Secure and Energy-Efficient Proximity-Based Pairing for IoT Devices

Yaqi He, Kai Zeng, Brian L. Mark, and Khaled N. Khasawneh
Department of Electrical and Computer Engineering
George Mason University, Fairfax, VA, U.S.A.

Abstract—**Internet of Things (IoT) devices are largely resource-constrained embedded devices with limited user interface and battery capacity. As a consequence, bootstrapping a secure connection between an IoT device and a wireless network (e.g., WiFi network) becomes a challenging problem since the traditional Pre-Shared Key (PSK) based authentication cannot be directly applied. Proximity-based device authentication is a promising mechanism to enable secure pairing of an IoT device to a wireless network. However, existing solutions do not deliberately consider energy-efficiency nor the tradeoff between energy consumption and security strength in the pairing process. This paper fills this gap by enhancing the energy-efficiency and studying the tradeoff between energy consumption and security strength of an existing proximity-based IoT device authentication protocol, called Move2Auth. An optimization problem is formulated to minimize the energy consumption incurred by the pairing protocol while satisfying the desired security performance. Experimental results based on Raspberry Pi devices show the energy-efficiency advantage of the proposed scheme over the existing one.**

Index Terms—**Internet of Things, Device pairing, D2D communication, Energy-efficiency, Authentication and verification, Elliptic Curve Cryptography.**

I. Introduction

In the past decade, the Internet of Things (IoT) has grown incredibly in various fields. Moreover, its continuous boosting with the deployment of 5G networks opens up many new IoT applications, ranging from smart factories to telehealth. An IoT device usually connects to the Internet through a wireless access network, such as WiFi. Due to the lack of a user interface (e.g., keyboard or keypad), when a new IoT device needs to join a WiFi network, the user cannot input the password directly to the device. Typically, the IoT device first pairs with a user's smartphone, which then transfers a password through an established secure channel. Thus, the problem reduces to IoT-to-smartphone authentication, in which the smartphone must ensure that the password is transmitted to the pairing IoT device rather than a man-in-the-middle (MITM) attacker. The challenge arises from the fact that the IoT device does not have a pre-shared secret with the smartphone to enable authentication in the first place.

To address this challenge, proximity-based authentication mechanisms are widely used for secure device pairing without relying on a pre-shared secret [1]–[5]. These mechanisms either provide one-way (e.g., [3], [5]) or two-way (e.g., [1], [2], [4]) authentication by proving location proximity of the pairing device(s), leveraging auxiliary sensors on either one or both pairing devices. Although the security and usability of existing proximity-based authentication schemes have been extensively studied, the existing solutions do not deliberately consider energy-efficiency nor the tradeoff between energy consumption and security strength in the pairing process. However, since many IoT devices are battery powered, energy-efficient design should be considered in the IoT computing and communication protocols (including the device pairing process) in order to prolong the lifetime of their operations.

This paper aims to fill this gap by enhancing the energy efficiency and studying the tradeoff between energy consumption and security strength of a representative proximity-based IoT device authentication protocol, called Move2Auth [5]. Our proposed device pairing scheme provides a one-way IoT device to smartphone authentication without requiring any particular sensing measurement information from the IoT device or holding or movement of the IoT device. Therefore, it provides a usable solution for a wide range of IoT device types and pairing scenarios. We need to emphasize that although our study is mainly based on Move2Auth, the methodology proposed in this paper can be applied to analyze and enhance the energy efficiency of other device pairing protocols.

The major contributions of this paper are summarized below.

- We study the tradeoff between energy consumption and security strength of Move2Auth, and jointly optimize the number of transmission packets and the correlation coefficient decision threshold to minimize energy consumption while achieving a desirable security performance.
- We enhance the security strength of Move2Auth by adding an extra step to detect a MITM attacker who might impersonate the smartphone.
- An elliptic curve cryptography (ECC)-based key agreement scheme is adopted to further enhance the energy efficiency of the Move2Auth scheme, which applies the Rivest-Shamir-Adleman (RSA) public key cryptosystem.
- We have implemented an enhanced Move2Auth protocol on Raspberry Pi embedded devices and conducted experiments to validate the security and energy-efficiency of the protocol.

