Towards Security Policy and Architecture for Managing Implantable Medical Devices

Ram Krishnan and Eugene John
Dept. of Electrical and Computer Engineering
University of Texas at San Antonio
Email: Ram.Krishnan@utsa.edu and Eugene.John@utsa.edu

Manoj Panday
School of Medicine
University of Texas Health Science Center at San Antonio
Email:panday@uthscsa.edu

SAM Track: Security Management

***Abstract*—Implantable cardiac rhythm management devices (CRMDs) such as permanent pacemakers and internal cardioverter defibrillators (ICDs) utilize embedded computers and radios to monitor chronic disorders and treat patients. Life-saving devices like ICDs, for instance, include pacemaker technology and are designed to communicate wirelessly with a nearby external device programmer (EDP) that can remotely read data and change settings without the need for surgery. An ICD implanted in a patient can sense a rapid heartbeat and administer an electric shock to restore normal heart rhythm. It is has been shown that current ICDs in the market can be reverse engineered and are prone to software radio-based attacks. The ICDs can be remotely disabled or be made to administer an electric shock at random. Existing defense mechanisms include a simple cryptographic approach where a symmetric-key based challenge-response protocol is used between the ICD and an authorized EDP. This approach does not scale. In the real world, large scale deployment and management of shared key material amongst various entities such as CRMDs, EDPs, hospitals, clinics, and ambulances is a major issue. In this paper, we investigate security policy issues applicable to the CRMD ecosystem and issues for architectures that enforce the policy. Given the nature of this domain, these solutions will need to balance security, privacy and risk. For instance, an unauthorized EDP may need to issue a command to the ICD in emergency situations.**

***Index Terms*—Security Policy, Architecture, Key Management, Implantable Medical Devices**

I. Introduction

This paper is motivated by the need to provide security and privacy for various cardiac rhythm management devices (CRMDs) that are being deployed in the order of millions in the market today. Specifically, the CRMDs that are of interest are those that can perform some computation to provide patient data to medical personnel and/or administer some type of treatment. A well-known example of such a device is the implantable cardioverter defibrillator (ICD). ICDs utilize embedded computers and radios to monitor chronic heart disorders and treat patients. They include pacemaker technology and are capable of wireless communication with a nearby external device programmer (EDP). EDPs can wirelessly read patient data and change settings without the need for surgery. An ICD implanted in a patient can sense a rapid heartbeat and administer an electric shock to restore normal heart rhythm.

Figure 1 shows a chest x-ray of a patient with a dual chamber ICD implanted via the left subclavian vein. An EDP can be used to configure such an ICD in a patient. The configuration could dictate under what conditions the ICD should activate and regulate heart rhythm by administering corrective electric shocks. An ICD can communicate with the EDP when a magnetic field is generated in its vicinity which closes a switch in the ICD. Subsequently, the EDP can be used to perform diagnostics, read and write private patient data and modify therapy settings.

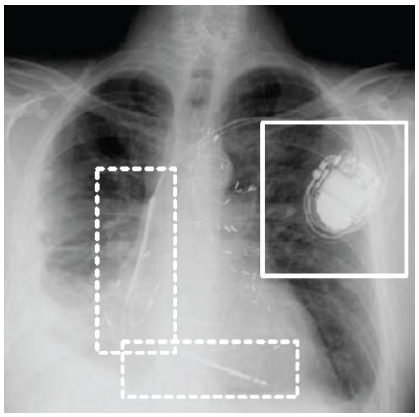

Fig. 1. Chest x-ray of a patient with a dual chamber ICD implanted via the left subclavian vein (source: [1]).

Figure 2 shows the timeline of a sample communication between an EDP and ICD. As shown, after the application of a magnetic field, the EDP can query the ICD for telemetry and patient data and issue commands to change its configurations. The protocol is not cryptographically protected. In [1], the authors show how such a protocol can be reverse engineered using standard lab equipments. The ICD can be made to respond to unauthorized EDPs, and its configurations can be changed with relative ease. Thus, an attacker could employ an unauthorized EDP to administer an electrical shock.

