Computer Systems Science & Engineering
DOI: 10.32604/csse.2023.029356
Article

Tech Science Press

Secure e-Prescription Management System: Mitigating Blended Threat in IoBE

Deukhun Kim[1], Heejin Kim[2] and Jin Kwak[3,*]

[1]ISAA Lab., Institute for Information and Communication, Ajou University, Suwon, 16499, Korea
[2]Korea Orphan & Essential Drug Center (KOEDC), Seoul, 04523, Korea
[3]Department of Cyber Security, Ajou University, Suwon, 16499, Korea
*Corresponding Author: Jin Kwak. Email: security@ajou.ac.kr

Received: 01 March 2022; Accepted: 06 April 2022

Abstract: New information and communication technologies (ICT) are being applied in various industries to upgrade the value of the major service items. Moreover, data collection, storage, processing, and security applications have led to the creation of an interrelated ICT environment in which one industry can directly influence the other. This is called the "internet of blended environments" (IoBE), as it is an interrelated data environment based on internet-of-things collection activities. In this environment, security incidents may increase as size and interconnectivity of attackable operations grow. Consequently, pre-emptive responses to combined security threats are needed to securely utilize IoBE across industries. For example, the medical industry has more stringent information protection measures than other industries. Consequently, it has become a major target of attackers, as more clinician–patient interactions occur over the internet owing to COVID-19. Therefore, this study aims to acquire security for IoBE while focusing on the medical industry. Among the various types of medical ICT services, this study analyzes data flow and potential security threats from the e-prescription lifecycle perspective, which is highly utilized, strongly data-centric, and has numerous security issues. Based on our analysis, we propose a secure authentication and data-sharing scheme.

Keywords: Authentication scheme; blended threat; e-prescription; internet of blended environments

1 Introduction

With the continuous development of information and communication technologies (ICTs), new internet of things (IoT) devices, cloud-based storage and operations, big data and business intelligence services, mobile applications, and artificial intelligence (AI) models are continuously emerging. Notably, these technologies are being utilized in a more interrelated, blended manner than ever and have enabled data convergence among the formerly segregated industrial fields [1,2]. This new environment is called the "internet of blended environment" (IoBE), and it adds layers of complexity to security risks as the interconnectivity of attackable data-dependent operations grow [3]. Thus, new types of security threats are being identified. Therefore, studies to create a secure IoBE must be urgently conducted.

From the several complex security risks that spread across the IoBE, the current study focuses on the medical industry, which regularly creates, manipulates, and stores sensitive patient data. Hence, the medical industry has the largest damage potential and has become a major target of the attackers, as an increasing number of clinician–patient interactions are taking place over the internet owing to COVID-19 restrictions [4].

Among the plethora of data-heavy medical services, the current study focuses on the e-prescription system, which is heavily targeted by cybercriminals [5,6], and incident rates are continuously increasing. For example, in one incident, approximately 700 million patient and prescription records were compromised, and at least 400 million were confirmed as sold. In another incident, approximately 78 million prescription records were illegally harvested and sold to pharmacies as contact information.

Therefore, this study analyzes various security threats affecting the generation, collection, and processing of e-prescription data to provide a sharing and authentication scheme to improve IoBE security.

The rest of this paper is organized as follows. Section 2 provides an overview of the e-prescription system and analyzes the security risks that may occur in the analyzed environment. Section 3 proposes a secure sharing and authentication scheme for the e-prescription system. Section 4 verifies the proposed authentication and data-sharing technology, and Section 5 presents our conclusions.

2 Related Works

2.1 E-Prescription System

The e-prescription system allows clinicians and patients to avoid handling paper prescriptions and dealing with clinician-to-pharmacy coordination and printing. In the legacy prescription process, patients receive a paper prescription after diagnosis and obtain medications and equipment at a separate dispensary. However, paper prescriptions face counterfeiting, forgery, and loss/theft risks. To overcome these problems, e-prescription methods that electronically manage prescriptions have recently received great attention. With the ubiquity and lowering costs of ICT and the rapid utilization of smart devices, the construction of e-prescription systems is accelerating. However, with fast growth comes great risk, as the medical information handled on such networks is among the most sensitive in all e-commerce. See the examples in Tab. 1. When medical information is leaked, patient privacy is violated, and they encounter added personal risks. Hence, focusing on the e-prescription aspect, potent and quickly available security measures are needed.

Table 1: Types of sensitive e-prescription information

Category	Collected sensitive information	Example
Patient	Name	Deuk-Hun Kim
	Resident registration number	891206-1234567
Diagnosis and prescription	Disease classification code	(H) Number/(H) Number
	Name of prescribed medicines	(Drug Number) Name/quantity of drugs

2.2 Security Threats in the E-Prescription System

In this section, e-prescription system security risks are presented and analyzed [7]. Agents that participate in e-prescription processes consist of patients, clinicians, pharmacists, and e-prescription management center (ePMC) agents who manage cloud-based services. The security threats and mitigation requirements of e-prescription activities per agent are summarized in Tab. 2. Furthermore, the roles of each are explained as follows:

- **Patients**: Patients engage clinicians to receive diagnoses and treatment options via an e-prescription, and they receive medications and equipment from pharmacists accordingly. A patient may request changes to their prescription options using the e-prescription system after full authentication.
- **Clinicians**: Clinicians fully authenticate their identity on the e-prescription system to provide patients with medical services and prescriptions.
- **Pharmacists**: Pharmacists fully authenticate their identity on the e-prescription system to dispense drugs according to the corresponding prescription.
- **ePMC agents**: ePMC agents fully authenticate their identity on the e-prescription system so that patients, doctors, and pharmacists can perform their roles. ePMC agents also manage the storage, modification, deletion, and recovery of e-prescriptions, and they generate unique codes and verify validity periods, availability, and other authentication activities.

Table 2: Security threats affecting e-prescription system

Security threats	Description	Security requirement	Countermeasure
Sniffing	♦ The contents of the exchanged information may be exposed (e.g., authentication information, e-prescription)	Confidentiality	Cryptography
Falsification	♦ The contents of the exchanged information may be falsified (e.g., authentication information, e-prescription)	Integrity	Message digest technique
Spoofing	♦ Illegal acts may be committed by spoofing entities (e.g., user spoofing, illegal issuance of e-prescription)	Authentication	Authentication technique
Repudiation	♦ The exchanged information can be repudiated (e.g., authentication result, issuance of e-prescription)	Non-Repudiation	Digital signature
Reuse	♦ Illegal acts may be committed by reusing information that has been already used (e.g., reusing authentication information, re-issuance of e-prescription)	Validation/ One time	Expiry date validation
Invasion of privacy	♦ Privacy may be violated if the contents of exchanged information are leaked (e.g., leakage of biometric authentication information, leakage of sensitive information on e-prescription)	Anonymity	De-Identification

2.2.1 Analysis of the E-Prescription Issuance Process

The e-prescription issuance process is detailed next so that the phases of security threats in Tab. 2 can be further analyzed [8,9].

□ Entire process for issuing e-prescription

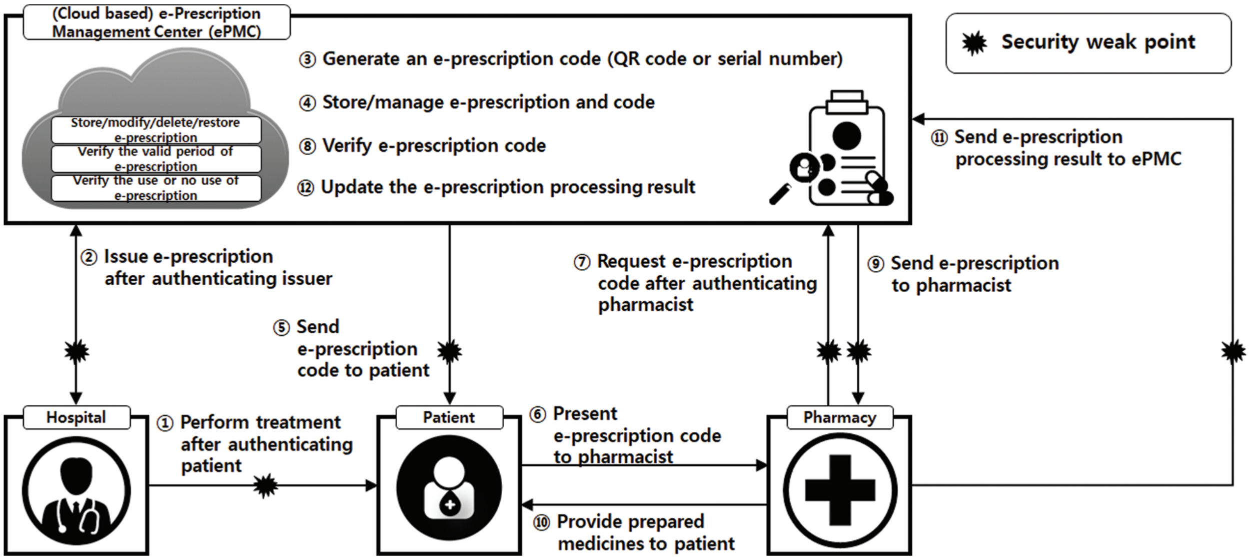
(Cloud based) e-Prescription Management Center (ePMC)
Store/modify/delete/restore e-prescription
Verify the valid period of e-prescription
Verify the use or no use of e-prescription
③ Generate an e-prescription code (QR code or serial number)
④ Store/manage e-prescription and code
⑧ Verify e-prescription code
⑫ Update the e-prescription processing result
Security weak point
⑪ Send e-prescription processing result to ePMC
② Issue e-prescription after authenticating issuer
⑤ Send e-prescription code to patient
⑦ Request e-prescription code after authenticating pharmacist
⑨ Send e-prescription to pharmacist
Hospital
① Perform treatment after authenticating patient
Patient
⑥ Present e-prescription code to pharmacist
⑩ Provide prepared medicines to patient
Pharmacy

Figure 1: Entire process for issuing e-prescription

Step 1: Patient authenticates and receives treatment from clinician.

