EPR Access Authorization of Medical Teams Based on Patient Consent

Sigurd Eskeland and Vladimir Oleshchuk

Agder University College,
Department of ICT, Grooseveien 36,
4876 Grimstad, Norway
{sigurd.eskeland,vladimir.oleshchuk}@hia.no

Abstract: Electronic patient records (EPR) may contain highly confidential and personal medical information. It is therefore essential that medical data is properly protected and managed. Today, it is widely recognized that patients have a right to self-determination and to exert control of their own medical data by consent. In this paper, we present a cryptographic EPR access authorization scheme that incorporates patient consent as a basis for granting EPR access to medical teams or practitioners. This ensures that only the medical practitioners specified by a consenting patient are granted EPR access. If a patient is unconscious, the variation of the scheme allows an emergency or security team to act on behalf of the patient.

1 Introduction

With the emergence of information technology in health care, there has been extensive focus on security and confidentiality issues of electronic patient records (EPR) in medical environments [Rin97, BB96, FIG06, PEH97]. An important issue here concerns proper access control. A basic criterion for this is legitimacy, i.e., only medical personnel providing medical care to a given patient (or patients) should access only the necessary medical data of the concerning patient they are providing care to [AMA]. Another significant security issue concerns secure and confidential management, handling and storage of personal medical information [Rin97].

In a typical medical information scenario, electronic patient records could be stored in EPR servers that are managed and controlled by one or few security administrators. These administrators would normally possess all privileges with respect to the patient data. They would perform functions such as authorizing and assigning medical practitioners access to the EPRs of the concerning patients that are to be provided care for. Consequently, each security administrator would have full access to all personal medical data. However, as patient records may contain highly sensitive and confidential personal information, it is very important to ensure that such information remains confidential. In this scenario, the patients have no actual control over their medical data and are in practice left no other option than to simply trust that their data will not be disclosed to illegitimate personnel nor manipulated. However, *patient consent* identifies what nowadays has generally been recognized as patients' rights to exert control over their own medical data [GN05, CC04, BBPH07, AMA]. Patient consent has today become an important principle in medical ethics and access control policies. Even though this has lately been a widely recognized aspect concerning patients' self-determination and right to exert control over their own personal medical data, patient consent is in practise enforced by means of filling out paper forms. Since this does not impose an actual obstacle against illegitimate EPR access, it is therefore important that patient consent should be integrated in medical access control systems.

As medical data in general should be protected from disclosure to unauthorized personnel, certain data are more sensitive than others. Since medical records may possibly contain 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 essential that such information should be protected from disclosure including to security personnel except when legitimately needed by medical practitioners. To ensure the privacy of medical data, the EPRs could be stored encrypted at the EPR server. Alternatively, assuming that the EPR is arranged into blocks or modules, a proper arrangement could be that only certain EPR modules containing particulary sensitive data are encrypted. Encryption imposes, however, the problem of secure key storage and management. For example, if a cryptokey is revealed, the encrypted data can be decrypted and revealed. If a cryptokey is lost, the data is lost. A straight-forward solution is that one or few security administrators would control all EPR cryptokeys. Due to the fact that security administrators would be individually entrusted with the responsibility of managing possibly thousands of secret EPR keys, which could impose a considerable risk of human error, fraud, attacks and possibly high workloads.

A naive and insufficient solution could be to use a threshold-based (t, n) secret sharing scheme where $t < n$ and the corresponding key is split into *n* secret shares [Sha79]. The shares are distributed to *n* authorities, so that each individual holds one share. The secret key can only be reconstructed when at least *t* of the participants pool their shares together. However, there are at least three shortcomings with this approach: 1) The same secret key is associated to all EPRs. 2) When reconstructed, the single secret key is revealed once and for all. 3) The participants must reveal their secret shares to each other in order to reconstruct the secret key. Thus, there is no confidentiality regarding the secret shares.

