

Red Alert: Break-Glass Protocol to Access Encrypted Medical Records in the Cloud

Marcela Tuler de Oliveira
dept. *KEBB/BMEP, Amsterdam UMC*
University of Amsterdam
Amsterdam, Netherlands
m.tuler@amsterdamumc.nl

Antonis Michalas
Computing Sciences
Tampere University
Tampere, Finland
antonios.michalas@tuni.fi

Adrien E. D. Groot
Neurology, Amsterdam UMC
University of Amsterdam
Amsterdam, Netherlands
a.e.groot@amsterdamumc.nl

Henk A. Marquering
dept. *BMEP, Amsterdam UMC*
University of Amsterdam
Amsterdam, Netherlands
h.a.marquering@amsterdamumc.nl

Silvia Delgado Olabariaga
dept. *KEBB, Amsterdam UMC*
University of Amsterdam
Amsterdam, Netherlands
s.d.olabariaga@amsterdamumc.nl

Abstract—Availability of medical records during an emergency situation is of paramount importance since it allows healthcare professionals to access patient’s data on time and properly plan the next steps that need to be taken. Cloud storage has the potential to provide a solution to the problem of data unavailability during an emergency situation. However, sharing medical records raises several concerns about security and privacy. In this paper, we study the problem of how to share encrypted patients’ data during an emergency situation. To this end, we propose a protocol through which a team of healthcare professionals can securely decrypt the medical records of a patient who is under an emergency situation (e.g. acute stroke). Furthermore, our protocol ensures that a team of healthcare professionals will only have access to the patient’s data for the time needed to complete a specific process related to the patient’s situation (e.g. transfer patient to the hospital). In our study, the dynamically granting and revoking data access during an emergency treatment is the main novelty.

Index Terms—Attribute-Based Encryption; e-Health Privacy; Electronic Medical Records; Emergency care, Secure Cloud Storage.

I. INTRODUCTION

The use of Electronic Medical Records (EMR) improves the overall quality of care that a patient receives since their use can lead to a substantial reduction of unnecessary investigations and improvement of communication between the healthcare professionals that are involved in the treatment [1]. Therefore, the availability of EMR, especially when a patient is under an emergency situation, is of paramount importance. For example, in stroke treatment, the phrase ‘*Time is brain*’ conveys the idea that minutes can make the difference between life and death [2]. Hence, guaranteeing that the patient’s EMR will be available to healthcare professionals involved in the acute

stroke care can save time and improve the efficiency of decision-making processes, leading to higher quality of care.

EMR management on cloud infrastructure increases the availability of data. However, this poses new challenges for security and privacy [3]. Initially, researchers proposed to send sensitive data to the cloud service provider, where it would be encrypted and stored. In this case, the key used for data encryption is known by the cloud provider, which does not protect the EMR against internal attacks [4]. To overcome this, other studies rely on encrypting the EMR with a secret key before storing it in the cloud. In most scenarios, the secret key needs to be pre-shared with all users that wish or need to access the EMR [5], [6]. However, this makes access control inefficient. In particular, when a user needs to be revoked, the EMR must be re-encrypted with a fresh key and the new key must be distributed to the other legitimate users.

In the case of acute stroke care, having access to patient data is vital. Therefore, it is necessary to provide access to encrypted data - even if the patient cannot consent explicitly. The so-called ‘break-glass’ access mechanism provides emergency access to the patient’s EMR in such situations. Although some studies approach the break-glass access to encrypted EMR [7]–[9], the revocation after emergency is still a problem. For security and privacy sake, immediately after the emergency situation ends, the access needs to be revoked. In addition, revoking a user must not affect the other users.

Our Contribution: In this paper, we describe a protocol to provide access to a patient’s encrypted EMR during acute stroke treatment with an additional security mechanism that ensures authorisation only for the period which the access is necessary. The protocol securely enables sharing of medical records among multi-treatment teams through a cloud platform. The proposed solution adopts the concept of Attribute-Based Encryption (ABE) associated with policies defined explicitly for the emergency situation. Additionally, it adopts tokenized authentication to dynamically grant and

This work was funded by the ASCLEPIOS project (Advanced Secure Cloud Encrypted Platform for Internationally Orchestrated Solutions in Healthcare) of the European Unions Horizon 2020 research and innovation program under grant agreement No. 826093.

revoke access during the timeline of acute stroke treatment.