Step 2: Clinician sends e-prescription to ePMC agent.

Step 3: ePMC agent generates an e-prescription code.

Step 4: ePMC agent stores/manages e-prescription and code.

Step 5: ePMC agent sends e-prescription code to patient.

Step 6: Patient presents the e-prescription code to pharmacist.

Step 7: Pharmacist sends e-Prescription code to ePMC agent for verification.

Step 8: ePMC agent verifies the e-prescription code.

Step 9: ePMC agent sends e-prescription authorization to pharmacist.

Step 10: Pharmacist dispenses prescribed medications or equipment.

Step 11: Pharmacist sends e-prescription processing results to ePMC agent.

Step 12: ePMC agent updates the e-prescription processing result.

□ Patient-to-hospital process for issuing e-prescription

Step 1: Patient requests treatment from a clinician.

Step 2: Clinician requests patient authentication.

Step 3: Patient authenticates to prove their identity.

Step 4: Clinician treats patient.

Step 5: Clinician requests ePMC agent issuance of e-prescription.

Step 6: ePMC agent requests clinician authentication.

Step 7: Clinician authenticates to prove their identity.

Step 8: ePMC agent issues e-prescription.

Steps 9~11: ePMC agent manages e-prescription code/stores and manages e-prescription and code/sends e-prescription code. See entire process for issuing e-prescription.

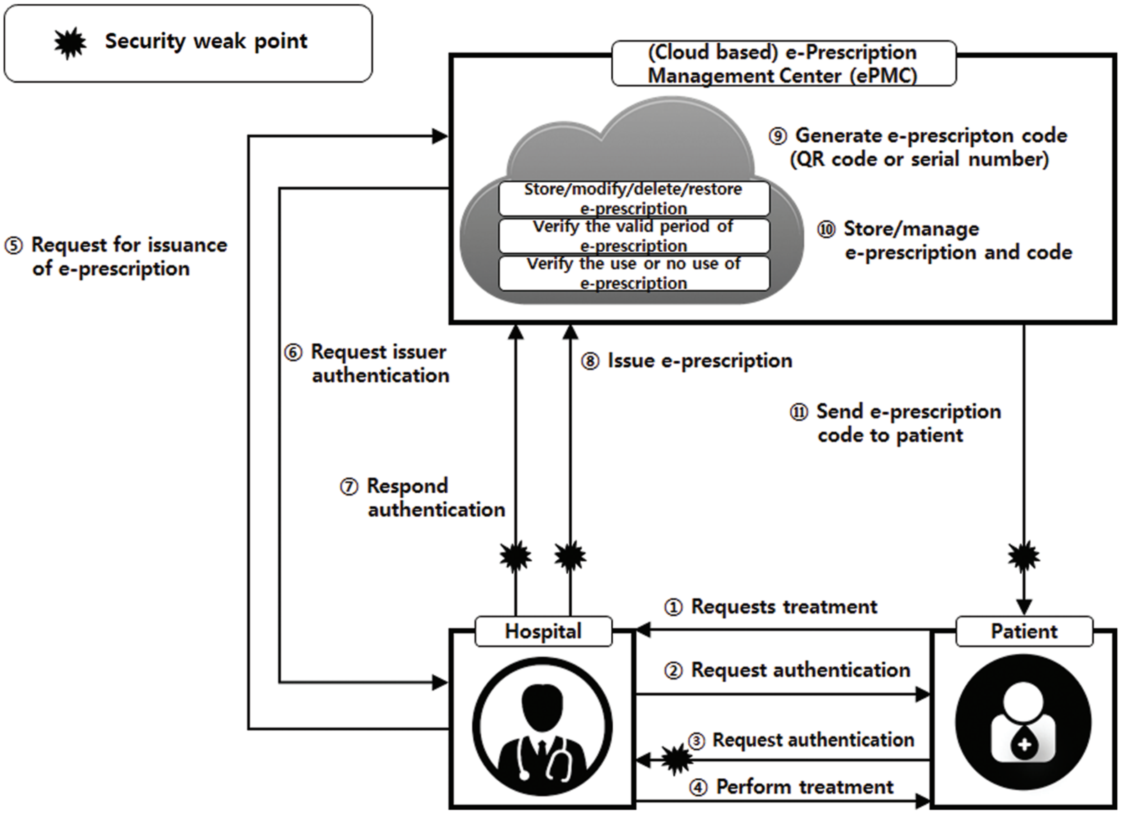
Security weak point
(Cloud based) e-Prescription Management Center (ePMC)
Store/modify/delete/restore e-prescription
Verify the valid period of e-prescription
Verify the use or no use of e-prescription
⑨ Generate e-prescripton code (QR code or serial number)
⑩ Store/manage e-prescription and code
⑤ Request for issuance of e-prescription
⑥ Request issuer authentication
⑦ Respond authentication
⑧ Issue e-prescription
⑪ Send e-prescription code to patient
Hospital
Patient
① Requests treatment
② Request authentication
③ Request authentication
④ Perform treatment

Figure 2: Patient-to-hospital process for issuing e-prescription

□ Patient-to-pharmacy process for issuing e-prescription

Step 1~2: Patient presents e-prescription code/requests inquiry of e-prescription code. This is identical to the corresponding process for issuing e-prescription.

Step 3: ePMC agent requests authentication of pharmacist.

Step 4: Pharmacist authenticates to prove their identity.

Steps 5~9: Pharmacist sends e-prescription code/sends e-prescription/provides prepared medicines or equipment/sends e-prescription processing result; ePMC agent updates e-prescription processing result. This is identical to the corresponding process for issuing e-prescription.

□ Patient-to-ePMC process for issuing e-prescription

Step 1: Patient requests e-prescription inquiry of ePMC agent.

Step 2: ePMC agent requests authentication of patient.

Step 3: Patient authenticates to prove identity.

Step 4: ePMC agent examines patient's e-prescription.

Step 5: ePMC agent provides the content the of e-prescription to the patient.

Step 6: ePMC agent takes necessary action (modification, deletion, or recovery).

Step 7: ePMC agent updates the e-prescription processing result.

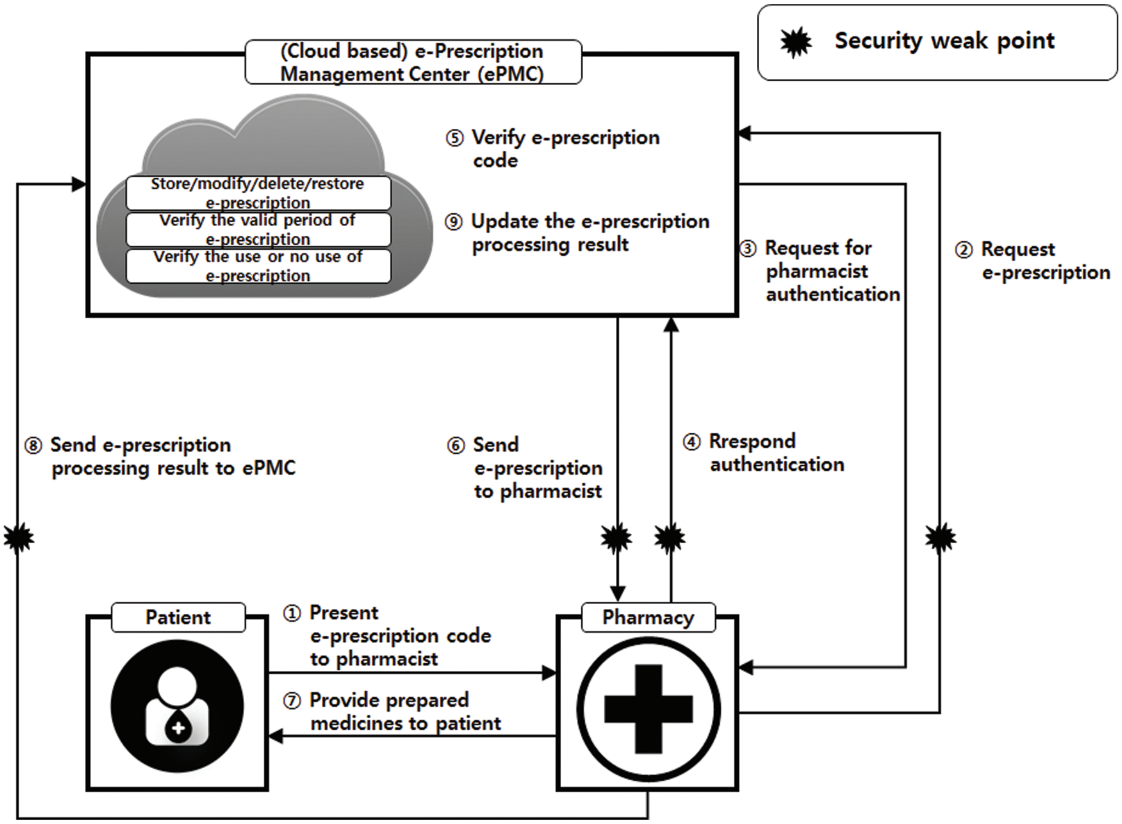
Security weak point
(Cloud based) e-Prescription Management Center (ePMC)
Store/modify/delete/restore e-prescription
Verify the valid period of e-prescription
Verify the use or no use of e-prescription
⑤ Verify e-prescription code
⑨ Update the e-prescription processing result
③ Request for pharmacist authentication
② Request e-prescription
⑧ Send e-prescription processing result to ePMC
⑥ Send e-prescription to pharmacist
④ Rrespond authentication
Patient
Pharmacy
① Present e-prescription code to pharmacist
⑦ Provide prepared medicines to patient