In this paper, we present a cryptographic access authorization scheme that incorporates the concept of patient consent. We consider *the function of patient consent to be equivalent to the function of granting*. By granting, we mean that an entity, i.e., a patient, has the authority to grant another entity, i.e., a medical team, access to his or her EPR. Therefore, we use the terms grant and patient consent interchangeably. Moreover, we consider that the term *to grant access* is semantically equivalent to the term *authorize access*.

Note that EPR access could be granted to individuals, except to the patients themselves, instead of teams like to a specific doctor (specialist, general practitioner, etc.). This provides proper distribution of trust since the patient is in charge of disclosing his or her EPR by consent. However, the medical data cannot be accessed by the patients themselves without special arrangements.

We assume that each EPR is encrypted by a unique and distinct key unknown to all participants including the pertaining patient. The scheme provides secure and confidential establishment of EPR cryptokeys for subsequent decryption of the pertaining medical records. There are no cryptokey tables, but the cryptokey for a given EPR is temporarily restored at the EPR server for each session by means of the consenting patient holding a secret user key (not the EPR cryptokey) in conjunction with the EPR server. The scheme is secure and prohibits deduction of secret user keys or EPR cryptokeys. Accordingly, medical data is protected due to that electronic patient records (or modules) can be stored encrypted at an EPR server, prohibiting medical data to be disclosed without the collaboration of the consenting patient and a medical team. *Encryption* of updated medical data could be done at the EPR server or by medical practitioners by means of a corresponding public key.

The EPR cryptosystem presented in [Esk06] seems to be the only EPR cryptosystem incorporating patient consent for EPR authorization of medical teams. However, a serious security weakness about this cryptosystem is that the EPR server does not have an active function in EPR cryptokey reconstruction, enabling a colluding patient and team to reveal the secret EPR cryptokey independently of the EPR server.

The rest of the paper is organized as follows: In Section 2, we give a brief introduction to threshold cryptography. In Section 3, we present the cryptographic scheme. In case a patient is unconscious, he or she would not be in a position to actively and consciously grant anybody EPR access. In Section 4, we present a variant of this scheme for the emergency case, allowing a coalition of security administrators or emergency team to grant medical personnel EPR access on behalf of the patient.

2 Group-orientation and threshold cryptosystems

The motivation of threshold cryptosystems is to provide flexibility by allowing a minimum number of participants, i.e. a minimum arbitrary composed subset of members of a group, department or organization, to carry out a cryptographic operation instead of requiring *all* the members for this. Thus, the term *threshold* denotes the minimum number of participants of the group or team that must collaborate in order to carry out the cryptographic operation. This is desirable in scenarios where some sort of separation of duty is required, for example that the holder or originator of some sensitive information like a secret key, is only willing to disclose it as result of the agreement and cooperation of a given number of designated participants. Accordingly, it is precluded that single individuals can obtain the secret on their own. As a practical example, we can consider access to a bank vault where it is not desirable that one person alone would possess and control the key to the vault due to the risk of fraud, robbery and extortion, but the participation of at least 2 or 3 persons out of for instance 4, each holding a unique and secret key, should be required in order to unlock the vault. Common for such cryptosystems is that each active participant performs some partial computations that they succeedingly ”pool” together in order to complete the cryptographic operation. This is a desirable property in security systems involving collaboration of several participants.

Typical threshold-oriented applications are threshold decryption and threshold signatures. A threshold decryption cryptosystem is a cryptosystem requiring an arbitrary composed subset of a minimum number of participants of a given group to collaboratively perform decryption. Represented by a public key, outsiders can confidentially address the group. Only by collaboration where the active group members are providing partial computations, the encrypted message can be decrypted [DF89, Ped91, SG99]. Likewise, regarding threshold signatures [Har94, LHL94], only a minimum subset of the team can compute signatures due to the threshold requirement.