Organization: Section II discusses related works, and Section III summarizes the flow of patient information during stroke emergency. Section IV defines the cryptographic primitives used throughout the paper. In Section V we present the main entities that participate in our system model and in Section VI we define both the problem statement and the considered threat model. In Section VII we describe our protocol and in Section VIII we analyze its security against malicious behaviour. Section IX presents a discussion and final remarks.

II. RELATED WORK

Many studies have addressed access control to medical records. Here, however, we focus on emergency situations where the break-glass condition is valid. Very few research works have considered this requirement.

One of the earliest arguments for a break-glass concept was formulated by Povey [10]. It states that the basic approach of an optimistic security system is to assume that any emergency situation requesting data access is legitimate and should be granted. Petrisch and Bruker presented in [11] a generic break-glass model where the data subjects are permitted to override specific access control permissions. In [12] Zhang *et al.* proposed a concrete break-glass method based on two-factor encryption: password-based encryption and master secret key based encryption. In [13] the authors presented 'Rampole', a model that implements access permissions in a fine-grained manner using a declarative query language to specify a break-glass decision procedure. All the approaches described above do not support attribute-based access control.

In [7] the authors leveraged ABE techniques to control access to patient data. This study approaches break-glass access under emergency scenarios using a unique authority which authenticates the medical staff to access the data. To revoke access, the data needs to be re-encrypted with a new key. Brucker *et al.* [8] presented an integration of fine-grained break-glass concepts into a system based on ABE. The authors present multi-levelled break-glass access control; however, the solution does not enable revoking access after it has been granted. Yang *et al.* [9] proposed an ABE access control in which the patient pre-shares her password with the emergency contact person. When the patient is in an emergency situation, the contact person utilizes the password to derive the break-glass key and to decrypt the patient's medical files.

Even though [7]–[9] present interesting solutions for the break-glass situation, they do not provide a concrete and efficient solution for access revocation. Our approach overcomes the revocation problem by using a Ciphertext-Policy ABE (CP-ABE) scheme combined with an access control token scheme to grant and revoke access dynamically without re-encrypting the patient EMR. In addition, our protocol supports the involvement of multi-treatment teams, even from different institutions, which brings the solution closer to a real emergency scenario.

III. PATIENT DATA SHARING DURING STROKE EMERGENCY

Emergency care for acute stroke involves professionals at the emergency call centre, ambulance service and primary and comprehensive stroke hospitals, all of which need to share information on a break-glass access mechanism. Below we describe a typical scenario, providing high-level information of involved parties and the basic information exchange that takes place between them.

When a patient suffers a stroke, the patient or someone who is present with the patient is the first to contact the emergency call centre by phone. The call centre team is composed of trained healthcare workers who are able to determine if there is an emergency situation, and how to address it best. During the phone call, the call centre professional follows a triage process, collecting information about the time of the stroke onset, personal data (e.g. age, gender etc.) and impressions about the patient conditions.

When there is a suspicion of stroke, the call centre professional contacts the ambulance service and shares the collected information about the patient. An ambulance that is closer to the event location is sent from the closest regional centre and, as soon as the ambulance arrives, the ambulance team continues the triage process. They perform examinations, measurements and medical procedures at the patient's location and during the travel to the hospital. When the ambulance arrives at the hospital, all the information about patient's condition is orally shared with the hospital team.

During the travel, the ambulance professional communicates by phone with the proposed destination hospital. Therefore, the hospital prepares the emergency room and gathers the team with the relevant experts to receive the patient (e.g. neurologist, interventional neuroradiologist, anesthesiologist, nurses). If the patient already had a record in this hospital local system, the hospital team can access the patient's medical record. If there is no information about the patient, the hospital team may attempt to contact other hospitals to retrieve the patient's medical record. Furthermore, the hospital team collects additional data about the patient, which is stored at the patient's EMR at the treating hospital. In the case of a patient with a large vessel occlusion eligible for additional endovascular treatment, the patient needs to be transported by a second ambulance to the comprehensive stroke hospital. Therefore, it is necessary that the teams in the second ambulance and second hospital also access and update the patient's medical record.

Note that in this case three or more teams are involved in the treatment, requiring access to the patient's EMR and generating new content for it. Therefore, to improve accessibility to medical records and protect patient's privacy, it is necessary to dynamically grant and revoke access to the EMR.

IV. CRYPTOGRAPHIC PRIMITIVES

We first define the basic cryptographic primitives that are used throughout the paper and then we continue with the definition of a CP-ABE scheme as described in [14].

The set of all binary strings of length n is denoted by $\{0, 1\}^n$, and the set of all finite binary strings as $\{0, 1\}^*$.

