POKs Based Secure and Energy-Efficient Access Control for Implantable Medical Devices

Chenglong Fu, Xiaojiang Du, Longfei Wu, Qiang Zeng, Amr Mohamed, and Mohsen Guizani

1 Temple University, Philadelphia PA 19122, USA {chenglong.fu,xjdu}@temple.edu
2 Fayetteville State University, Fayetteville NC 28301, USA lwu@uncfsu.edu
3 University of South Carolina, Columbia SC 29208, USA zeng1@cse.sc.edu
4 Qatar University, Doha, Qatar {amrm,mguizani}@ieee.org

Abstract. Implantable medical devices (IMDs), such as pacemakers, implanted cardiac defibrillators and neurostimulators are medical devices implanted into patients' bodies for monitoring physiological signals and performing medical treatments. Many IMDs have built-in wireless communication modules to facilitate data collecting and device reprogramming by external programmers. The wireless communication brings significant conveniences for advanced applications such as real-time and remote monitoring but also introduces the risk of unauthorized wireless access. The absence of effective access control mechanisms exposes patients' life to cyber attacks. In this paper, we present a lightweight and universally applicable access control system for IMDs. By leveraging Physically Obfuscated Keys (POKs) as the hardware root of trust, provable security is achieved based on standard cryptographic primitives while attaining high energy efficiency. In addition, barrier-free IMD access under emergent situations is realized by utilizing the patient’s biometrical information. We evaluate our proposed scheme through extensive security analysis and a prototype implementation, which demonstrate our work’s superiority on security and energy efficiency.

Keywords: Implantable Medical Devices · Physical Obfuscation Keys · Access Control.

1 Introduction

Implantable medical devices (IMDs) are electronic devices that can be either partially or fully implanted into patients’ bodies for collecting patients’ physiological data and delivering timely treatment. With advantages of providing ongoing diagnosis and treatment, IMDs have been pervasively used for treating chronic medical disorders and are proved to be effective in coping with sudden deaths caused by cardiac arrests and ventricular arrhythmia. Most IMDs are equipped with radio modules for wireless communication with external devices called programmers which are used by physicians for data exporting and IMD reprogramming. However, the wireless communication brings attackers extra arsenal to threaten the users’ lives with cyber attacks. Jay Radcliffe and Branaby
Jack have demonstrated the feasibility to remotely hack the insulin pump in [37] and [1] respectively. Coincidentally, former U.S. Vice President Dick Cheney had the wireless connection of his implanted defibrillator disabled due to the concern of cyber attacks launched by terrorists [19].

Although countermeasures against unauthorized IMD access are critical, the design is challenging due to two technical difficulties. First, requirements for utility and security are conflicting. On the one hand, the scheme must be robust enough to defeat all malicious access. On the other hand, during the emergent situation, first-aiders may need to access or reprogram a patient’s IMD immediately without any hindering caused by security mechanisms. Second, the IMD security mechanism must be extremely energy efficient because IMDs are implanted into human’s bodies via surgeries and rely on their embedded batteries to operate many years.

To cope with these security problems of medical devices, a series of research works are proposed [23, 24, 45]. Recent researches tend to address the first difficulty with the touch-to-access principle which is based on a reasonable assumption that attackers having physical contact with victims can harm them directly (rather than utilizing IMDs). Following this, many proposed solutions [16, 41, 47] implement simple access control policies by verifying physical access or proximity for IMD access attempts. However, these proposed works have three drawbacks. (1) Their enforcement of the touch-to-access policy is not based on provable security and may be breached by newly-developed attack techniques [40]. (2) They assume IMDs are equipped with special sensing or communication capabilities like ECG measurement and piezo broadcasting. (3) Simple touch-to-access access control policy cannot deal with complicated scenarios like hierarchical privileges.

Our goal is to design an IMD access control solution that provides provable security without high energy consumption. To that end, we present a Physical Obfuscation Keys (POKs) based IMD access control system. Leveraging a POKs enabled IC card for secure credential storage, we design a lightweight access control protocol with minimal computation and communication overhead on IMDs. For emergent access, We follow the touch-to-access principle and verify physical contact by requiring the patient’s IC card and iris image. Our design is built on standard cryptographic operations to provide provable security and does not need any special sensing or communication capability of the IMD. Moreover, an online Hospital Authentication Server (HAS) is integrated in our system to authenticate the programmer’s identity and realize the dynamic and fine-grained access control.

In summary, our work has the following contributions:

– A novel POKs based key agreement scheme: We innovatively use the POKs enabled IC card to design a secure key agreement scheme for IMDs. In our proposed protocol, computation-intensive operations are offloaded to the IC card and the Hospital Authentication Server to reduce the energy consumption of the IMD.
– Biometrics based emergent access: We design a highly secure method that uses the patient’s iris for barrier-free emergent access.
– **Real-device implementation:** We implement the IMD’s logic on the TelosB sensor mote, analyze the security properties of the design, and evaluate the energy consumption to show our design’s advantage of energy efficiency as well as its speed and memory consumption.

The rest of the paper is organized as follows. We first describe some background on POKs and our key generation in Section 2. Then we present the system and threat model in section 3. Our access control scheme for normal access and emergent access is described in Sections 4 and 5, respectively. After that, we analyze the security of our work in Section 6 and the results of the overhead consumption evaluation are presented in Section 7. The review of the related works is in Section 8. Finally, we conclude in Section 9.