Motivation The current solution involves sharing a secret key (symmetric key) between an ICD and authorized EDPs. This allows one to build a cryptographic strength protocol where the ICD can throw a challenge to any EDP that wants to interact with it. Only if the EDP provides a correct response that is based on the shared secret key, the ICD would accept further commands.

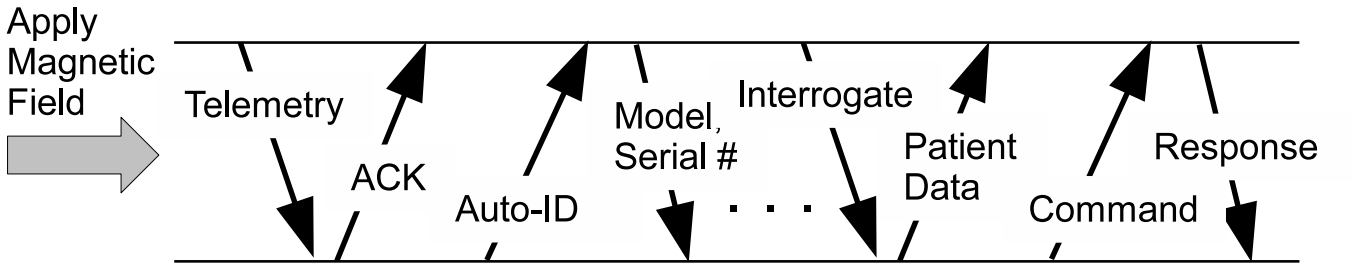

Fig. 2. An example interaction between an ICD and EDP. The interaction is not cryptographically protected. This allows a malicious party to easily issue unauthorized commands to the ICD.

In the real world, this approach does not scale. Large-scale deployment of shared key material amongst various entities such as CRMDs, EDPs, hospitals, clinics, and ambulances poses an unacceptable amount of risk for key compromise At the same time, it would be naive to expect that keys be shared only between the patient's ICD and the corresponding doctor's EDP. Due to nature of this domain, inaccessibility to an ICD in times of emergency could be fatal. Consider a situation where a physician who is new to the patient responds to an emergency situation but is unable to access the ICD since his/her EDP does not have the shared secret key.

Key Challenges The key challenge is to develop intuitive, yet effective and scalable models for managing security keys amongst a large number of disparate entities that may belong to different administrative domains. For instance, the patient may consult doctors from different hospitals, various nurses may work with the doctors and patients may be treated by emergency response personnel who may not have prior agreement to access the patient's ICD.

This paper will investigate the requirements of a scalable framework for secure, reliable and risk-aware cryptographic key management for CRMDs and its ecosystem. The techniques developed will need to balance security, privacy and risk due to the sensitive nature of this domain. Security is concerned about integrity of commands exchanged between external control devices and CRMDs, in addition to the confidentiality of data exchanged. Privacy is concerned about appropriate access of collected sensitive data by authorized parties. Risk is concerned about balancing security and privacy in scenarios where the mission is more important (for example, in emergency situations, an unauthorized EDP may need to issue a command to the ICD).

Relevance to Real-World Threat Although no known attack has been reported, implantable medical devices are only increasing in sophistication. For instance, CRMDs can now communicate over a much longer range than those developed in the late 90s. Thus, this tremendously increases the threat surface and attack practicality. Furthermore, the knowledge of such vulnerabilities is psychologically troubling to patients that undergo implantation. If not maliciously, there is always room for accidental unauthorized modification of ICD settings and/or the ability to read private patient data. Thus this paper addresses an important problem that has broad impact.

Prior Work To the best of our knowledge, prior work in this area does not address this problem either directly or addresses them inadequately. Existing solution to this problem is to simply share a secret key between every ICD and EDP [1]. While conceptually acceptable, this does not scale to real-world scenarios as argued earlier. The challenges involved in manufacturing and providing safe computer-based medical treatments in the presence of unintentional failures have been investigated. However, the proposed work addresses such challenges in the context of intentional failures due to passive and active attacks by a malicious entity. Securing patient data in databases has been studied in the past [7]. Pervasive healthcare security including medical sensor security has been investigated in [10]. There is also ample literature on wireless security in low-power environments. See for example: [2], [8], [11].

