A Lightweight Security Protocol for Ultra-low Power ASIC Implementation for Wireless Implantable Medical Devices

Saied Hosseini-Khayat
Department of Electrical Engineering
Ferdowsi University of Mashhad, Iran
Email: shk@ieee.org

Abstract— **The newest generation of Implantable Medical Devices (IMDs) employs wireless communication with a nearby base station in order to provide better treatment and monitoring of the patients. However, a wireless connection opens a host of potential security threats to the privacy and safety of patients. This paper proposes a lightweight security protocol providing authentication and confidentiality to wireless energy-limited IMDs that operate on small energy sources such as a battery for many years. Adding security features to these devices can impose an unacceptable overhead. The protocol presented here employs lightweight encryption and is suitable for implementation on ultra-low power ASIC chips.**

I. Introduction

An emerging class of implantable medical devices (IMDs) employs wireless technology to allow patients to move around their living places freely while receiving continuous remote monitoring and treatment for very extended periods of time. But this convenience comes with its potential hazards. Recent research [3], [4], [5], [6], [8], [11] has triggered the industry and research community to think more seriously about the security and privacy implications of wireless connection. While the issues related to medical safety of IMDs seem to receive due attention, the security and privacy issues and solutions are yet to be explored and developed.

The risk of attack on patients with IMD may seem negligible at present, but the consequences of not having appropriate safeguards in place can be devastating. A team of researchers recently demonstrated [9] that using an inexpensive software radio, it was possible to intercept signals sent from a certain model of a commercial cardiac IMD. The group, who had no access to the proprietary design of the device, was able to obtain information about a hypothetical patient, including name, diagnosis, date of birth and medical ID number. Researchers could also determine the make and model of the device and access real-time electrocardiogram results as well as data on the hypothetical patient's cardiac activity. The team then mounted several attacks. Researchers were able to turn off the therapy settings stored in the implantable device, making it incapable of responding to dangerous cardiac events. Additional commands were delivered, resulting in the delivery of a shock that could induce a potentially lethal arrhythmia.

The purpose of the present paper is to propose a lightweight protocol that provides authentication and confidentiality to wireless battery-operated IMDs that operate for many years without the possibility of battery replacement. The protocol is suitable for implementation on ultra-low power ASIC chips. In Section II, we discuss the security requirements of IMDs and enumerate the design constraints. In Section III our lightweight security protocol is presented. In Section IV, we discuss our design rationales.

II. IMD Design Requirements

The topmost security goals in the context of IMDs are the following [4]:

- *Privacy*: An attacker should not be able to exploit the properties of a device to read private information, such as patient's name and records of bio-signals that are stored in or transmitted from an IMD.
- *Integrity*: An unauthenticated person should not be able to reprogram the device settings or issue unauthorized treatments. The person can be a malicious attacker or even the patient him/herself.
- *Availability*: An attacker should not be able to deactivate a device entirely and render it ineffective. This is considered as a denial-of-service attack aimed at draining the energy resources of an IMD too quickly.

This paper aims at achieving the first two of the above goals. The computer security researchers have recently started to develop new security mechanisms customized for wireless IMDs [5], [8], [12]. One should note that many standard data security algorithms and protocols (such as IPSec, AES and SHA-x) that work effectively in other application areas require too much processing and do not fit within the tiny energy budget of an IMD. The closest application area is sensor networks. However, IMDs are more demanding in that they perform continuous life-critical monitoring and treatment and are expected to last much longer time (up to 10 years or more).

An IMD contains electronic circuits that perform data processing and control functions on an extremely small energy budget (e.g., a small ion-lithium battery storing about 3000 joules of energy) for a very extended period of time (e.g., 10 years). Therefore, architecture and circuit design for an IMD must be extremely conservative in terms of energy consumption. As a result, an IMD can only perform ultra-lightweight security algorithms that are acceptably strong.

978-1-4244-9444-6/11/$26.00 ©2011 IEEE

Due to its limited energy, a wireless IMD can only communicate with a base station located in its close vicinity (up to a few tens of meters). There are two mode of communication for an IMD:

1) *Receive Mode*: An authorized base station sends a message containing a command to the IMD. This command can be of device configuration type or of query type. In either case, the command (a certain bit string) is encrypted and encapsulated in a predefined packet format and transmitted by the base station to the IMD. The IMD must then decrypt and verify authenticity of each received packet before interpreting and acting upon the command.
2) *Transmit Mode*: An IMD sends a message to an authorized base station. This message can be a response to a query made by the authorized base station or can be a piece of telemetry bio-data. The message is a bit string that is encrypted and encapsulated in a predefined packet format and sent over the RF channel. On receiving each packet, the base station must decrypt and verify authenticity of the packet.