The remainder of the paper is organized as follows. In Section II, we review the state-of-the-art in IoT device pairing. In Section III, we describe our system model. In Section IV, our enhanced Move2Auth protocol is proposed and an optimization problem for energy-efficiency with security constraints is formulated. Section V presents experimental

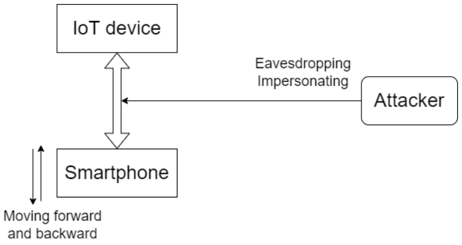

Fig. 1. System model.

performance evaluation results from our implementation of the device pairing protocol. Finally, concluding remarks are given in Section VII.

II. Related Work

Bootstrapping a secure communication between two wireless devices without a pre-shared secret is typically called a secure device pairing (SDP). Proximity-based authentication is widely used for SDP by leveraging an out-of-band (OOB) channel to verify the location proximity of the pairing device. Major proximity-based mobile or IoT device authentication schemes include "Shake Well Before Use" [1], MagPairing [4], Good Neighbor [3], Touch-To-Pair (T2Pair) [2], and Move2Auth [5]. "Shake Well Before Use" requires accelerometers on both pairing devices and MagPairing requires magnetometers. Both schemes require the user to tap the pairing devices together and shake them. Although Good Neighbor does not require any sensors on pairing devices, it relies on multiple antennas well separated in distance on one device. T2Pair proposes the concept of Universal Operation Sensing (UOS), which allows IoT devices to sense user operations and uses timestamps to describe them without requiring inertial sensors. The user needs to touch the IoT device by pressing a button, twisting a knob, or swiping a touchscreen with intentional random pauses in these operations.

Similar to T2Pair, Move2Auth does not depend on any type of inertial sensors on the IoT device. The difference lies in that it does not require the user to make multiple touches on the IoT devices nor hold or move the IoT device. Neither does Move2Auth require multiple antennas. These features make it a promising universal contactless solution for IoT device pairing under various application scenarios. Therefore, we choose Move2Auth as a baseline design and aim to enhance its energy-efficiency and security strength.

To the best of our knowledge, this is the first work studying the energy-efficiency of a device pairing scheme. The motivation comes from the fact that many IoT devices are battery-powered, so energy-efficiency must be considered in all the IoT device communication and computation operations to maximize its operational lifetime. The methodology used in this paper can also be applied or customized to other device pairing protocols to jointly optimize energy-efficiency and security strength.

III. System Model

The system model considered in this paper is depicted in Fig. 1, in which we have three parties: IoT device, smartphone, and attacker. The goal is to establish a secure channel (i.e., generate a shared secret key) between the IoT device and smartphone without a pre-shared secret, while achieving energy-efficiency and a desirable security strength under both passive and active attacks. We assume the IoT device may not be moved, held, or touched by the user during the pairing process. The only assumption on the IoT device is that it has a wireless interface (e.g., WiFi) and sufficient computation capability to perform cryptographic operations. We assume the smartphone also has a WiFi interface and is equipped with an accelerometer, which is very common in practice. If the pairing is successful, the smartphone transmits a WiFi password to the IoT device through the established secure channel, allowing the device to securely join the user's WiFi network.

A. *Attack Model*

We mainly consider passive eavesdropping and active MITM attacks, which are described as follows.

Eavesdropping: A passive eavesdropping attacker does not interrupt the device pairing process but tries to steal or determine the password that the smartphone transmits to the IoT device. To defend against such an attack, the password should be encrypted using the secret key established during the pairing process.