3 EPR access authorization based on patient consent

In this section, we present the cryptographic EPR access authorization scheme. It assumes that the medical records are stored encrypted on a server. Each EPR is encrypted by a unique secret key and there are *no* cryptokey tables. The proposed scheme has mainly two purposes: The first is to enable patients to securely grant EPR access to medical teams and medical practitioners. The second purpose is to provide secure and temporarily reconstruction of the secret cryptokey for a given EPR at the EPR server from the process of a patient granting a medical team access to his or her EPR.

The scheme enables reconstruction of a predefined EPR cryptokey (which thus is the same for each session), based on the computations involving the secret keys of the pertaining patient and the EPR server. The server subsequently decrypts the given EPR. The protocol prevents disclosure and deduction of restored EPR cryptokeys to any party other than the EPR server. It moreover prevents that any secret inputs or keys of the participants can be deduced by any participating or external party.

The patient grants a medical team EPR access by basically generating a secret cryptographic challenge in agreement with the public key of the pertaining team. The EPR server will only be able to reconstruct the secret EPR cryptokey provided a valid response. Since only associated members of the addressed medical team can collaboratively provide the correct response to the challenge, this ensures that *no one other than the genuine team can obtain access to the patient's EPR*. Otherwise, the pertaining EPR cryptokey cannot be restored. An eligible minimum number of active team participants is defined by applying a threshold mechanism.

3.1 Protocol initializations

A trusted authority (TA) is responsible for providing the required public key infrastructure. Let $\mathcal{U} = \{P_1, \ldots, P_n\}$ denote a medical team of *n* members. The TA defines the minimum number of active participants *t* that are required in order to obtain the EPR access granted by the patient. This subcoalition is denoted $T \subseteq \mathcal{U}$ where $|T| \geq t$. According to the Shamir secret sharing scheme [Sha79], the TA generates a unique secret polynomial of degree $(t-1)$:

$$f(x) = \sum_{j=0}^{t-1} a_j \, x^j$$

that represents the team $\mathcal{U}$. The TA computes one personal long-term secret share for each team member as follows: For each $P_i \in \mathcal{U}$, the TA arbitrarily selects a input x_i from $\mathbb{Z}_q$, and computes the secret user share

$$s_i = f(x_i) \pmod q$$

where *q* is a large public prime. The team $\mathcal{U}$ is externally represented by the public key

$$y = \alpha^{a_0} \pmod p$$

where $a_0 = f(0)$ and α is a generator to $\mathbb{Z}_p$. Note that $p = 2 \cdot q + 1$ is a large public prime.

Let *S* denote the EPR security server and G_i denote the patient (the granting entity). The TA moreover provides the EPR server *S* with the secret key k_s and each patient G_i with the secret key k_i where $k_s, k_i \in \mathbb{Z}_q$. The TA computes the secret EPR cryptokey

$$K_i = \alpha^{k_s \, k_i} \pmod p$$

by which the TA encrypts the EPR of G_i by means of a proper cryptographic algorithm. The TA deletes K_i subsequently.

3.2 Protocol description

In this section, we describe the cryptographic EPR access authorization scheme. This is moreover presented in Figure 1, and goes as follows:

Step 1. The protocol is initiated by *S* that generates the secret random numbers $r_1, r_2 \in \mathbb{Z}_q$, and computes for G_i

$$a_s = \alpha^{r_1} \pmod p, \quad b_s = \alpha^{k_s} \alpha^{r_1 r_2} \pmod p$$

Step 2. The patient G_i grants EPR access to a medical team $\mathcal{U}$ (the grantee) by means of the team's public key y. G_i generates a random secret number $r_i \in \mathbb{Z}_q$, computes and returns (c_i, d_i, R_i, y) to *S* where

$$c_i = b_s^{k_i} \pmod p, \quad d_i = y^{r_i} \, a_s^{k_i} \pmod p \quad \text{and} \quad R_i = \alpha^{r_i} \pmod p$$

The public key *y* is included in the message, indicating to the medical team that G_i claims to be the grantee. Whether *y* is the *genuine* key applied in the computation of d_i, is certified according to the correctness of K_i computed in Step 5. Note that since y^{r_i} is an unknown factor of d_i and r_i is secret, this can only be resolved by the partial computations of team members holding the secret user shares, collectively computing $(\alpha^r)^{a_0}$ which corresponds to y^{r_i}.