### 2 Background and Key Generation

#### 2.1 Background: Physical Obfuscation Keys

Modern cryptographic primitives have their security based on the confidentiality of secret credentials. Once the credentials are uncovered, attackers can maliciously impersonate legitimate users or retrieve sensitive information from encrypted communications. Thus, secure secret key storage components such are critical to the security of all kinds of applications. However, recently developed physical tampering attacks such as micro-probing attack and electrical glitching attack have already showed their effectiveness on retrieving secret credentials kept in statical storage devices.

To defeat these physical attacks targeting secret credentials, researches propose the Physical Unclone Functions (PUFs) [38] as a hardware-based cryptographic component for authentication and secret key storage. PUFs rely on unique physical characteristic variation as the secret challenge-response pattern. When given an input, the PUF responds with a corresponding unpredictable output. These patterns are determined by the PUF’s unique and unclonable physical properties such as the integrated circuit’s gate propagation delay [31] and SRAM cell initial status [25] which are introduced by uncontrollable variations during the IC fabrication process. Generally, PUFs are divided into two broad categories: strong PUFs and weak PUFs, differing in the number of available Challenge-Response Pairs (CRPs). Typically, strong PUFs can have a large set of CRPs while weak PUFs only support a limited number of CRPs. The Physically Obfuscated Keys (POKs) [15] is an application of weak PUFs that could be considered as a secure key storage technique. The appearance of this technique brings a reliable solution to protect the secret key from being compromised by invasive side-channel attacks. Recent POKs-based secure key generators have reached a bit error rate of less than 1% [27] and require no special processing during the chip fabrication, making it a cost-efficient alternative to expensive secure storage components like EEPROM.

Bringer, et al. make a further step by proposing POKs compatible cryptographic algorithms [3] to defeat the runtime **memory scanning** attacks which
aim to steal the credentials after they are loaded to the memory. They achieve it by splitting scalar product and exclusive OR (XOR), two basic operations used in the linear-feedback shift register (LFSR), into multiple steps. During each step, only part of the secret stored in the POKs are loaded to the memory with all intermediate results obfuscated. Based on the specially designed LFSR, secure POKs enabled stream ciphers like Trivium \cite{6} is realized. Therefore, compared with the traditional primitives, POKs enabled cryptographic operations are resilient to memory scanning attacks. Moreover, invasive tampering attacks that try to unpack the chip will change physical features the keys rely on and thus destroy the the secret inside it permanently. Thus, revealing keys via invasive tampering attacks also fail.

### 2.2 Key Pair Generator

For multi-party communication, key management (e.g., \cite{10,12,46} is important for medical and IoT devices. We design a key generator to synchronously generate temporary keys for each access. As shown in Figure 1, the key generator has the input of a master key and an initial value fed into a Trivium stream generator. During each access, 256 bits are truncated from the output stream and then be processed by the SHA-256 module to get a 256-bit temporary key. In the system, we have two different long-term master keys $Key_1$ and $Key_2$ to derive two temporary keys $SA_i$ and $SB_i$. $Key_1$ and $Key_2$ are stored by the patient’s IC card and the HAS (Hospital Authentication Server), respectively, and the IMD stores the two master keys.

![Fig. 1. The key pair generator. $SA_i$ and $SB_i$ are generated by patient’s IC card and the hospital authentication server respectively.](image)

On the IC card, we have a POK module to store the master key $Key_1$ and a POKs-based Trivium generator to generate temporary keys. As illustrated in the Figure 2, the IC card has the POK module, the Trivium stream generator, and the CPU packed on one chip. The secret stored by the POK module can only be accessed by the Trivium generator which outputs the bitstream directly to the embedded CPU for cryptographic operations. The One-Time Programming (OTP) interface \cite{32} is added to enable one-time access to the secret during the IC card commissioning. After the first access, the OTP interface is physically disabled to prevent any direct access to the master key.

The doctor’s IC card also has a POK module to store the master key $Key_3$ that is used for generating 256-bit message authentication code (MAC). The
MAC algorithm is based on LFSR as described in [3], which is inherently resistant to memory probe attacks and invasive attacks.

3 System Overview and Threat Model

3.1 System Overview

As shown in Figure 3, our access control system involves the following components: the IMD, the programmer, the Hospital Authentication Server (HAS), and IC cards for patients and doctors. IC cards are contact cards with embedded POKs modules. The HAS is a centralized authentication server that is expected to be common in modern hospital and e-health systems. Two pre-defined master keys are shared by the patient’s IC card and the HAS with the IMD respectively. Each master key is used as the secret seed to generate different temporary keys for each access. Then the programmer authentication and the session key generation are all conducted with the temporary keys.

In our design, the patient’s and doctor’s IC cards are commissioned as the representation of their identities via the enrollment phase. During the regular
access, the doctor firstly plugs his IC card into the programmer to prove his identity to the HAS. Then, the programmer forwards the challenge message received from the IMD to the IC card and the HAS. Finally, the programmer generates the correct response message with the help of the IC card and the HAS and establishes a secure communication channel with the IMD. The HAS enforces fine-grain access control policies according to the doctor’s identity and the access operation type.

3.2 Assumptions

For devices, we assume the IMD and IC card are capable of running basic cryptographic operations including SHA-256 and HMAC. While the programmer has Internet access for communicating with the Hospital Authentication Server (HAS). The IC cards are contact cards without wireless communication capability. We can safely assume IC cards are taken with patients and physicians all the time considering its small size easy maintenance (no charging required).

As for the emergent situation when no Internet connection and valid physician’s IC card are available, we assume first-aiders can find the patient’s IC card have the equipment to acquire the patient’s iris code.

3.3 Threat Model

In this paper, we assume a powerful adversary with abilities to eavesdrop all wireless communications between the programmer and the IMD and send arbitrary messages to the IMD remotely.

