4-24-2020

Vulnerabilities of the Artificial Pancreas System and Proposed Cryptographic Solutions

D. J. Cooke
Boise State University

Kristen Garcia
Boise State University

Andres Guzman
Boise State University

Lindsey Kim
Boise High School

Brooklyn Mesia
Boise State University

See next page for additional authors

Vulnerabilities of the Artificial Pancreas System and Proposed Cryptographic Solutions

Abstract

Type I Diabetes Mellitus is the most common form of diabetes in people under the age of 30. Current treatment for Type I Diabetes Mellitus includes lifelong monitoring of blood glucose levels and administration of insulin injections, but medical advances in the hybrid closed-loop artificial pancreas are a possible improvement in the maintenance of this disease. Our goal is to build a simulation of the artificial pancreas using three Raspberry Pi computers and an implementation of the OpenAPS algorithm. We will also build an artificial pancreas system using two Raspberry Pi computers, a Medtronic insulin pump, and an implementation of the OpenAPS algorithm. We are investigating the vulnerabilities of the two artificial pancreas systems by using common hacking resources such as Kali Linux equipped with Wireshark and other tools. One challenge with securing the artificial pancreas system and other implantable medical devices is the limitations of the computational power and energy storage. Through an analysis of the vulnerabilities of the system, we will design and perform experiments to propose a lightweight cryptographic algorithm that ensures the security of the data transmissions while operating with constrained resources.

Authors

D. J. Cooke, Kristen Garcia, Andres Guzman, Lindsey Kim, Brooklyn Mesia, Jacob Palmer, Shawn Shields, Milan Zanussi, Liljana Babinkostova, Marion Scheepers, Jay Radcliffe, and Robert Erbes

Security Vulnerabilities of the Artificial Pancreas

Daniel J. Cooke, Andres Guzman, Brooklyn Mesia, Jacob Palmer, Shawn Shields, and Milan Zanussi

BOISE STATE UNIVERSITY

Introduction

We live in a world of cyber-enabled, wireless devices that enhance many aspects of life, including treatment of diabetes. Type I Diabetes is a chronic autoimmune disorder characterized by the destruction of pancreatic B-cells and subsequent deficiency of insulin - a crucial hormone in the regulation of blood glucose levels. Current treatment includes monitoring blood glucose levels and administration of insulin injections, but development of an Artificial Pancreas is automating the maintenance of this disease. Implantable Medical Devices (IMD) are shrinking in physical size which limits their storage, power, and processing capacity resulting in the unsecure transmission of data. The National Institute of Standards and Technology (NIST) has called for encryption algorithms to be considered as the lightweight cryptographic standard to combat these vulnerabilities. Our team implemented one encryption algorithm with our simulation of an Artificial Pancreas.

Artificial Pancreas

(1) Continuous Glucose Monitor (CGM): records ongoing blood sugar readings and measures glucose concentrations in the interstitial fluid of the patient's cells
(2) Control Algorithm (CAD): sensor readings are received and analyzed to calculate the correct insulin dose required
(3) Insulin pump (cannula inserted under the skin): administration of the correct insulin dose per that time
(4) Patient Effect: the patient's glucose levels respond to the delivery of insulin

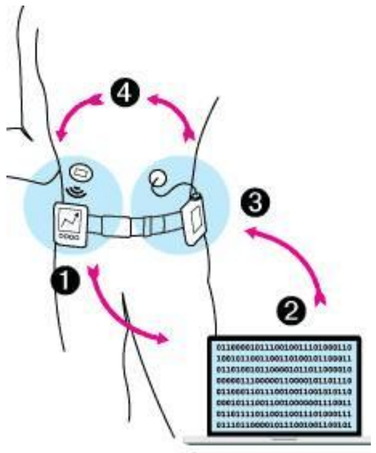

Figure 1: The loop and communication channels of the Artificial Pancreas System [5]

Challenges and Limitations of Artificial Pancreas System

- Insulin sensitivity varies due to time of day, stress levels, exercise/activity levels, and food intake.
- Once insulin is in the body it cannot be removed
- Hypoglycemia
- Computational power and battery life
- Securing transmission of data between devices

Experimental Design

We built a simulation of the Artificial Pancreas to demonstrate the vulnerabilities and to mitigate unencrypted data and unauthorized access. The simulation uses Raspberry Pi computers to imitate a control algorithm device and insulin pump. Another Raspberry Pi uses WireShark, a packet analyzer, to intercept the communication between devices. This design allows us to encrypt the transmitted data and measure the additional resources consumed by the security features.

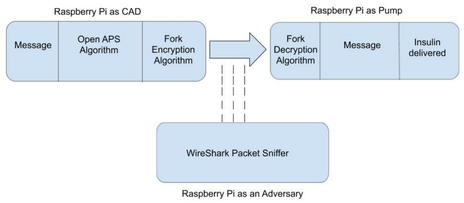

Figure 2: Components of each Raspberry Pi in the simulation

OpenAPS Algorithm

OpenAPS is an open-source project to build an Artificial Pancreas using older medical devices and a Raspberry Pi computer to monitor glucose levels and deliver insulin.