Man-in-the-middle (MITM): The MITM attack is the most insidious attack on device pairing. In this attack, the attacker impersonates the IoT device and/or smartphone in order to control/hijack the communication between the pairing devices. All the information exchanged between the IoT device and smartphone will be disclosed to the attacker if it is successful. We assume the MITM attacker is at a different location from the IoT device beyond a certain distance (e.g., at least 0.5 m). Otherwise, it will be easily detected physically by the user.

IV. Device Pairing Protocol Design

In this section, we first review the principle mechanisms of Move2Auth and then propose enhancements with respect to energy-efficiency and security.

A. *Move2Auth Primer*

In Move2Auth, the smartphone first establishes a secure channel with the IoT device and then transmits the encrypted WiFi password to the IoT device, which can be used to securely connect to the user's WiFi network. In order to establish a secure channel, the IoT and smartphone execute the following processes for information exchange:

- IoT device: Sends a series of identical packets containing its random public key.
- Smartphone: Held by a user, performs simple movements (e.g., back-and-forth) recorded by its embedded accelerometer, measures the received signal strength (RSS) of each packet, and records their reception times.
- Smartphone: Derives the instantaneous distance between itself and the IoT device based on RSS.
- Smartphone: Derives the accelerations based on the instantaneous distances and timestamps by taking the second derivative.

- Smartphone: Computes and compares the correlation coefficient between the derived acceleration trace and the acceleration readings from the accelerometer, and examines the RSS variations.

Only an IoT device in close proximity can lead to a good match between the derived acceleration trace and accelerometer readings, and at the same time, cause a large RSS variation. A MITM attacker will not pass both checks. Therefore, when both conditions are met, i.e., both the correlation coefficient and RSS variation are higher than the corresponding threshold, the smartphone verifies that the identical packets are sent from the IoT device in close proximity. Then the smartphone verifies the public key by decrypting the packets and checking their contents. Upon successful decryption and content checking, the smartphone encrypts a random shared session key using the verified public key and sends it to the IoT device. Finally, a secure channel is established between the smartphone and the IoT device. The smartphone further encrypts the WiFi password with the session key and sends it to the IoT device, which will decrypt it using the established key and then securely connect to the user's WiFi network.

B. Enhancements to Move2Auth

We identify three major issues that Move2Auth has not well addressed as follows. *First*, although Move2Auth provides a usable proximity-based authentication solution for SDP, energy-efficiency is not deliberately considered in its design. *Second*, it only provides one-way authentication, meaning the smartphone can authenticate and verify IoT devices, but not vice versa. In this case, the MITM attacker could impersonate the smartphone and falsify the messages sent to the IoT device. For example, the MITM attacker can use an IoT device's public key to encrypt a password that corresponds to a rogue access point (AP) to launch session hijacking attacks. *Third*, Move2Auth uses RSA to encrypt the messages and password. However, RSA operation is commonly considered computationally expensive.

Next, we propose mechanisms to address these issues: 1) to reduce transmission energy consumption at IoT device by minimizing the number of identical packets transmissions; 2) to improve security by adding extra steps at the end of the Move2Auth protocol to detect a MITM attacker who might impersonate the smartphone; and 3) to enhance the computational energy-efficiency by replacing RSA with ECC.

1) ***Minimize number of identical packets****:* By examining the Move2Auth protocol, we can find that in the beginning of the protocol, the IoT device needs to send a series of identical packets (i.e., smartphone's MAC address encrypted by the private key) to the smartphone. The smartphone will obtain an RSS trace based on these packets, derive the acceleration information, and then check whether the derived acceleration matches the accelerometer readings in order to verify if it is communicating with the IoT device in proximity.

Intuitively, if we transmit the identical packets at a higher frequency, we will get more samples in the RSS trace given a relatively fixed time period for the pairing process (e.g., about 3 seconds as reported in [5]). This will potentially give us a better estimate of acceleration and result in higher confidence in the computed correlation coefficient, which translates to better detection/verification performance. However, higher transmission frequency also implies higher energy consumption on the IoT device. Therefore, there is a tradeoff between energy-efficiency and security strength in the packet transmission process. We attempt to determine a transmission frequency that can be implemented in the device pairing process to achieve good performance while also maintaining low energy consumption as much as possible.