Figure 3: Patient-to-pharmacy process for issuing e-prescription

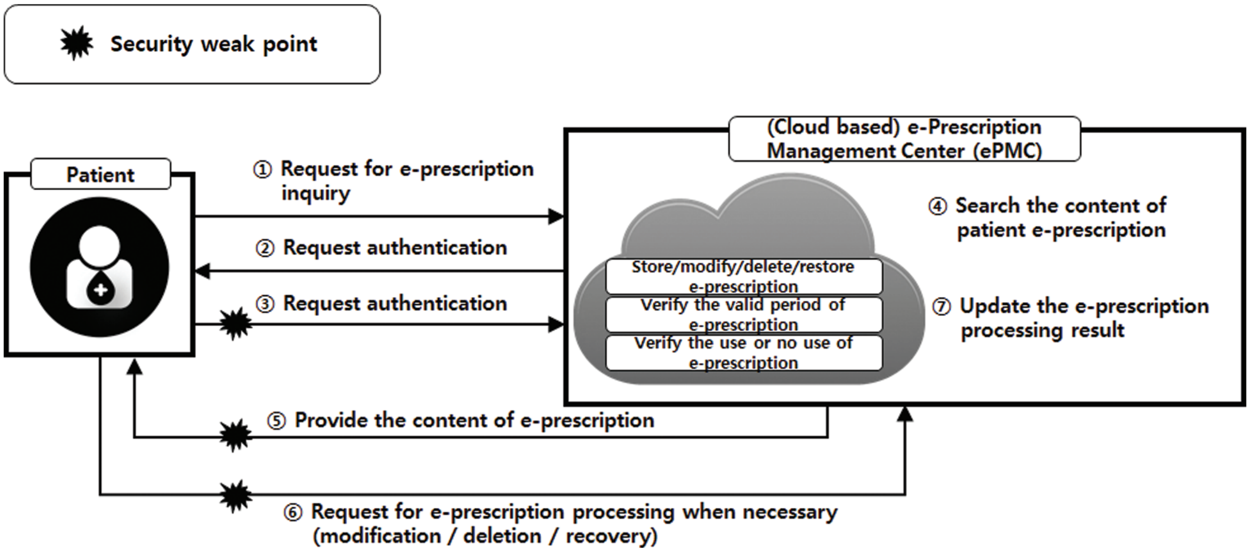
Security weak point
Patient
(Cloud based) e-Prescription Management Center (ePMC)
① Request for e-prescription inquiry
② Request authentication
③ Request authentication
Store/modify/delete/restore e-prescription
Verify the valid period of e-prescription
Verify the use or no use of e-prescription
④ Search the content of patient e-prescription
⑦ Update the e-prescription processing result
⑤ Provide the content of e-prescription
⑥ Request for e-prescription processing when necessary (modification / deletion / recovery)

Figure 4: Patient-to-ePMC process for issuing e-prescription

2.2.2 Detailed Analysis of Security Threats for E-Prescription Issuance Process

In the e-prescription issuance process analyzed in Section 2.2.1, security threats exist in the process of sending and receiving information (e.g., issuances, transfers, and inquiries). The risk areas are expanded in Tabs. 3–6.

Table 3: Analysis of security threats for the entire process for issuing e-prescription

Weak point	Security threats					
	Sniffing	Falsification	Spoofing	Repudiation	Reuse	Invasion of privacy
Step 1.	✓	✓	✓	✓	✓	✓
Step 2.	✓	✓	✓	✓	✓	✓
Step 5.	✓	✓	–	✓	✓	–
Step 7.	✓	✓	✓	✓	✓	✓
Step 9.	✓	✓	–	✓	✓	✓
Step 11.	✓	✓	✓	✓	–	–

The e-prescription system security threats (Fig. 1) are shown in Tab. 3.

In Step 1, the exchange of patient authentication information is threatened by sniffing, falsification, spoofing, reuse, and repudiation attacks, and invasion of privacy is possible due to the leakage of sensitive information.

In Step 2, the exchange of clinician authentication information and the issuance of e-prescriptions are threatened by sniffing, falsification, spoofing, reuse, and repudiation attacks, and invasion of privacy is possible.

In Step 5, the issuance of a unique e-prescription code issued by the ePMC agent is threatened by sniffing, falsification, reuse, and repudiation. In this case, spoofing and invasion of privacy are not considered, as the ePMC agent is a trusted agent, and the unique e-prescription code does not contain sensitive information.

In Step 7, the exchange of pharmacist authentication information and e-prescription code requests is threatened by sniffing, falsification, spoofing, reuse, and repudiation attacks.

In Step 9, the exchange of e-prescription information is threatened by sniffing, falsification, reuse, and repudiation attacks, and invasion of privacy is possible. Spoofing is not considered in this case, as the ePMC agent is a trusted agent.

In Step 11, the exchange of e-prescription processing results is threatened by sniffing, falsification, and spoofing attacks.

Table 4: Analysis of security threats for the patient-to-hospital process for issuing e-prescription

Weak point	Security threats					
	Sniffing	Falsification	Spoofing	Repudiation	Reuse	Invasion of privacy
Step 3.	✓	✓	✓	✓	✓	✓
Step 7.	✓	✓	✓	✓	✓	✓
Step 8.	✓	✓	✓	✓	✓	✓
Step 11.	✓	✓	–	✓	✓	–

The security threats of the patient-to-clinician e-prescription issuance exchange in Fig. 2 are detailed in Tab. 4. The authsentication threats of Steps 3 and 7 are identical to those of Steps 1 and 2 in Tab. 3. The e-prescription issuance security threats in Step 8 are identical to those of Step 2 in Tab. 3. The e-prescription code transmission threats of Step 11 are identical to those of Step 5 in Tab. 3.

Table 5: Analysis of security threats for the patient-to-pharmacy process for issuing e-prescription

Weak point	Security threats					
	Sniffing	Falsification	Spoofing	Repudiation	Reuse	Invasion of privacy
Step 2.	✓	✓	✓	✓	✓	–
Step 4.	✓	✓	✓	✓	✓	✓
Step 6.	✓	✓	–	✓	✓	✓
Step 8.	✓	✓	✓	✓	–	–

The security threats of the patient-to-pharmacist e-prescription issuance process in Fig. 3 are detailed in Tab. 5. The e-prescription code threats in Steps 2 and 4 are identical to those of Step 7 in Tab. 3. However, Step 2 is not considered as it does not have a privacy invasion risk. The e-prescription transmission threats in Step 6 are identical to those of Step 9 in Tab. 3. The e-prescription processing result exchange threats in Step 8 are identical to those of Step 11 in Tab. 3.

Table 6: Analysis of security threats for the patient-to-ePMC process for issuing e-prescription

Weak point	Security threats					
	Sniffing	Falsification	Spoofing	Repudiation	Reuse	Invasion of privacy
Step 3.	✓	✓	✓	✓	✓	✓
Step 5.	✓	✓	–	✓	✓	✓
Step 6.	✓	✓	✓	✓	✓	–

The security threats of the patient-to-ePMC agent e-prescription issuance process in Fig. 4 are detailed in Tab. 6. The patient authentication threats in Step 3 are identical to those of Step 1 in Tab. 3. The e-prescription issuance process of Step 5 is identical to those of Step 9 in Tab. 3, which can occur when the e-prescription is transmitted. In Step 6, threats of sniffing, falsification, spoofing, reuse, and repudiation attacks for patient requests to the ePMC agent also exist.

From this mapping, the proposed protection scheme can now be illustrated.

3 Proposed Scheme

In this section, a more secure e-prescription management scheme is proposed according to the issuance requirements of each entity (agent). The entities participating in the e-prescription system consist of patients, clinicians, pharmacists, and ePMC agents. They are set as entities, depending on the similarity of the authentication and e-prescription-related processing behaviors of the analyzed processes. The scheme is executed by the ePMC agent, who manages the overall e-prescription process. There are three subprocesses: registration phase, authentication phase, and data transfer phase. Tab. 7 shows the notations used.

Table 7: Notation of our scheme

Notation	Description
ID_X	Identity of an entity X
PW_U	Password of an user client
Bio_U	Biometric information of an user client (Such as finger, face, etc.)
AID_U	Anonymous identity of an user client
Sig_X	Digital signature of an entity X
N	Random nonce
x	Secret value of an user client
y	Secret value of e-Prescription Management Center(ePMC)
ρ, σ	Value of authentication challenge for each entity
B_U	Value of Biometric authentication for an user client
$H(\cdot)$	One-way hash function
$PRNG(\cdot)$	Pseudo Random Number Generator
PU_X/PR_X	Encryption/Decryption function of public key cryptosystem of an entity X
T_X	Time stamp of an entity X
$\oplus$	Exclusive-OR operation
$\|\|$	Concatenation operation
$Data_X$	data provided by entity X
$Auth.Token(\cdot)$	Authentication token
C_{num}/P_{num}	Encrypted/Decrypted message

3.1 Registration Phase

Fig. 5 shows the registration phase proposed to perform authentication between entities comprising the e-prescription issuance and processing, and the process is described below.

