Anonymity Preserving Authorization Granting In Medical Information Networks

Sigurd Eskeland and Vladimir Oleshchuk
Agder University College
Grooseveien 36
N-4876 Grimstad, Norway

Abstract: Due to the sensitivity of personal medical information, this paper addresses the need of hiding patient identities — in contrast to only keeping their medical data confidential. Thus, it is desirable that personal and meaningful patient identity information like names, addresses, personal identity numbers, etc., are not to be linked to disclosed electronic patient records (EPR). To achieve this, we propose a scheme that enables patients to anonymously grant medical teams authorization to access their EPRs without revealing their identities to the teams providing medical care. An essential benefit is that it enables patients to exert control over their own medical data. A security evaluation is included.

1 Introduction

With the emerge of information technology in health care, there has been extensive focus on the security issues of electronic patient records (EPR) in medical environments. These issues include how to ensure that *only* legitimate personnel can access no more than the required electronic patient records in order to provide medical care to the concerning patients, and moreover, how to ensure that medical information is preserved and managed confidentially.

Although medical patient data remain confidential, it may be cases when it is desirable that the identities of patients remain confidential as well, even after disclosure of patient data. Concerning personal information about AIDS/HIV status, sexual transmittable diseases, emotional problems, psychiatric illnesses, sexual divergencies, genetic predispositions to diseases, information about toxic addictions, and so on [Rin97], it is likely that some patients wish to remain anonymous. Thus, meaningful identity information such as names, birthdays, personal identity numbers, addresses etc., must not be linked to corresponding disclosed EPRs. Likewise, for purposes such as medical research, disclosed medical data should not be linked to the concerning patients. Access control should nevertheless be maintained properly so that EPRs are not accessible to other medical employees than legitimate medical teams providing medical care to the concerning patients.

The problem boils down to how to link patients with their respective EPR without revealing personal identity information. In this paper, we propose a cryptographic solution that enables patients to grant medical teams authorization to access their corresponding EPRs

in such a way that the real identity of the patients are not disclosed. Thus, patients remain anonymous, and patients, teams and EPR security server can nevertheless authenticate each other.

Patients granting authorization implicitly state consent and exert control over their own medical data by controlling who can access their corresponding EPRs. Moreover, by removing personal identity information from EPRs, privacy is preserved when medical data is disclosed for medical research.

In the case that a patient is unconscious and therefore unable to consent access to the EPR, the security system can include an emergency mode where one, or preferably a minimum coalition of two (or three) security administrators, approve a medical team access on behalf of the concerning patient.

2 Previous work

It has been previously proposed to remove personal identity information (names, addresses, phone numbers, etc.) to provide anonymity in medical environments [TER04]. However, this may not rule out the possibility that it may be possible to correlate anonymous medical data to the corresponding individuals. Sweeney [Swe97] proposes methods of substitution of data to prohibit such correlations. In order to enforce access control, explicit identifiers are required to link data records. It is assumed in this paper that the medical data themselves contain no explicit references to the patient.

In [KDTC04b, KDTC04a], the authors propose anonymization of patient data based on encrypted anonymous identifiers or pseudonyms. Patients can consent to disclose their medical data by supplying their pseudonyms, but have, however, no technological enforcement about who is to access the medical data. Another problem is the staticness of the pseudonyms, and that there is no challenge-response mechanisms ensuring online certification of requests. Thus, an adversary can obtain an encrypted anonymous identifier by eavesdropping. By replaying this, the security server will not be able to distinguish whether a request originated from the patient or not. Moreover, there is no mechanism that ensures the security server that no other than legitimate medical professionals, and not an adversary, is targeted to access the EPR.

3 Anonymous authentication and authorization

3.1 A framework for patient anonymity

In medical networks, access to EPRs includes known identities of patients. This could be meaningful public identifiers like names and/or personal security numbers. Such identifiers would then be known to the parties involved in the authentication and authorization process. Consequently, the medical personnel that provides medical care depends on

Entity	*Identifier*
STA	$TAPI_i$, SSN_i
S	$TAPI_i$, $AEID_i$
P_i	$TAPI_i$, SSN_i
T	$TAPI_i$

Figure 1: Relationship between identifiers known by which entities

knowing such identities in order to provide the right care to the right patients.

