CRYPTOGRAPHIC APPROACH TO PATIENT RECORDS PRIVACY PROTECTION IN EMERGENCY SITUATIONS

Kosmas Kapis, Jan C.A. van der Lubbe, Kathy Cartrysse
{K.Kapis; J.C.A.vanderLubbe; K.Cartrysse}@ewi.tudelft.nl
DELFT UNIVERSITY OF TECHNOLOGY

This paper presents cryptographic mechanisms for securing electronic medical patient records, EPRs in emergency situations. The mechanism maintains the EPR confidentiality during an emergency when a patient is unconscious and located outside any health provider, HP. In such situations any authorized health care practitioner, HCP can access all relevant medical records when needed. However, after such an emergency, it is impossible for anyone to access the patient's EPR without the patient's consent. This can be achieved by using a modified forward-secure signature and encryption algorithm used in two directions. After an emergency treatment, the patient's consent is restored.

INTRODUCTION

An Electronic Patient Record, EPR is a collection of electronic medical data for the same treatment or by the same medical practitioner. The increase of large networks for access to medical records has raised concerns about security, confidentiality and privacy. Currently, medical doctors control the patient's medical data. Therefore a patient has no privacy in the sense that doctors have access to patient's records at any time. Privacy is only protected by means of policies and practice statements (medical ethics). One of the solution to provide privacy is for the patient to have control over his medical data [1,2,3] such that access to medical data is only possible with the patient's consent. The few studies done on the combination of EPR and patient privacy focus on a conscious patient physically able to grant the medical practitioner access to his records. Yet, the patient privacy and confidentiality problems have remained partially unsolved [5].
It is a nice idea for patients to control their medical records, but it becomes more risky during emergency situations, because under such circumstances the the means used to access the EPR must be exposed. To get the insight of the problem, assume that an unconscious person is located at a large distance from the nearest health provider while in need of medical treatment and an emergency

Health Care Provider, HCP (e.g. ambulance personnel) is present after an emergency call of a bystander. To treat this person efficiently, the HCP needs medical data of this person. However, the network in which the EPRs are stored must be available at this location. Furthermore, the HCP must somehow be able to retrieve the EPRs in order to read them without the patient's consent. The HCP must again be restricted to access the EPRs after the patient is conscious and physically able to communicate.
From the scenario above the following risks are anticipated: a) The HCP may access the the wrong EPR; b)The EPR may not be available at a given time ; c)It may take unacceptable delay to access the EPR; d)The means to access the EPR may not be available at that moment; and e)The EPR access means must be exposed. In this paper the concentration is on risks a), c), d),and e). This paper provides solutions to risks especially present in emergency scenario by applying cryptographic techniques.

CRYPTOGRAPHIC BUILDING BLOCKS

The general idea behind the proposed scheme is that the patient remains himself the part-owner of and jointly responsible for his own EPR. This does justice to the privacy rights and nowadays emancipation of the patient. Encryption techniques are used for providing confidentiality. By using both digital signatures and encryption techniques it is also possible to provide privacy. In principle, this works when a patient is able to communicate. A patient's digital signature is applied to guarantee the integrity and authenticity of the EPR. In an emergency event all access means have to be temporarily bypassed and restored afterwards. Two schemes are used. The modified forward secure (FS) encryption solution is used by a patient when he is conscious and physically able to communicate with the HCP. It is used to generate and access medical records. The master key solution is used by the HCP to access the patient's EPR in an emergency event. In both solutions the forward signature algorithm is used.
The principle of a forward signature algorithm is that several messages can be signed with a signing key $KSIG_j$ that eventually will be revoked and replaced with $KSIG_{j+1}$. The verifier is still able to verify the messages signed with $KSIG_j$ for $1 \leq i \leq j$ using verifying key $KVER$. In case of exposure of $KSIG_j$ to third parties, only an update of $KSIG_j$ is needed.
This scheme uses the scheme originally proposed by Itkis and Reyzin ,(IR) [4]. The IR algorithm has no encryption function. While forward-secure, (FS) encryp-

tion algorithms can be obtained from different literature sources, in this proposal; the IR algorithm is modified such that an encryption function is integrated. The modified IR scheme is chosen based on the fact that a symmetric encryption is needed for non-emergency situation where oblivious access of EPRs also must be possible. This modification enables incorporation of a symmetrical algorithm like AES in the forward-secure system. The details of the two solutions are given in the following section.