Step 3. *S* checks that *y* is a public key of a genuine and approved medical team or medical practitioner. Otherwise, G_i could grant EPR access to illegitimate persons, knowingly or unknowingly. If *y* is accepted, *S* broadcasts the challenge

$$u_s = (\alpha\, R_i)^{r_2} \pmod p$$

Otherwise, terminate.

Step 4. To correctly respond to the challenge u_s, the partial computations of a subcoalition $T \subseteq \mathcal{U}$ of at least *t* participants are required. Each team member $P_j \in T$ receives u_s, and computes and returns the partial computations $z_j = u_s^{s_j} \pmod p$.

Step 5. Key computation. *S* applies Lagrange interpolation to the partial computations of *T* according to

$$Y_i = \prod_{j \in I_T} z_j^{b_j} \pmod p \quad \text{where} \quad b_i = \prod_{\substack{j \in I_T \\ i \neq j}} \frac{x_j}{x_j - x_i} \pmod q$$

and $I_T = \{i \mid P_i \in T\}$. By means of the secret r_2, only *S* is capable of reconstructing the EPR cryptokey as follows:

$$\begin{aligned} K_i &= \alpha^{k_i k_s} = c_i \cdot Y_i \cdot (d_i \cdot y)^{-r_2} \pmod p \\ &= (\alpha^{k_s \cdot k_i} \cdot \alpha^{r_1 r_2 k_i}) \cdot (\alpha^{r_i r_2 a_0} \cdot \alpha^{r_2 a_0}) \cdot (y^{r_i} \cdot \alpha^{r_1 k_i} \cdot y)^{-r_2} \\ &= (\alpha^{k_s \cdot k_i} \cdot \alpha^{r_1 r_2 k_i}) \cdot (y^{r_i r_2} \cdot y^{r_2}) \cdot (y^{-r_i r_2} \cdot \alpha^{-r_1 r_2 k_i} \cdot y^{-r_2}) \end{aligned}$$

by which it subsequently decrypts the pertaining EPR. The EPR can be securely transferred to $T \subseteq \mathcal{U}$, for example by encrypting it with public key of $\mathcal{U}$.

Given that r_2 is secret prohibits anyone but *S* to obtain K_i. Note that in Step 2, G_i indicates the public key of the grantee. Since this is included in the key reconstruction phase, the correctness of K_i is ensured according to that only $T \subseteq \mathcal{U}$ represented by *y* can provide the correct response in Step 4.

Note that the function of this protocol is very different from the function of key establishment protocols which provide secure establishment of non-predefined random secret shared keys over insecure networks. Participants of such protocols are usually authenticated towards their public keys. Our protocol differs since its function is to securely establish a *predefined* secret key whereof its correctness implies authenticity of the participants. For example, only G_i, holding k_i, can contribute to establish the correct K_i, disregarding the emergency case. Further user/key authentication is thus not required.

3.3 Security discussion

In this subsection, we will show that the proposed protocol preserves both authenticity and key secrecy.