Given a set V , we refer to the i^{th} element as v_i . Additionally, we use the following notations for cryptographic operations throughout the paper:

- For an arbitrary message $m \in \{0, 1\}^*$, $c = \text{Enc}(K, m)$ denotes a symmetric encryption of m using the secret key $K \in \{0, 1\}^*$, and $m = \text{Dec}(K, c) = \text{Dec}(K, \text{Enc}(K, m))$ is the corresponding symmetric decryption operation.
- We denote by pk/sk a public/private key pair for an IND-CCA2 secure public key encryption scheme PKE. An encryption of message m under the public key pk is denoted by $c = \text{Enc}_{\text{pk}}(m)$ and the corresponding decryption operation by $m = \text{Dec}_{\text{sk}}(c) = \text{Dec}_{\text{sk}}(\text{Enc}_{\text{pk}}(m))$.
- $\sigma = \text{Sign}_{\text{sk}}(m)$ denotes a digital signature over a message m . The corresponding verification operation for a digital signature is denoted by $b = \text{Verify}_{\text{pk}}(m, \sigma)$, where $b = 1$ if the signature is valid, and $b = 0$ otherwise.
- A one-way hash function (H) over a message m is denoted by $H_m = H(m)$.
- We denote by $\tau = \text{RAND}(n)$ a random binary sequence of length n , where $\text{RAND}(n)$ represents a random function that takes a binary length argument n as input and gives a random binary sequence of this length in return¹.

A CP-ABE scheme is a tuple of the following four algorithms:

1. CPABE.Setup is a probabilistic algorithm that takes as input a security parameter λ and outputs a master public key MPK and a master secret key MSK. We denote this by $(\text{MPK}, \text{MSK}) \leftarrow \text{Setup}(1^\lambda)$.
2. CPABE.Gen is a probabilistic algorithm that takes as input a master secret key, a set of attributes $\mathcal{A} \in \Omega$ and the unique identifier of a user, and it outputs a secret key that is bound both to the corresponding list of attributes and the user. We denote this by $(\text{sk}_{\mathcal{A},i}) \leftarrow \text{Gen}(\text{MSK}, \mathcal{A}, u_i)$.
3. CPABE.Enc is a probabilistic algorithm that takes as input a master public key, a message m and a policy $P \in \mathcal{P}$. After a proper run, the algorithm outputs a ciphertext c_P which is associated to the policy P . We denote this by $c_P \leftarrow \text{Enc}(\text{MPK}, m, P)$.
4. CPABE.Dec is a deterministic algorithm that takes as input a user's secret key and a ciphertext and outputs the original message m iff the set of attributes \mathcal{A} that are associated with the underlying secret key satisfies the policy P that is associated with c_P . We denote this by $\text{Dec}(\text{sk}_{\mathcal{A},i}, c_P) \rightarrow m$.

V. SYSTEM MODEL

The system model presented here is based on the model introduced in [15]. Below we present an overview of the main entities and the most relevant communication between them.

¹We assume that a true random function is replaced by a pseudo-random function, the input-output behaviour of which being "computationally indistinguishable" from that of a true random function.

Cloud Service Provider (CSP): The cloud computing environment is based on a trusted Infrastructure-as-a-Service (IaaS) provider. The IaaS platform consists of cloud hosts that operate virtual machine (VM) guests and communicate through a network. In our model, we require that the IaaS runs a protocol similar to the one described in [16], where the integrity of the underlying CSP is verified. In principle, such integrity verification can be added to any IaaS. A CSP stores patients' EMR encrypted under a CP-ABE scheme. Additionally, the CSP is responsible for controlling the access to the encrypted EMR.

Registration Authority (RA): responsible for the registration of all healthcare entities and users. The RA is responsible for generating user attributes that will be used for the proper authorization (e.g. membership to a particular treatment team). The RA can run as a separate third party, but can be also implemented as part of the CSP.

Master Authority (MA): has a master secret key MSK and a public key MPK. The master key is kept private, while the public key is known to everyone. Additionally, the MA uses MSK to generate CP-ABE secret keys for users, based on user attributes to authorize recovery of encrypted EMR. The MA is also responsible for granting and revoking the access tokens.