In designing ASIC chips for an IMD (which performs biomedical signal processing as well as cryptographic processing), one should note that in some applications, power consumption is the dominant limitation while speed and area do not present severe constraints. There is also a consensus among electronic designers that crypto-processing in software consumes more energy than performing the same task in dedicated optimized hardware. Therefore, our protocol is aimed at ASIC implementation.

III. Proposed Protocol

Now we present our lightweight protocol that is designed for ultra-low power ASIC implementation. It meets the following security objectives: (a) All messages between sender and receiver must be confidential. No third party not having a shared secret key should be able to read those messages. (b) All messages originating from an unauthorized sender must be recognized and rejected. (c) All messages originating from an authorized sender but modified by a third party must be recognized and rejected. (d) All messages originating from an authorized sender and not modified but replayed by a third party must be recognized and rejected.

We make the following assumptions in our protocol:

a) A secret key (denoted K) is shared between an IMD and its designated base station. One way the secret key can be shared is the following: At the time of manufacturing, a random secret key is generated and is written into a small non-volatile memory inside the IMD. The same key is safely sealed and sold along with the IMD package. At the time of implanting, the seal is broken by a trusted healthcare professional. The key is then inserted into the designated base station. The key document is also handed to the patient or his/her healthcare professional who will keep it secret.

b) The encryption algorithm employed in the protocol is not broken and can safely withstand the intended type of attackers. The center piece of our protocol is one of the ultra-lightweight block ciphers, such as PRESENT-80 [1] or KATAN [2]. These ciphers have been designed for implementation on hardware with extremely limited resources. They use a key size of 80 bits which provides sufficient security as long the encryption algorithm is not broken. They operate on 64-bit data blocks. The choice of 64 and 80 bits for block and key size especially suits the resource-constrained environment of IMDs because larger sizes increase the complexity of the cipher which negatively impact their power consumption.

c) Each IMD has a unique serial number (denoted S) that makes it unique among all IMDs using the same protocol. A serial number of 32 bits seems to be sufficiently large for the purpose. This number can be written into the IMD's non-volatile memory at the time of manufacturing. It is also written on a document accompanying the IMD. The number is entered into the designated base station by a trusted healthcare professional. The serial number S is not required to be kept secret.

d) The IMD and its designated base station each have a counter (denoted A, B, respectively) that are initially set to zero at the time of installation. This counter is employed to prevent replay attacks and it should be large enough not to roll over during the intended lifetime of the IMD. Our suggested size for A and B counters is 32 bits as a 32-bit counter would take over 10 years before it rolls over if approximately 49000 messages is sent per hour on the average. This number should not ever happen in a battery-operated wireless IMD as it will quickly drain the battery.

e) The messages between an IMD and its designated base station are in the form of data packets. The packets contain commands and data whose meaning and format is irrelevant to our security protocol (adhering to the idea of protocol layering). Since our proposed protocol uses a 64-bit block cipher, we assume that the messages are either at most 64 bits long or, if larger, they have been already broken down into 64-bit chunks using a suitable segmentation protocol.

In describing our protocol, we use the following notations:

- IMD and BASE will denote the IMD and its designated base station whose communications are to be secured.
- $\{M\}_K$ will denote a message that is encrypted with the key K using a pre-specified encryption algorithm.
- $X\&Y$ will denote the concatenation of two bit strings X and Y.
- *Interleave*(X, Y) is the function that bit-interleaves two equally-sized bit strings $X = x_1x_2 \ldots x_k$ and $Y = y_1y_2 \ldots y_k$ and returns the bit string $x_1y_1x_2y_2 \ldots x_ky_k$.
- *Split*(X) is the function that splits a bit string $X = x_1x_2 \ldots x_{2k}$ and returns two bit strings $X_1 = x_1x_2 \ldots x_k$ and $X_2 = x_{k+1}x_{k+2} \ldots x_{2k}$.

A. *Protocol Description*

In this section, we describe our protocol in full detail. The protocol described here covers the receive mode of an IMD. The counters A (at IMD) and B (at base station) are set to zero at installation time.

1) BASE has a message X of size m ($m \leq 64$) bits. If $m < 64$, the message is padded with a random bit string to make it 64 bits long. If $m > 64$, the message is segmented into 64-bit chunks.
2) BASE increments its counter: $B = B{+}1$. (Or optionally, BASE increments its counter by a small random integer in $\{1, 2, \cdots, \alpha\}$, where α is a small integer).
3) BASE produces the message $M = \textit{Interleave}(X, S\&B)$.
4) BASE produces the message $[M_1, M_2] = \textit{Split}(M)$.
5) BASE sends the message $\{M_1\}_K \& \{M_2\}_K$ to IMD.
6) IMD receives $\{M_1\}_K \& \{M_2\}_K$.
7) IMD decrypts $\{M_1\}_K$ and $\{M_2\}_K$ using the key K to recover M_1 and M_2.
8) IMD un-interleaves $M_1 \& M_2$ to recover X, S, and B.
9) IMD compares the received number S to its own S. If they are identical, it will continue to the next step. Otherwise it drops X and take no more action.
10) IMD compares the received counter value B to its own counter value A. If $B > A$, then IMD sets $A = B$ and it accepts X as a fresh and authentic message. Otherwise, it drops X and takes no action.