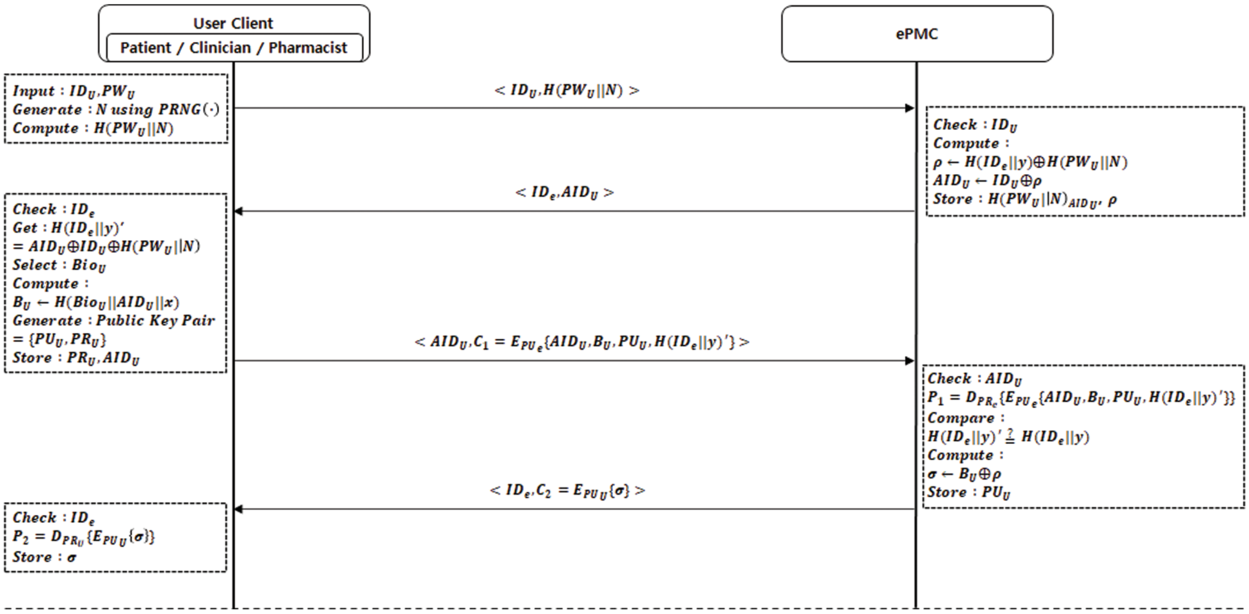
User Client
Patient / Clinician / Pharmacist
ePMC
Input : ID_U, PW_U
Generate : N using $PRNG(\cdot)$
Compute : $H(PW_U||N)$
$< ID_U, H(PW_U||N) >$
Check : ID_U
Compute :
$\rho \leftarrow H(ID_e||y) \oplus H(PW_U||N)$
$AID_U \leftarrow ID_U \oplus \rho$
Store : $H(PW_U||N)_{AID_U}, \rho$
$< ID_e, AID_U >$
Check : ID_e
Get : $H(ID_e||y)'$
$= AID_U \oplus ID_U \oplus H(PW_U||N)$
Select : Bio_U
Compute :
$B_U \leftarrow H(Bio_U||AID_U||x)$
Generate : Public Key Pair
$= \{PU_U, PR_U\}$
Store : PR_U, AID_U
$< AID_U, C_1 = E_{PU_e}\{AID_U, B_U, PU_U, H(ID_e||y)'\} >$
Check : AID_U
$P_1 = D_{PR_e}\{E_{PU_e}\{AID_U, B_U, PU_U, H(ID_e||y)'\}\}$
Compare :
$H(ID_e||y)' \overset{?}{=} H(ID_e||y)$
Compute :
$\sigma \leftarrow B_U \oplus \rho$
Store : PU_U
$< ID_e, C_2 = E_{PU_U}\{\sigma\} >$
Check : ID_e
$P_2 = D_{PR_U}\{E_{PU_U}\{\sigma\}\}$
Store : σ

Figure 5: Registration phase in our scheme

Step 1. [*User client sends <$ID_U, H(PW_U||N)$> to ePMC*]

The user client inputs ID_U and PW_U and generates a random nonce N using $PRNG(\cdot)$. Hence, $H(PW_U||N)$ is computed. The user client then sends their identifier, ID_U, and $H(PW_U||N)$ to the ePMC and requests registration.

$$\begin{cases} Input : User\ agent\ input\ to\ \boldsymbol{ID_U}, \boldsymbol{PW_U} \\ Generate : Random\ nonce\ \boldsymbol{N}\ using\ \boldsymbol{PRNG}(\cdot) \\ Compute : \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N})\ using\ User\ client's\ \boldsymbol{PW_U}\ and\ random\ nonce\ \boldsymbol{N} \\ Send : \boldsymbol{ID_U}, \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N}) \end{cases}$$

Step 2. [*ePMC checks ID_U and sends <ID_e, AID_U> to User client*]

The ePMC checks the identifier ID_U of the user client and computes ρ, which is used as the authentication challenge value using the received $H(PW_U||N)$, their own identifier ID_e, and a secret value, y, as follows:

$$\rho \leftarrow H(ID_e||y) \oplus H(PW_U||N)$$

Then, the AID_U, an anonymous identifier of the user client, is computed using the computed ρ, as follows:

$$AID_U \leftarrow ID_U \oplus \rho$$

Afterward, the ePMC stores the $H(PW_U||N)_{AID_U}$ obtained by indexing the $H(PW_U||N)$ received from the user client and the authentication challenge value, ρ, and sends their identifier, ID_e, and an anonymous identifier, AID_U, to the user client.

$$\begin{cases} Check : User\ client's\ identity\ \boldsymbol{ID_U} \\ \begin{cases} Compute : \rho \leftarrow \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y}) \oplus \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N})\ using\ \boldsymbol{ID_e}, \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N})\ and\ ePMC's\ \boldsymbol{y} \\ \qquad \boldsymbol{AID_U} \leftarrow \boldsymbol{ID_U} \oplus \rho\ using\imath\boldsymbol{ID_U}\ and\ authentication\ challenge\ value\ \rho \end{cases} \\ Store : \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N})_{\boldsymbol{AID_U}}, \rho \\ Send : \boldsymbol{ID_e}, \boldsymbol{AID_U} \end{cases}$$

Step 3. [*User client checks ID_e and sends <AID_U, C_1> to ePMC*]

The user client checks the ePMC's identifier, ID_e, and obtains $H(ID_e||y)'$ by calculating the following equation using the received anonymous identifier, AID_U, their identifier, ID_U, and $H(PW_U||N)$:

$$H(ID_e||y)' \leftarrow AID_U \oplus ID_U \oplus H(PW_U||N)$$

Then, the user client selects their biometric information, Bio_U (e.g., fingerprint or face scan). Then, to prevent the leakage of biometric information during authentication, B_U, which is used as the biometric authentication value, is calculated as follows using the user's anonymous identifier, AID_U, and secret value, x:

$$B_U \leftarrow H(Bio_U||AID_U||x)$$

Then, the user client generates their public key pair $\{PU_U, PR_U\}$, stores their anonymous identifier, AID_U, received from the ePMC and their private key, PR_U. They then send $C_1 = E_{PU_e}\{AID_U, B_U, PU_U, H(ID_e||y)'\}$, which was obtained by encrypting the anonymous identifier, AID_U, the biometric authentication value, B_U, and the public key, PU_U, to obtain $H(ID_e||y)'$ using the ePMC's public key, PU_e, and the anonymous identifier AID_U to the ePMC.

$$\begin{cases} \mathit{Check} : \mathit{ePMC's\ identity}\ \boldsymbol{ID_e} \\ \mathit{Get} : \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y})' \leftarrow \boldsymbol{AID_U} \oplus \boldsymbol{ID_U} \oplus \boldsymbol{H}(\boldsymbol{PW_U}||\boldsymbol{N}) \\ \mathit{Select} : \boldsymbol{Bio_U}\ \mathit{such\ as\ face, finger\ print\ etc.} \\ \mathit{Compute} : \boldsymbol{B_U} \leftarrow \boldsymbol{H}(\boldsymbol{Bio_U}||\boldsymbol{AID_U}||\boldsymbol{x})\ \mathit{using}\ \boldsymbol{Bio_U}, \boldsymbol{AID_U}\ \mathit{and\ User\ client's}\ \boldsymbol{x} \\ \mathit{Generate} : \mathit{Public\ key\ pair}\{\boldsymbol{PU_U}, \boldsymbol{PR_U}\} \\ \mathit{Store} : \boldsymbol{PR_U}, \boldsymbol{AID_U} \\ \mathit{Encrypt} : \boldsymbol{E_{PU_e}}\{\boldsymbol{AID_U}, \boldsymbol{B_U}, \boldsymbol{PU_U}, \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y'})\}\ \mathit{with}\ \boldsymbol{AID_U}, \boldsymbol{B_U}, \boldsymbol{PU_U}, \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y'})\ \mathit{using}\ \boldsymbol{PU_e} \\ \qquad\qquad\qquad \boldsymbol{C_1} = E_{PUe}\{AID_U, B_U, PU_U, H(ID_e||y)'\} \\ \mathit{Send} : \boldsymbol{AID_U}, \boldsymbol{C_1} \end{cases}$$

Step 4. [*ePMC checks AID_U, P_1 and sends <ID_e, C_2> to User client*]

The ePMC checks the anonymous identifier, AID_U, of the user client and obtains the plain text, P_1, by performing $D_{PR_e}\{E_{PU_e}\{AID_U, B_U, PU_U, H(ID_e||y)'\}\}$, which decrypts the received encrypted message, C_1, using the user's private key, PR_e. The ePMC verifies this by comparing $H(ID_e||y)'$ to the existing value, $H(ID_e||y)$, and computes σ, which is used as another authentication challenge based on the received biometric authentication value, B_U, and the authentication challenge value, ρ, as follows:

$\sigma \leftarrow B_U \oplus \rho.$

Afterward, the ePMC stores the public key, PU_U, received from the user client and sends $C_2 = E_{PU_U}\{\sigma\}$, obtained by encrypting the authentication challenge value, σ, with the user client's public key, PU_U, and its identifier, ID_e, to the user client.