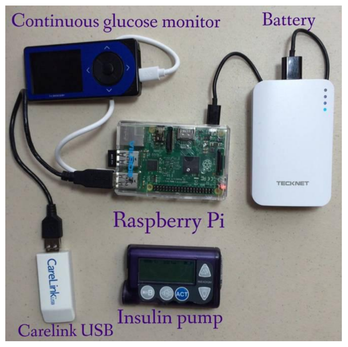

Figure 3: Example of the hardware used in an OpenAPS system [4]

Securing Data Transmission

To demonstrate the need for added security features, we show how an adversary can remotely suspend delivery of insulin, command the pump to dispense insulin, and view personal medical information broadcasted over an unencrypted channel.

```
root@vip400:~/myopenaps# openaps use pump suspend_pump
{
  "status": "suspended",
  "recieved": true,
  "enacted_at": "2020-03-08T20:03:33.980331"
}
```

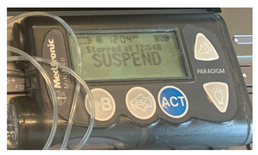

Figure 4: Highlighted in red is the command to suspend the delivery of insulin and the pump's response.

```
root@vip400:~/myopenaps# openaps use pump read_settings
{
  "low_reservoir_warn_point": 15,
  "keypad_lock_status": 0,
  "maxBasal": 4.1,
  "temp_basal": {
    "percent": 100,
    "type": "Units/hour"
  },
  "rf_enable": true,
  "auto_off_duration_hrs": 0,
  "block_enable": false,
  "timeformat": 0,
  "insulin_action_curve": 4,
  "audio_bolus_size": 0,
  "selected_pattern": 0,
  "patterns_enabled": true,
  "maxBolus": 25.0,
  "paradigm_enabled": 1
}
```

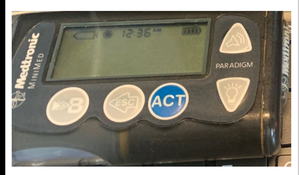

Figure 5: The left shows asking the pump for setting information. Highlighted in red are the max Basal and Bolus settings that can be exploited when commanding the pump to deliver insulin. On the right, the pump has no indication that these settings are being accessed.

```
root@vip400:~/myopenaps# openaps use pump bolus -h
usage: openaps-use pump bolus [-h] input
Send bolus command. [#warning!!!]
positional arguments:
  input
optional arguments:
  -h, --help  show this help message and exit
  Beware! This is a powerful command because it can give a lot of
  insulin.  Please be careful!
  Not a part of oref0.
  ---------------------
  Requires json input with the following keys defined:
    * `units` - Number of units to bolus.
  Zero point one units:
  { "units": 0.1 }
  Two units:
  { "units": 2 }
```

Figure 6: The pump can be commanded to deliver insulin. The terminal warns that this command is very dangerous, even though no further authentication is required.

Fork: Authenticated Encryption

ForkAE is a lightweight authenticated encryption scheme optimized for short messages. Fork is a 2nd-round candidate for the NIST Lightweight Cryptographic Standard. In [1], it is claimed that Fork is an appropriate cryptosystem for resource constrained devices including IMDs. Figure 7 depicts the mathematical structure of one round of Fork.

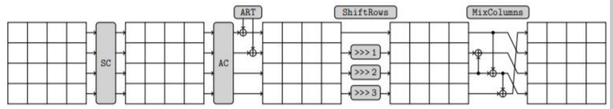

Figure 7: Structure of one round of Fork [1]

Fork: Security Analysis

The forking procedure introduced in Fork creates two separate copies of ciphertext for the same corresponding plaintext, but adds an extra constant to one of the copies of the block and encodes both blocks using the same encryption algorithm and key. Each branch can be regarded as its own permutation of the same set of messages.

Theorem: Let F_1: $\{0,1\}^n \rightarrow \{0,1\}^n$ be the permutation performed by the first branch and F_2: $\{0,1\}^n \rightarrow \{0,1\}^n$ be the permutation performed by the second branch. Then the subgroup generated by F_1 and F_2 is a subgroup of the alternating group on $\{0,1\}^n$.

Future Work

The preliminary actions of this project created versatility in the experimental potential of lightweight devices. Further research is required to demonstrate the vulnerabilities of the current communication protocols, measure the resource consumption of the APS design enhanced with security features, and to compare the security and resource consumption of ForkAE to other encryption schemes. Our goal is to propose a solution to securing the Artificial Pancreas, transferable to other IMDs like pacemakers, neurostimulators, and cardioverter defibrillators.

References

[1]Andreeva E. et al., "ForkAE: Lightweight AEAD Submission to NIST", *NIST Lightweight Cryptography Standardization*, (2018).
[2]Andreeva, Elena, et al. "Forkcipher: a new primitive for authenticated encryption of very short messages." *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, Cham, 2019.
[3] Banik, Subhadeep, et al. "Cryptanalysis of ForkAES." *International Conference on Applied Cryptography and Network Security*, 2019.
[4] Lewis, Dana. "How I Designed a 'DIY' Closed Loop Artificial Pancreas." *DIYPS.org*, OpenAPS, 12 May 2016.
[5]Zaccardi F et al., *Pathophysiology of Type 1 and Type 2 Diabetes Mellitus: a 90-year Perspective* Postgraduate Medical Journal, Vol. 92, pp. 63-69, (2016).

Acknowledgements

We thank the Boise State College of Innovation and Design for their support of the project as well as our mentors Dr. Liljana Babinkostova, Robert Erbes (Idaho National Lab), Jay Radcliffe (Thermo Fisher Scientific), Dr. Marion Scheepers and William Unger. This research was supported by NSF REU Site Grant DMS-169872.