According to existing empirical studies (e.g., [6]), the energy consumption of packet transmission can be modeled as a linear function of the number N of transmitted packets as follows:

$$E_{\text{tot}}(N) = E_{\text{tx}} N + \kappa, \tag{1}$$

where E_{tx} is the per-packet energy consumption for packet transmission and κ accounts for the packet-independent energy consumption overhead. We would like to choose (N, ρ_{th}) so as to minimize $E_{\text{tot}}(N)$ while satisfying expected security performance, i.e., achieving a false alarm rate below a threshold $\alpha = 5\%$ and a detection rate above a threshold $\beta = 95\%$. Note that ρ_{th} is a decision threshold. If the computed correlation coefficient between the derived acceleration from RSS trace and the readings from accelerometer is higher than ρ_{th}, Move2Auth assumes there is no attack, otherwise an alarm is raised and the pairing process has to be rebooted.

Therefore, our optimization problem (OPT) is formulated as follows:

$$\text{OPT:} \quad \min_{N, \rho_{\text{th}}} E_{\text{tot}}(N)$$

subject to:

$$\int_{-1}^{\rho_{\text{th}}} f(r, N, \rho_0)\ dr < \alpha_{th} \text{ [false alarm rate]} \tag{2}$$

$$\int_{-1}^{\rho_{\text{th}}} f(r, N, \rho_1)\ dr > \beta_{th} \text{ [detection rate]} \tag{3}$$

$$N_{\min} \leq N \leq N_{\max}, \quad \rho_1 < \rho_{\text{th}} < \rho_0, \tag{4}$$

where ρ_0 and ρ_1 are the population correlation coefficient under no attack and attack cases, respectively. These two values can be estimated from empirical tests. A typical setting can be $\rho_0 = 0.8$ and $\rho_1 = 0.3$. Here, $N_{\min}$ and $N_{\max}$ are the lower and upper bounds of the number of transmitted identical packets, and are tunable parameters. The value of $N_{\min}$ should be larger than the minimum number of packets needed to compute accelerations (e.g., to derive one acceleration value, three RSS measurements are needed) and the correlation coefficient. The upper bound $N_{\max}$ is determined by the physical packet transmission capacity and packet size. It can range from a few hundred to thousands. The functions $f(r, N, \rho_0)$ and $f(r, N, \rho_1)$ are probability density functions (pdfs) of the sample correlation coefficient (r) under the no attack (ρ_0) and attack (ρ_1) cases, respectively. We note that as more samples are taken, the pdf $f(r, N, \rho_i)$ becomes more concentrated around its population correlation coefficient ρ_i, which implies a more accurate estimate.

In principle, $f(r, N, \rho)$ depends on the distribution of the samples used to compute the correlations coefficient as well. Usually, no closed-form representation of $f(r, N, \rho)$ exists. To facilitate numerical analysis, we assume that the derived acceleration trace and the measured acceleration trace follow a bivariate normal random distribution given as follows [7]:

$$f(r,N,\rho) = \frac{1}{\pi}(N-2)(1-r^2)^{\frac{N-4}{2}}(1-\rho^2)^{\frac{N-1}{2}}\sqrt{\frac{\pi}{2}} \cdot \frac{\Gamma(N-1)}{\Gamma(N-\frac{1}{2})}(1-\rho r)^{-(N-\frac{3}{2})}{}_2F_1\left(\frac{1}{2},\frac{1}{2},\frac{2N-1}{2},\frac{\rho r+1}{2}\right), \quad (5)$$

where $\Gamma(\cdot)$ is the gamma function, ${}_2F_1(\cdot,\cdot,\cdot)$ is the hypergeometric function, and ρ is the population correlation coefficient.

According to the property of $f(r, N, \rho)$, if we cannot find a threshold ρ_{th} to satisfy constraints (2) and (3) at some N, we will not be able to find a threshold to satisfy the constraints at a smaller N. On the other hand, if we can find a threshold ρ_{th} to satisfy constraints (2) and (3) at some N, we will always find a threshold to satisfy the constraints at all larger values of N. Leveraging this fact, we design a binary search algorithm to find the minimum N^* and an appropriate threshold ρ_{th} for Problem OPT as shown in Algorithm 1. The search step s for finding ρ_{th} can be set as 0.01 to achieve a desirable precision.