Procedures of FS-encryption scheme

One of the capabilities the scheme must have, is the protection of patient's confidentiality. This is provided by encryption. The patient has to own the encryption key in order to control the access to his records. The patient's EPRs integrity and authentication need also to be assured by the scheme, hence patient signature is needed. A signature also provides patient's consent. Due to the earlier stated risks of encryption and signature keys exposure, the scheme must be forward secure.

Apart from the signature key present in the IR algorithm [4], an encryption key need to be included. A patient identification key, KID is therefore introduced. The IR algorithm based on Strong RSA assumption is used to obtain KID in the FS-encryption-scheme. Strong RSA assumption postulates that, it is intractable, given n and a value $\alpha \in \mathbb{Z}_n^*$, to find $\beta \in \mathbb{Z}_n^*$ and $r > 1$ such that $\beta^r \equiv \alpha \pmod{n}$ in which n is a product of two 'strong' primes and r is a random number $[r \overset{\mathrm{R}}{\leftarrow} \mathbb{Z}_n^*]$. In the IR scheme as well as this scheme, the maximum number of key updates T, is defined a priori to determine the validity duration of a given $KVER$. From IR scheme, the following expressions are co-opted to simplify the scheme; $f_i = (e_i e_{i+1} \cdots e_T) \; mod \varphi(n)$ represents a parameter in a signing factor equation; while $\varrho_e = \{e_1, e_2, \cdots, e_T\}$ is a set of all e. Where all e_i are random numbers relative prime to each other and $\varphi(n)$, representing assigning positions of signature validity periods. This scheme uses the IR expression $y_j \leftarrow r^{e_j}(modn)$ to represent the key KID_j. Where j is an instant in a given period T.

Let M be the message that is to be signed and encrypted, KID_j be the j^{th} subkey for AES encryption, k, be a security parameter, and l be a security parameter that influences the sizes of e_i for each i. Now the protocol is as follows;

(1a) ***Keys generation procedure***: Input: T, l, k. Output: $KSIG1$, $KVER$.

1) Generate random $\lceil((k/2) - 1)\rceil$-bit primes q_1, q_2. Also generate two strong

(safe) primes p_1 and p_2 for $i = 1, 2$: $p_i \xleftarrow{\mathrm{R}} \mathbb{Z}_n^*$ primes s.t. $p_i = 2q_i + 1$ are primes.
2) Determine n by multiplying the strong primes. $n \leftarrow p_1 \cdot p_2 \Rightarrow \varphi(n) = 4q_1 \cdot q_2$.
3) Generate t_1, a random number mod$n$. $t_1 \xleftarrow{\mathrm{R}} \mathbb{Z}_n^*$.
4) Generate primes e_i s.t. $2^l(1 + \frac{i-1}{T}) \leq e_i < 2^l(1 + \frac{i}{T})$ for $i = 1, 2, , T$.
5) Multiply all e_i for $i \geq 2$ and assign it f_2 for determining signing factor s_1.
$f_2 \leftarrow \prod_{i=2}^{T} e_i \bmod \varphi(n)$.
6) Determine signing factor s_1: The f_2^{th} power of t_1. $s_1 \leftarrow t_1^{f_2} \bmod n$.
7) Determine the verifying factor v out of s_1 and e_1. $v \leftarrow s_1^{-e_1} \bmod n$.
8) Replace t_1 for t_2 such that s_1 cannot be derived anymore.$t_2 \leftarrow t_1^{e_1} \bmod n$.
9) Determine and return the signing key $KSIG_1$. $KSIG_1 \leftarrow (1, T, n, s_1, t_2, \varrho_e)$.
10) Determine and return the verifying key $KVER$. $KVER \leftarrow (n, v, T)$.
(1b) ***Signature and encryption procedure***
Input: $KSIG_j$, $M = ID_p$. Output: (C_j, z, σ, j, e_j).
1)Define $KSIG_j$. Let $KSIG_j = (j, T, n, s_j, t_{j+1}, \varrho_e)$.
2) Generate a random number r mod n. $r \xleftarrow{\mathrm{R}} Z_n^*$.
3) Determine the AES key KID_j. $KID_j \leftarrow r_e^j(\mathrm{mod}n)$.
4) AES encrypts ID_p. $C_j = \mathbf{e}KID_j(ID_p)$. Where $\mathbf{e}$ is encryption using key KID_j and message ID_p.
5) Let σ be the hash of ID_p, contrary to the original IR scheme. $\sigma \leftarrow h(ID_p)$ This is done for ID_p verification purposes.
6) Determine z, with which ID_p can be verified. $z \leftarrow rs_j^{\sigma} \bmod n$.
(1c) ***Verification and decryption procedure*** Input: $KVER, Cj, z, \sigma, j, e_j$. Output: ID_p or false.
1) Define $KVER$. Let $KVER = (n, v, T)$.
2) Check if the correct e_j is received. If $e_j \notin \left\langle 2^l\ , 2^l(1 + \frac{j}{T})\right]$ or e_j is not prime then return false.
3) Check if z has a proper value. If $z \equiv 0\ (\mathrm{mod}n)$ then return false.
4) Determine what KID_j would have been. $KID_j^* \leftarrow z^{e_j} v^{\sigma} \bmod n$.
5) Determine what ID_p would have been. $ID_p^* = dKID_j^*(C_j)$.
6) Determine if ID_p is correct. If $\sigma = h(ID_p^*)$ then return ID_p else return false.
(1d)***Update procedure***.
Input: $KSIG_j$. Output: $KSIG_{j+1}$ or error.
1) Define $KSIG_j$. Let $KSIG_j = (j, T, n, s_j, t_{j+1}, \varrho_e)$.
2) Stop if j has reached its maximum. If $j = T$ then return error.
3) Replace s_j. $s_{j+1} \leftarrow t_{j+1}^{e_{j+2} \cdot \ldots \cdot e_T} \bmod n$.