Exactly the same protocol can be used in transmit mode, except for a minor modification which is necessary to make the IMD-to-base messages incompatible with the base-to-IMD messages. The modification will be at Step 3 as follows: $M = \textit{Interleave}(X, B\&S)$.

IV. Design Rationales

In this section, we discuss the design rationales behind our protocol.

We do not use a challenge-response type of authentication. That type of protocol has the following steps: (1) a message is sent by the base station to IMD, (2) IMD processes that message and creates a "challenge", (3) IMD sends the challenge to the base station, (4) the base station creates a response to the challenge and sends it to the IMD, (5) IMD processes the response to determine authenticity. Thus this protocol requires two message processing steps and one message transmission by the IMD. Our protocol requires only one message processing and no transmissions by IMD; therefore this is more energy-efficient than challenge-response.

Our protocol does not use a "nonce" as a means to prevent replay attacks. In a nonce-based protocol, the base station should send a random number to IMD with each message in order to uniquify it. The idea is that no nonce should be repeated. To verify that a received nonce has not been used in the past, the IMD must maintain a list of last n nonces, and should compares each newly received nonce to those in the list. This technique requires a nonce memory in IMD and n compare operations. To make replay attack difficult, n should be chosen large. However, memories are known to be consume large amount power. Therefore in our protocol, we do not use nonce technique and instead use only two counters A (in IMD) and B (in base station). Each new message sent by the base station is uniquified by insertion of a new counter value B. The IMD checks to detect a replay attack by comparing its own current counter value A with the received counter value B. The only requirement for acceptance of the message is $B > A$. If an adversary replays an old authentic message, it will be rejected by IMD. Since each message in encrypted with the secret key K that is not known by an adversary, it is impossible for him to forge a message that contains S value in the right field of the message and a B value that is greater than A. This protocol requires only one compare operation and one counter for A in IMD, thus it conserves energy.

Our protocol does not employ a stream cipher, although this class of encryption algorithms usually require less logic resources than block ciphers. The problem with stream ciphers is that the cipher modules at the sending end and receiving end have to be bit-level synchronized. Otherwise, decryption is not possible. If synchronization is lost due to packet loss (which is possible), a synchronization recovery circuit is needed which increases IMD complexity and power consumption. A block cipher, on the other hand, does not need synchronization. Reference [1] reports that the resource requirements of the block cipher PRESENT-80 on an ASIC chip is very minimal compared to full-fledged block ciphers and is comparable to the lightweight stream cipher GRAIN-80 [10].

One method to prevent replay attacks is by using the concept of synchronized *time* at both ends (which is implemented by means of synchronized free running counters). In this method, the sender always time-stamps its messages, and the receiver verifies that a received message has a time-stamp value that is within a predefined offset with respect to its own 'time.' However, time synchronization between sender and receiver is difficult to maintain for a long time, and its recovery adds to IMD complexity. As a result, our protocol avoids using synchronized 'time.' Instead, it uses two loosely coupled counters (A and B) that are maintained in such a way that always $A < B$. This method is robust in the face of packet loss which can de-synchronize two counters that should advance in lock-step.

Optionally, every time the base station increments its counter by a small random integer (instead of just 1). This can enhance security over time, as it becomes increasingly more difficult for an adversary to guess the sender's counter value. The price to be paid for having random increments is a hardware pseudorandom number generator (PRNG) module, such as a small Linear Feedback Shift Register (LFSR). This PRNG does not need to be cryptographically secure since the counter value is transmitted as part of an encrypted message.

The interleaving procedure applied in Step 3 of the protocol spreads the 64-bit message X and the 64-bit string $S\&B$ along two messages M_1 and M_2, 64 bits each. Since a 64-bit block cipher is used, this procedure makes each of the encrypted messages $\{M_1\}_K$ and $\{M_2\}_K$ separately useless.

A dangerous alternative would be to create and send two encrypted messages $\{X\}_K$ and $\{S\&B\}_K$. In that case, an attacker would be able to swap an old $\{X\}_K$ with a new $\{X\}_K$.

