EGR 4830 – Senior Project

Secure Robotic Environment for NASA Artemis Program using XMSS Algorithm

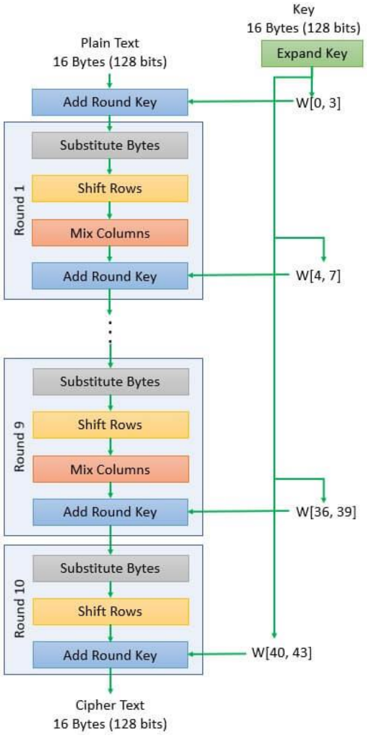
Luna, Michael
012981748

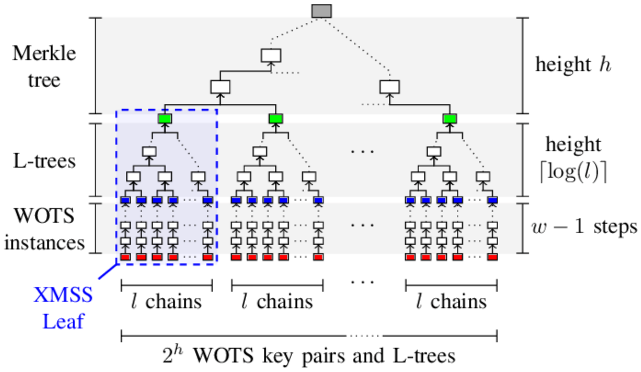
Mohamed El-Hadedy
Computer Engineering
05/09/2023

1. How long did it take to complete this executive summary? 2 days
2. Did you attend or watch the lesson provided by Mr. Paul Hottinger regarding Information Literacy (check the correct box)? ☒YES ☐ NO

Secure Robotic Environment for NASA Artemis Program using XMSS Algorithm

As technology continues to advance, the need for secure communication channels becomes increasingly crucial. With the emergence of quantum computing, current cryptographic schemes are at risk of being broken, threatening the security of sensitive data and critical applications. Post-quantum cryptography offers a solution to this problem by providing cryptographic primitives that are resistant to attacks by both classical and quantum computers. In this paper, we present a project plan for implementing post-quantum security on embedded IoT devices using AES encryption and XMSS signatures.

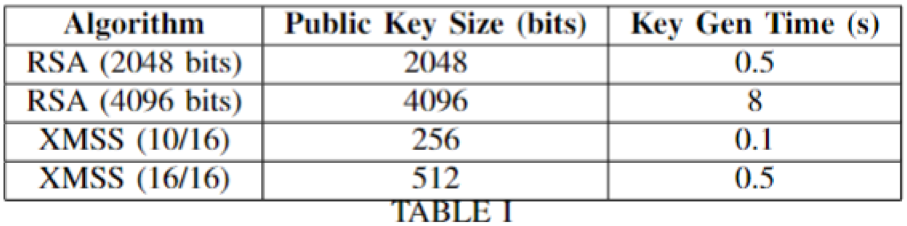




Our proposed solution focuses on hardware implementation, utilizing the PYNQ platform, and incorporates a color detection algorithm for drone verification. Throughout the project, we faced several challenges related to hardware and software issues. These challenges included the PYNQ SD card becoming corrupted, communication protocol limitations, and power distribution difficulties within the UGV system. By identifying and addressing these issues, we were able to improve the reliability and performance of the system. Our goal is to provide a robust and secure post-quantum cryptographic solution to protect embedded IoT devices used in critical applications against the threat of quantum computing. This paper not only presents our project plan but also highlights the challenges we faced during the design and implementation process, emphasizing the importance of careful attention to both hardware and software considerations in the development of complex systems.

Different perspectives on the implementation of post-quantum cryptography in embedded IoT devices for critical applications can be found in the literature. Traditional cryptographic algorithms, such as RSA and ECC, have been widely used to secure these devices. However, the emergence of quantum computing poses a significant threat to the security of these schemes. In response, researchers have begun exploring post-quantum cryptography as a solution to ensure long-term security for embedded IoT devices. Some researchers argue that traditional cryptographic schemes, such as RSA and ECC, can still provide temporary protection against quantum attacks by increasing key sizes and optimizing cryptographic algorithms. However, others advocate for the adoption of post-quantum cryptographic schemes, such as lattice-based cryptography, hash-based signatures, and code-based cryptography, to ensure long-term security for embedded IoT devices against the threat of quantum computing. In our project, we have chosen to focus on the implementation of post-quantum security using the XMSS signature scheme. We believe that XMSS provides a viable solution to the security challenges posed by quantum computing, as it offers faster key generation and smaller public key sizes compared to traditional cryptographic algorithms such as RSA. Furthermore, the use of XMSS in conjunction with AES encryption and the PYNQ platform allows for efficient hardware implementation, ensuring that our solution is not only secure but also practical for deployment in embedded IoT devices used in critical applications.