II. Approach for CRMD Management

This section briefly provides an overview of the technical approach of the proposed solution. From a methodology standpoint, security policy issues will be clearly separated from security enforcement issues. The policy model is concerned about "what" needs to achieved while the enforcement model is concerned about "how" the policy can be realized. This allows one to address issues at an appropriate level of abstraction. The goal of this paper is to develop effective key management techniques for CRMDs. (From here on, we will use the more general term CRMD, for implantable *medical devices*, instead for ICD.) Thus various policy models need to be specified so that appropriate key management techniques can be developed at enforcement level.

The policy issues concerning this problem include developing models to specify information sharing policies amongst various entities that are involved in a CRMD ecosystem. For example, it may be the case that a specific CRMD implanted in a patient can communicate with an EDP only in the presence of his/her doctor unless it is an emergency. In another scenario, the policy could be that the CRMD can only communicate with a pre-authorized set of EDPs regardless of who employs them. Yet another policy could specify authorization with respect to the personnel that employ them instead of the identity of the EDPs that are employed. Note that such varying policies require different key management techniques.

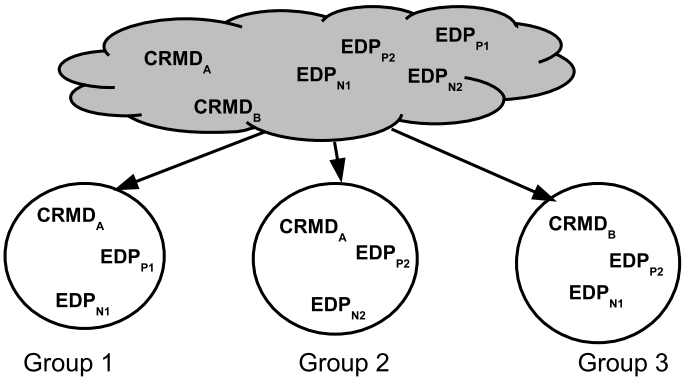

Fig. 3. Various groups can be formed between CRMDs and ICDs from the global set. The groups indicate which EDPs can communicate with which ICDs. Group 1 is formed between a CRMD for patient A and her physician P1 and nurse N1. The same CRMD can belong to another group 2.

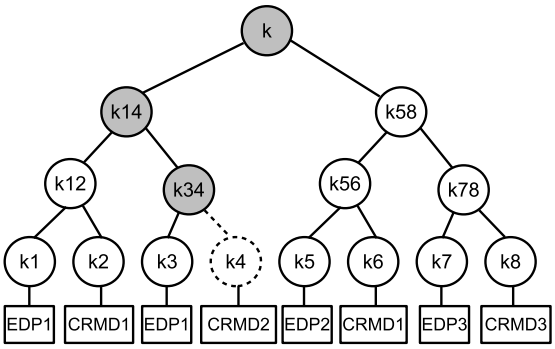

Fig. 4. A Logical Key Hierarchy [12] for group key management. The shaded circles indicate the keys that need to be changed when a new member (indicated by boxes) CRMD2 joins a group in which EDP1 belongs.

The policy models will build upon our preliminary work in abstract policy models for Group-Centric Secure Information Sharing (g-SIS) [3], [9], [4], [6], [5]. In g-SIS, a security policy can be specified for entities that form a group. In our scenario, a group can be formed between a CRMD and the EDPs that may communicate with that CRMD. Furthermore, a CRMD can belong to multiple groups. For example, a CRMD can belong to a group of physician EDPs while also belonging to another group of nurse EDPs. Furthermore, the CRMD can also belong to yet another group of EDPs that will be used by emergency response personnel. A major challenge is in managing such a large number of CRMDs and EDPs and their group memberships. In the prior work in g-SIS, formal models for a single group has been developed. We need to develop models to manage multiple groups that are specific to this domain and those that can specify inter-relationship between groups. For example, we may specify that a CRMD can be a member of group A as long as it is also a member of group B. Thus if membership in group B ceases, the CRMD will lose membership in group A as well.