Also, we reasonably assume the adversary cannot replicate the IC card or retrieve the secret from it due to the POKs’s unclonable feature. The secret keys stored in the IMD is also inaccessible because the IMDs are implanted into patients’ body and are physically inaccessible. Also, the HAS is considered as secure because the HAS has plenty of resources for enforcing security schemes and is managed by professional security administrators.

4 The Authentication Protocol

In this section, we present our IMD authentication and access control protocol, which is based on the a pair of temporary keys $SA_i$ and $SB_i$ and the doctor’s master key $Key_3$. The protocol is composed of four phases: Enrollment, Service Request, Authorization, and Session Establishment. The following parties are involved: an Implantable Medical Device (IMD), a Hospital authentication Server (HAS), a programmer (with the doctor’s IC card plugged in), and the patient’s IC card. Table 1 summarizes all the symbols and notations used in our description. Figure 3 and Figure 4 illustrate the enrollment phase and all the following phases, respectively.
Table 1. Symbols and Notations.

| Notation | Description |
|----------|-------------|
| $Key_3$  | doctor’s master key |
| $SA_i$, $SB_i$ | 256-bits temporary key pair |
| $R$   | 32-bits Service Request Code |
| $i$   | 32-bits counter |
| $ID_1$, $ID_P$, $ID_d$ | 32-bits identity for patient and doctor |
| $T_1$ | 32-bits Time stamp |
| $TS$ | 32-bits Time window threshold |
| $T$ | current time |
| $token(i)$ | token for the $i$th cycle |
| $S_{key}(i)$ | Session key for the $i$th cycle |

4.1 Enrollment

During the enrollment phase, all parties in our protocol are initialized through two independent steps as shown in Figure 3.

Doctor’s Registration Since we assume the programmer is not bound with the doctor’s identity. Hence, registration is required to set up accounts for doctors and link their identities with IC cards.

The doctor’s account should contain detailed profile information and the master key $Key_3$ extracted from the doctor’s IC card through the One-time programming (OTP) interface. Aside from this, a corresponding identification number $ID_P$ and password are also configured for login through the programmer. Based on the profile information, HAS can check doctors’ privilege of accessing a certain patient’s IMD for carrying out dynamic access control. For example, all doctors that are in charge of a certain patient have the permission to export monitoring data from the patient’s ICD, but only the chief physician can reconfigure it.

Patient’s Registration Similar to the doctors’ registration, patients also need to set up accounts on the HAS which contain their profile information, the initial value ($IV$) and the master key $Key_2$.

Before the implantation, the doctor extracts the master key $Key_1$ from a new IC card through the One-time programming (OTP) interface and then fusing the OTP to prevent further access. After that, two master keys and $IV$ are loaded into the IMD and the cycle counter $i$ is set to 1. Once the implantation is completed, the IC card is delivered to the patient.

4.2 Service Request

During normal access circumstances, a patient is conscious, and the wireless connection to the Internet is available. The doctor first inserts his/her IC card into the programmer and the programmer then logs in with his account name and password. Once the doctor’s identity is successfully verified, a TLS session
between the programmer and HAS is established to protect all communications between them, and the session is tagged by the doctor’s identity number $ID_P$. Thereafter, with the patient’s permission, the doctor has the patient’s IC card inserted into the programmer.

As shown by step (i) in Figure 4, the service request is initiated by the programmer sending the request code and the doctor’s identity number to the IMD. When the IMD wakes up from a periodical hibernation, it first retrieves the cycle counter $i$ and produces the current temporal key pair $SA_i$ and $SB_i$. Then, a response containing the patient’s identity number $ID_I$, counter $i$, current timestamp $T_1$, and two HMACs over these variables are sent back to the programmer (step (ii)) as the challenge message.

4.3 Authorization

When the programmer receives a response from the IMD in step (ii), it forwards the two HMACs to the IC card and HAS respectively, as marked by step (iii) and (v) in Figure 4. Specifically, for message (v), the programmer makes use of the doctor’s IC card to generate a UMAC (a universal hashing based message authentication [2]) for the entire message with the master key $Key_3$ as a proof of the doctor’s identity.

For the patient’s IC card, it generates the temporary key $SA_i$ according to the received cycle counter $i$. With $SA_i$, it verifies the received HMAC on $T_1$. If the verification succeeds, the IC card sends back the HMAC on the timestamp $T_1$ cascaded by request code $R$ and increases its counter $i$ by 1.

Similarly, the HAS generates the temporary key $SB_i$ and verifies the HMAC of concatenated timestamp $T_i$ and two identities: $ID_I$ and $ID_P$. If the HMAC is valid, the HAS performs the following checks:

1. Check the temporal validity indicated by $T_1$ and time window length $TS$. 

Fig. 4. The authentication workflow.
2. Check the doctor’s identity by verifying the UMAC with corresponding Key.
3. Check the doctor’s privilege level of accessing the the IMD with the identity ID.

If all checks are successful, the HAS sends back the HMAC of a timestamp, two identity numbers and the request code R as described by the step (vi) in Figure 4. Finally, HAS’s cycle counter i is increased by 1.