*2) **Detect MITM attacker who might impersonate smartphone:*** As mentioned at the beginning of Section IV-B, Move2Auth only provides one-way IoT-to-Smartphone authentication. A MITM attacker can still impersonate a smartphone and send an IoT device a falsified session key followed by a WiFi password corresponding to a rogue AP. To counter such an attack, we propose an extra challenge-response process (CRP) at the end of the Move2Auth protocol as follows:

- CRP$_1$ step: After the smartphone verifies the IoT device's public key (K_{pub}), it encrypts the session key (K_s) and a challenge (n_c, a nonce) and sends the ciphertext $E_{K_{pub}}(K_s||n_c)||E_{K_s}(\text{password})$ to the IoT device, where || denotes concatenation.
- CRP$_2$ step: The IoT device decrypts the received message using its private key (K_{priv}) and then uses K_s to decrypt the WiFi password. It further encrypts the nonce n_c and sends $E_{K_s}(n_c)$ as a response to the challenge.
- CRP$_3$ step: After receiving the response, the smartphone decrypts it using K_s, and then checks if the plaintext is n_c. If it is, it means the IoT device received the session key generated by smartphone as well as the correct password. Otherwise, a MITM attack is detected and the pairing process will be rebooted.

Algorithm 1 Solve OPT (binary search)

```
Input: $\alpha$, $\beta$, $\rho_0$, $\rho_1$, $s$, $N_{\min}$, $N_{\max}$
Output: $N^*, \rho_{\text{th}}$
beg = $N_{\min}$; end = $N_{\max}$; $N^* = -1$;
while beg $\leq$ end do
    mid = $\lfloor(\text{beg} + \text{end})/2\rfloor$; find = 0;
    for $\rho = \rho_1$ to $\rho_0$ with step $s$ do
        if (mid, $\rho$) satisfies constraints (2) and (3) then
            end = mid − 1;        ▷ To search the lower half
            $N^*$ = mid; $\rho_{\text{th}} = \rho$; find = 1;
            break;
        end if
    end for
    if find == 0 then
        beg = mid + 1;        ▷ To search the higher half
    end if
end while
```

*3) **Replace RSA with ECC:*** It is well known that Elliptic Curve Cryptography (ECC) is much more energy-efficient than RSA [8]–[10]. However, ECC cannot be directly used to encrypt a message. We adopt the ECC-based hybrid encryption/decryption cryptosystem [11], which uses Elliptic Curve Diffie-Hellman Key Exchange (ECDH) to derive the session key for AES-GCM symmetric encryption. Now we assume the IoT device generates an ECC public and private key pair (I_a, I_b) and sends identical packets carrying its public key (I_a) to smartphone. We can use the same mechanism used in Move2Auth to verify that this public key is sent from an IoT device in proximity to the smartphone. In order to apply ECC, we modify steps CRP$_1$ to CRP$_3$ as follows.

- ECC-CRP$_1$: The smartphone generates its own ECC public key and private key pair (S_a, S_b) and then generates the session key by multiplying its own private key and the IoT device's public key: $K_s = I_a \star S_b$. Note that this multiplication operation is point multiplication defined in ECC. Then the smartphone sends $E_{K_s}(n_c||\text{password})||S_a$ to IoT device.
- ECC-CRP$_2$: Upon receiving the message from the smartphone, the IoT device derives the session key $K_s = I_b \star S_a$, then decrypts n_c and WiFi password, and then sends n_c back to the smartphone.
- ECC-CRP$_3$: The smartphone verifies whether the received response equals n_c, which implies that a session key K_s is established between the IoT device and the smartphone and the password has been successfully transmitted to the IoT device.