In circumstances where patients require confidentiality and want to remain anonymous, consequently, it is essential that names and social security numbers (SSN) cannot be associated with their corresponding EPRs. This means that it is not desirable to have fully trusted administrators who can obtain such relations. Moreover, the EPR server must not link (or contain a table that links) patient names/SSNs with corresponding EPRs. In contrary, in this paper we propose an approach where the EPR server associates each EPR entry with a specially generated anonymous EPR identifier (AEID). The AEID is partly based on a secret long-term key held by the corresponding patient. This secrecy of this key prohibits any other party including administrative personnel from obtaining associations between patients and their EPRs. Basically, given a patient name or SSN, it must be infeasible to obtain the corresponding EPR entry and the EPR of the patient without knowing the key.

A semi-trusted administrator (STA) may be needed to assign available and appropriate medical professionals for treatment of incoming patients. It may be necessary that the STA knows names and SSN of hospitalized patients, but there is no need for this party to access EPRs. In our approach, the STA, being an administrative and coordinating entity having knowledge of names and SSN of hospitalized patients, has not a role and authority to assign authorizations on behalf of patients for medical professionals access their EPRs, nor to be able to associate names/SSNs of patients with their medical records.

The EPR server contains a table that links the anonymous EPR identifiers (AEID) and the associated EPR entries of each patient. This table does not provide any relationships between patient names and AEIDs. Furthermore, none of the involved parties, including patients, know or can obtain the association between an AEID and actual patient identities.

To ensure patient anonymity, medical personnel is not given any name/SSN information of patients, but patients are instead referenced by means of a temporary anonymous patient identifier (TAPI). The duration of the hospitalization reflects the lifetime of a TAPI, and a new unique TAPI must be established for each hospitalization.

The correspondence between TAPI and AEID can only be obtained by the EPR server. Thus, if STA has obtained access to the AEID/EPR entry table where each record is referenced by AEID, the STA has no way to determine which record that is associated by any TAPI. Fig. 1 shows which identifiers known by the involved entities, and the association of the identifiers. The entity STA denotes a semi-trusted administrator, *S* denotes the EPR

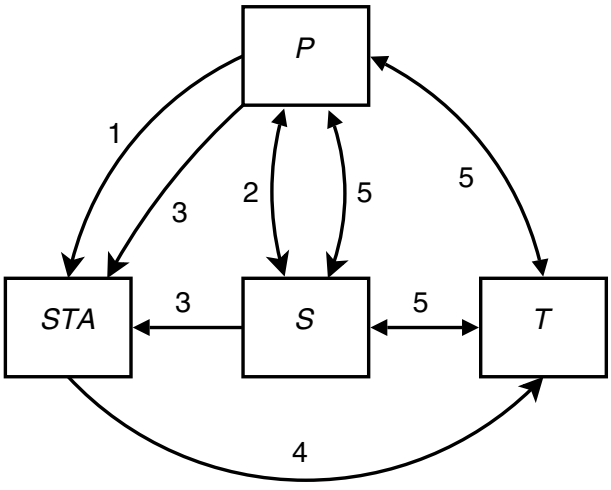

Figure 2: The hospitalization process

server, P_i denotes a patient and T a medical team.

The hospitalization process of a patient is summarized in Fig. 2 and as follows:

1. First at hospitalization, the patient may identify himself by name and SSN to the STA for administrative purposes like billing.
2. The patient anonymously certifies himself the EPR server by means of his or her secret key, and generates TAPI.
3. The patient supplies the value of TAPI to STA. The EPR server acknowledges by supplying TAPI to STA. The STA compares the to received values – match indicates that the patient is legitimate.
4. STA assigns a medical team for the patient referenced by TAPI.
5. The team, EPR server and patient (by means of TAPI) authenticates each other. By completing the authentication, the patient is granting the medical team authorization to access his EPR.

