

Rendezvous-based Access Control for Medical Records in the Pre-hospital Environment

Feike W. Dillema and Simone Lupetti
Department of Computer Science, University of Tromsø
Tromsø, Norway
{dillema,simone}@cs.uit.no

ABSTRACT

We present rendezvous-based access control for access control in the pre-hospital environment. Rendezvous-based access control is a simple cryptographic access control method that provides access if and only if patient and health worker meet in the physical world. Access is provided locally and does not depend on connectivity with remote systems. It is therefore suitable in an environment with small mobile devices that have local connectivity but may be disconnected now and then from remote systems. It is designed to protect against aggregation threats without letting the patients carry their own medical data. A system can then be implemented where the tokens carried by the patients are simple and robust which is easily managed. We believe that our mechanism provides a useful alternative to remote access to a centralized system and to patients carrying their own medical record (on a smartcard e.g.).

Categories and Subject Descriptors

J.3 [Life and Medical Sciences]: Medical information systems; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms

Security, Design

Keywords

Security model, aggregation threat, capabilities, electronic health records, pre-hospital environment

1. INTRODUCTION

Availability of medical records wherever medical care is required may save lives. A rescue team in a remote area treating avalanche victims, an ambulance team trying to stabilize a car crash victim, are examples where quick access to medical data can be of life-saving value. The progressive shift from paper records to digital ones, associated

with the proliferation of sophisticated mobile computing devices, offers a whole range of new opportunities to make this possible.

Granting health workers ubiquitous access to the medical data of all patients they may encounter in the field is certainly feasible and appealing, but this may exacerbate threats to privacy of this data [2, 3]. In closed environments such as hospitals, enforcement of access control is relatively easy, even though the complexity of the security mechanisms can already become overwhelming [11]. In such environments, roles can be defined in a meaningful way, workers and patients can easily be identified in a semi-permanent way, and physical security of data (private intranets, locked server-rooms, etc.) can offer adequate levels of protection.

Access to medical data is an open-ended scenario where proper planning of access control is frustrated because it is impossible to predict which health worker will treat which patient. Despite of this, it is still necessary to implement proper access control to medical data in order to defend against attackers interested in the medical data of a specific patient (journalists looking for medical data about celebrities, employers screening prospective employees, etc.). But, maybe most importantly, we need protection against attackers that have an interest in large collections of medical records (e.g. insurance companies that want to refuse risky customers).

We propose an access control architecture to allow access to medical data only when a particular patient and health worker physically meet. We believe that this property makes it suitable for the pre-hospital environment enabling ubiquitous access by health workers to medical data of *those and only those* patients that they are actually treating. Our access control mechanism has the additional benefit of enabling off-line access to medical data at the time of treatment by allowing its full or partial replication, in a secure way, to untrusted equipment in the field. The data could be replicated at storage servers stationed at every GSM base station, for example. Replica storage servers could also be placed in ambulances and police cars equipped that can then provide localized access to health workers using WiFi wireless connectivity. Depending on the size of the aggregate data involved and the quickly increasing storage capacity of mobile devices, it may soon even be feasible to keep replicas of the data with each and every health worker. Reducing the dependency on continuous connectivity to a central database represents, in our opinion, a desirable feature in pre-hospital environments such as emergency scenarios, rural environments and in cases of catastrophic events that

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

HealthNet'07, June 11, 2007, San Juan, Puerto Rico, USA.
Copyright 2007 ACM 978-1-59593-767-4/07/0006 ...\$5.00.

may take out or overload communication infrastructure. Finally, our architecture accounts both for external threats (theft of records by outsiders) and internal threats (misbehavior of a health operator) to patient privacy.

2. SECURITY MODEL

The British Medical Association (BMA) commissioned a study of the threats to personal health information [1] and a security policy model [4] suitable to protect against them. This model, which we will refer to as the BMA model, tries to translate traditional ethics of the profession into a concise set of rules. We will use the BMA model in two ways. First, we will use the BMA model as inspiration and starting point for a model specifically covering the security requirements of medical data in the pre-hospital environment. Second, we assume an electronic health record (EHR) system [10] designed after the guidelines of the BMA model to be present in the hospital environment and to be the main system where data is stored and maintained. The infrastructure that allows access to medical records in the field will therefore represent an extension to this.