Security Analysis: If the attacker intercepts the challenge message in CRP$_1$ by changing it to $E_{K_{pub}}(K'_s||n'_c)$, the IoT device will not be able to generate a legitimate response $E_{K_s}(n_c)$ in CRP$_3$. The user can detect this. The attacker could also launch a denial-of-service (DoS) attack by jamming/corrupting the response sent in CRP$_2$ and then the smartphone will not receive any response in CRP$_3$. To detect such an attack, the smartphone can trigger a timer after CRP$_1$. If the timer expires before a response comes back, a DoS attack is suspected and then the user can reboot the pairing process as well, or adopt other anti-jamming mechanisms. Defending against jamming attacks in the device pairing process is beyond the scope of this paper, but is a worthwhile topic for further investigation.

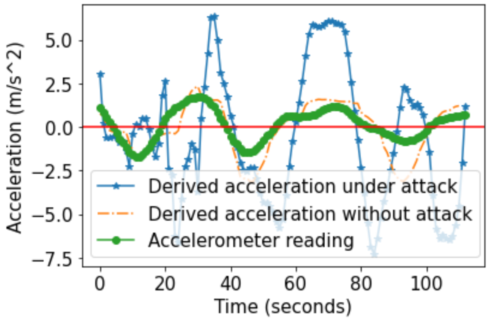

Fig. 2. Derived acceleration traces and accelerometer readings

V. Performance Evaluation

In this section, we present results from the experimental evaluation of our proposed device pairing protocol and demonstrate its energy-efficiency.

A. *Experimental Setup*

Our experiment used three Raspberry Pi4s to simulate the IoT, smartphone, and attacker, respectively. The connection between IoT device and smartphone is set up by utilizing Panda PAU05 WiFi dongles with ad-hoc mode enabled. The smartphone uses a monitor interface to extract the RSS and timestamp of each received packet using TCPDump [12]. The acceleration trace is derived from the RSS data using the same method described in Move2Auth. The smartphone also obtains accelerometer readings from a Sense HAT(B) [13] attached to the Pi4s. Both the baseline Move2Auth protocol and the enhanced version with the additional CRP steps and the adoption of ECC were implemented.

B. *Correlation Coefficient*

To evaluate the reliability under an insecure environment, we simulate a MITM attack with a third Raspberry Pi 4 during the pairing process, which is placed one meter away from the two pairing devices. The malicious device duplicates identical packets in the channel so that the smartphone captures two different sets of RSS data and converts them into the distance, velocity, and eventual acceleration data. The two different acceleration data are compared with the acceleration reading from the smartphone accelerometer via the correlation coefficient. Figure 2 shows the three acceleration traces derived from the IoT device, the MITM device, and the accelerometer reading, respectively. We can see the IoT derived acceleration data matches the accelerometer readings very well. However, the MITM derived acceleration data does not match the accelerometer readings. This validates the proximity-based authentication concept of Move2Auth.

C. *Optimal Selection of N and ρ_{th}*

We present numerical results on the optimal joint selection of the correlation coefficient threshold ρ_{th} and the transmission packet number N according to problem OPT. We set the population correlation coefficient values as $\rho_0 = 0.8$ and $\rho_1 = 0.3$, corresponding to the no attack and attack scenarios, respectively. In Fig. 3, we have plotted the receiver operating characteristic curves (ROC) for $N = 10, 20, 30, 50, 80$. The vertical line at $x = 0.05$ and the horizontal line at $y = 0.95$ represent the constraint lines on the false alarm and detection rates thresholds (i.e., $\alpha = 5\%$ and $\beta = 95\%$), respectively. From Fig. 3 we see that the constraint lines intersect the ROC curve corresponding to $N = 20$. Note that the ROC curves above this $N = 20$ curve all satisfy the constraint conditions with larger values of N. Applying Algorithm 1, we obtain $N^* = 20$ and $\rho_{\text{th}} = 0.62$ as the optimal parameter values. With different population correlation coefficients ρ_0 and ρ_1, we find ROC curves similar to those in Fig. 3.