An important property here is that the patient's involvement is required in order to obtain access to his or her EPR. Since only the corresponding patient holds the private key, this provides the patient the authority to grant arbitrary teams of medical professions access to his EPR.

3.2 Security properties and requirements

In the problem setting considered in this paper, there are three active entities: The medical team T, the granting patient P, and the EPR security server S. We have the following security requirements:

Authentication: The involved parties must mutually authenticate each other.

Authorization: Only P can authorize T to access his or her EPR.

Anonymity: The identity of P must be hidden from T and S.

Unlinkability: It is infeasible to deduce that various TAPIs may refer to the same P.

The first security requirement embraces the following aspects:

- T and S must authenticate themselves to P, preventing that an adversary masquerading as T or S could illegitimately obtain authorization
- P anonymously authenticates himself to S to prohibit that an adversary may successfully masquerade as P. Likewise, T must authenticate themselves to S.
- S must authenticate itself to T and thereby confirming whether P is a valid patient and whether T is granted authorization. Recall that the real identity of P is hidden to T, and that P is referenced by TAPI.

Successful and completed execution of the proposed anonymity-preserving authentication and authorization protocol provides granted authorization. Thus, the patient must authenticate the medical team and security server first, since the patient is the granting entity. Subsequently, the EPR server authenticates the patient and team.

The third security requirement means that the given patient is not referenced by his name/SSN, but rather a temporary TAPI. It should be infeasible to obtain associations between any patients and their TAPIs. This is related to the fourth which is to ensure unlinkability between TAPI and AEID of the same patient for all other than patients and EPR server. It is likewise infeasible to obtain relationships between any names/SSNs and AEIDs.

Assuming that the network is not secure, it can be assumed that all messages exchanged over the network can be eavesdropped by an adversary. Thus, the protocol must resist passive attacks like eavesdropping, and active attacks by manipulation and substitution of messages where the adversary may be masquerading as a legitimate entity.

Public keys should be certifiable, for example represented by digital certificates or be identity-based so that substitution attacks of public keys will be detected. Successful replacement of public keys would consequently break the security of the protocol (and any other protocol).

Since all entities hold private keys that are actively involved in the authentication and authorization process, these should be held and stored securely, for example in secure

tamper-free devices. Computations should be done in these devices so that the personal keys are never disclosed. Personal smartcards are a possible solution to satisfy these requirements. However, this part is outside the scope of this paper.

4 The anonymity preserving authorization protocol

The objective of this protocol is to enable patients to anonymously grant medical teams authorization to access their EPRs without revealing identities. The protocol is threefold. Initially, patients are anonymously registered at the current hospital where an EPR is created for each patient associated with a unique anonymous EPR identifier (AEID) (Fig. 3).

For each subsequent hospitalization, each patient establishes a valid temporary anonymous identifier (TAPI) (Fig. 4) by which the patient can anonymously grant medical personnel EPR access according to the anonymity-preserving authentication and authorization (AAA) protocol (Fig. 5).

Intuitively, a naive way for P_i to anonymously convince *S* about ownership of a certain EPR is by using some kind of reference that anonymously link to patient's EPRs. Let aid_i denote such reference. Assuming that such a reference aid_i is known only to the patient P_i and *S*, P_i could use it to authenticate himself or herself to *S*.

P_i could send aid_i encrypted to *S* along with a timestamp. *S* could then decrypt, check the timestamp, and authenticate the validity of aid_i by checking if there exists an EPR that is referenced by aid_i. The problem about such authentication is two-fold: 1) the EPR table referencing the aid_i's and the corresponding EPRs must be kept secret. 2) aid_i is equivalent to a long-term secret user key, whereas secret keys should never be disclosed or leave the owner. An adversary getting hold of aid_i could obtain the same by encrypting aid_i along with a valid timestamp. Thus, this approach would not be appropriate. In the protocols of Fig. 3 and Fig. 4, anonymous authentication is achieved by exchanging blinded messages without submitting identifying information directly. Blinding refers to hiding data usually by multiplication of a secret secret number. This allows subsequent operations like exponential operations to be performed on the blinded data by another party while effectively hiding the original data.