Figure 3 shows an example scenario. Membership of EDPs in a group indicates that they can communicate with the ICDs in that group. Various permissions can be specified in the group. For example, the physician and nurse in group 1 can read patient data and update ICD's settings while those in group 2 may only read data.

III. Key Management

Following concrete and intuitive models under which various information sharing policies can be specified using formal logic,[1] various enforcement models that meet the policy specification can be developed. Since the policy models specify information sharing at the group level, group key management techniques can be employed. Specifically, various techniques will be employed for scalable group-centric key management including centralized, decentralized and distributed approaches.

In all of these approaches, the critical problem is in handling group creation and membership changes. For example, when an EDP is taken out of a group, the group key for all the remaining members should be updated. In centralized approach, we assume that a group manager exists for each group that will manage the keys shared amongst various entities. In decentralized approach, more than one group manager may exist. In the distributed approach, there is no group manager. Every member can both be a manager and a regular member.

We discuss how logical key hierarchy (figure 4) can be employed to manage keys in this context. In the figure, the boxed nodes indicate entities that need to securely exchange messages. The rest of the figure forms a key graph. Each node contains a symmetric key. Each boxed node stores every key that is in a path from itself to the root. For example, CRMD1 stores keys k2, k12, k14 and k. The key in the leaf node indicates a unique key for each boxed node. Given this setting, data can be securely exchanged between CRMD1 and EDP1 using the key k12. This is because both the entities have knowledge of k12. Similarly, data can be securely exchanged between EDP1 and CRMD2 using k34. Note that data protected using k14 is readable by all the members that fall under that subtree—specifically EDP1, CRMD1 and CRMD2. Finally, data encrypted using k can be decrypted by every member in the system: EDP1, EDP2, CRMD1 and CRMD2.

Such a key graph is typically stored in a server that manages the keys for all participating entities in the CRMD ecosystem. Note that the key graph can be fully customized to accommodate the needs of a specific scenario with respect to group level sharing. For example, by adjusting the degree of each node and the height of the tree, several groups and subgroups can be formed for secure communication.

In practice, under normal circumstances each entity may use its group key to communicate with other entities in the same group. In case of an emergency, if an EDP does not belong to a specific group but still needs to communicate with a CRMD in that group, it can use a key at a higher level. For

[1]For example, First-Order Linear Time Temporal Logic was used in the preliminary work on g-SIS [3].

example, under emergency situations, CRMD3 can be used to communicate with EDP2 using k58.

Membership management is an important issue. For example, a new member, say EDP1, may need to be added to the group that is managed using key k78. Also, a member may leave a group. Different policies may be required in such scenarios. Forward secrecy ensures that after a member leaves a group, it cannot read any new data exchanged the remaining members. Backward secrecy ensures that when a new member joins a group, it is only able to read new data exchanged between the group members and not any data exchanged before the member joined. Thus to ensure forward secrecy, whenever a member leaves, the remaining group members need to change the group key. Similarly, to ensure backward secrecy, whenever a member joins, the group key needs to updated to ensure that the new member is unable to access past data.

Let us consider member management in logical key hierarchy. In figure 4, when CRMD2 joins the group in which EDP1 belongs, a new node with a new unique key (k4) for CRMD2 is created in the key graph. (We assume that the server has prior knowledge of this unique key.) To guarantee backward secrecy, every key in the path from k4 to the root needs to be updated. Thus keys k, k14 and k34 need to be updated. Note that updated versions of these keys (say k', k14' and k34') in the key graph need to share with other devices in the graph. As seen in the figure, this means that EDP1 needs to get k34', k14' and k' and CRMD1 needs to get k14' and k'. This can be achieved by encrypting the new keys with appropriate old keys. For instance, the new keys can be encrypted using k3 to be sent securely to EDP1 and can be encrypted with the old k12 to be sent securely to all members in that subgroup. Similarly, when EDP1 leaves the group with k34, the keys k, k14 and k34 need to be updated and shared with members that are affected by this membership change.

Thus membership management can be efficiently handled in logical key hierarchy. Specifically, when one member leaves, it does not require key updates to every other member.