$$
\begin{array}{lcl@{\qquad}l}
\underline{S} & & \underline{G_i} & \underline{P_j \in T} \\
r_1,\, r_2 \in_R \mathbb{Z}_q & & r_i \in_R \mathbb{Z}_q & \\
a_s = \alpha^{r_1} & & & \\
b_s = \alpha^{k_s + r_1 r_2} & \xrightarrow{\quad (a_s,\, b_s) \quad} & c_i = b_s^{k_i} & \\
 & & d_i = y^{r_i} \cdot a_s^{k_i} & \\
 & \xleftarrow{\quad (c_i, d_i, R_i, y) \quad} & R_i = \alpha^{r_i} & \\
u_s = (\alpha \cdot R_i)^{r_2} & \multicolumn{2}{c}{\xrightarrow{\qquad\qquad\qquad u_s \qquad\qquad\qquad}} & \\
 & \multicolumn{2}{c}{\xleftarrow{\qquad\qquad\qquad z_j \qquad\qquad\qquad}} & z_j = u_s^{s_j} \\
K_i = c_i \cdot Y_i \cdot (d_i \cdot y)^{-r_2} & & &
\end{array}
$$

Figure 1: EPR cryptokey reconstruction due to the proposed scheme. All computations are in $\mathbb{Z}_p$.

Authenticity. The protocol provides a legitimate user to securely reconstruct the secret EPR cryptokey associated to him or her at the EPR server and no one else. Thus, if an illegitimate user tries to establish a given key, it will fail. It is essential that the protocol preserves the authenticity of the users, resisting any masquerading attack so that no entity (internal or external) may successfully masquerade as another entity. Since the goal is to establish a fixed secret EPR cryptokey, explicit user authentication is not required since the key is established as a function of the secret user keys held by the participants and is therefore implicitly provided.

Like most cryptographic authentication protocols, user authentication is provided on the assumption that only the genuine user and no one else is holding a specific secret whereof the genuineness of his or her identity is based. The protocol provides the ability for the user to prove that he or she actually holds the specific secret according to the correctness of the result. Accordingly, the protocol must prevent that anybody else can establish or obtain the correct result and therefore illegitimately obtain access to a patient's EPR.

Note that (a_s, b_s) are cryptographically bound to the secrets r_1 and r_2 only known by S. This binding prevents replay attacks where an adversary attempts to successfully run the protocol by masquerading. An adversary replaying the numbers c_i, d_i, R_i from a former session would cause inconsistency in the key recovery phase since r_2 is distinct and unique for each session, and only the genuine value of r_2 can resolve the EPR cryptokey.

Key secrecy. There are two aspects regarding key-secrecy. First, it is required that no secret user keys or secret user shares can be deduced from the messages. Secondly, it must be infeasible to deduce EPR cryptokeys for anybody except S.

Regarding the first key secrecy requirement, no user input must be revealed from the computations. This is obtained due to the Discrete Logarithm Problem that protects the secret key k_s of S in Step 1, the secret key k_i of G_i in Step 2 and the secret user shares of each $P_j \in T$ in Step 4.

Considering the secrecy of the EPR cryptokey, disclosure of α^{k_s} must be prevented, otherwise a patient could compute $K_i = (\alpha^{k_s})^{k_i}$. Regarding $a_s = \alpha^{k_s}\alpha^{r_1 r_2}$ and $b_s = \alpha^{r_1}$, due to the Diffie-Hellman assumption, it is computationally infeasible to obtain $\alpha^{r_1 r_2}$ given α^{r_1} and α^{r_2} where r_1 and r_2 are unknown. However, an adversary could try to attack the protocol by returning α^{-r_1} to S in Step 3. Since S would compute $u_s = \alpha^{-r_1 r_2}\alpha^{r_2}$ in Step 4, the adversary would only obtain $a_s \cdot u_s = \alpha^{k_i} \cdot \alpha^{r_2}$ where α^{r_2} is unknown. Thus, the attack would fail.

The EPR cryptokey is the first factor in $c_i = \alpha^{k_s k_i} \cdot \alpha^{r_1 r_2 k_i}$. Likewise, it is protected by the unknown second factor $\alpha^{r_1 r_2 k_i}$ where it is computationally infeasible to obtain $\alpha^{r_1 r_2}$.