### 4.4 Session Establishment

After the programmer receives the returned HMACs from the patient’s IC card and the HAS, the access token is generated by performing XOR operation on the two HMACs as described in (1). Then the programmer sends the hash of timestamp $T_1$ and the token back to IMD (step (vii)).

$$token(i) = \text{HMAC}_{SA_i}(T_1||R) \oplus \text{HMAC}_{SB_i}(T_1||ID_P||ID_I||R)$$  \hspace{1cm} (1)

On receiving the response from the programmer, the IMD first records the current time $T$ and checks whether the response is returned within the time window $TS$. If the response is timely enough, the IMD generates the token token' (as shown in (1)) by itself independently (the IMD possesses both two master keys to produce both two temporary keys $SA_i$ and $SB_i$). After that, the IMD compares the received Hash($T_1||Token$) with the generated Hash($T_1||Token'$). If two results match, the programmer is authenticated and authorized to access the IMD, and the cycle counter in the IMD is increased by 1. Otherwise, the IMD discards all completed steps and goes back to listening mode. Once the authentication and authorization is completed, a secret number $SKey(i)$ is generated independently by the IMD and the programmer. The two parties calculate the XOR of two hashed HMACs as described in (2). The secret key can be used as the seed for deriving session keys to encrypt the communication between the programmer and the IMD.

$$SKey(i) = \text{Hash}(\text{HMAC}_{SA_i}(T_1||R)) \oplus \text{Hash}(\text{HMAC}_{SB_i}(T_1||ID_P||ID_I||R))$$  \hspace{1cm} (2)

### 5 Enhancement

#### 5.1 Emergent Access

Apart from the normal access control described in section 4, special access mode is necessary for the emergent situation when the patient is unconscious and require emergency care. Under this situation, first aiders may need to access the
patient’s IMD to measurement the patient’s physiological signal and perform timely treatment. However, the response message generated by the HAS may be inaccessible due to the lack of valid privilege or the absence of the Internet connections. To address this problem, we design an offline emergent access control scheme to use the temporal keys $SB_i$ cached inside the patient’s IC card as an alternative of the HAS. For the security concern, the cached keys are obfuscated using the patient’s iris code.

The HAS can generate patient specified number of temporal keys for future authentication rounds in advance and load them into the patient’s IC card during the patient’s normal visit to the medical center. With the cached temporal key $SB_i$, the programmer can independently generate the correct response message as the step (vi) in Figure 4. When cached temporal keys are invalidated by regular accesses, the patient can easily the cached updated by asking the HAS to generate more.

5.2 Biometric Encryption of the Cached Temporary Key

The drawback of the cached keys is the possible leakage of temporary keys. Since the cache keys are used as a substitution of HAS, disclosure of them would undermine the effectiveness of HAS. For instance, if attackers steal the patient’s IC card, they can initiate access to the IMD immediately because both two temporary keys can be obtained from the IC card. Under this situation, the patient has no way to protect himself even he/she realizes the lost and report it to HAS manager. As a result, cached temporal keys must be encrypted to prevent unauthorized accesses. We denote each cache item as

$$\text{cache}(i) = i||\text{Enc}_{C_k}(SB_i)$$

where $i$ is the counter and the $C_k$ is the cache encryption key. The cache encryption key $C_k$ is secured with the biometric encryption where the patient’s iris code is exploited to obfuscate the original message as described in [21].

Here, we give some brief retelling about the biometric encryption we use. First, a binary string is derived from an infrared image of patient’s iris image by demodulating the phase information with complex-valued 2D Gabor wavelets [5]. The generated reference string, denoted as $\Theta_{ref}$, is acquired in the patient register phase. After that, the HAS generates the cache encryption key $C_k$ and the key is encoded by Hadamard code and Reed-Solomon code. The encoded $C_k$ is then obfuscated by the iris reference $\Theta_{ref}$ and the resulted $\Theta_{lock}$ is stored in patient’s IC card.

During emergent situation, first aiders can use a digital camera (with infrared mode) to capture the patient’s iris image and generate the sample binary string $\Theta_{sam}$ to decrypt $\Theta_{lock}$ with the XOR operation and then acquire $C_k$ by doing Hadamard and Reed-Solomon decoding. After getting the XORed cache encryption key $C_k$, first aiders are capable of decrypting the temporary key $SB_i$ and access the patient’s IMD locally without the HAS’s support. The robustness of cache key regeneration is guaranteed by Hadamard and Reed-Solomon code which deals with errors in a binary level and block level respectively.
5.3 Recovery Mode

To deal with the issue of patients losing their IC cards, we design a recovery mechanism to reset the IMD and pair it with a new IC card. For the security concern of the recovery mode, the patient needs to go to the hospital to report the lost and sign the related documents in person before starting the reset procedure. A programmer operated by the security administrator is granted with the reset permission by the HAS. The recovery mode is triggered by the programmer with a designated service request code. Upon receiving the reset request, the IMD begins to continuously challenge the programmer as shown in Figure 6. During the challenge process, the IMD asks the programmer to provide the temporary key $SB$ for the cycle counter $K$. The challenge is conducted continuously for $2 \times S$ times where $S$ is the maximum number of cached temporary keys for the emergent access. The reset command is accepted and executed by the IMD if the programmer can provide correct responses to all those challenges with the help of the HAS. When the reset succeeds, the IMD rollback to the registration status to accept the configuration of new master keys $Key_1$ and $Key_2$. Thus, the doctor can re-do the patient’s registration procedures to pair the IMD with a new IC card, and the old stolen IC card is invalidated permanently.

6 Security Analysis

6.1 Active Attacks

In active attacks, adversaries intercept the transmission between the IMD and the programmer and then manipulate the message to deceive the them. However, the identities of the IMD, the programmer, and the HAS are secured by
master keys or kept in the server or the POKs enabled IC card. Even if adversaries can successfully intercept those messages, there is no way to retrieve the valid temporal keys for the current cycle. Without temporal keys, adversaries cannot generate valid HMAC for the tampered message. The reply attack is also not feasible due to the implementation of the time window. The IMD and the programmer would reject all duplicate packets that are outside of the time window.