The BMA model can be summarized by nine principles that express what are the requirements for an EHR system. The first of these principles states that “A clinician may open a record with herself and the patient on the access control list”. For the pre-hospital environment, an access control regime based on ACLs is not a viable solution because predicting what health worker will encounter some patient seems impossible, and with it the *a priori* construction of meaningful ACLs. We therefore specify the following principle:

Principle 1. Health workers in the pre-hospital environment may access only the records of patients that they are treating.

Another principle of the BMA model (the fourth in the original paper) requires the consent of a patient before somebody is granted access to his medical data except in emergency or in case of statutory exemptions. This raises the question whether treatment in the pre-hospital environment can be regarded as emergency access and as such be exempted from patient consent. While in these settings medical treatment will likely be unexpected and unpredictable, it is a well-defined situation where the kind of parties involved (a health worker and a patient) and the kind of access they want (read and write access for the health worker) are known. Patients should therefore be able to decide *in advance* whether they consent to such ‘emergency’ access or not. This is synthesized in the following principle:

Principle 2. A patient must be able to control whether medical data about him is accessible in the pre-hospital environment.

At the time the medical data is generated in the pre-hospital environment, it’s relevance for further treatment or auditing may not at all be obvious. Health workers should not be burdened (nor trusted) with deciding who should be allowed to access this data. A sound approach is then to automatically place all data generated in the pre-hospital environment under the same access regime as the regular (in-hospital) medical record for the patient. We therefore add the following principle to our model:

Principle 3. Medical data produced in the pre-hospital environment shall be made available to those on the ACL of the patient’s medical record in the hospital EHR system.

Finally, prevention of aggregation of medical data is a critical requirement for access in the pre-hospital environment. Because the dramatically increased availability of patients’ records will likely make them much more exposed to attacks, aggregation of medical data is unacceptable. The notification of patients of such aggregation as proposed by one of the principles of the BMA model may not constitute an adequate or an effective measure here, and stronger protection needs to be devised. So, our version of this last principle is simply:

Principle 4. Aggregation of personal health information in the pre-hospital environment must be prevented.

These four principles are tailored for the pre-hospital environment and address therefore its peculiarities but are, being derived from the BMA model, compatible with and complimentary to the BMA model.

3. SYSTEM DESIGN

Because we limit the scope of our design to provide access to medical data in the pre-hospital environment only, we will not discuss other parts of an EHR system in use in a hospital, nor how access is or should be provided within the hospital environment itself. To this end, we assume that an EHR system provides access within the hospital environment and to general practitioners (GP).

3.1 Design choices

From the system design point of view, two main issues must be addressed: *where* access control decisions are made, and *what* is used as basis for these decisions. We present the design choices regarding these issues and the rationale behind them.

Trusted computing base. In the case that access control decisions are made in the pre-hospital environment, then (a part of) the equipment in use there becomes part of the *trusted computing base* (TCB) of the system [9]. Extending the TCB to include part of the pre-hospital environment means that this equipment must be protected against direct physical compromise. Because such protection may be hard and expensive to provide, we require the equipment used by health workers in the pre-hospital environment to *not* be part of the *trusted computing base* of the system. This makes that attacks on this equipment alone cannot cause violations of the security policy of the system.

Data availability. One way to make medical data available in the pre-hospital environment is to provide full connectivity between paramedics and the hospital’s EHR system. Alternatively, or in addition, one could replicate relevant medical data to keep it close to where it is needed and thus provide off-line access to it. This alternative would be more robust because it would not depend on connectivity with the hospital information system, connectivity that may be hard or expensive to provide in all scenarios at all times. Today’s computing and storage technology certainly makes a mobile, distributed storage system feasible. Of course, the many replicas must be synchronized on a regular basis (maybe upon each return to the hospital or during periods of

connectivity with it). A large body of literature is available on how such consistency can be maintained or restored [6].

Co-location of health worker and patient. Granting health workers in the pre-hospital environment access to the medical data of just any patient they may encounter may constitute an unacceptable form of aggregation. To prevent such a threat, we require access to the medical data of a patient to be conditional on the presence of that patient. In this way, health workers can only access medical data of those they treat, but of no other. This can be achieved by equipping *the patient* with a capability which authorizes access to his medical record only [13]. Because the patient under emergency treatment may be incapacitated, these capabilities must be easily transferable from patient to health worker. It follows that the medium used to store these capabilities must be carried by the patient and that it can only be protected by preventing physical access to it.