Note that α^{k_i} could be a public key of G_i though it would have no function in this protocol. An adversary would have no use of this due to that knowledge of the secret k_i is required for the exponentiations for computing c_i and d_i in Step 2.

4 The emergency case

There could be situations when patients are in a coma, or situations of car accidents, fire, terrorist acts, etc., where patients may be unconscious and may therefore not be able to actively grant any medical practitioners access to his or her EPR. In this section we describe a modified version of the protocol presented in the previous section to handle such emergency cases. In emergency cases, a coalition of security administrators or an emergency team could act on behalf of the patient to grant EPR access. Note that there should be a minimum threshold in order to prohibit that any single individual may solely grant or obtain access to personal medical data.

In normal situations, the patient would by means of his or her secret key grant any team access to his or her EPR. For the emergency case, each patient could be represented by an associated public parameter or identifier that the security team would use to reference the patient.

The emergency case could be handled as follows: The TA defines the minimum threshold t' of security administrators that is required to actively grant on behalf of a pertaining patient that is disabled. The TA generates a random secret polynomial $g(x)$ of order $(t' - 1)$ that represents the team of security administrators SA. The TA computes for each administrator $A_i \in SA$ a secret share according to

$$t_i = g(i) \pmod q$$

The secret keys of each patient G_i is computed according to

$$k_i = g(h(G_i)) \pmod q$$

where G_i denotes the identity of the patient and *h* is a secure one-way function.

The protocol is as the previous except that SA coalition acts on behalf of the patient which introduces a second team aspect. The protocol goes as follows:

Step 1. The protocol is initiated by *S* that generates the secret random numbers $r_1, r_2 \in \mathbb{Z}_q$, and forwards to SA the challenges

$$a_s = \alpha^{r_1} \pmod p, \quad b_s = \alpha^{k_s + r_1 r_2} \pmod p$$

Step 2. The SA grants a medical team $\mathcal{U}$ EPR access on behalf on G_i by means of the team's public key y. Each $A_j \in SA$ generates a random secret number $r_j \in \mathbb{Z}_q$, and computes and returns (c_j, d_j, R_j) to *S* where

$$c_j = b_s^{t_j\, b_j} \pmod p, \quad d_j = a_s^{t_j\, b_j} y^{r_j} \pmod p \quad \text{and} \quad R_j = \alpha^{-r_j} \pmod p$$

Note that

$$b_j = \prod_{\substack{k \in I_{SA} \\ j \neq k}} \frac{h(G_i) - k}{j - k} \pmod q$$

and $I_{SA} = \{j \mid A_j \in SA\}$. Also note that the computations of SA agree to Lagrange interpolation on exponents (Step 3) which corresponds to applying k_i as an exponent to (a_s, b_s) as in Section 3.

Step 3. *S* receives the messages from SA and completes the Lagrange interpolation on the exponents by multiplication

$$c_i = \prod_{j \in I_{SA}} c_j \pmod p, \quad d_i = \prod_{j \in I_{SA}} d_j \pmod p \quad \text{and} \quad R = \prod_{j \in I_{SA}} R_j \pmod p$$

and forwards the challenge R to $\mathcal{U}$.

Step 4. To correctly respond to the challenge R, the partial computations of a subcoalition $T \subseteq \mathcal{U}$ of at least *t* participants are required. Each $P_j \in T$ receives R, and computes and returns $Y_j = R^{s_j} \pmod p$.

Step 5. Due to the fact that r_i is secret, *S* is required to obtain Y_i by Lagrange interpolation of the partial computations of *T* according to

$$Y_i = y^{r_i} = \prod_{j \in I_T} Y_j^{b_j} \pmod p \quad \text{and} \quad b_i = \prod_{\substack{j \in I_T \\ i \neq j}} \frac{x_j}{x_j - x_i} \pmod q$$

where $I_T = \{i \mid P_i \in T\}$. Finally, *S* reconstructs the secret key

$$K_i = c \cdot (d \cdot Y_i)^{-r_2} = (\alpha^{k_s \cdot k_i} \cdot \alpha^{r_1 r_2 k_i}) \cdot (y^{-r_i r_2} \alpha^{-r_1 k_1 r_2}) \cdot (y^{r_i r_2}) = \alpha^{k_i k_s} \pmod p$$

by which it subsequently decrypts the pertaining EPR.