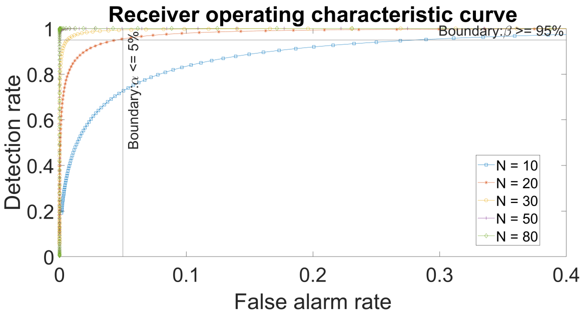

Fig. 3. ROC curve of detection rate vs. false alarm rate.

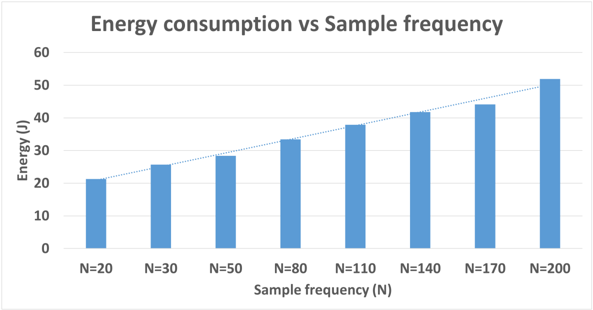

Fig. 4. Energy consumption vs. number of packets.

D. *Energy Consumption Measurement*

We used the UM25C voltage and current meter [14] to measure the energy consumption with different numbers of transmitted packets in the pairing protocol. This result validates the energy consumption model we adopted in the OPT

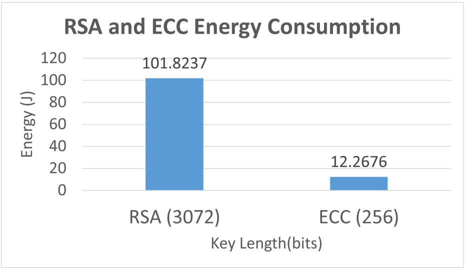

Fig. 5. RSA and ECC energy consumption

problem. Fig. 4 displays the relationship between energy consumption and number of packets. It is obvious that energy consumption rises as the number of samples increases. A higher sampling frequency can improve the pairing process but also consumes more energy. Based on our empirical studies, we found that $N = 20$ provided a good tradeoff between performance and energy consumption. In our experiment with $N = 20$, the total energy consumption was 21.25 J. When $N = 200$, this value more than doubled to 51.92 J.

Figure 5 compares the energy consumption between the CRP process (based on RSA) and ECC-CRP (based on ECC) at a 128-bit security level. Clearly, the RSA-based mechanism consumes almost 10 times the energy of the ECC-based hybrid solution. This validates the energy-efficiency enhancement proposed in Section IV-B3.

VI. Discussion

We have focused on energy consumption incurred by packet transmissions and cryptographic operations during the movement-based device pairing protocol. The Move2Auth protocol also requires the smartphone to measure the acceleration using an accelerometer. A higher sensing frequency will also incur higher energy cost on the smartphone. On the other hand, a higher sensing frequency provides a larger number of acceleromenter readings, which leads to a more accurate estimate of the sample correlation coefficient, resulting in better security performance. Therefore, there is a tradeoff between energy consumption due to sensing and security performance. The sensing-based energy consumption could be integrated into the objective function of problem OPT to account for this tradeoff. Our methodology could also be applied to analyze and optimize the energy consumption of other SDP protocols the require either transmission of identical packets (e.g., "Good Neighbor" [3]) or use of inertial sensors (e.g., "Shake Well Before Use" [1] and "MagPairing" [4]).

The impact of energy depletion attacks (EDAs) on the device pairing protocol deserves further investigation. Such attacks can be launched from the physical layer to the application layer [15]. An EDA attacker could corrupt MAC layer frames, causing an excessive number of retransmissions, which in turn could could consume significantly more energy than in normal operation. The MITM attack and large junk encrypted data can incur considerable energy costs at the MAC and physical layers as well as interruption of the device sleep mode or packet forwarding actions, resulting in additional energy consumption at the network layer.