$$\begin{cases} \mathit{Check} : \mathit{User\ client's\ anonymous\ identity}\ \boldsymbol{AID_U} \\ \mathit{Decrypt} : \boldsymbol{D_{PR_e}}\{\boldsymbol{E_{PU_e}}\{\boldsymbol{AID_U}, \boldsymbol{B_U}, \boldsymbol{PU_U}, \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y})'\}\}\mathit{using}\ \boldsymbol{PR_e} \\ \qquad\qquad\qquad \boldsymbol{P_1} = D_{PRe}\{C_1\} \\ \mathit{Compare} : \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y})'? = \boldsymbol{H}(\boldsymbol{ID_e}||\boldsymbol{y}) \\ \mathit{Compute} : \sigma \leftarrow \boldsymbol{B_U} \oplus \rho\ \mathit{using\ biometric\ authentication\ value}\ \boldsymbol{B_U}\mathit{and} \\ \qquad\qquad \mathit{authentication\ challenge\ value}\ \rho \\ \mathit{Store} : \boldsymbol{PU_U} \\ \mathit{Encrypt} : \boldsymbol{E_{PU\,U}}\{\sigma\}\ \mathit{with\ authentication\ challenge\ value}\ \sigma\ \mathit{using}\ \boldsymbol{PU_U} \\ \qquad\qquad\qquad \boldsymbol{C_2} = E_{PU\,U}\{\sigma\} \\ \mathit{Send} : \boldsymbol{ID_e}, \boldsymbol{C_2} \end{cases}$$

Step 5. [*User client checks ID_e, P_2 and the end of registration phase*]

The user client checks the ePMC's identifier, ID_e, and obtains the σ of the plain text, P_2, by performing $D_{PR_U}\{E_{PU_U}\{\sigma\}\}$, which decrypts the received encrypted message, $C_2\frac{2}{2}$, using their private key, PR_U. Then, the user client stores the authentication challenge value, σ, and completes the registration phase.

$$\begin{cases} \mathit{Check} : \mathit{ePMC's\ identity}\ \boldsymbol{ID_e} \\ \mathit{Decrypt} : \boldsymbol{D_{PR\,U}}\{\boldsymbol{E_{PU\,U}}\{\sigma\}\}\ \mathit{using}\ \boldsymbol{PR_U}\ \ \boldsymbol{P_2} = D_{PR\,U}\{C_2\}. \\ \mathit{Store} : \sigma \end{cases}$$

3.2 Authentication Phase

Fig. 6 shows the authentication phase proposed to perform authentication between entities comprising the issuance and processing of e-prescriptions, and the process is described below.

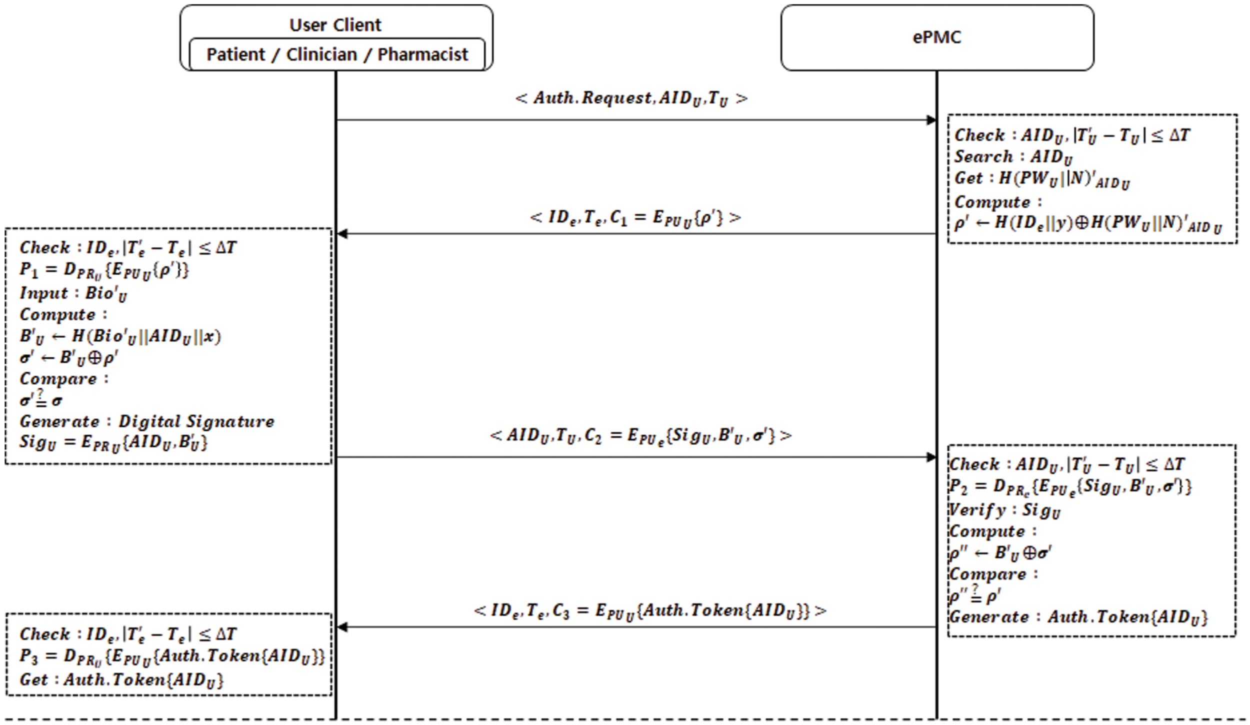
User Client
Patient / Clinician / Pharmacist
ePMC
$< Auth.Request, AID_U, T_U >$
$Check : AID_U, |T'_U - T_U| \leq \Delta T$
$Search : AID_U$
$Get : H(PW_U||N)'_{AID_U}$
$Compute :$
$\rho' \leftarrow H(ID_e||y) \oplus H(PW_U||N)'_{AID_U}$
$< ID_e, T_e, C_1 = E_{PU_U}\{\rho'\} >$
$Check : ID_e, |T'_e - T_e| \leq \Delta T$
$P_1 = D_{PR_U}\{E_{PU_U}\{\rho'\}\}$
$Input : Bio'_U$
$Compute :$
$B'_U \leftarrow H(Bio'_U||AID_U||x)$
$\sigma' \leftarrow B'_U \oplus \rho'$
$Compare :$
$\sigma' \overset{?}{=} \sigma$
$Generate : Digital\ Signature$
$Sig_U = E_{PR_U}\{AID_U, B'_U\}$
$< AID_U, T_U, C_2 = E_{PU_e}\{Sig_U, B'_U, \sigma'\} >$
$Check : AID_U, |T'_U - T_U| \leq \Delta T$
$P_2 = D_{PR_e}\{E_{PU_e}\{Sig_U, B'_U, \sigma'\}\}$
$Verify : Sig_U$
$Compute :$
$\rho'' \leftarrow B'_U \oplus \sigma'$
$Compare :$
$\rho'' \overset{?}{=} \rho'$
$Generate : Auth.Token\{AID_U\}$
$< ID_e, T_e, C_3 = E_{PU_U}\{Auth.Token\{AID_U\}\} >$
$Check : ID_e, |T'_e - T_e| \leq \Delta T$
$P_3 = D_{PR_U}\{E_{PU_U}\{Auth.Token\{AID_U\}\}$
$Get : Auth.Token\{AID_U\}$

Figure 6: Authentication phase in our scheme

Step 1. [*User client sends* $<Auth.Request, AID_U, T_U>$ *to ePMC*]

To request authentication, the user client sends their anonymous identifier, AID_U, their time-stamp value, T_U, and their request message, *Auth.Request*, to the ePMC.

$$\begin{cases} Request : User\ client\ \boldsymbol{Auth.Request}\ to\ ePMC \\ Send : \boldsymbol{AID_U}, \boldsymbol{T_U} \end{cases}$$

Step 2. [*ePMC checks* AID_U, T_U *and sends* $<ID_e, T_e, C_1>$ *to User client*]

The ePMC checks the validity of the user client's anonymous identifier, AID_U, and their time-stamp value, T_U, and searches AID_U to obtain the $H(PW_U||N)'_{AID_U}$ stored in Step 2 of the registration phase. Using this, the ePMC computes the authentication challenge value, ρ', as follows:

$$\rho' \leftarrow H(ID_e||y) \oplus H(PW_U||N)'_{AID_U}.$$

Afterward, the ePMC sends $C_1 = E_{PU\,U}\{\rho'\}$, obtained by encrypting the computed authentication challenge value, ρ', with the public key of the user client, ID_e, and the time-stamp value, T_e, to the user client.

$$\begin{cases} Check : User\ client's\ anonymous\ identity\ \boldsymbol{AID_U}\ and \\ \qquad\qquad\qquad\qquad timestamp\ validation\ |\boldsymbol{T'_U} - \boldsymbol{T_U}| \leq \boldsymbol{\Delta T} \\ Search : \boldsymbol{AID_U}\ to\ get\ H(PW_U||N)'_{AID_U} \\ Get : \boldsymbol{H(PW_U||N)'_{AID_U}} \\ Compute : \rho' \leftarrow \boldsymbol{H(ID_e||y)} \oplus \boldsymbol{H(PW_U||N)'_{AID_U}}\ using\ \boldsymbol{ID_e}, \boldsymbol{H(PW_U||N)'_{AID_U}}\ and\ ePMC's\ \boldsymbol{y} \\ Encrypt : \boldsymbol{E_{PU_U}}\{\rho'\}\ with\ \rho'\ using\ \boldsymbol{PU_U} \\ \qquad\qquad\qquad\qquad\qquad\qquad \boldsymbol{C_1} = E_{PU_U}\{\rho'\} \\ Send : \boldsymbol{ID_e}, \boldsymbol{T_e}, \boldsymbol{C_1} \end{cases}$$

Step 3. [*User client checks* ID_e, T_e, P_1 *and sends* $<AID_U, T_U, C_2>$ *to ePMC*]