Another type of active attack is extracting credentials from the IC card through invasive ways. There have been a lot of methods to recover security credentials by tampering IC card circuits [13, 33, 42]. However, with the master key secured by POKs, any attempt to tamper the circuit would fail and destroy the secret credential permanently.

6.2 Desynchronization attack

A lot of PUFs-based authentication protocols that rely on synchronization are subject to the desynchronization attack. Normally, the desynchronization attack is achieved by intentional interruptions during normal authentication, which results in the temporary key generator in different components running at malposed cycles. In our work, the desynchronization attack could be avoided by applying one-step cache in the IC card and the HAS for the previous used temporary keys $SA_{i-1}$ and $SB_{i-1}$. Since the IMD only increases its cycle counter $i$ when authorized by the IC card and the HAS succeeds, the desynchronization condition only happens when the IMD’s counter is one step slower than the IC card or the HAS. When the IC card and the HAS receive the counter from the programmer at step (iii) and (v) in Figure 4, they can easily recover from the desynchronization status with the help of cached keys.
6.3 Identity impersonation attack

The adversaries may disguise themselves as legitimate physicians and use a programmer to access patients’ IMDs. This kind of identity fraud is prevented in our scheme by the doctor’s identity proof at step (v) in Figure 4. To impersonate a legitimate doctor, the attacker must get the corresponding doctor’s identity IC card. Even if the attacker can steal a valid IC card from doctors, the stolen identities are not universally applicable to all IMDs, which restrict the range of damage. Also, this kind of attack cannot bypass the touch-to-access assumption because of requirement of the patient’s IC card. It is difficult to use the IC card without the patient’s permission. Through physical contacts with the patient, attackers are highly likely to be spotted and recorded by other people or video monitoring systems which greatly undermines the power of attacks.

6.4 Security of the emergent access

Many previous attempts introduced additional risk by adding an emergent access mode. For example, cloaker [17] and IMDGuard [47] rely on external devices to secure the IMD. When these external devices are removed, the IMD switches to open access. This can cause potential risk when the external device is stolen or the patient forgets to take it. Instead of trying to let the IMD be aware of the emergent situation, the IMD in our design always runs in the same mode, which means the aforementioned touch-to-access assumption always holds in emergent access since the temporary key $SA_i$ can only be acquired from the patient’s IC card. To exploit the emergent access, attackers must get the patient’s IC card and iris image. Even if the IC card is obtained by attackers without breaching the touch-to-access assumption (e.g., lost by the patient), the iris image must be captured in front of the patient. Additionally, cached keys in the IC card can be easily invalidated after the patient loses his/her card by rolling the normal access for multiple rounds to override the cached $SB$s. After that, attackers will not be able to access the patient’s IMD even if they get the patient’s iris code. The security of the iris code has been extensively discussed and many counterfeit iris detection schemes are proposed in [14, 29, 44]. However, most of the counterfeit iris attack (e.g., cosmetic lens with texture, printed iris on paper) has the prerequisite of acquiring high resolution of the victim’s Iris NIR image. The NIRs that are used to illuminate the iris only has the effective range of 50~70 cm [22]. The successful iris code generation requires a resolution of at least 50 pixels in iris radius for the iris picture [1]. As a result, it is hard for attackers to get the usable iris image without breaching the touch-to-access assumption.

6.5 Security of Trivium

Unlike the block cipher such as AES, there is not a widely recognized secure stream cipher. Trivium has been selected as part of the eSTREAM project [39]. Up to now, no effective analysis attack is proposed that is better than the brute
force search. Existing attacks utilizing the side channel like Differential Power attack \cite{28} and Fault Injection attack \cite{35} all require hardware tampering of the cipher IC. This kind of hardware attack is impractical in our design because of the use of tampering-resistant POKs. Once the circuit is tampered, the master key will be destroyed permanently. Moreover, the passive analysis is also not applicable to our design. The output of the Trivium stream generator, the temporary key $SA_i$, and $SB_i$ are never transmitted directly over the air. Instead, they are only used as keys for HMAC operations.

7 Evaluation

Because commercial IMDs are not opensourced for customization, we implement our design on our own testbed which comprises a TelosB sensor mote, a Raspberry Pi and a Laptop to simulate the IMD, programmer, and HAS respectively. As stated in \cite{47}, Telosb sensor mote is built on the similar low-energy platform as commercial IMDs making it a good choice to simulate the overhead of our system on real IMDs. With the prototype implementation, we evaluate the overhead of the consumption of energy, time, memory, and storage for both the computation process and communication.

7.1 Overhead Statistics

To evaluate the overhead, we make statistics about all computing and communicating operations conducted by the IMD during an access cycle as shown in Table \ref{table:2} and Table \ref{table:3}.

| Steps                      | Operation | Amount | Length (bits) |
|----------------------------|-----------|--------|---------------|
| Service Request            | HMAC      | 2      | 32,96         |
| Keypair Generation         | Trivium   | 2      | 256,256       |
| Token Generation           | HMAC      | 2      | 32,128        |
| Session Key Generation     | SHA-256   | 2      | 256,256       |

| Steps                      | Operation | Length (bits) |
|----------------------------|-----------|---------------|
| Service Request            | Receiving | 64            |
|                           | Sending   | 64            |
| Token verifying            | Receiving | 256           |

7.2 Experiment Results

We split the design into several parts, one for the key generator and others for the HMAC and SHA-256 operations. By deploying asynchronous counters, we
record the time consumption of each part with the accuracy of 1 millisecond. With the energy consumption model \[7,36,43\] for TelosB as shown in Table 4, the energy consumption for each operation is calculated from the time consumption.

