. *B* connects to the web and logs in to download the file sent by *A* from the mailbox. *B* decrypts the downloaded file and views the original file after downloading it.

5 CONCLUSIONS

This study proposed a system of sharing a file securely using FIDO technology to prevent the leak of internal documents to the outside and view internal documents from outside. The proposed system was designed to conduct biometric authentication using FIDO technology and complete the identification procedure. It provides a platform by which files are encrypted and decrypted between specific groups to share the files. The proposed system is expected to solve inconveniences such as key generation, sharing, and memorization -- the drawbacks of the existing password-based authentication methods -- to enable the users to have simpler and secure file sharing and increase security against internal document leak.

To date, this study has implemented most components in the system except server interlocking. For future studies, service will be implemented based on the proposed system, and standards provided by the FIDO Alliance and work environments will be expanded to offline environments including online. Performance comparison experiments will also be conducted with existing encryption systems.

ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2020-2018-0-01799) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation) and by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT : Ministry of Science and ICT) (No. NRF-2020R1A2C1012187).

REFERENCES

[1] Hyun-Jo Lee, Han-Jin Cho, Yong-Ki Kim, Cheol-Joo Chae. 2020. A study on the FIDO authentication system using OpenSource. *Journal of the Korea Convergence Society* 11, 5 (2020), 19-25. DOI: https://doi.org/10.15207/JKCS.2020.11.5.019

[2] Jaeyeon Park, Jaeyoung Lee, Hyoungseok Lee, Jiwon Kang, Hyukjin Kwon, Dongil Shin, Dongkyoo Shin. 2019. Design of Military Information System User Authentication System Using FIDO 2.0-based Web Browser Secure Storage. *KCSA(Korea Convergence Security Association)* 19, 4(Oct, 2019), 43-53. DOI: https://doi.org/10.33778/kcsa.2019.19.4.043

[3] Jaejung Kim. 2015. Study on the password-free certification system using the FIDO. *KIISE* 33,5(May, 2015), 9-12.

[4] Yong-Ki Kim, Cheol-Joo Chae, Han-Jin Cho. 2018. User Authentication Method using EEG Signal in FIDO System. *Journal of the Korea Convergence Society* 9,1 (2018), 465-471, DOI: https://doi.org/10.15207/JKCS.2018.9.1.465