User: We consider three different types of users: patients, healthcare professionals and healthcare entities. The set of all patients registered at RA is denoted by $\mathcal{U} = \{u_1, \dots, u_{N_u}\}$ and the set of all registered healthcare professionals is denoted as $\mathcal{S} = \{s_1, \dots, s_{N_s}\}$. A healthcare entity is a special type of user represented by an attested smart device. This device serves to confirm the following treatment team locations: Emergency Call Centre (e), Ambulance (a) and Hospital (h). A treatment team is a group of professionals co-located at one of the entities that attest each other's involvement in the emergency situation.

Each user from \mathcal{U}, \mathcal{S} and the healthcare entities has a unique public/private key pair (pk/sk) used to communicate securely through an IND-CCA2 secure public key encryption scheme PKE and an EUF-CMA secure signature scheme sign.

VI. PROBLEM STATEMENT AND THREAT MODEL

A. Problem Statement

Let u_i be a patient from the set \mathcal{U} and $s_j \in \mathcal{S}$ be a member of one of the stroke treatment teams. Let's assume that u_i has a set of N different files stored in the CSP. We denote this set of files as $D_i = \{d_1^i, \dots, d_N^i\}$. The problem is to find a way to achieve the following:

1. Enable access to the content of each $d_l^i \in D_i$ to s_j involved in the treatment of u_i ;
2. User s_j has access to D_i if and only if she has a legitimate role in the treatment team of u_i at the time, as given by a valid policy;
3. Access control to D_i should be granted and revoked dynamically as requested for the patient's treatment. This should not require to decrypt and re-encrypt the file with a fresh key, and it should not affect the access by the rest of the legitimate users.

Additionally, two steps are internal to the MA: RAP.GenKey and RAP.GenToken. An overview of all messages is presented in Table I. We assume that during the protocol, each party verifies the integrity and freshness of the message.

RAP.Setup : Each model entity (MA, CSP, and users) obtains a public/secret key pair (pk, sk) and publishes its public key while keeping the private key secret. The following keys are generated: (pk_{CSP}, sk_{CSP}) for the CSP and (pk_{MA}, sk_{MA}) for the MA. Furthermore, MA runs CPABE.Setup to generate a master public/secret key pair (MPK, MSK).

Each user (from \mathcal{U} or \mathcal{S}) is registered through a central RA². During registration, each user receives a unique identifier i and a public/private key pair (pk_i/sk_i) . Furthermore, a set of attributes \mathcal{A} based on user's personal data is created. For patients, identifying attributes such as name and surname could be used. For healthcare professionals, attributes include identification and function in the organization, and in particular the membership and role in an emergency treatment team. This set of attributes is sent to MA, and stored for future use to generate ABE secret keys $sk_{A,i}$ to decrypt patient EMR.

RAP.StoreData : A patient u_i stores her EMR in the CSP as a ciphertext c_P^i . The generation of c_P^i requires running $CPABE.Enc(MPK, d_i^i, P)$, where d_i^i is the file that the user wishes to encrypt and P is a policy defining who can access d_i^i . In this paper we explicitly focus on the problem of how only authorized users can access a patient's EMR during an emergency session. To this end, P needs to always contain a condition that will allow a user s_j to successfully decrypt $d_i^i \in D_i, \forall l \in [1, |D_i|]$. Among other conditions in P , the following should be added for u_i : "...OR (Emergency=TRUE AND TreatmentTeamMember=TRUE AND UserInEmergency=i)". A professional s_j will be then granted access to the EMR of u_i only when her attributes satisfy this policy.

RAP.BreakGlass (II) : Through this process the MA acknowledges the emergency event for a patient u_i and begins the emergency session. This takes place when a patient, or someone on her behalf, contacts the call centre team by phone. The emergency call is received by $s_e \in \mathcal{S}$. After identifying patient u_i , s_e contacts MA to notify the emergency event and requests to become part of the session by sending the message $m_1 = \langle r_1, Enc_{pk_{MA}}(u_i, t, s_e), \sigma_{s_e}(H(r_1||u_i||t||s_e)) \rangle$, where r_1 is a random number generated by s_e and t is the registered time of the call. Upon reception, MA confirms that s_e is indeed part of the call centre team. In this proposal we trust that s_e will contact the MA only if she receives a legitimate phone call from the patient or someone on behalf of the patient. The phone call authentication is very important, but it is considered outside the scope of this paper. MA includes s_e to the emergency session and generates the ABE

key (using RAP.GenKey) and token (using RAP.GenToken) for the emergency call centre team.

The MA then sends the following message to s_e : $m_2 = \langle r_2, \tau_e, Enc_{pk_{sk_e}}(sk_{A_e,i}), \sigma_{MA}(H(r_2||\tau_e||sk_{A_e,i})) \rangle$, where $sk_{A_e,i}$ is the ABE key including emergency attributes, and τ_e is the access token for the call centre team.