Use-once authorization. The risks of loss and theft of patient tokens must be taken into account and the resulting damage must be minimized. Limiting or even preventing damage due to lost or stolen tokens is typically performed by invalidation/revocation of the token as soon as the theft or loss is discovered. Such a dependence on detection of loss or theft of a token is acceptable in the cases where somebody is able to discover the loss in a reasonably short time, either when trying to use the token or by detecting misuse (for example in a credit-card invoice). Patients may only rarely use their token, if ever at all, so quick detection of theft or loss will have to depend on the detection of misuse. Accesses to medical records can (and probably should) be logged and regularly audited to uncover misuse [7], but it may be quite expensive to do so in a timely manner. We choose to rely on a simpler technical solution instead, which is to give the patient’s token a use-once semantics. Using the token then means that a new token must be issued to the patient. The patient will therefore be automatically informed of any kind of (mis-)use of his token. Use-once semantics also minimize damage of loss or theft by making the possible damage independent on the detection time interval. Use-once semantics is reasonable in our setting because after legitimate use, the patient will likely end-up in the hospital or at his GP where his token can easily be renewed.

Unlinkability of patient and his record. The patient tokens should not assist attackers in identifying the patient. A lost token is then no threat to the privacy of the patient, provided the finder has no other means of linking the medical record to its patient (e.g. by social engineering or by the content of the record itself). We require then that the patient tokens do not identify their owner.

3.2 Rendezvous-based access control

Our access control framework uses capabilities. These were first described by Dennis and Van Horn [8] and a capability is defined by them as a “a token, ticket or key that gives the processor permission to access an entity or object in a computer system”. Classical capabilities tie a unique object identifier to access rights [18, 17], and users are granted access by proving possession of the appropriate capability to the reference monitor guarding the object [15]. Our approach to access control does not involve such a reference monitor but uses capabilities to access cryptographically protected data directly. We name our scheme *rendezvous-based access control* because it requires the co-location of the

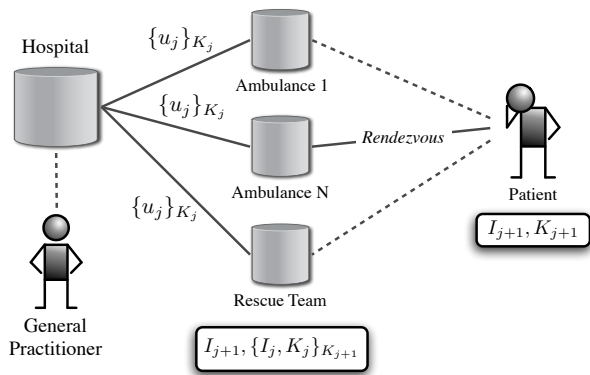


Figure 1: System model for access in the pre-hospital environment.

patient and a health worker in the pre-hospital environment to gain access to the medical data of the patient. At the heart of our scheme lies the protection of two components namely the token carried by the patient and the encrypted data carried by health workers.

Capability-based authorization raises the question of how principals acquire such capabilities. While these do not require identification of principals that use them, principals are often required to authenticate themselves to acquire the capabilities they need. Following one of the principles of the BMA model, we assume a single clinician to be responsible for a patient’s medical record. This person is the only one who may alter the access control list for his medical record. For the sake of this presentation, we assume that patients have a stable relationship with a General Practitioner (GP) who is the responsible party for his patients’ medical records. We also assume that a GP can authenticate his patients, and we make him responsible for handing out capabilities/tokens to his patients (during patient consultations and visits, for example).

4. PROTOCOL

We use a system model featuring EHR systems in the hospitals, general practitioners (GPs), health workers in the pre-hospital environment (paramedics) and patients as shown in Figure 1.

4.1 Medical records

We assume that when a patient is assigned to a GP, the GP constructs the medical record to be used in the pre-hospital environment. This record could, in theory, be the same as the record for the patient in the EHR system. We believe, however, that a record specifically tailored for the use in the pre-hospital environment may be more useful and more appropriate. Not only would this reduce the storage requirements for the equipment in use, it would also allow patients to request the exclusion of data items they deem to private for use in the pre-hospital environment. Apart from this, we assume our records to have properties typical of EHR systems, for example as those specified by the openEHR architecture [5]. In particular, they:

- use globally unique identifiers (GUIDs) to identify medical records,

- make use of versioning; a record consists of a tree of updates or change-sets,
- separate demographic data from the medical data (such that a patient is not easily identified from it).