In the rest of this paper, we assume that all cryptographic computations are in a finite field $\mathbb{Z}_p$ determined by a large prime *p* where α is a generator to *p*. Both *p* and α are public.

4.1 Initialization

The initialization protocol (Fig. 3) is utilized when a patient P_i for the first time is registered at the hospital and his or her EPR is created. In the protocol, $AEID_i$ is blindly established jointly by P_i and *S*, and being associated to the his or her new EPR.

In the first step, the patient randomly generates a private long-term key x_i, which should

1)	$S \rightarrow P_i:$	$r_S = (\alpha \cdot N)^{x_S}, N$
2)	$P_i \rightarrow S:$	$E_S(r_S^{x_i}, N^{x_i}, N)$
	$S:$	$AEID_i = r_S^{x_i} \cdot (N^{x_i})^{-x_S} = \alpha^{x_S x_i}$

Figure 3: The $AEID_i$ initialization protocol for P_i

preferably be stored in tamper-proof hardware like a personal smartcard, and cannot be disclosed. Alternatively, it could be $x_i = h(P_i'\text{s password})$ where h denotes a secure hash function. The value α^{x_i} is essential for long-term anonymous identification of P_i, because the anonymous $AEID_i$ is computed by S according to $AEID_i = (\alpha^{x_i})^{x_S}$ where x_S is the private key of *S*. In this protocol, the disclosure of α^{x_i} is insignificant.

Initially, S generates a nonce N. Then S computes and submits $r_S = (\alpha \cdot N)^{x_S}$ and N to the patient P_i. Based on this, P_i computes $r_S^{x_i}$ and N^{x_i}, and returns $E_S(r_S^{x_i}, N^{x_i}, N)$. (The notation $E_X(m)$ denotes that a message m is encrypted with the public key of entity X.)

S receives the message from P_i, decrypts it and obtains $r_S^{x_i}$ and N^{x_i} and N. S verifies whether the nonce N matches with what was sent to P_i in the previous message. In case of no match, S aborts initialization. Otherwise, S computes

$$\begin{aligned} AEID_i &= r_S^{x_i} \cdot (N^{x_i})^{-x_S} = (\alpha \cdot N)^{x_S x_i} \cdot (N^{x_i})^{-x_S} \\ &= \alpha^{x_S x_i} \cdot N^{x_S x_i} \cdot N^{-x_i x_S} = \alpha^{x_S x_i} \end{aligned}$$

It is computationally infeasible to for S to obtain x_i due to the Discrete Logarithm Problem. Then S creates an empty EPR for P_i, and creates a new row in the AEID/EPR entry table linking the anonymous $AEID_i$ and the new EPR.

The difference of proposed initialization protocol from the Diffie-Hellman key agreement protocol [DH76] lyes in the fact that $AEID_i$ acts as an identifier, while the purpose of Diffie-Hellman is for two parties to establish secret keys that are not to be known by other than the two parties. Although that the table of $AEID_i$ should be kept secret at the EPR server, only knowledge of the private key x_i can provide EPR authorization, knowledge of the respective $AEID_i$ cannot.

4.2 TAPI establishment

When patients are getting hospitalized, a semi-trusted administrator (STA) provides related administrative tasks and coordination by assigning available and appropriate medical personnel to provide care to the patient. The patient may (or may not) identify themselves with names and SSN to the STA for administrational purposes like billing, but the STA will be unable to link any names to the records of the patients.

The protocol for establishing an temporary anonymous patient identifier (TAPI) (Fig. 4) is similar to the one described in the previous subsection, except that in this one, the patient

1)	$S \rightarrow P_i:$	$r_S = \alpha^{x_S} \cdot N^{x_S}, N$
	$P_i:$	$TAPI_i = \alpha^{p_i}$
2)	$P_i \rightarrow S:$	$E_S(r_S^{x_i}, N^{x_i}, N, TAPI_i)$
	$S:$	$AEID_i = r_S^{x_i} \cdot (N_1^{x_i})^{-x_S}$
3)	$S \rightarrow P_i:$	$TAPI_i$

Figure 4: The TAPI establishment protocol

P_i is anonymously authenticated towards $AEID_i$, and that the patient provides $TAPI_i$ by which the patient will be referenced throughout the current hospitalization.