4) Replace t. Note that s_j cannot be derived anymore. $t_{j+2} \leftarrow t_{j+1}^{e_{j+1}} \bmod n$.
5) Determine and replace the new signing key. $KSIG_{j+1}$=$(j{+}1, T, n, s_{j+1}, t_{j+2}, \varrho_e)$.
This scheme works when a patient is conscious and physically able to communicate. The patient identification number, PIN, gives access to $KSIG_j$ which can be used to access the EPR. He then can sign and encrypt using his $KSIG_j$ on the records entered by HCP such that a consent is given. For emergency situations when a patient cannot communicate the following solution is proposed.

Master key scheme

In an emergency situation when a patient cannot communicate, the EPR cannot be accessed using the FS-encryption scheme. A master key, KMASTER to circumvent the patient's EPR access PIN, is therefore needed. In the course of the event, the patient's privacy must be exposed, and has to be restored after a patient gains consciousness. Therefore, the KMASTER must be updatable, such that the same KMASTER cannot be used to access the EPR after the patient becomes physically and conscious able to communicate. The same FS-encryption scheme procedures are used to derive the master key scheme. The difference is in some key designations and parameters' values. A signature and encryption key called generation key KGEN, is also introduced in the scheme. The KGEN generates the EPR encryption keys, $KEPR_{\tilde{j}}$ which are AES keys and the signature. Using the KMASTER it is possible to derive the EPR decryption keys, $KEPR_{\tilde{j}}$ from each $\tilde{C}_{\tilde{j}}$. Where $\tilde{C}_{\tilde{j}}$ is the encrypted patient medical records. Many patients can encrypt/decrypt their messages using $KEPR_j$. The $KEPR_{\tilde{j}}$ is updated every after usage. The scheme assures that all $\tilde{C}_{\tilde{j}}$ in the patient's EPR can be decrypted independently because each EPR has its encryption key. Even when all or some of $KEPR_{\tilde{j}}$ are lost, $KMASTER$ can determine all $KEPR_{\tilde{j}}$, making it possible to decrypt and verify the EPR data. Procedures of the master key scheme are as follows;
(2a) ***Keys generation***: On input parameters $\tilde{k}, \tilde{l}, \tilde{T}$ this procedure outputs the signature and encryption key $KGEN_1$ and the master key, $KMASTER$. The scheme assures that all $\tilde{C}_{\tilde{j}}$ in the patient's EPR can be decrypted independently because each EPR has its encryption key. Even when all or some of $KEPR_{\tilde{j}}$ are lost, $KMASTER$ can determine all $KEPR_{\tilde{j}}$, making it possible to decrypt and verify the message cum EPR data. This procedure is exactly the same as procedure (1a) except for $KSIG_1$ which becomes $KGEN_1$ and $KVER$ becomes