In addition to this, our records have a GUID I_j associated with *each version* u_j of the record. Each of these versions is encrypted with a different encryption key K_j . A whole medical record R is then defined as an ordered set of encrypted updates:

$$R = \{\{u_0\}_{K_0}, \{u_1\}_{K_1}, \dots, \{u_j\}_{K_j}\}$$

Associated with the record R as a whole is a symmetric encryption key K_R . The GP initializes a record for a new patient by generating the GUID I_R of the record and the record's encryption key K_R . This encryption key K_R is stored in the patient's medical record in the hospital's EHR system. As we will see later, K_R is used to implement Principle 3 of our model.

4.2 Record and token creation

We replicate the encrypted records to the equipment of all health workers that may need access to it, and issue tokens to patients that give access to the encryption key for this data. Because the tokens carried by patients will be difficult and expensive to access by the system's management, we choose to minimize the access requirements on these tokens beyond normal use, e.g. for revocation purposes. The GP issues an access token to a patient with the following steps:

1. Creates or fetches the current version u_j of the record R from the EHR system. It is identified by GUID I_j .
2. Generates the symmetric encryption key K_j for the version u_j of the record.
3. Encrypts u_j with K_j using a symmetric block cipher such as AES [12] or Blowfish [16] resulting in $\{u_j\}_{K_j}$.
4. Distributes $\{u_j\}_{K_j}$ in the pre-hospital environment.
5. Generates a new GUID I_{j+1} and a new encryption key K_{j+1} for the next update u_{j+1} to the record.
6. Generates the read-capability $(I_{j+1}, \{I_j, K_j\}_{K_{j+1}})$ and distributes it in the pre-hospital environment.
7. Generates the update-capability $(I_{j+1}, \{K_{j+1}\}_{K_R})$ and adds it to the patient's record in the EHR system.
8. Creates the patient's token (I_{j+1}, K_{j+1}) .

Note how we store the identity I_{j+1} and the encryption key K_{j+1} of the *next* update to the record on the patient token, instead of those of the current version I_j . The identity I_j and encryption key K_j of the current version are placed in the read-capability $(I_{j+1}, \{I_j, K_j\}_{K_{j+1}})$ that is encrypted with the encryption key K_{j+1} . The key K_{j+1} is subsequently placed on the patient's token while the read-capability is replicated to the health workers. The main reason for this is to minimize what we need to have in a patient's token. In this way, we avoid storing two identities and two keys on the same token; i.e. one for the current version and one for the update. The extra level of indirection is also useful to decouple the GUID present in a patient's token from the one of his record and for revocation purposes as we will see in Section 4.5 and Section 5.

4.3 Access upon rendezvous

When a health worker meets a patient in the pre-hospital environment, he takes the token (I_{j+1}, K_{j+1}) from this latter and uses it on to access the medical record of the patient as follows:

1. Fetches the read-capability $(I_{j+1}, \{I_j, K_j\}_{K_{j+1}})$ using the GUID I_{j+1} found in the patient's token.
2. Decrypts the read-capability $\{I_j, K_j\}_{K_{j+1}}$ using key K_{j+1} from the token to retrieve GUID I_j and key K_j .
3. Uses I_j from step 2 to fetch the encrypted current version of the record $\{u_j\}_{K_j}$ from the record's database.
4. Uses K_j from step 2 to decrypt $\{u_j\}_{K_j}$ and read u_j

Assuming here that the health worker generates new data, if not only the information that he treated the patient in question, this must be tied to the patient's medical record:

1. Encrypts the newly generated data u_{j+1} with the encryption key K_{j+1} from the patient's token,
2. Stores result $(I_{j+1}, \{u_{j+1}\}_{K_{j+1}})$ in the EHR system.

The new update u_{j+1} does not have to be distributed in the pre-hospital environment. It can therefore simply be sent or (physically) transported to the EHR system. There it can be tied to (and possibly included in) the medical record of the patient in the EHR system such that those having access to the medical record will be granted access to the update.

4.4 Update verification

The new encrypted update u_{j+1} can be verified and accessed by all that know the key K_R associated with the record as a whole, using the following steps:

1. Fetches $(I_{j+1}, \{u_{j+1}\}_{K_{j+1}})$.
2. Fetches the update-capability $\{K_{j+1}\}_{K_R}$ for I_{j+1} .
3. Uses K_R to decrypt the update-capability.
4. Decrypts/verifies the update using resulting key K_{j+1} .