Initially, S generates a nonce N, computes $r_S = (\alpha \cdot N)^{x_S}$, and sends r_S and N to P_i. P_i computes $r_S^{x_i}$ and N^{x_i}. P_i generates a large secret random number p_i, and computes $TAPI_i = \alpha^{p_i}$. Then P_i encrypts $r_S^{x_i}$, N^{x_i}, N, $TAPI_i$ with the public key of S.

The nonce N ensures S that message 2 is not a replay of a previous session, and blinds α^{x_S} (due to the Discrete Logarithm Problem). S decrypts the message, verifies correctness of the received nonce N, and computes

$$\begin{aligned} AEID_i = r_S^{x_i} \cdot (N^{x_i})^{-x_S} &= (\alpha \cdot N)^{x_S x_i} \cdot (N^{x_i})^{-x_S} \\ &= \alpha^{x_S x_i} \cdot N^{x_S x_i} \cdot N^{-x_i x_S} = \alpha^{x_S x_i} \end{aligned}$$

Then S checks in the AEID/EPR table whether there is an entry that matches $AEID_i$. If such entry exists, then S sends back $TAPI_i$ as an acknowledgment to STA and P_i to confirm that a record exists according to the request.

$TAPI_i$ is an anonymous identifier that identifies the patient P_i during the hospitalization where its validity is approved by S. However, in the proposed approach, it is not only for referencing the patient anonymously, but also to function as an anonymous ephemeral public key. During subsequent execution of the AAA protocol, the patient is authenticated by means of $TAPI_i$ (which is already approved by S), acting as a anonymous ephemeral public key where p_i is the corresponding private key. Correspondingly, both S and T are associated with respective public keys, although not anonymous.

For subsequent execution of the AAA protocol, the EPR server needs to make a temporary association between $TAPI_i$ and $AEID_i$ to facilitate subsequent quick database look-up. However, S must not disclose this association to ensure unlinkability.

4.3 Anonymity-preserving authentication and authorization

Fig. 5 shows the anonymous authentication protocol. The notation $[m]_X$ denotes *m* signed by entity *X*. N_X denotes a nonce generated by X. The protocol is circular in the sense that the three parties, P_i, *S*, *T*, form a circle where all messages are sent in one direction. The messages flow according to $P_i \rightarrow S \rightarrow T \rightarrow P_i \rightarrow S$. Thus, each entity receives only

1)	$P_i \rightarrow S:$	$N_{P_i}, TAPI_i$
2)	$S \rightarrow T:$	$N_S, [N_S, N_{P_i}, TAPI_i]_S$
3)	$T \rightarrow P_i:$	$N_T, [N_T, N_S, S]_T, \left[N_T, [N_S, N_{P_i}, TAPI_i]_S\right]_T$
4)	$P_i \rightarrow S:$	$[N_{P_i}, N_T, T]_{P_i}, \left[N_{P_i}, [N_T, N_S, S]_T\right]_{P_i}$
5)	$S \rightarrow T:$	$\left[N_S, [N_{P_i}, N_T, T]_{P_i}\right]_S$

Figure 5: The anonymity-preserving authentication and authorization protocol

messages from the preceding entity, and sends messages only to its succeeding entity.

In this protocol whenever the patient P_i need to sign a message he/she utilizes $TAPI_i = \alpha^{p_i}$ as a public key and p_i as a private key (see previous subsection). Due to the form of the public/private key, the choice of possible signature schemes are restricted to an ElGamal-based signature scheme [Elg85] or similar like Schnorr's signature scheme [Sch89] or the Digital Signature Standard scheme [Nat00] which are all based on the Discrete Logarithm Problem.