RAP.JoinTeam (III) : In this step the MA associates users in a treatment team to an existing emergency session. This is initiated by some user who is already part of the emergency session, and therefore in possession of an ABE key and a valid token. At first the call centre includes the ambulance team, which later invites the hospital team, to join the session (see figure 2). To add a new team to the emergency session, the following message is sent by a user s_j to the MA: $m_3 = \langle r_3, Enc_{pk_{MA}}(x, s_{x1}, s_{x2}), \sigma_{s_j}(H(r_3||x||s_{x1}||s_{x2})) \rangle$, where $x \in \{a, h\}$ is the attested device in the ambulance a or hospital h , and s_{x1}, s_{x2} are two³ users co-located with x .

After receiving m_3 , MA confirms the identity of x and the users. Additionally, s_{x1} and s_{x2} need to prove that they are indeed together in the same location of x , which has been specified by s_j . For this confirmation, MA creates a challenge for x, s_{x1} and s_{x2} in the following way: MA generates a random number v , splits it into three random shares such that $v = v_0 + v_1 + v_2$, and creates challenge = $\langle Enc_{pk_x}(v_0), Enc_{pk_{s_{x1}}}(v_1), Enc_{pk_{s_{x2}}}(v_2) \rangle$. It then creates the message $m_4 = \langle r_4, challenge, \sigma_{MA}(H(challenge||r_4)) \rangle$ and sends it to x, s_{x1} and s_{x2} . Upon reception, each one recovers its own share (i.e., x uses sk_x to recover v_0 , s_{x1} uses sk_{x1} for v_1 , etc.). Then, each one sends back its location and the solved challenge value to the MA. For example, x sends $m_5 = \langle r_5, Enc_{pk_{MA}}(v_0, l_x), \sigma_x(H(r_5||v_0||s_{x1}||s_{x2}||l_x)) \rangle$ where l_x stands for the current location of x . Analogous messages are sent by s_{x1} and s_{x2} . After receiving all m_5 messages, MA checks if the locations are the same (i.e., $l_x = l_{x1} = l_{x2}$), and verifies the identities of all three users. MA then calculates v' by adding the received shares and checks if $v = v'$, in which case the verification is completed successfully. At this moment, MA includes all the team members (x, s_{x1}, s_{x2}) into the emergency session. It then generates the ABE key (using RAP.GenKey) and access token (using RAP.GenToken) for the team, and sends these to the users through a message analogous to m_2 presented above.

RAP.RetrieveData (IV) : After having received the ABE key and token, s_j from team $x \in \{e, a, h\}$ is ready to access the patient's data. First s_j sends a request to the CSP to access all ciphertexts in the EMR of u_i : $m_6 = \langle r_6, \tau_x, \sigma_{s_j}(H(r_6||\tau_x)) \rangle$, where τ_x is the token for team x . Note that the token is already encrypted with the CSP public key, and that it contains the identity of team members, u_i and token expiration time. The CSP verifies the identity of s_j and the token validity and authenticity.

After successful verification, the CSP retrieves u_i 's ciphertexts and forwards them to s_j through this message:

²For the sake of simplicity we assume a single RA for all organizations. Also, the same format is used to store EMR shared during emergency.

³Here we assume that at least two professionals are part of the team, but more could be included in similar way.

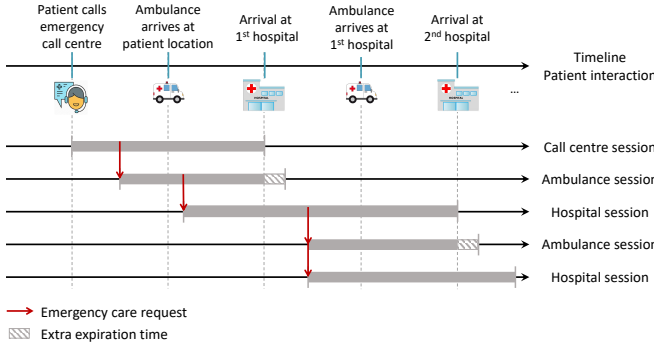


Fig. 2. Stroke care and data access session timelines. The top line represents events of interaction between healthcare provider entities and the patient. The others show the period of time that each team has access to the patient data.