**Table 4. TelosB power consumption model.**

| Operation       | Power consumption (mW) |
|-----------------|------------------------|
| Transmit        | 69                     |
| Listen          | 60                     |
| Receiving       | 61                     |
| Computing (active) | 4.8                 |
| Computing (idle)  | 4.5                   |
| Sleep           | 0.035                  |

The total computing time for a complete authentication cycle is 367 ms, if the transmission time is not taken into account. The overhead for different parts is listed in Table 5.

**Table 5. Overhead for computation.**

| generator | HMAC | SHA256 |
|-----------|------|--------|
| Time (ms) | 52   | 46     |
| Energy (µJ) | 240 | 220.8  |
| ROM (bytes) | 14290 | 6022 |
| RAM (bytes) | 312 | 282    |

Unlike the computational part, the overhead for transmission is much more complex because of the utilization of low power listening and collision avoiding mechanisms in the MAC layer. The listening time is uncertain depending on the environment noise, and the power consumption on synchronizations is difficult to specify. Therefore, to make things easier, we test the overhead of broadcasting packets with no retransmission. The results of the transmission test are listed in Table 6.

**Table 6. Overhead for communication.**

| Receiving (320 bits) | Sending (608 bits) |
|----------------------|---------------------|
| Time (ms)            | 40                  |
| Energy (µJ)          | 2440                |
| ROM (bytes)          | 11140               |
| RAM (bytes)          | 504                 |

With the operation specific evaluation results, ensemble time and energy overhead are calculated as illustrated in Equation 3 and 4.
\[
E = E_{\text{comp}} + E_{\text{Tx}} + E_{\text{Rx}}
\]
\[
= E_{\text{gen}} + 4 \cdot E_{\text{HMAC}} + 3 \cdot E_{\text{Sha256}} + E_{\text{Tx}} + E_{\text{Rx}}
\]
\[
= 249 \mu J + 4 \cdot 220.8 \mu J + 3 \cdot 72 \mu J + 2440 \mu J + 1518 \mu J
\]
\[
= 5306 \mu J
\]  

\[
T = T_{\text{comp}} + T_{\text{Tx}} + T_{\text{Rx}}
\]
\[
= T_{\text{gen}} + 4 \cdot T_{\text{HMAC}} + 3 \cdot T_{\text{Sha256}} + T_{\text{Tx}} + T_{\text{Rx}}
\]
\[
= 52 ms + 4 \cdot 46 ms + 3 \cdot 15 ms + 40 ms + 22 ms
\]
\[
= 343 ms
\]

The results indicate that our design introduces orders of magnitude lower energy and time overhead to the IMD compared with state-of-art physiological-feature-based solutions: The energy consumption of the OPFKA [26] ranges from about 70 mJ to 1000 mJ according to the coffer size. The IMDguard [47] needs at least 45 seconds to measure more than 21 heart beats. While, the Heart-to-heart [41] uses TLS communication where the RSA encryption itself takes 5000000 cpu cycles (equivalent to 100 ms and 3038 \mu J on the platform used by the author).

8 Related Work

8.1 Pre-loaded-Key Based Solutions

In some early works of IMD security, a long-term and device-specific credential is pre-loaded into the IMD and the programmer must possess the corresponding credential to pass the authentication. Halperin et al. [20] propose a acoustic side channel based solution where programmer need a valid master key to access the IMD. Li et al. [30] present a rolling code based authentication scheme, in which the IMD and the programmer share a pair of encryption keys that are used to encrypt a sequence number. Liu et al. Denning et al. [8] uses visual objects to carry a static credential for the IMD authentication. This type of IMD authentication/authorization schemes are obsoleted because the difficulty to protect the pre-loaded credential.

8.2 Physiological Feature Based Solutions

Recent researches tend to establish temporary keys by extracting time-varying information from patient’s physiological signals. Most of them are also based on the touch-to-access assumption as our work does because the physiological features used for key generation can only be measured when the programmer is physically close to the patient. In [26, 41, 47, 49], the patient’s ECG features are utilized to derive credentials due to its high level of randomness. In [22], the
access credentials are extracted from the patient’s biometric features such as the fingerprints, iris and height.

This type of solutions face the problem of reliability: accurate measurement of physiological value is difficult because of the noise and distortion which results in long time consumption for feature measurement. Also, all these requires the IMDs to be equipped with special sensors to measure physiological signals such as ECG which not only increase the cost of IMDs but also causes excessive power consumption.

8.3 Proxy Based Solutions

Some researches propose to utilize additional devices to authenticate the programmer on behalf of the IMD for reducing the IMD’s power consumption. The common design of proxy-based security schemes [9, 18, 47, 48] includes (1) the proxy device jams the signal of external programmers until they are authenticated; (2) the IMD allows open-access when the proxy is not in proximity.

The Shield [18] jams the programmer’s signal, but also the IMD’s signal. Equipped with a full-duplex radio with two antennas, Shield is able to receive and decode the IMD’s signal, meanwhile jamming it so that the programmer cannot receive and decode the complete IMD’s message. A secured communication channel is assumed to have been set up between the Shield and the legitimate programmer. Therefore, Shield actually serves as a relay between the IMD and the programmer. Similarly, Cloaker [9] is also a mediator that forwards all authorized communications between the IMD and programmer.

In the other schemes [47, 48], the proxy devices are mainly considered as an external authenticator, which stores the public keys of all authorized programmers and are able to verify the identity of the programmer that requests for access using its digital signature. The communication channel between the IMD and the programmer is established, after the programmer is successfully authenticated. In [47], a pair of lightweight symmetric keys are issued to the IMD and the programmer, to encrypt their future communications. In [48], the authentication proxy is embedded into a gateway.