The protocol proceeds as follows:

1. P_i generates a nones N_{P_i}, and sends $TAPI_i$, N_{P_i} to S.

2. S generates a nonce N_S, signs $[N_S, N_{P_i}, TAPI_i]$, and then sends $N_S, [N_S, N_{P_i}, TAPI_i]_S$ to *T*.

3. Upon receiving message 2, *T* extracts $[N_S, N_{P_i}, TAPI_i]$ and verifies the consistency of the message by comparing N_S of the signature with the unsigned N_S.

   If both match, T generates a nonce N_T, then signs $[N_T, N_S, S]$ and $\left[N_T, [N_S, N_{P_i}, TAPI_i]_S\right]$, and sends $N_T, [N_T, N_S, S]_T, \left[N_T, [N_S, N_{P_i}, TAPI_i]_S\right]_T$ to P_i.

4. Upon receiving message 3, P_i extracts $[N_T, N_S, S]$, and $\left[N_T, [N_S, N_{P_i}, TAPI_i]_S\right]$ by using public key of T. Then P_i extracts $[N_S, N_{P_i}, TAPI_i]$ using public key of *S*, and verifies the correctness of the challenge N_{P_i}. P_i then verifies that the message is consistent by checking that all three instances of N_T match, and that both instances of N_S match.

   If they are correct, P_i signs $[N_{P_i}, N_T, T]$ and $\left[N_{P_i}, [N_T, N_S, S]_T\right]$ with his private key p_i, and sends $[N_{P_i}, N_T, T]_{P_i}, \left[N_{P_i}, [N_T, N_S, S]_T\right]_{P_i}$ to S.

5. Upon receiving message 4, S extracts $[N_{P_i}, N_T, T]$ and $\left[N_{P_i}, [N_T, N_S, S]_T\right]$ using $TAPI_i$, the anonymous public key of P_i. *S* extracts moreover $[N_T, N_S, S]$ by using the public key of *T*, and verifies the correctness of the challenge N_S. *S* then verifies that the message is consistent by checking whether both instances of N_{P_i} match with N_{P_i} of message 2, and whether both instances of N_T match.

If they are correct, *S* signs $\left[N_S, [N_{P_i}, N_T, T]_{P_i}\right]$, and sends $\left[N_S, [N_{P_i}, N_T, T]_{P_i}\right]_S$ to *T*.

6. Upon reception of message 5, *T* extracts $\left[N_S, [N_{P_i}, N_T, T]_{P_i}\right]$ by using public key of S and verifies that N_S match the values of N_S of message 2. Then *T* moreover extracts $[N_{P_i}, N_T, P]$ from $[N_{P_i}, N_T, T]_{P_i}$ by using $TAPI_i$, and verifies that N_{P_i} match N_{P_i} of message 2, and the correctness of the challenge N_T.

The protocol complies to that the participating entities authenticates each other in such a way the P_i remains anonymous. The team *T* is represented as an entity, but correspondingly consists of a number of collaborating team members. In order for *S* and *P* to be certain that indeed a certain minimum number of the participants of *T* actually are present, and that not just one individual is acting on behalf of *T*, a threshold group-oriented cryptosystem should be employed by *T*. In a threshold group-oriented cryptosystem, a group is represented by a public key, but a specific minimum number of the participants must collaborate to decrypt messages encrypted by the threshold-based public key [Des93, SG99].

5 Security analysis

5.1 The TAPI protocol

First, we need to show that using $TAPI_i$ cannot be associated with P_i and therefore utilization of $TAPI_i$ as the patient's public key will preserve patient's anonymity. According to the algorithm, the patient generates the private p_i randomly. Therefore, the corresponding $TAPI_i = \alpha^{p_i}$ is also random. The association between $TAPI_i$ and patient P_i (via $AEID_i$) is sent encrypted to S with the public key of *S*. Assuming that the employed public key cryptosystem is secure, we can conclude that $TAPI_i$ preserves the patient anonymity, and unlinkability as it is defined in the beginning of this paper is provided.