Our protocol is designed so that it requires as few hardware modules as possible. Mainly it uses a 64-bit ultra-lightweight block cipher decryptor (e.g., the PRESENT-80 cipher), a padding circuit (if necessary), a 32-bit counter, a 32-bit register, an optional small LFSR, some control logic, and no random-access memory. Memories are known to be large consumers of energy. An ultra-lightweight block cipher such as PRESENT-80 does not need a random access memory. The interleaving and splitting functions consume virtually no logic resources. Minimizing the required logic resources in a low-speed design such as an IMD correlates very well with reducing power consumption. This is especially true if the chip is implemented in deep submicron technology where static power dominates in the overall power consumption at low operating frequencies.

V. Conclusion

This paper presented a security protocol for wireless IMDs that operate on extremely limited energy sources for a very extended period of time. It can provide privacy and authentication and is aimed at ultra-low power implementation on ASIC chips. The protocol employs one of the ultra-lightweight encryption algorithms and requires very few hardware modules for its implementation. Our future step will be to design an ASIC chip that implements the protocol. The actual power measurements, when the chip is fabricated, will determine how effectively this protocol performs with respect to power consumption, speed and area.

Acknowledgment

The author would like to thank Prof. Mohamad Sawan of École Polytechnique de Montréal for his kind support and helpful comments on an earlier draft of this paper while the author was a visiting faculty at Polystim Laboratory.

References

[1] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. B. Robshaw, Y. Seurin and C. Vikkelsoe, "PRESENT: An Ultra-Lightweight Block Cipher," Cryptographic Hardware and Embedded Systems - CHES 2007, Lecture Notes in Computer Science, Volume 4727/2007, pp. 450-466, 2007.

[2] Christophe De Canniere, Orr Dunkelman, and Miroslav Knezevic, "KATAN and KTANTAN A Family of Small and Efficient Hardware-Oriented Block Ciphers," Cryptographic Hardware and Embedded Systems - CHES, 2009, Lecture Notes in Computer Science, 2009, Volume 5747/2009, pp. 272-288, 2009.

[3] Tamara Denning, Kevin Fu, and Tadayoshi Kohno, "Absence makes the heart grow fonder: New directions for implantable medical device security," USENIX Workshop on Hot Topics in Security (HotSec), July 2008.

[4] Tamara Denning, Yoki Matsuoka, and Tadayoshi Kohno, "Neurosecurity: Security and Privacy for Neural Devices," *Neurosurgical Focus*, Vol. 27, July 2009.

[5] Tamara Denning, Alan Borning, Batya Friedman, Brian T. Gill, Tadayoshi Kohno, and William H. Maisel, "Patients, Pacemakers, and Implantable Defibrillators: Human Values and Security for Wireless Implantable Medical Devices," in proceedings of 28th Conference on Human Factors in Computing Systems (CHI), Atlanta, GA, April 2010.

[6] Kevin Fu, "Inside risks, reducing the risks of implantable medical devices: A prescription to improve security and privacy of pervasive health care," *Communications of the ACM*, 52(6):25–27, June 2009.

[7] Paul Gerrish, Erik Herrmann, Larry Tyler, and Kevin Walsh, "Challenges and Constraints in Designing Implantable Medical ICs," *IEEE Transactions on Device and Materials Reliability*, vol. 5, no. 3, september 2005

[8] Daniel Halperin, Tadayoshi Kohno, Thomas S. Heydt-Benjamin, Kevin Fu, and William H. Maisel "Security and Privacy for Implantable Medical Devices," *IEEE Pervasive Computing*, Vol. 7, No. 1, January - March 2008.

[9] Daniel Halperin, Thomas S. Heydt-Benjamin, Benjamin Ransford, Shane S. Clark, Benessa Defend, Will Morgan, Kevin Fu, Tadayoshi Kohno, and William H. Maisel, "Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses," IEEE Symposium on Security and Privacy, May 2008.

[10] Martin Hell, Thomas Johansson and Willi Meier, "Grain: A Stream Cipher for Constrained Environments," eSTREAM, 2006. Currently available at http://www.ecrypt.eu.org/stream/grainp2.html.

[11] William H. Maisel, and Tadayoshi Kohno, "Improving the security and privacy of implantable medical devices," *New England Journal of Medicine*, April 2010.

[12] Kasper B. Rasmussen, Claude Castelluccia, Thomas Heydt-Benjamin, and Srdjan Capkun, "Proximity-based Access Control for Implantable Medical Devices," CCS09, Chicago, Illinois, USA, November 2009.

[13] Liu Zhenglin, Zeng Yonghong, Zou Xuecheng, Han Yu, and Chen Yicheng, "A High-security and Low-power AES S-Box Full-custom Design for Wireless Sensor Network," in proceeding of International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2007.