5 Conclusion

In this paper, we have presented a cryptographic EPR access authorization scheme that incorporates patient consent as a basis for granting EPR access. This ensures that only the medical practitioners specified by a consenting patient are granted EPR access. If a patient is unconscious, a variation of the scheme allows an emergency or security team to act on behalf of the patient.

The security scheme assumes that electronic patient records (or specific parts of patient records) are stored encrypted at the EPR server and each EPR is encrypted with a unique and secret key. The key management problem is precluded due to the fact that there are no cryptokey tables and no one, including patients, hold or can obtain the cryptokey that can decrypt his or her EPR. However, each patient holds a long-term secret user key. Instead, the protocol enables secure reconstruction of a secret EPR cryptokey at the EPR server from the cryptographic interaction between the EPR server and the pertaining patient granting a medical team access to his or her EPR. This allows the EPR server to subsequently decrypt the pertaining EPR. The scheme is secure in the sense that it prohibits that secret user key and EPR cryptokeys can be deduced and disclosed.

References

[AMA] American Medical Association. Patient Confidentiality. See http://www.ama-assn.org/ama/pub/category/4610.html.

[BB96] J. Biskup and G. Bleumer. Cryptographic protection of health information: cost and benefit. *International Journal of Bio-Medical Computing*, 43:61–67, 1996.

[BBPH07] J. Bergmann, O. Bott, D. Pretschner, and R. Haux. An e-consent-based shared EHR system architecture for integrated healthcare networks. *International Journal of Medical Informatics*, 76(2-3):130–136, 2007.

[CC04] E. Coiera and R. Clarke. e-Consent: The design and implementation of consumer consent mechanisms in an electronic environment. *Journal of the American Informatics Association*, 11:129–140, 2004.

[DF89] Y. Desmedt and Y. Frankel. Threshold cryptosystems. In *CRYPTO '89: Proceedings on Advances in cryptology*, pages 307–315. Springer-Verlag New York, Inc., 1989.

[Esk06] Sigurd Eskeland. Access control by secure multi-party EPR decryption in the medical scenario. In *Communication, Network, and Information Security (CNIS)*, pages 99–103. IASTED/ACTA Press, 2006.

[FIG06] FIGO Committee for the Ethical Aspects of Human Reproduction and Women's Health. Confidentiality, privacy and security of patients' health care information. *International Journal of Gynecology & Obstetrics*, 93,2:184–186, 2006.

[GN05] P. A. B. Galpottage and A.C. Norris. Patient consent priciples and guidelines for e-consent: a New Zealand perspective. *Health Informatics Journal. SAGE Publications*, 11, 1:5 – 18, 2005.

[Har94] L. Harn. Group-oriented (t, n) threshold digital signature scheme and digital multisignature. 141(5):307–313, 1994.

[LHL94] C. M. Li, T. Hwang, and N. Y. Lee. Threshold-Multisignature Schemes where Suspected Forgery Implies Traceability of Adversarial Shareholders. In *Eurocrypt*, pages 194–204, 1994.

[Ped91] T. Pedersen. A threshold cryptosystem without a trusted party (Extended Abstract). In *Eurocrypt '91, LNCS*, volume 547, pages 522–526. Springer-Verlag, 1991.

[PEH97] *Committee on Maintaining Privacy and Security in Health Care Applications of the National Information Infrastructure. For the Record: Protecting Electronic Health Information.* National Academies Press, 1997.

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

[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.

[Sha79] A. Shamir. How to share a secret. *Commun. ACM*, 22(11):612–613, 1979.