$m_7 = \langle r_7, \text{Enc}_{\text{pk}_{s_i}}(c_P^i), \sigma_{\text{CSP}}(H(r_7 || c_P^i)) \rangle$, where c_P^i is the set of ciphertexts. Finally, through a secure application, s_j uses the emergency key $\text{sk}_{\mathcal{A}_e, i}$ to recover u_i 's EMR from c_P^i .

RAP.AddData (V): During and after patient's treatment, all teams may upload new files d_i^j to the patient EMR. These files include the report of the emergency treatment and need to be encrypted with the same policy P used by in **RAP.StoreData**. To do so, a professional s_j runs $\text{CPABE.Enc}(\text{MPK}, d_i^j, P)$ and the output is the ciphertext c_P^j . After that, s_j sends a message to CSP: $m_8 = \langle r_8, \tau_x, \text{Enc}_{\text{pk}_{\text{CSP}}}(c_P^j), \sigma_{s_j}(H(r_8 || \tau_x || c_P^j)) \rangle$, where τ_x is a token for the team x of which s_j is a member. The CSP checks the token validity and stores c_P^j .

RAP.RevokeAccess (VI): Access needs to be revoked when it is no longer necessary for patient treatment. This happens when a treatment team leaves the emergency session, which in our solution requires revocation of the token for that treatment team. After that moment, the CSP will no longer allow access to the ciphertexts of the patient to the users in that treating team.

In stroke treatment, the moments when the patient arrives and leaves the emergency care of the hospital define the end of involvement of treatment teams. When the patient arrives at the first hospital, the call centre team and the ambulance teams leave the emergency session. For the call centre, revocation of τ_e should be immediate. The ambulance team, however, is granted extra time after arrival at the hospital to add their reports into the medical record. The revocation of τ_a is therefore delayed (see figure 2). In principle, τ_h needs to be revoked when the patient leaves the hospital emergency care. However, if the patient needs to be transferred for treatment, the token for the first hospital will be revoked as soon as the patient arrives at the second hospital. For this reason, as soon as the patient arrives or leaves the hospital, s_h informs MA by sending $m_9 = \langle r_9, \text{E}_{\text{pk}_{\text{MA}}}(t), \sigma_{s_h}(H(r_9 || t)) \rangle$, where t can be used as the time that the patient has been delivered to the hospital or the time the patient leaves the hospital emergency care. As soon as the MA knows the moment when the patient arrived or left the hospital, it sends a message to the CSP as follows: $m_{10} = \langle r_{10}, \tau_x, \sigma_{\text{MA}}(H(r_{10} || \tau_x)) \rangle$,

where τ_x is the token for the team x which is revoked from the emergency session. The emergency session ends when all tokens associated with it have expired or explicitly revoked. After this, no new team is allowed to join the session anymore.

RAP.GenKey: The MA runs $\text{CPABE.Gen}(\text{MSK}, \mathcal{A}_e, u_i)$ to generate an ABE key for a treatment team. As attributes, among others, MA inserts in \mathcal{A}_e the following: "[Emergency, TreatmentTeamMember, i]". This guarantees that the generated key will satisfy the policy bound to u_i 's ciphertexts. The ABE key is sent to the users in a treatment team that has joined the emergency session.

RAP.GenToken: The MA generates a token for each team involved in the emergency session (call centre e , ambulances a and hospitals h). These tokens are used to prove to the CSP that a user s_j has a valid permission to access u_i 's data. Professionals in the different teams share the token for that team. A token for team $x \in \{e, a, h\}$ is defined as follows: $\tau_x = (t_{\text{gen}}, t_{\text{exp}}, \text{Enc}_{\text{pk}_{\text{CSP}}}(r, s_{x1}, s_{x2}, u_i), \sigma_{\text{MA}}(H_\tau))$, where r is a random number, t_{gen} is the time when the token was generated by MA, t_{exp} is the default expiration time and H_τ is the hash of $(t_{\text{gen}} || t_{\text{exp}} || r || x || s_{x1} || s_{x2} || u_i)$. Note that the tokens also include identification about all team members certified through the challenge (see **RAP.JoinTeam**), enabling the CSP to implement user authentication and traceability.

Tokens need to be used within a predefined time interval from the time that are generated (t_{gen}). This expiration time can vary and will be defined by each implementation of our system based on the specific needs of the underlying environment.

VIII. SECURITY ANALYSIS

In this section, we analyze the security of the proposed 'Red Alert Protocol' against several malicious behaviours. For the needs of this analysis, we assume that all the involved cryptographic schemes are semantically secure. Hence, our analysis is focused on looking at protocol's exchanged messages.