In both the AEID and TAPI protocols, the value of $AEID_i$ is blindly established where P_i signs by means of his or her private key x_i the blinded value r_S issued by *S*. Since only *S* holds the secret key x_S, *S* can recover $AEID_i$ as an application of the ElGamal cryptosystem [Elg85].

5.2 The AAA protocol

The function of the AAA protocol is for the three parties, P_i, *S*, and *T*, to mutually authenticate each other. P_i is anonymously represented by his or her public key $TAPI_i$. Each party generates a nonce acting as a challenge. For all three parties, the two other parties sign the nonce issued by a first party. Subsequent verification of the signatures establishes the authenticity of the participants. Each signature includes the nonce issued by the originating party in order to prevent that an adversary can obtain illegitimate signatures [BM03, p. 109].

In a given session, each participant issues a unique random nonce in messages 1–3. Each of the succeeding messages is cryptographically linked to each other by means of the nonces. Redundancy of nonces in each messages (except for message 1) ensures that each message is consistent. P_i receiving message 3 can certify the authenticity of *S* and *T* according to the signed challenge N_{P_i} originated at P_i. Moreover, the nonces N_S and N_T ensures that the consistency of message 3 can be certified. By *consistency*, we mean that the parts of the message must correspond, where certification of consistency reveals that a message has been tampered with by modification of substitution of a part of the message. Likewise *S* can, upon receiving messages 1 and 4, certify P_i and *T* and certify the consistency of the message. Finally, *T* upon receiving messages 2 and 5, certifies P_i and *S*.

6 Conclusion

In this paper, we have addressed the need of keeping patient identities anonymous in regard to their medical data, in contrast to only keeping their medical data confidential. This is essential in cases where patients wish that their identities (like names, addresses, personal identity numbers, etc.) cannot to be associated with information about for example related diseases, physical or genetic defects, drug addictions, etc., stated in their respective patient records. To achieve that personal identity information is not to be linked with disclosed medical data, we have proposed a scheme that enables patients themselves to *anonymously* grant medical teams authorization to access their EPRs without revealing their identities to the teams, by instead using certifiable anonymous identifiers. Another benefit of this scheme is that it allows patients to exert control over their own medical records.

References

[BM03] C. Boyd and A. Mathuria. *Protocols for authentication and key establishment*. Springer-Verlag, 2003.

[Des93] Y. Desmedt. Threshold cryptosystems. *Advances in Cryptology - Auscrypt '92, Springer-Verlag*, LNCS 718:3 – 14, 1993.

[DH76] Whitfield Diffie and Martin E. Hellman. New Directions in Cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, 1976.

[Elg85] T. Elgamal. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Transactions on Information Theory*, 31(4):469–472, 1985.

[KDTC04a] A. Kalam, Y. Deswarte, G. Trouessin, and E. Cordonnier. A generic approach for healthcare data anonymization. In *Proc. of the 2004 ACM workshop on Privacy in the electronic society*, pages 31 – 32. ACM Press, 2004.

[KDTC04b] A. Kalam, Y. Deswarte, G. Trouessin, and E. Cordonnier. Smartcard-based Anonymization. In *CARDIS*, pages 49–66, 2004.

[Nat00] National Institute of Standards and Technology. *FIPS PUB 186-2: Digital Signature Standard (DSS)*. January 2000.

[Rin97] T. Rindfleich. Privacy, information technology and health care. *Communications of the ACM*, 40,8, 1997.

[Sch89] Claus P. Schnorr. Efficient identification and signatures for smart cards. In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 239–252, New York, NY, USA, 1989. Springer-Verlag New York, Inc.

[SG99] S. Saeednia and H. Ghodosi. A self-oriented group-oriented cryptosystem without a combiner. In *Proc. of the 4th Australasian Conference on Information Security and Privacy*, pages 192 – 201. Springer-Verlag, 1999.

[Swe97] L. Sweeney. Guaranteeing anonymity when sharing medical data, the datafly system. *Journal of the American Medical Informatics Association*, 1997.

[TER04] A. Tveit, O. Edsberg, and T. Røst. Anonymization of General Practioner's Patient Records. In *Proc. of HelseIT'04*, 2004.