The user client checks the validity of the ePMC's identifier, ID_e, and the time-stamp value, T_U, and obtains the authentication challenge value, ρ', of the plain text, P_1, by performing $D_{PR_U}\{E_{PU_U}\{\rho'\}\}$, which decodes the encrypted message, C_1, with their private key, PR_U. Afterward, the user client inputs the biometric information selected during the registration phase to compute the biometric authentication value, B'_U, using the anonymous identifier, AID_U, and secret value, x, as follows:

$$B'_U \leftarrow H(Bio'_U||AID_U||x)$$

Furthermore, the user client computes another authentication challenge value, σ', using the computed B'_U and the authentication challenge value, ρ', obtained by decrypting, as follows:

$$\sigma' \leftarrow B'_U \oplus \rho'$$

After comparing and verifying the computed σ' and the authentication challenge value, σ, stored in Step 5 of the registration phase, the user client generates the electronic signature value, $Sig_U = E_{PRU}\{AID_U, B'_U\}$, for the anonymous identifier, AID_U, and the biometric authentication value, B'_U, using their private key, PR_U. Afterward, the user client sends $C_2 = E_{PUe}\{Sig_U, B'{}_U, \sigma'\}$, obtained by encypting their electronic signature value, Sig_U, biometric authentication value, B'_U, and authentication challenge value, σ', with the ePMC's public key, their anonymous identifier AID_U, and their time-stamp value, T_U, to the ePMC.

$$\left\{\begin{array}{l}
Check : ePMCs'\ identity\ \boldsymbol{ID_e}\ and\ timestamp\ validation\ |\boldsymbol{T_e^{'}} - \boldsymbol{T_e}| \leq \Delta\boldsymbol{T} \\
Decrypt : \boldsymbol{D_{PR_U}}\{\boldsymbol{E_{PU_U}}\{\rho'\}\ using\boldsymbol{PR_U}\}\boldsymbol{P_1} = D_{PR_U}\{C_1\} \\
Input : \boldsymbol{Bio'} \\
\left\{\begin{array}{l}
Compute : \boldsymbol{B'}_U \leftarrow \boldsymbol{H}(\boldsymbol{Bio'_U}||\boldsymbol{AID_U}||\boldsymbol{x})using\ \boldsymbol{Bio'_U}, \boldsymbol{AID_u}and\ Userclient's\ \boldsymbol{x} \\
\sigma^{'} \leftarrow B_U^{'} \oplus \rho^{'}\ using\ biometric\ authentication\ value\ \boldsymbol{B_U^{'}}\ and\ authentication\ challenge\ value\ \rho'
\end{array}\right. \\
compare : \sigma'? = \sigma \\
Generate : Digital\ signature\ \boldsymbol{E_{PRU}}\{\boldsymbol{AID_U}, \boldsymbol{B_U^{'}}\}with\ \boldsymbol{AID_U}, \boldsymbol{B_U^{'}}\ using\ \boldsymbol{PR_U} \\
\qquad\qquad\qquad\qquad\qquad\qquad \boldsymbol{Sig_U} = E_{PRU}\{AID_u, B_U^{'}\} \\
Encrypt : \boldsymbol{E_{PUe}}\{\boldsymbol{Sig_U B_U^{'}}\sigma^{'}\}\boldsymbol{with\ Sig_U}, \boldsymbol{B_U^{'}}, \sigma^{'} using\boldsymbol{PU_e}\ \boldsymbol{C_2} = E_{PUe}\{Sig_U, B_U^{'}, \sigma^{'}\} \\
Send : \boldsymbol{AID_U}, \boldsymbol{T_U}, \boldsymbol{C_2}
\end{array}\right.$$

Step 4. [*ePMC checks* AID_U, T_U, P_2 *and sends* <ID_e, T_e, C_3> *to User client*]

The ePMC checks the validity of the user client's anonymous identifier, AID_U, and time-stamp, T_U, and obtains the plain text, P_2, by performing $D_{PR_e}\{E_{PU_e}\{Sig_U, B'_U, \sigma'\}\}$, which decrypts the received encrypted message, C_2, with their private key, PR_e. Afterward, the ePMC verifies the anonymous identifier, AID_U, of the electronic signature value, Sig_U, and the biometric authentication value, B'_U, using their public key, PU_U, and computes another authentication challenge value, ρ'', using the received biometric authentication value, B'_U, and authentication challenge value, σ', as follows:

$$\rho'' \leftarrow B'_U \oplus \sigma'.$$

The ePMC then compares and verifies the computed $\rho''$with the authentication challenge value, ρ', computed in Step 2 of the authentication phase. Then, it generates the authentication token, $Auth.Token\{AID_U\}$, for the anonymous identifier, AID_U. Afterward, the ePMC sends $C_3 = E_{PUU}\{Auth.Token\{AID_U\}\}$, obtained by encrypting the generated authentication token, $Auth.Token\{AID_U\}$, with the public key of the user client, its identifier, ID_e, and its time-stamp value, T_e, to the user client.

$$\begin{cases} \mathit{Check} : \mathit{User\ client's\ anonymous\ identity}\ \boldsymbol{AID_U} \\ \qquad\qquad\quad \mathit{and\ timestamp\ validation}\ |\boldsymbol{T'_U} - \boldsymbol{T_U}| \le \boldsymbol{\Delta T} \\ \mathit{Decrypt} : \boldsymbol{D_{PR_e}}\{\boldsymbol{E_{PU_e}}\{\boldsymbol{Sig_U}, \boldsymbol{B'_U}, \sigma'\}\}\ \mathit{using}\ \boldsymbol{PR_e P_2} = D_{PR_e}\{C_2\} \\ \mathit{Verify} : \boldsymbol{AID_U}, \boldsymbol{B'_U} \leftarrow \boldsymbol{Sig_U}\ \mathit{using}\ \boldsymbol{PU_U} \\ \mathit{Compute} : \rho'' \leftarrow \boldsymbol{B'_U} \oplus \sigma'\ \mathit{using\ biometric\ authentication\ value}\ \boldsymbol{B'_U}\ \mathit{and} \\ \qquad\qquad\qquad \mathit{authentication\ challenge\ value}\ \sigma' \\ \mathit{Compare} : \rho''? = \rho' \\ \mathit{Encrypt} : \boldsymbol{E_{PU_U}}\{\boldsymbol{Auth.Token}\{\boldsymbol{AID_U}\}\}\ \mathit{with}\ \boldsymbol{AID_U}\ \mathit{using}\ \boldsymbol{PU_U} \\ C_3 = E_{PU_U}\{Auth.Token\{AID_U\}\} \\ \mathit{Send} : \boldsymbol{ID_e}, \boldsymbol{T_e}, \boldsymbol{C_3} \end{cases}$$

Step 5. [*User client checks* ID_e, T_e, P_3 *and the end of authentication phase*]

The user client checks the validity of the ePMC's identifier, ID_e, and its time-stamp value, T_e, to obtain the authentication token, $Auth.Token\{AID_U$, of the plain text, P_3, by performing $D_{PR_U}\{E_{PU_U}\{Auth.Token\{AID_U\}\}\}$, which decrypts the received encrypted message, C_3, with its private key, PR_U. Then, the user client terminates the authentication phase.

$$\begin{cases} \mathit{Check} : \mathit{ePMC's\ identity}\ \boldsymbol{ID_e}\ \mathit{and\ timestamp\ validation}\ |\boldsymbol{T'_e} - \boldsymbol{T_e}| \le \Delta T \\ \mathit{Decrypt} : \boldsymbol{D_{PR_U}}\{\boldsymbol{E_{PU_U}}\{\boldsymbol{Auth.Token}\{\boldsymbol{AID_U}\}\}\}\ \mathit{using}\ \boldsymbol{PR_U} \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad P_3 = D_{PR_U}\{C_3\} \\ \mathit{Get} : \boldsymbol{Auth.Token}\{\boldsymbol{AID_U}\} \end{cases}$$

3.3 Data-Transfer Phase

Fig. 7 shows the data transfer phase proposed to perform data storage and request among entities comprising the issuance and processing of e-prescriptions, and the process is described below.

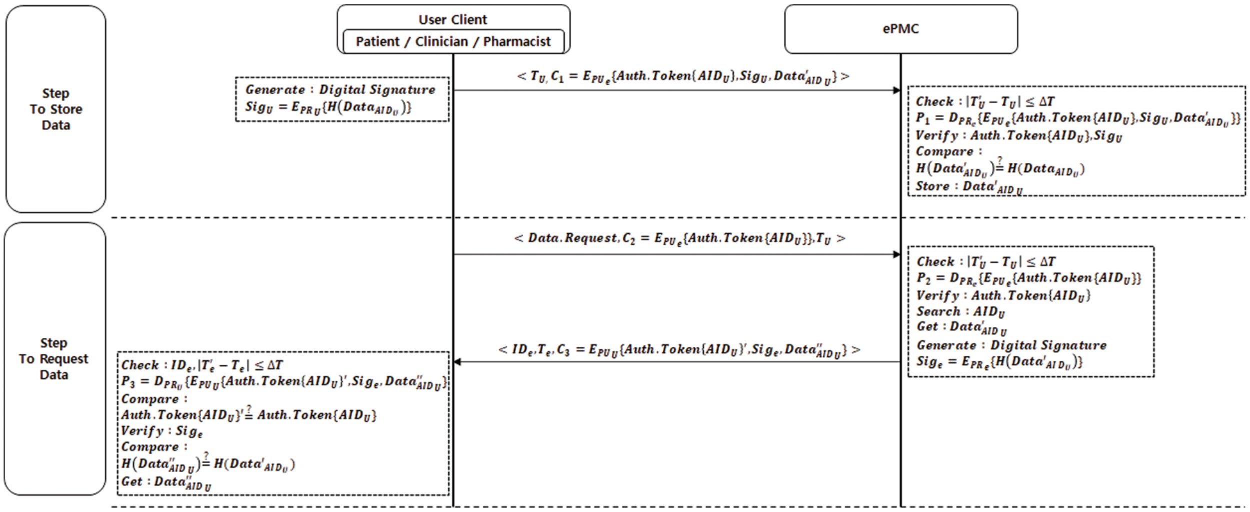
Step
To Store
Data
User Client
Patient / Clinician / Pharmacist
ePMC
$Generate : Digital\ Signature$
$Sig_U = E_{PR_U}\{H(Data_{AID_U})\}$
$< T_U, C_1 = E_{PU_e}\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID\,U}\} >$
$Check : |T'_U - T_U| \le \Delta T$
$P_1 = D_{PR_e}\{E_{PU_e}\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}\}\}$
$Verify : Auth.Token\{AID_U\}, Sig_U$
$Compare :$
$H(Data'_{AID_U}) \overset{?}{=} H(Data_{AID_U})$
$Store : Data'_{AID\,U}$
Step
To Request
Data
$< Data.Request, C_2 = E_{PU_e}\{Auth.Token\{AID_U\}\}, T_U >$
$Check : |T'_U - T_U| \le \Delta T$
$P_2 = D_{PR_e}\{E_{PU_e}\{Auth.Token\{AID_U\}\}$
$Verify : Auth.Token\{AID_U\}$
$Search : AID_U$
$Get : Data'_{AID\,U}$
$Generate : Digital\ Signature$
$Sig_e = E_{PR_e}\{H(Data'_{AID_U})\}$
$< ID_e, T_e, C_3 = E_{PU_U}\{Auth.Token\{AID_U\}', Sig_e, Data''_{AID\,U}\} >$
$Check : ID_e, |T'_e - T_e| \le \Delta T$
$P_3 = D_{PR_U}\{E_{PU_U}\{Auth.Token\{AID_U\}', Sig_e, Data''_{AID\,U}\}$
$Compare :$
$Auth.Token\{AID_U\}' \overset{?}{=} Auth.Token\{AID_U\}$
$Verify : Sig_e$
$Compare :$
$H(Data''_{AID\,U}) \overset{?}{=} H(Data'_{AID_U})$
$Get : Data''_{AID\,U}$