First, we consider the case where a corrupted user \mathcal{ADV} had been legitimately part of a treatment team in the past. This implies that at some point \mathcal{ADV} received a valid token τ to access the medical records of a legitimate user's u_l . \mathcal{ADV} may now try to alter τ to either access u_l 's data after the token expiration or use τ to access the data of another patient u_k ,

TABLE I
PROTOCOL MESSAGES

Index	Message
m_1	$\langle r_1, \text{Enc}_{\text{pk}_{\text{MA}}}(u_i, t, s_e), \sigma_{s_e}(H(r_1 u_i t s_e)) \rangle$
m_2	$\langle r_2, \tau_e, \text{Enc}_{\text{pk}_{s_e}}(\text{sk}_{\mathcal{A}_e, i}), \sigma_{\text{MA}}(H(r_2 \tau_e \text{sk}_{\mathcal{A}_e, i})) \rangle$
m_3	$\langle r_3, \text{Enc}_{\text{pk}_{\text{MA}}}(x, s_{x1}, s_{x2}), \sigma_{s_j}(H(r_3 x s_{x1} s_{x2})) \rangle$
m_4	$\langle r_4, \text{challenge}, \sigma_{\text{MA}}(H(r_4 \text{challenge})) \rangle$
m_5	$\langle r_5, \text{Enc}_{\text{pk}_{\text{MA}}}(v_0, l_x), \sigma_{\text{MA}}(H(r_5 v_0 s_{x1} s_{x2} l_x)) \rangle$
m_6	$\langle r_6, \tau_x, \sigma_{s_j}(H(r_6 \tau_x)) \rangle$
m_7	$\langle r_7, \text{Enc}_{\text{pk}_{s_j}}(c_P^j), \sigma_{\text{CSP}}(H(r_7 c_P^j)) \rangle$
m_8	$\langle r_8, \tau_x, \text{Enc}_{\text{pk}_{\text{CSP}}}(c_P^j), \sigma_{s_j}(H(r_8 \tau_x c_P^j)) \rangle$
m_9	$\langle r_9, \text{E}_{\text{pk}_{\text{MA}}}(t), \sigma_{s_h}(H(r_9 t)) \rangle$
m_{10}	$\langle r_{10}, \tau_x, \sigma_{\text{MA}}(H(r_{10} \tau_x)) \rangle$

$l \neq k$. To this end, \mathcal{ADV} needs to generate an altered token τ' with a new t'_{gen} , t'_{exp} and/or u_k , but she cannot alter the hash in the signature of MA. When \mathcal{ADV} sends m_6 or m_8 using τ' to access patient data, the CSP will drop the connection because the signature of MA will not match with the hash of altered token.

Second, we consider the case where \mathcal{ADV} tries to use a token τ issued to another legitimate healthcare professional s_l to add a false ciphertext to the patient's EMR. To capture τ , \mathcal{ADV} could overhear the communication between a s_l and MA to intercept m_2 , or between s_l and CSP to intercept m_6 or m_8 . However, the intercepted τ contains the identity of s_l , which cannot be changed because \mathcal{ADV} cannot generate a valid signature for the modified token, as described above. The only remaining alternative for \mathcal{ADV} is to use the τ directly. To do so, \mathcal{ADV} uses the CSP public key to encrypt a false ciphertext and generates m_8 . However, this message has to be signed by s_l , which cannot be achieved by \mathcal{ADV} because of the assumption that the signature scheme is unforgeable.

Third, we consider the case where the \mathcal{ADV} wants to access the EMR of a legitimate patient u_l . To do this, the \mathcal{ADV} needs to have an emergency key and a valid token, which requires her inclusion in the emergency session for u_l . This attack would need the collaboration of another corrupted user s_c who is already part of the session. More precisely, s_c invites \mathcal{ADV} to the session by sending m_3 to MA with the identity of \mathcal{ADV} and the others in the team. After that, MA sends m_4 with the challenge, which requires all users to authenticate the team from the same location. Thus, if at least one user denies the solution to the challenge, the attack will be prevented. As a result, the whole team will not receive access to the emergency key or to the token.

IX. DISCUSSION AND FINAL REMARKS

In this paper, we present our initial solution to address the break glass situation for accessing encrypted data to support emergency treatment. The proposed protocol, Red Alert, is based on CP-ABE and allows healthcare professionals from different teams to decrypt patient's medical records only during emergency treatment. As the main novelty, access to the medical record is implemented through an access token that is issued to a particular healthcare professional by a semi-trusted authority. The access tokens expire at the end of the emergency session, revoking access to the medical record after that. The security of the proposed protocol has been shown by analyzing several malicious behaviours.