VII. Conclusion

We presented a new method for device pairing that enhances the energy-efficiency and security of an existing proximity-based scheme, Move2Auth, by minimizing the number of packet transmissions and finding an appropriate correlation coefficient decision threshold. We also replaced the RSA encryption used in Move2Auth with ECC-based hybrid encryption, which further reduces energy consumption for heterogeneous IoT pairing process. ECC maintains the same security level with a smaller key size and is a promising approach to defend against future quantum cryptographic attacks under these resource-constrained platforms [16]. For our future work, we will further investigate energy consumption from inertial sensors and examine the impact of EDAs on device pairing protocols and design EDA countermeasures.

VIII. Acknowledgement

This material is based on research sponsored by Defense Advanced Research Projects Agency (DARPA) under agreement number HR001120C0154. The views, opinions, and/or findings contained in this article are those of the author(s) and should not be interpreted as representing the official views or policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the Department of Defense.

References

[1] R. Mayrhofer and H. Gellersen, "Shake well before use: Intuitive and secure pairing of mobile devices," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 792–806, 2009.

[2] X. Li, Q. Zeng, L. Luo, and T. Luo, "T2pair: Secure and usable pairing for heterogeneous IoT devices," in *ACM SIGSAC*, 2020, pp. 309–323.

[3] L. Cai, K. Zeng, H. Chen, and P. Mohapatra, "Good Neighbor: Secure pairing of nearby wireless devices by multiple antennas," in *NDSS*, 2011.

[4] R. Jin, L. Shi, K. Zeng, A. Pande, and P. Mohapatra, "Magpairing: Exploiting magnetometers for pairing smartphones in close proximity," in *IEEE Conf. on Comm. and Netw. Security*, 2014, pp. 445–453.

[5] J. Zhang, Z. Wang, Z. Yang, and Q. Zhang, "Proximity based IoT device authentication," in *IEEE INFOCOM*, 2017.

[6] P. Kryszkiewicz, A. Kliks, L. Kulacz, and B. Bossy, "Power consumption for single technology wireless transceivers," in *IEEE WoWMoM*, 2020.

[7] J. F. Kenney and E. S. Keeping, *Mathematics of Statistics*. D. van Nostrand, 1951.

[8] Z. Vahdati, S. Yasin, A. Ghasempour, and M. Salehi, "Comparison of ECC and RSA algorithms in IoT devices," *Journal of Theoretical and Applied Information Technology*, vol. 97, no. 16, 2019.

[9] K. S. Kumar and R. Sukumar, "Achieving energy efficiency using novel scalar multiplication based ECC for Android devices in Internet of Things environments," *Cluster Computing*, vol. 22, no. 5, pp. 12 021–12 028, 2019.

[10] L.-Y. Yeh, P.-J. Chen, C.-C. Pai, and T.-T. Liu, "An energy-efficient dual-field elliptic curve cryptography processor for Internet of Things applications," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 9, pp. 1614–1618, 2020.

[11] S. Nakov, M. Stefanov, and M. Shideroff, *Practical Cryptography for Developers*, 2018. [Online]. Available: https://cryptobook.nakov.com/

[12] "TCPdump," https://www.tcpdump.org/, [Online, accessed 19-July-2022].

[13] "Waveshare Sense HAT B," https://www.waveshare.com/wiki/Sense_HAT_(B), [Online, accessed 13-July-2022].

[14] "MakerHawk UM25C USB Tester," https://www.makerhawk.com/, [Online, accessed 10-July-2022].

[15] V.-L. Nguyen, P.-C. Lin, and R.-H. Hwang, "Energy depletion attacks in low power wireless networks," *IEEE Access*, vol. 7, pp. 51 915–51 932, 2019.

[16] T. M. Fernández-Caramés, "From pre-quantum to post-quantum IoT security: A survey on quantum-resistant cryptosystems for the Internet of Things," *IEEE Internet of Things Journal*, vol. 7, no. 7, pp. 6457–6480, 2019.