The system must implement write-once semantics for updates. Once an update with identity I_{j+1} has been received and verified, further updates with that specific identity should be discarded (even if found encrypted with K_{j+1}). A minimal implementation of this would keep a log of the identities of new updates. This effectively revokes write access using the patient token. Note that this does not mean that it is or should be impossible to add multiple pieces of data possibly by different health workers. But at a certain point in time, when the patient arrives at the hospital for example, the update should be committed and no further additions or other updates should be accepted.

4.5 Revocation

Each update made in the pre-hospital environment requires the use of a different token. When an update is received by the hospital EHR system, the GP of the patient is notified about it and must issue a new token. If no new relevant data has been added, this involves only the final 3 steps of Section 4.2.

Once the new token, say (I_{j+2}, K_{j+2}) , has been issued it is appropriate to revoke the old one (I_{j+1}, K_{j+1}) which may

not have been securely destroyed by the patient. Instead of leaving it the sole responsibility of patients to destroy old tokens, our system can actively render an old token useless by removing the old read-capability $(I_{j+1}, \{I_j, K_j\}_{\kappa_{j+1}})$ associated with the token. This makes K_j unavailable in the pre-hospital environment which effectively revokes read access even in the case that the encrypted record is still available. The old token (I_{j+1}, K_{j+1}) still enables access to the encrypted update u_{j+1} made with it, but this latter can easily be removed from the pre-hospital environment once it has been received and its contents copied by the hospital EHR system.

We believe that removing the read-capabilities instead of the encrypted record represents a more efficient solution because the combined size of the read-capabilities is relatively small and it grows linearly (and predictably) with the number of patients. For example, in a typical implementation where identifiers and keys are 128 bits in size, the read-capability can be made as small as 32 bytes per patient. A 1GB memory card would then be enough for over 30 million patients. This makes it possible to implement automatic revocation procedures. For example, one can store the read-capabilities on a battery-powered volatile memory device that ‘self-destructs’ within a day, leaving health workers to get a new fully charged one at the start of each working day.

5. DISCUSSION

We believe that our rendezvous-based access control represents a practical solution for access to medical data in the pre-hospital environment. It provides suitable protection against the most important threats against medical records while providing good availability to this data at the same time. It is practical due to its conceptual simplicity and its simplicity of implementation and management. Our solution can also grow with the increasing (storage) capabilities of mobile equipment; technological advances simply means replication closer to the health worker becomes feasible without requiring changes to our design.

Because our patient tokens do not identify patients, we must consider the problem that the token found on or near an incapacitated patient may simply not refer to him. To counter this, a photo or other personal information like age, gender and length can be added to the encrypted medical record for the pre-hospital environment so that health workers can check these with the patient they are treating. This raises an obvious trade-off: the more information the record contains to identify the patient, the less anonymous becomes the record. We believe however that the solution space for this issue is large enough to allow for a good compromise.

In an implementation where identifiers and keys are 128 bits in size, the patient token only needs to encode 32 bytes of data. A token can then even be implemented as a simple and robust barcode. Because of its small size its encoding format can provide additional redundancy to increase the chances that it is be readable even when badly damaged. Barcodes can simply and cheaply be replicated by the patient himself (by scanning or photocopying them) so that he can protect himself against misplacement, loss and destruction of a single copy. Barcodes (or other visual codifications for the access token) have the advantage that a health worker or even the patient himself can read such a token even using off-the-shelves technology like mobile phones with built-in cameras (see e.g. [14]).

The use-once semantics of our tokens do not imply that a medical record can be updated only once in the pre-hospital environment. All data produced for a patient being treated can be collected and encrypted with the same key from the patient token. But once the treatment in the pre-hospital environment ends and the patient arrives at the hospital, this key can and should not be used again. This means that multiple health workers can add their data to the ‘single’ allowed update per token. For integrity protection and auditing purposes, health workers can be equipped with the means to digitally sign the data they produce. This is, however, a function that is complementary to our access control mechanisms.

6. CONCLUSIONS

Patient privacy and patient consent often seem or are presented as conflicting with the great benefits that widely available electronic health records would bring, and that a tradeoff between the two must always be made. The BMA security policy model already documented how patient privacy and patient consent practices can be preserved when medical data is being made available in electronic form. We have presented a security policy model summarized by 4 principles, that aims to do the same for access to medical data in the pre-hospital environment.