The proposed protocol, however, has a few weak features. For example, while the use of CP-ABE offers great flexibility in terms of access management, the size of the produced ciphertexts and the time required to both encrypt and decrypt a file grows proportionally with the complexity of the underlying policy. Also, although Red Alert has approached the problem of granting and revoking data access for emergency treatment, the protocol was modelled for the stroke emergency procedure currently adopted in The Netherlands. Improvement of these limitations is the topic of our future research.

REFERENCES

- [1] D. P. Manca, "Do electronic medical records improve quality of care?: Yes;" *Canadian Family Physician*, vol. 61, no. 10, pp. 846–847, 2015.
- [2] J. L. Saver, "Time is brain-quantified," *Stroke*, vol. 37, no. 1, pp. 263–266, 2006.
- [3] A. Michalas, N. Paladi, and C. Gehrman, "Security aspects of e-health systems migration to the cloud," in *2014 IEEE 16th International Conference on e-Health Networking, Applications and Services (Healthcom)*. IEEE, 2014, pp. 212–218.
- [4] A. Abbas and S. U. Khan, "A review on the state-of-the-art privacy-preserving approaches in the e-health clouds," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 4, pp. 1431–1441, 2014.
- [5] D. Mashima and M. Ahamad, "Enhancing accountability of electronic health record usage via patient-centric monitoring," in *Proceedings of the 2nd ACM SIGHT International Health Informatics Symposium*. ACM, 2012, pp. 409–418.
- [6] R. Zhang and L. Liu, "Security models and requirements for healthcare application clouds," in *2010 IEEE 3rd International Conference on cloud Computing*. IEEE, 2010, pp. 268–275.
- [7] M. Li, S. Yu, K. Ren, and W. Lou, "Securing personal health records in cloud computing: Patient-centric and fine-grained data access control in multi-owner settings," in *International conference on security and privacy in communication systems*. Springer, 2010, pp. 89–106.
- [8] A. D. Brucker, H. Petritsch, and S. G. Weber, "Attribute-based encryption with break-glass," in *IFIP International Workshop on Information Security Theory and Practices*. Springer, 2010, pp. 237–244.
- [9] Y. Yang, X. Zheng, W. Guo, X. Liu, and V. Chang, "Privacy-preserving smart iot-based healthcare big data storage and self-adaptive access control system," *Information Sciences*, vol. 479, pp. 567–592, 2019.
- [10] D. Povey, "Optimistic security: a new access control paradigm," in *Proceedings of the 1999 workshop on New security paradigms*. ACM, 1999, pp. 40–45.
- [11] A. D. Brucker and H. Petritsch, "Extending access control models with break-glass," in *Proceedings of the 14th ACM Symposium on Access Control Models and Technologies*, ser. SACMAT '09. New York, NY, USA: ACM, 2009, pp. 197–206.
- [12] T. Zhang, S. S. Chow, and J. Sun, "Password-controlled encryption with accountable break-glass access," in *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016.
- [13] S. Marinovic, R. Craven, J. Ma, and N. Dulay, "Rumpole: a flexible break-glass access control model," in *Proceedings of the 16th ACM symposium on Access control models and technologies*. ACM, 2011.
- [14] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, ser. SP'07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 321–334.
- [15] A. Michalas, "Sharing in the rain: Secure and efficient data sharing for the cloud," in *2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*. IEEE, 2016.
- [16] N. Paladi, C. Gehrman, and A. Michalas, "Providing user security guarantees in public infrastructure clouds," *IEEE Transactions on Cloud Computing*, vol. 5, no. 3, pp. 405–419, July 2017.
- [17] A. Michalas, "The lord of the shares: Combining attribute-based encryption and searchable encryption for flexible data sharing," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, ser. SAC '19. New York, NY, USA: ACM, 2019, pp. 146–155. [Online]. Available: <http://doi.acm.org/10.1145/3297280.3297297>
- [18] D. Dolev and A. C. Yao, "On the security of public key protocols," *Information Theory, IEEE Transactions on*, vol. 29, no. 2, 1983.
- [19] A. Michalas, N. Komninos, and N. Prasad, "Multiplayer game for ddos attacks resilience in ad hoc networks," in *Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE), 2011 2nd International Conference on*, Feb 2011, pp. 1–5.