$KMASTER$.
(2b) ***Signature and encryption procedure***: This procedure takes as input $KGEN_{\tilde{j}}, \tilde{M}$ for $\tilde{j} < \tilde{T}$ and the message $\tilde{M}$ to be signed, and returns the signature and encrypted message $\tilde{C}_{\tilde{j}}, \tilde{z}, \tilde{\sigma}, \tilde{j}, \tilde{e}_{\tilde{j}}$ for time period $\tilde{j}$. Where $\tilde{M} = D_{EPR}$ and $\tilde{\sigma} = \mathbf{h}(D_{EPR})$. This procedure is obtained from procedure (1b) by replacing $KSIG_j$ with $KGEN_{\tilde{j}}$, and KID_j with $KEPR_{\tilde{j}}$.
(2c) ***Verification and decryption procedure:*** The $KMASTER$, and the signed-encrypted data $\tilde{C}_{\tilde{j}}, \tilde{z}, \tilde{\sigma}, \tilde{j}, \tilde{e}_{\tilde{j}}$ are the inputs returning at the output true, D_{EPR} or false. The procedure equals procedure (1c) if $KVER$ is replaced with $KMASTER$.
(2d) ***Update procedure***: The algorithm takes $KGEN_{\tilde{j}}$ as input for the current period $\tilde{j} < \tilde{T}$ and returns a new signature and encryption key $KGEN_{\tilde{j}+1}$ for the next period $\tilde{j} + 1$. Replacing $KSIG_j$ with $KGEN_{\tilde{j}}$ in (1d) this procedure can be obtained. In this scheme no one, except the patient, can access both $KGEN_{\tilde{j}}$ and $KMASTER$ at the same moment. However, parties having $KGEN_1$ can derive $KMASTER$ according to procedure (2a) in which $\tilde{v}$ is derived from $\tilde{s}_1$ and $\tilde{e}_1$. This is solved by running procedure (2d) once, such that $\tilde{j} = 2$ (or at least $\tilde{j} > 1$) and $\tilde{s}_1$ cannot be derived anymore from $KGEN_{\tilde{j}}$ thus $\tilde{v}$ cannot be derived. Using the smartcard reader and the patient's smartcard the emergency HCP on duty can login the respective mobile network and access the EPR. The following section gives general picture of how the emergency EPR scheme functions.

EMERGENCY EPR SCHEME

Emergency health care services can be needed anywhere and anytime. This is because accidents and situations that need urgent medical attention are independent of time and location. Therefore the need for a mobile environment is inevitable.

For the HCP to be able to access the patient's EPR from any location in a given area, several assumptions have to be made. Four parties, namely the patient, HCP, EPR location, Health Provider, HP (e.g.hospital), and the *Trusted Party*, TP participate. Several smart gateways are also used to facilitate mobility. A smartcard is provided to every patient by the TP as the patient's means of identification and access to EPRs. The TP is a party that is also by law obliged to take an oath of secrecy. It also monitors the mobile gateway G_{MOBILE}. The G_{MOBILE} role is to locate the HP containing EPRs. The total topology of the proposed system with all parties involved is shown in figure 1. For the scheme to

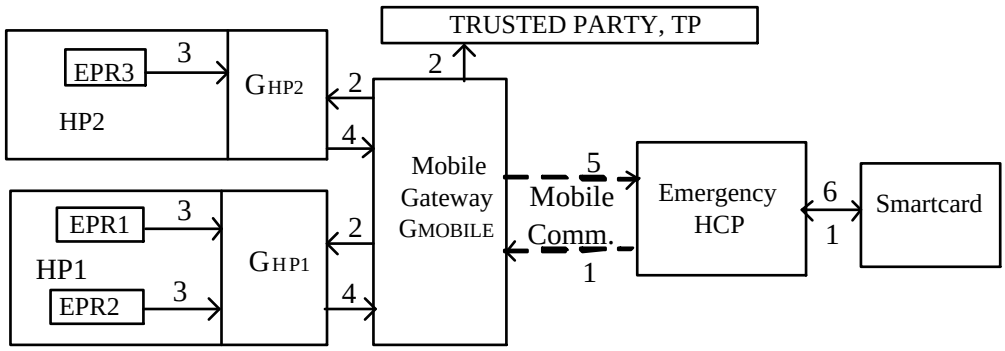

Figure 1: Topology of parties involved during an emergency EPR access at the network level. Numbers represent the sequence of events.

work, it requires EPR initialization, creation, emergency access, and key update steps. In this paper only the EPR emergency access is described.