We show that, with fairly simple means, patient privacy can be protected and patient consent be supported while making medical data available when and wherever needed. We presented a cryptographic access control mechanism that implements our security model. We termed this mechanism *rendezvous-based access control* because it requires physical co-locality of a health worker and a patient in order to access that patient’s medical data. This property prevents aggregation of access to medical data, and with it protection against the worst threat against patient privacy. This is achieved without requiring patients to carry their own medical data around (on a smartcard for example).

An important property of our access control mechanism is its simplicity. Simplicity is often regarded as a prerequisite for security, but our mechanism is also simple in the sense that it places low demands on the tokens carried by patients. Simple tokens are more robust and easier to manage leading to a more trustworthy, manageable and cheaper system.

7. ACKNOWLEDGEMENTS

This work has been supported by the Norwegian Research Council through the PENNE project (project nr. 158596/431). Thanks to Tage Stabell-Kulø and the referees for their useful feedback.

8. REFERENCES

- [1] R. Anderson. Security in clinical information systems. Published by the British Medical Association, 1996.
- [2] R. J. Anderson. NHS-wide networking and patient confidentiality. *BMJ*, 311(6996):5–6, 1995.
- [3] R. J. Anderson. Clinical system security: interim guidelines. *BMJ*, 312(7023):109–111, 1996.
- [4] R. J. Anderson. A security policy model for clinical information systems. In *Proceedings of the 1996 IEEE Symposium on Security and Privacy*. IEEE Computer Society, 1996.

- [5] T. Beale, S. Heard, D. Kalra, and D. Lloyd. openEHR Architecture Overview. <http://www.openEHR.org>, Mar 2006.
- [6] S. B. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in a partitioned network: a survey. *ACM Comput. Surv.*, 17(3):341–370, 1985.
- [7] M. A. C. Dekker and S. Etalle. Audit-based access control for electronic health records. *Electron. Notes Theor. Comput. Sci.*, 168:221–236, 2007.
- [8] J. Dennis and E. V. Horn. Programming semantics for multiprogrammed computations. *Communications of the ACM*, 9(3):143–155, Mar. 1966.
- [9] Department of Defense. DoD 5200.28-STD: Department of defense (DoD) trusted computer system evaluation criteria (TCSEC), 1985.
- [10] M. Eichelberg, T. Aden, J. Riesmeier, A. Dogac, and G. B. Laleci. A survey and analysis of electronic healthcare record standards. *ACM Comput. Surv.*, 37(4):277–315, 2005.
- [11] P. G. Goldschmidt. HIT and MIS: implications of health information technology and medical information systems. *Commun. ACM*, 48(10):68–74, 2005.
- [12] V. R. Joan Daemen. *The Design of Rijndael: AES - The Advanced Encryption Standard*. Springer Verlag, 2002.
- [13] B. Lampson. Protection. In *Proceedings of the Fifth Princeton Symposium on Information Sciences and Systems*, pages 437–443, Princeton University, Mar. 1971. Reprinted in *ACM Operating Systems Review*, 8, 1, January 1974, pp. 18–24.
- [14] J. M. McCune, A. Perrig, and M. K. Reiter. Seeing-is-believing: Using camera phones for human-verifiable authentication. In *SP '05: Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 110–124, Washington, DC, USA, 2005. IEEE Computer Society.
- [15] B. Neuman. Proxy-based authorization and accounting for distributed systems. In *Proceedings of the 13th International Conference on Distributed Computing Systems*, pages 283–291, Pittsburgh, May 1993.
- [16] B. Schneier. Description of a new variable-length key, 64-bit block cipher (blowfish). In *Fast Software Encryption, Cambridge Security Workshop*, pages 191–204, London, UK, 1994. Springer-Verlag.
- [17] A. Tanenbaum, S. Mullender, and R. van Renesse. Using sparse capabilities in a distributed operating system. In *Proceedings of the 6th International Conference on Distributed Computing Systems (ICDCS)*, pages 558–563, Washington, DC, 1986. IEEE Computer Society.
- [18] M. Wilkes and R. Needham. *The Cambridge CAP computer and its operating system*. Operating and Programming System Series. Elsevier, North Holland, 1979.