Figure 7: Data transfer phase in our scheme

3.3.1 Data Transfer Process to Store Data (Step to Store Data)

Step 1. [*User client sends* $<T_U, C_1>$ *to ePMC*]

To store data at the ePMC, the user client generates the electronic signature value, $Sig_U = E_{PR_U}\{H(Data_{AID_U})\}$, for data $Data_{AID_U}$. Then, the user client sends $C_1 = E_{PU\,e}\{Auth.Token\{AID_U\},$

$Sig_U, Data'_{AIDU}\}$, obtained by encrypting the authentication token, $Auth.Token\{AID_U\}$, the electronic signature value, Sig_U, and data, $Data'_{AIDU}$, which were obtained during the authentication phase using the ePMC's public key and the time-stamp value, T_U, to the ePMC.

$$\begin{cases} \mathit{Generate}: \mathit{Digital\ signature}\ \boldsymbol{E_{PRU}\{H(Data_{AIDU})\}}\ \mathit{with}\ \boldsymbol{Data_{AID_U}}\ \mathit{using}\ \boldsymbol{PR_U} \\ \qquad\qquad\qquad\qquad \boldsymbol{Sig_U} = E_{PRU}\{H(Data_{AIDU})\} \\ \mathit{Encrypt}: \boldsymbol{E_{PU_e}\left\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}\right\}} \\ \qquad\qquad\qquad \mathit{with}\ \boldsymbol{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}}\ \mathit{using}\ \boldsymbol{PU_e} \\ \qquad\qquad \boldsymbol{C_1} = E_{PU_e}\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}\} \\ \mathit{Send}: \boldsymbol{T_U, C_1} \end{cases}$$

Step 2. [*ePMC checks* T_U, P_1, *store* $Data'_{AID_U}$ *and the end of data transfer phase*]

The ePMC checks the validity of the user client's time stamp value, T_U, and obtains the authentication token, $Auth.Token\{AID_U\}$, and electronic signature value, Sig_U, of the plain text, P_1, by performing $D_{PR_e}\left\{E_{PU_e}\left\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}\right\}\right\}$, which decrypts the received encrypted message, C_1, with its private key. Afterward, the ePMC verifies the hash value of the electronic signature value, Sig_U, and the authentication token, $Auth.Token\{AID_U\}$, generated during the authentication phase using the user client's public key, PU_U. At this point, the ePMC compares and verifies the hash value, $H(Data_{AIDU})$, and the hash value, $H(Data'_{AIDU})$, for data $Data'_{AID_U}$ of plain-text P_1, stores the received data $Data'_{AIDU}$, and terminates the data transfer phase.

$$\begin{cases} \mathit{Check}: \mathit{User\ client's\ timestamp\ validation}\ |\boldsymbol{T'_U} - \boldsymbol{T_U}| \le \Delta T \\ \mathit{Decrypt}: \boldsymbol{D_{PR_e}\left\{E_{PU_e}\left\{Auth.Token\{AID_U\}, Sig_U, Data'_{AID_U}\right\}\right\}}\ \mathit{using}\ \boldsymbol{PR_e} \\ \qquad\qquad\qquad\qquad \boldsymbol{P_1} = D_{PR_e}\{C_1\} \\ \mathit{Verify}: \boldsymbol{Auth.Token\{AID_U\}}\ \mathit{and} \\ \qquad\qquad\qquad \boldsymbol{H(Data_{AIDU})} \leftarrow \boldsymbol{Sig_U}\ \mathit{using}\ \boldsymbol{PU_U} \\ \mathit{Compare}: \boldsymbol{H(Data'_{AID_U})}? = \boldsymbol{H(Data_{AIDU})} \\ \mathit{Store}: \boldsymbol{Data'_{AID_U}} \end{cases}$$

3.3.2 Data Transfer Process to Request Data (Step to Request Data)

Step 1. [*User client sends <Data.Request*, T_U, C_2*> to ePMC*]

To request data, the user client sends $C_2 = E_{PU_e}\{Auth.Token\{AID_U\}\}$, obtained by encrypting the authentication token, $Auth.Token\{AID_U\}$, with the ePMC's public key, time-stamp value, T_U, and data request message, *Data.Request* to ePMC.

$$\begin{cases} \mathit{Request}: \mathit{User\ client}\ \mathbf{Data}.\boldsymbol{Request}\ \mathit{to\ ePMC} \\ \mathit{Send}: \boldsymbol{T_U, C_2} \end{cases}$$

Step 2. [*ePMC checks* T_U, P_2 *and sends <*ID_e, T_e, C_3*> to User client*]

The ePMC checks the validity of the user client's time-stamp value, T_U, obtains and verifies the authentication token, $Auth.Token\{AID_U\}$, of the plain text, P_2, by performing $D_{PR_e}\{E_{PU_e}\{Auth.Token\{AID_U\}\}\}$, which decrypts the received encrypted message, C_2, with its private key, PR_e, and then obtains the stored data, $Data'_{AID_U}$, by searching AID_U.

Afterward, the ePMC generates the electronic signature value, $Sig_e = E_{PRe}\{H(Data'_{AIDU})\}$, for the data, $Data'_{AIDU}$, and sends $C_3 = E_{PUU}\{Auth.Token\{AID_U\}', Sig_e, Data''_{AIDU}\}$, obtained by encrypting the authentication token, $Auth.Token\{AID_U\}'$, the electronic signature value, Sig_e, and data, $Data''_{AIDU}$, with the user client's public key, its identifier, ID_e, and its time-stamp value, T_e, to the user client.

$$\left\{\begin{array}{l}
\mathit{Check}: \mathit{User\ client'timestamp\ validation}\ |\boldsymbol{T}'_{U}-\boldsymbol{T}_{U}|\le \Delta T\\
\mathit{Decrypt}: \boldsymbol{D}_{\boldsymbol{PR}_e}\{\boldsymbol{E}_{\boldsymbol{PU}_e}\{\boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}\}\}\ \mathit{using}\ \boldsymbol{PR}_e\\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad P_2 = D_{PR_e}\{C_2\}\\
\mathit{Verify}: \boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}\\
\mathit{Search}: \boldsymbol{AID}_U\ \mathit{to\ get}\ \mathit{Data}'_{AID_U}\\
\mathit{Get}: \boldsymbol{Data}'_{\boldsymbol{AID}_U}\\
\mathit{Generate}: \mathit{Digital\ signature}\boldsymbol{E}_{\boldsymbol{PR}_e}\left\{\boldsymbol{H}(\boldsymbol{Data}'_{\boldsymbol{AID}_U})\right\}\ \mathit{with}\ \boldsymbol{Data}'_{\boldsymbol{AID}_U}\mathit{using}\ \boldsymbol{PR}_e\\
\qquad\qquad\qquad\qquad\qquad \boldsymbol{Sig} = E_{PR_e}\left\{H(\mathit{Data}'_{AID_U})\right\}\\
\mathit{Encrypt}: \boldsymbol{E}_{\boldsymbol{PU}_U}\left\{\boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}', \boldsymbol{Sig}_e, \boldsymbol{Data}''_{\boldsymbol{AID}_U}\right\}\\
\qquad\qquad\qquad \mathit{with}\ \mathit{Auth.Token}\{AID_U\}', \mathit{Sig}_e, \mathit{Data}''_{AID_U}\ \mathit{using}\ PU_U\\
\qquad\qquad C_3 = \boldsymbol{E}_{\boldsymbol{PU}_U}\left\{\boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}', \boldsymbol{Sig}_e, \boldsymbol{Data}''_{\boldsymbol{AID}_U}\right\}\\
\mathit{Send}: \boldsymbol{ID}_e, \boldsymbol{T}_e, \boldsymbol{C}_3
\end{array}\right.$$

Step 3. [*User client checks* ID_e, T_e, P_3, *get* $Data''_{AID_U}$ *and the end of data transfer phase*]

The user client checks the validity of the ePMC identifier, ID_e, and the time-stamp value, T_e, and obtains the authentication token, $Auth.Token\{AID_U\}'$, and the electronic signature value, Sig_e, of the plain text, P_3, by performing $D_{PR_U}\left\{E_{PU_U}\left\{Auth.Token\{AID_U\}', Sig_e, Data''_{AID_U}\right\}\right\}$, which decrypts the received encrypted message, C_3, using their private key, PR_U. The user client then compares and verifies the obtained authentication token, $Auth.Token\{AID_U\}'$, and the existing authentication token, $Auth.Token\{AID_U\}$, and verifies the hash value, $H(Data'_{AIDU})$, of the electronic signature value, Sig_e, using the ePMC's public key, PU_e. At this point, the user client compares and verifies the hash values, $H(Data'_{AIDU})$ and $H(Data''_{AIDU})$, for the data, $Data''_{AID_U}$, of the plain text, P_3. It then obtains the received data, $Data''_{AIDU}$, and terminates the data transfer phase.