This type of solutions requires patients to carry the active proxy device all the time which is unpractical. Also, the jamming may cause interference on other medical devices.

9 Conclusion

Security and low power consumption are among the most critical goals when designing access control schemes for implantable medical devices, while how to attain both high energy efficiency and resiliency to powerful attacks, such as memory scanning and physical tampering, is still an unresolved challenge. We propose to take Physically Obfuscated Keys (POKs) as the hardware root of trust to establish a highly secure access control scheme for IMDs, and apply the idea of computation offloading to saving energy consumption on the IMD.
side. We have comprehensively analyzed the security of the proposed scheme and compared it with some well-known mechanisms. In addition to its security advantage, a prototype system implemented on the TelosB platform demonstrates its high energy efficiency.

Acknowledgments

This publication was made possible by NPRP grant #8-408-2-172 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. Barnaby jack hacks diabetes insulin pump live at hacker halted. [https://www.infosecurity-magazine.com/news](https://www.infosecurity-magazine.com/news) (2011)
2. Black, J., Halevi, S., Krawczyk, H., Krovetz, T., Rogaway, P.: Umac: Fast and secure message authentication. In: Annual International Cryptology Conference. pp. 216–233. Springer (1999)
3. Bringer, J., Chabanne, H., Icart, T.: On physical obfuscation of cryptographic algorithms. In: International Conference on Cryptology in India. pp. 88–103. Springer (2009)
4. Daugman, J.: The importance of being random: statistical principles of iris recognition. Pattern recognition 36(2), 279–291 (2003)
5. Daugman, J.: How iris recognition works. IEEE Transactions on circuits and systems for video technology 14(1), 21–30 (2004)
6. De Cannière, C.: Trivium: A stream cipher construction inspired by block cipher design principles. In: International Conference on Information Security. pp. 171–186. Springer (2006)
7. De Meulenaer, G., Gosset, F., Standaert, F.X., Pereira, O.: On the energy cost of communication and cryptography in wireless sensor networks. In: Networking and Communications, 2008. WIMOB’08. IEEE International Conference on Wireless and Mobile Computing,. pp. 580–585. IEEE (2008)
8. Denning, T., Borning, A., Friedman, B., Gill, B.T., Kohno, T., Maisel, W.H.: Patients, pacemakers, and implantable defibrillators: Human values and security for wireless implantable medical devices. In: SIGCHI Conference on Human Factors in Computing Systems (2010)
9. Denning, T., Fu, K., Kohno, T.: Absence makes the heart grow fonder: New directions for implantable medical device security. In: USENIX HotSec (2008)
10. Du, X., Guizani, M., Xiao, Y., Chen, H.H.: Transactions papers a routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks. IEEE Transactions on Wireless Communications 8(3), 1223–1229 (2009)
11. Du, X., Xiao, Y., Ci, S., Guizani, M., Chen, H.H.: A routing-driven key management scheme for heterogeneous sensor networks. In: 2007 IEEE International Conference on Communications. pp. 3407–3412. IEEE (2007)
12. Du, X., Xiao, Y., Guizani, M., Chen, H.H.: An effective key management scheme for heterogeneous sensor networks. Ad Hoc Networks 5(1), 24–34 (2007)
13. Fueki, S.: Semiconductor integrated circuit on ic card protected against tampering (Sep 26 2001), u.S Patent App. 09/962,224
14. Galbally, J., Marcel, S., Fierrez, J.: Image quality assessment for fake biometric detection: Application to iris, fingerprint, and face recognition. IEEE transactions on image processing 23(2), 710–724 (2014)
15. Gassend, B.L.: Physical random functions. Ph.D. thesis, Massachusetts Institute of Technology (2003)
16. Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., Fu, K.: They can hear your heartbeats: non-invasive security for implantable medical devices. ACM SIGCOMM Computer Communication Review 41(4), 2–13 (2011)
17. Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., Fu, K.: They can hear your heartbeats: non-invasive security for implantable medical devices. ACM SIGCOMM Computer Communication Review 41(4), 2–13 (2011)
18. Gollakota, S., Hassanieh, H., Ransford, B., Katabi, D., Fu, K.: They can hear your heartbeats: Non-invasive security for implantable medical devices. In: ACM SIGCOMM (2011)
19. Gupta, S.: Dick cheney’s heart. http://www.cbsnews.com/news/dick-cheney-s-heart/ (October 2013)
20. Halperin, D., Heydt-Benjamin, T.S., Ransford, B., Clark, S.S., Defend, B., Morgan, W., Fu, K., Kohno, T., Maisel, W.H.: Pacemakers and implantable cardiac defibrillators: Software radio attacks and zero-power defenses. In: IEEE S&P (2008)
21. Hao, F., Anderson, R., Daugman, J.: Combining crypto with biometrics effectively. IEEE transactions on computers 55(9), 1081–1088 (2006)
22. Hei, X., Du, X.: Biometric-based two-level secure access control for implantable medical devices during emergencies. In: IEEE INFOCOM (2011)