IV. Conclusion and Future Work

We investigated group-based security policies and corresponding key management strategies using logical key hierarchy for securely managing large-scale implantable medical devices such as ICDs. We strongly believe that this paper serves as starting point for research in this important area in a number of different avenues. First, a major challenge in implementing this approach for the CRMD ecosystem is ensuring key updates are carried out in a timely manner. This has a number of practical challenges. For instance, there are always periods of time during which a few entities will have outdated keys since they may not be always connected to the server.

Next, a number of other key management strategies can be explored. Logical key hierarchy is a centralized approach where the keys are managed by a single server. We plan to explore decentralized and distributed approaches to key management for the CRMD ecosystem. In the decentralized approach, there are more than one server to manage key updates. In a distributed approach, there is no specific server and client. Every entity is both a server and a client. We plan to investigate a hybrid approach for key management that is both practical and effective in terms of minimizing the time period during which members have outdated keys.

We also plan to explore different policy models and corresponding key management techniques for enforcement. For example, a conditional membership policy may require that a member's membership in a group is contingent upon its membership in another group. Similarly, membership could be hierarchical in which membership in a group automatically guarantees membership in other groups that are dominated by the joining group.

References

[1] D. Halperin, T.S. Heydt-Benjamin, B. Ransford, S.S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W.H. Maisel. Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In *Security and Privacy, 2008. SP 2008. IEEE Symposium on*, pages 129 –142, may 2008.

[2] Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: A link layer security architecture for wireless sensor networks. In *ACM SENSYS'04*. ACM, 2004.

[3] Ram Krishnan, Jianwei Niu, Ravi Sandhu, and William H. Winsborough. Group-centric secure information-sharing models for isolated groups. *ACM Trans. Inf. Syst. Secur.*, 14:23:1–23:29, November 2011.

[4] Ram Krishnan, Ravi Sandhu, Jianwei Niu, and William Winsborough. Towards a framework for group-centric secure collaboration. *Proceedings of IEEE International Conference on Collaborative Computing*, 2009.

[5] Ram Krishnan, Ravi Sandhu, Jianwei Niu, and William Winsborough. A conceptual framework for group-centric secure information sharing. *ACM Symposium on Information, Computer and Comm. Security*, March 2009.

[6] Ram Krishnan, Ravi Sandhu, Jianwei Niu, and William H. Winsborough. Foundations for group-centric secure information sharing models. In *Proc. of ACM symposium on access control models and technologies*, 2009.

[7] M. Meingast, T. Roosta, and S. Sastry. Security and privacy issues with health care information technology. In *Engineering in Medicine and Biology Society, 2006. EMBS '06. 28th Annual International Conference of the IEEE*, pages 5453 –5458, 30 2006-sept. 3 2006.

[8] Adrian Perrig, Robert Szewczyk, J. D. Tygar, Victor Wen, and David E. Culler. Spins: security protocols for sensor networks. *Wirel. Netw.*, 8:521–534, September 2002.

[9] Ravi Sandhu, Ram Krishnan, Jianwei Niu, and William Winsborough. Group-centric models for secure and agile information sharing. In *Computer Network Security: 5th International Conference, on Mathematical Methods, Models, and Architectures for Computer Network Security (MMM-ACNS 2010)*, Lecture Notes in Computer Science. Springer, 2010.

[10] K.K. Venkatasubramanian and S.K.S. Gupta. Security for pervasive health monitoring sensor applications. In *Intelligent Sensing and Information Processing, 2006. ICISIP 2006. Fourth International Conference on*, pages 197 –202, 15 2006-dec. 18 2006.

[11] S. Warren, J. Lebak, Jianchu Yao, J. Creekmore, A. Milenkovic, and E. Jovanov. Interoperability and security in wireless body area network infrastructures. In *Engineering in Medicine and Biology Society, 2005. IEEE-EMBS 2005. 27th Annual International Conference of the*, pages 3837 –3840, 2005.

[12] Chung Kei Wong, Mohamed Gouda, and Simon S. Lam. Secure group communications using key graphs. *IEEE/ACM Trans. Netw.*, 8:16–30, February 2000.