It is assumed that the patient's smartcard is available, but not the PIN code. The HCP uses the smartcard reader and patient's smartcard to access the EPR. The HCP will send to G_{MOBILE} the designation number of the smartcard N_{CARD} and patient's identification encrypted with the signature key, $\mathbf{e}KSIG_j(ID_P)$. The mobile gateway determines the correct $KVER$ from N_{CARD} and then runs the procedure (1c) with inputs $KVER$ and $\mathbf{e}KSIG_j(ID_P)$ obtaining the output ID_P*. Using the ID_P in its storage, the G_{MOBILE} checks whether ID_P*=ID_P. In this case G_{MOBILE} knows all HPs that have at least one relevant EPR of the patient. Therefore, the next is for the G_{MOBILE} to send all those G_{HP} a request for emergency access for ID_P. The TP gets notified of the event. All HPs contain $\sigma = h(ID_p)$. In this emergency situation the patient's consent provided by the FS-encryption scheme is exposed. Having located the correct EPR, every G_{MOBILE} copies all encrypted EPR data, $eKEPR_{\tilde{j}}(D_{EPR})$. It then sends the copy back to G_{MOBILE} with IR parameters $\tilde{z}, \tilde{\sigma}, \tilde{j}, \tilde{e}_{\tilde{j}}$. The G_{MOBILE} then sends all the received data and K_{MOBILE} to the HCP, where, K_{MOBILE} is a key used for accessing the EPRs decryption key, the KMASTER. Using the received K^*_{MOBILE} the master key can be deciphered; i.e. $\mathbf{d}KMOBILE^*(eKMOBILE(KMASTER))$ $= KMASTER^*$ and verified whether $\mathbf{h}(KMASTER^*) = \mathbf{h}(KMASTER)$. The HCP now can retrieve a copy of EPR by executing procedure (2c) for all encrypted EPR medical data, $\tilde{z}$, and $\tilde{\sigma}$, with inputs; $KMASTER, \mathbf{e}KEPR_{\tilde{j}}(D_{EPR}), \tilde{z}, \tilde{\sigma}, \tilde{j}, \tilde{e}_{\tilde{j}}$. This copy is supposed to be deleted after the emergency treatment of a patient. The EPR keys should be updated as soon as the patient is conscious and physi-

cally able to give his consent.

CONCLUSION AND DISCUSSION

This paper has introduced a modified forward-signature and an encryption solution for accessing EPRs. Two schemes have been used for emergency EPR access. The FS-encryption scheme has been used for the encryption of patient's identification, access request of emergency EPR, and the provision of patient's consent. The master key scheme has been used for the EPR signature and decryption. Patient's confidentiality is guaranteed by instantly updating all signature and encryption keys after every usage. The medical data integrity is also assured in the scheme. Timely availability is provided by the IR algorithm, which verifies optimally [4].

Changes introduced in the original IR scheme include among others a message integrity parameter and the AES encryption key. These introduced changes do not weaken the strength of the scheme in [4], because the signature key strength mainly depends on its continuous forward update of keys for every usage. The encryption key is also forward updated but can be accessed by a master key in emergency situations by authorized persons. Since the public key is also used for decryption it should be kept secret. In case the private key is compromised, the public key is protected by the factorization and discrete logarithm problem.

REFERENCES

[1] The Joint Computing Group of the General Practitioners' Committee: *Good Practice for General Practice Electronic Patient Records*, Ver. 2.6, UK. 2000.

[2] Health Canada, International Activities toward Electronic Health Records: *Unique Identification and PKI*, Ottawa, 2001.

[3] P. Charnley: *Security and Confidentiality Requirements for Cancer EHR*, Version 0.1, Wirral NHS Natural Community, 2001.

[4] Gene Itkis, Leonid Reyzin: *Forward-Secure Signatures with Optimal Signing and Verifying*, In: J. Kilian (ed.), Advances in Cryptology, LNCS, Vol. 2139, Springer-Verlag, Berlin, Germany, 2001, p. 332-354.

[5] G. Ateniese, R. Cutmola, B. de Meideiros and D. Davis: *Medical Information Privacy Assurance: Cryptographic and System Aspects*, In: Third Conference on Security in Communication Networks 2002, (SCN2002).