23. Hei, X., Du, X., Lin, S., Lee, I.: Pipac: Patient infusion pattern based access control scheme for wireless insulin pump system. In: 2013 Proceedings IEEE INFOCOM. pp. 3030–3038. IEEE (2013)
24. Hei, X., Du, X., Wu, J., Hu, F.: Defending resource depletion attacks on implantable medical devices. In: 2010 IEEE Global Telecommunications Conference GLOBECOM 2010. pp. 1–5. IEEE (2010)
25. Holcomb, D.E., Burleson, W.P., Fu, K.: Power-up sram state as an identifying fingerprint and source of true random numbers. IEEE Transactions on Computers 58(9), 1198–1210 (2009)
26. Hu, C., Cheng, X., Zhang, F., Wu, D., Liao, X., Chen, D.: Opfk: Secure and efficient ordered-physiological-feature-based key agreement for wireless body area networks. In: IEEE INFOCOM (2013)
27. Karpinskiy, B., Lee, Y., Choi, Y., Kim, Y., Noh, M., Lee, S.: 8.7 physically unclonable function for secure key generation with a key error rate of 2e-38 in 45nm smart-card chips. In: Solid-State Circuits Conference (ISSCC), 2016 IEEE International. pp. 158–160. IEEE (2016)
28. Kazmi, A.R., Afzal, M., Amjad, M.F., Abbas, H., Yang, X.: Algebraic side channel attack on trivium and grain ciphers. IEEE Access (2017)
29. Kohli, N., Yadav, D., Vatsa, M., Singh, R., Noore, A.: Detecting medley of iris spoofing attacks using desist. In: Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on. pp. 1–6. IEEE (2016)
30. Li, C., Raghunathan, A., Jha, N.K.: Hijacking an insulin pump: Security attacks and defenses for a diabetes therapy system. In: IEEE HealthCom (2011)
31. Lim, D., Lee, J.W., Gassend, B., Suh, G.E., Van Dijk, M., Devadas, S.: Extracting secret keys from integrated circuits. IEEE Transactions on Very Large Scale Integration (VLSI) Systems 13(10), 1200–1205 (2005)
32. Min, B.J., Lee, K.W., Lee, H.J., Kim, S.R., Oh, S.G., Jeon, B.G., Yang, H.H., Kim, M.K., Cho, S.H., Cheong, H., et al.: An embedded nonvolatile fram with electrical fuse repair scheme and one time programming scheme for high performance smart cards. In: Custom Integrated Circuits Conference, 2005. Proceedings of the IEEE 2005. pp. 255–258. IEEE (2005)
33. Ming-Chien, C., Chi-Wei, K.: Stolen-verifier attack on two new strong-password authentication protocols. IEICE Transactions on communications 85(11), 2519–2521 (2002)
34. Pope, A., Bouxsein, P., Manning, F.J., Hanna, K.E., et al.: Innovation and invention in medical devices: workshop summary. National Academies Press (2001)
35. Potestad-Ordóñez, F., Jiménez-Fernández, C.J., Valencia-Barrero, M.: Fault attack on fpga implementations of trivium stream cipher. In: Circuits and Systems (ISCAS), 2016 IEEE International Symposium on. pp. 562–565. IEEE (2016)
36. Prayati, A., Antonopoulos, C., Stoyanova, T., Koulamas, C., Papadopoulos, G.: A modeling approach on the telesb wsn platform power consumption. Journal of Systems and Software 83(8), 1355–1363 (2010)
37. Radcliffe, J.: Hacking medical devices for fun and insulin: Breaking the human scada system. In: Black Hat Conference presentation slides. vol. 2011 (2011)
38. Ravikanth, P.S.: Physical One-Way Functions. Ph.D. thesis, MASSACHUSETTS INSTITUTE OF TECHNOLOGY (2001)
39. Rolshaw, M.: The estream project. Lecture Notes in Computer Science 4986, 1–6 (2008)
40. Rostami, M., Burleson, W., Juels, A., Koushanfar, F.: Balancing security and utility in medical devices? In: IEEE Design Automation Conference (DAC) (2013)
41. Rostami, M., Juels, A., Koushanfar, F.: Heart-to-heart (h2h): authentication for implanted medical devices. In: ACM CCS (2013)
42. Skorobogatov, S.: Flash memory bumpingattacks. In: International Workshop on Cryptographic Hardware and Embedded Systems. pp. 158–172. Springer (2010)
43. Somov, A., Minakov, I., Simalatsar, A., Fontana, G., Passerone, R.: A methodology for power consumption evaluation of wireless sensor networks. In: Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on. pp. 1–8. IEEE (2009)
44. Wei, Z., Qiu, X., Sun, Z., Tan, T.: Counterfeit iris detection based on texture analysis. In: Pattern Recognition, 2008. ICPR 2008. 19th International Conference on. pp. 1–4. IEEE (2008)
45. Xia, Q., Sifah, E.B., Asamoah, K.O., Gao, J., Du, X., Guizani, M.: Medshare: Trust-less medical data sharing among cloud service providers via blockchain. IEEE Access 5, 14757–14767 (2017)
46. Xiao, Y., Rayi, V.K., Sun, B., Du, X., Hu, F., Galloway, M.: A survey of key management schemes in wireless sensor networks. Computer communications 30(11-12), 2314–2341 (2007)
47. Xu, F., Qin, Z., Tan, C.C., Wang, B., Li, Q.: Indguard: Securing implantable medical devices with the external wearable guardian. In: INFOCOM, 2011 Proceedings IEEE. pp. 1862–1870. IEEE (2011)
48. Zheng, G., Fang, G., Orgun, M.A., Shankaran, R.: A non-key based security scheme supporting emergency treatment of wireless implants. In: IEEE ICC (2014)
49. Zheng, G., Fang, G., Shankaran, R., Orgun, M., Zhou, J., Qiao, L., Saleem, K.: Multiple ecg fiducial points based random binary sequence generation for securing wireless body area networks. IEEE Journal of Biomedical and Health Informatics PP(99) (2016)