$$\left\{\begin{array}{l}
\mathit{Check}: \mathit{ePMC's\ identity}\ \boldsymbol{ID}_e\ \mathit{and\ timestamp\ validation}\ |\boldsymbol{T}'_e-\boldsymbol{T}_e|\le \Delta\boldsymbol{T}\\
\mathit{Decrypt}: \boldsymbol{D}_{\boldsymbol{PRU}}\{\boldsymbol{E}_{\boldsymbol{PUU}}\{\boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}', \boldsymbol{Sig}_e, \boldsymbol{Data''AID}_U\}\}\ \mathit{using}\ \boldsymbol{PR}_U\\
\qquad\qquad\qquad\qquad\qquad\qquad \boldsymbol{P}_3 = D_{PRU}\{C_3\}\\
\mathit{Compare}: \boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}'? = \boldsymbol{Auth.Token}\{\boldsymbol{AID}_U\}\\
\mathit{Verify}: \boldsymbol{H}(\boldsymbol{Data}'_{\boldsymbol{AIDU}}) \leftarrow \boldsymbol{Sig}_e\ \mathit{using}\ \boldsymbol{PU}_e\\
\mathit{Compare}: \boldsymbol{H}(\boldsymbol{Data}''_{\boldsymbol{AIDU}})? = \boldsymbol{H}(\boldsymbol{Data}'_{\boldsymbol{AIDU}})\\
\mathit{Get}: \boldsymbol{Data}''_{\boldsymbol{AID}_U}
\end{array}\right.$$

4 Security Analysis of the Proposed Scheme

In this section, a security analysis is performed to respond to the analyzed security threats of the e-prescription system.

4.1 Information Sniffing Prevention

Exchanged information can be leaked through data-packet sniffing during the authentication and data-transfer phases. To prevent this, the proposed scheme applies public key encryption using $\{PU_{Entity}, PR_{Entity}\}$ for data exchanged between entities. Thus, only valid entities can perform data encryption/decryption. Moreover, the information that can be obtained by attackers is $<Auth.Request, AID_U, ID_e, T_X, C_X>$ from the authentication process, and $<T_X, C_X, Data.Request, ID_e>$ from the data-sharing process. As AID_U is an anonymous identifier, attackers cannot identify the user client, even if they obtain the ID. Furthermore, additional information can be neither used nor identified, even if it is leaked from the e-prescription system. Thus, data leakage is prevented.

4.2 Information Falsification Prevention

Data can be falsified by attackers during authentication and data-transfer phases or as a result of errors during data communication. This presents reliability problems for the generated data collected and processed by the e-prescription system. During authentication, attackers can obtain the client's anonymous identifier, AID_U. Using this, they can falsify data $\{Data'_{AID_U}\}$ for sent data $\{Data_{AID_U}\}$. Then, the ePMC performs integrity $H(Data_{AID_U})? = H(Data'_{AID_U})$ verification based on the signature value, $Sig_U = E_{PRU}\{H(Data_{AIDU})\}$, for the data received from an authenticated user. If these data are falsified, this equation is not completed, and the data falsification threat is prevented.

4.3 Entity Spoofing Prevention

When data are generated, collected, and processed from unauthorized entities, the reliability problem of the e-prescription system is exacerbated, and violations occur. To prevent this, the proposed scheme performs mutual authentication during registration and authentication phases. In particular, Fast IDentity Online (FIDO) authentication is applied to prevent threats in network and physical environments, such as when agents steal a clinician's computer to issue illegal e-prescriptions.

Consequently, the biometric authentication element, $B_U \leftarrow H(Bio_U||AID_U||x)$, generated based on the biometric information, Bio_U, selected during the registration phase cannot be generated, even if the attacker obtains the user client's Bio_U, AID_U, as the secret value, x, cannot be known. Hence, even if the anonymous identifier, AID_X, and identifier, ID_X, of an entity are obtained by sniffing authentications, the entity is spoofed when $C_2 = E_{PUe}\{Sig_U, B'_U, \sigma'\}$ is sent, the encrypted message is decrypted by the ePMC, and mutual authentication is performed by comparing $\rho'' \leftarrow B'_U \oplus \sigma'$ and $\rho''? = \rho'$ through the biometric authentication element. Thus, threats are prevented.

4.4 Repudiation Prevention

A user client who uses the e-prescription system may repudiate e-prescription data reception. Consequently, authentication and data-transfer phases are carried out using the digital signature value, $Sig_U = E_{PRU}\{AID_U, B'_U\}$ or $Sig_U = E_{PRU}\{H(Data_{AIDU})\}$, and repudiation attacks can be prevented using the signer's private key, PR_X.

4.5 Prevention of Information Reuse

Attackers can cause availability violations in authentication and data processing using retransmission attacks on data exchanged between entities during authentication or data-transfer phases. Attackers can sniff information containing an *Auth.Request* message requesting authentication by a user client or information containing a *Data.Request* message. Then, when the attacker tries to access the data process used for e-prescriptions by reusing the information in the authentication or data-transfer process, the ePMC can prevent reuse attacks by performing $|T'_X - T_X| \leq \Delta T$ for the time stamp, T_X, which is the data validation element included in each message, making the attack invalid.

4.6 Privacy Invasion Prevention

The ripple effect is high when data are leaked from the e-prescription system because it uses sensitive data. Furthermore, when FIDO is used, the biometric information is replicated in the user client's smart devices. Hence, privacy invation can occur if the user client's biometric information leaks during authentication. Therefore, the user client's anonymous identifier, $AID_U \leftarrow ID_U \oplus \rho$, is used during the authentication phase to ensure anonymity, even if it is sniffed by an attacker. Thus, privacy invasion can be prevented, even if the biometric authentication element, B_U, for biometric information Bio_U is leaked. Hence, the actual biometric information, Bio_U, cannot be known.

5 Conclusions

This study proposed an authentication and data-sharing scheme for agents using a centralized data management center for an e-prescription system and verified its potential security capability. The innovative capability provided by this study improves the security of the growing e-prescription system in the medical industry. The security of medical services has been long-researched [10], but most studies viewed data flow in scope that was too large. Hence, detailed response measures to potential security threats were insufficient. Moreover, with the application of many new technologies, the surface area of attack has become larger, and the attack paths have become more diversified. Thus, a more advanced response to security threats is needed. Moreover, in countries with advanced medical welfare programs, such as Australia and Sweden, related information used in medical information management systems is transmitted and managed through centralized data management centers operated by the state. The innovations provided by this tudy will improve the security and efficiency of these systems over legacy medical service institutions. The security of medical services can be secured by preventing illegal access and leakage of medical information via proper authentication and data-sharing capabilities. Furthermore, the safety and security provided by e-prescription system services can be enhanced by constructing a centralized data management-centric environment and acquiring security techniques based on the proposals of this study. This study is expected to contribute to the security of IoBE worldwide by considering multi-industry linkages.

Funding Statement: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT; No. 2021R1A2C2011391).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] V. S. Naresh, S. S. Pericherla, P. S. R. Murt and S. Reddi, “Internet of Things in healthcare: Architecture, applications, challenges, and solutions,” *Computer Systems Science and Engineering*, vol. 35, no. 6, pp. 411–421, 2020.

[2] H. Su, X. Yuan, Y. Tang, R. Tian, E. Sun *et al.,* “A learning-based power control scheme for edge-based eHealth IoT systems,” *KSII Transactions on Internet and Information Systems*, vol. 15, no. 12, pp. 4385–4399, 2021.

[3] M. Lee, J. Jang-Jaccard and J. Kwak, “Novel architecture of security orchestration, automation and response in internet of blended environment,” *Computers, Materials & Continua*, 2022. http://dx.doi.org/10.32604/cmc.2022.028495.

[4] Q. A. Bui, W. B. Lee, J. S. Lee, H. L. Wu and J. Y. Liu, “Biometric-based key management for satisfying patient’s control over health information in the HIPAA regulations,” *KSII Transactions on Internet and Information Systems*, vol. 14, no. 1, pp. 437–454, 2020.

[5] M. Samadbeik, M. Ahmadi, F. Sadoughi and A. Garavand, “A comparative review of electronic prescription systems: Lessons learned from developed countries,” *Journal of Research in Pharmacy Practice*, vol. 6, no. 1, pp. 3–11, 2017.

[6] N. Noorbakhsh-Sabet, R. Zand, Y. Zhang and V. Abedi, “Artificial intelligence transforms the future of health care,” *The American Journal of Medicine*, vol. 132, no. 7, pp. 795–801, 2019.

[7] D. Kim and J. Kwak, “A Framework for preventing illegitimate e-prescribing practices,” in *Advances in Computer Science and Ubiquitous Computing:CSA-CUTE 17*, Singapore: Springer, pp. 648–653, 2018.

[8] D. Kim and J. Kwak, “The framework of 3P-based secure ehealth-information system,” in *Proc. IEEE Int. Conf. on Platform Technology and Service (PlatCon)*, Jeju, Republic of Korea, pp. 1–6, 2018.

[9] D. Kim and J. Kwak, “A study on secure information flow control in centralized ehealth-information system,” in *Proc. 12th Asia Pacific Int. Conf. on Information Science and Technology (APIC-IST)*, Chiang Mai, Thailand, pp. 120–122, 2017.

[10] X. Liu, Y. Li., J. Qu and Y. Ding, “A lightweight pseudonym authentication and key agreement protocol for multi-medical server architecture in TMIS,” *KSII Transaction on Internet and Information Systems*, vol. 11, no. 2, pp. 924–944, 2017.