To compare the performance of two different cryptographic algorithms, RSA and XMSS, we created a table detailing the public key size and key generation time for different parameter sets.

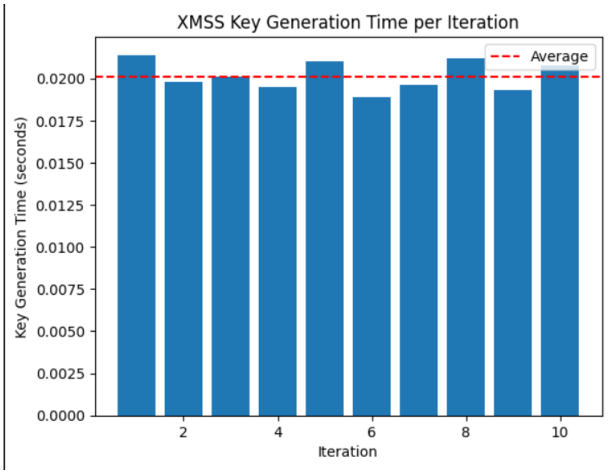


Algorithm	Public Key Size (bits)	Key Gen Time (s)
RSA (2048 bits)	2048	0.5
RSA (4096 bits)	4096	8
XMSS (10/16)	256	0.1
XMSS (16/16)	512	0.5

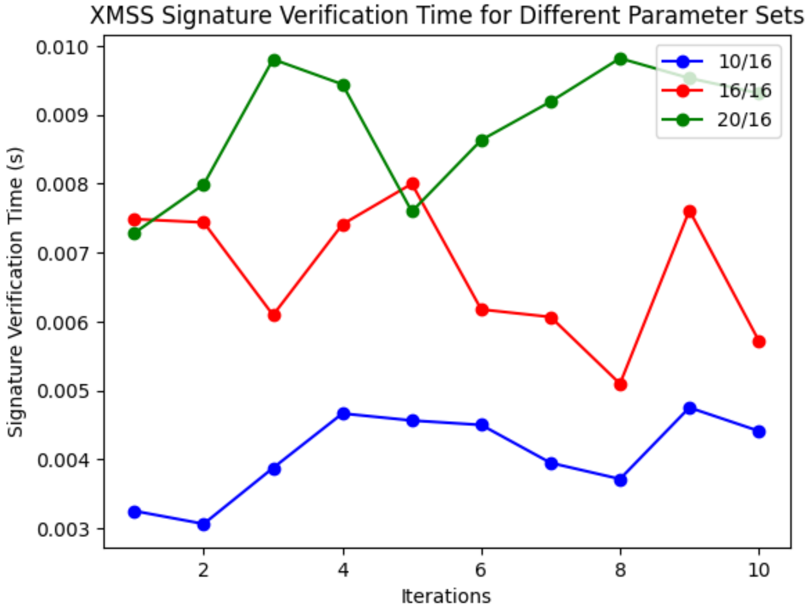
TABLE I

COMPARISON OF XMSS AND RSA PERFORMANCE

The results of our comparison as shown in the table, RSA with 2048-bit keys has a public key size of 2048 bits and a key generation time of 0.5 seconds, while an RSA signature with 4096 bits and a key generation time of 8 seconds. In contrast, XMSS with 10/16 parameter sets has a public key size of 256 bits and a key generation time of 0.1 seconds, while XMSS with 16/16 parameter sets has a public key size of 512 bits and a key generation of 0.5 seconds. The table shows that XMSS has a smaller public key size and faster key generation time than RSA for the parameter sets tested.



We conducted a key generation time test to measure the time it takes to generate a key pair using XMSS, we repeated the test 10 times to and recorded the key generation time for each iteration and found that the average key gen time was 0.0201 seconds. The figure below shows a bar graph of the key gen times for each iteration, with a red dashed line indicating the average key gen times for each iteration.



the signature verification time of the XMSS algorithm on the PYNQ Z1 board, we conducted multiple iterations of signature verification using different parameter sets. The objective was to measure the time it takes for the algorithm to verify a given signature with the corresponding public key. By recording the timestamps before and after the verification process, we obtained the signature verification time for each iteration. The results of the benchmarking test showed that the signature verification time varied depending on the parameter set used in the XMSS algorithm. In our test, we considered three parameter sets: 10/16, 16/16, and 20/16. The parameter set refers to the number of layers and the height of the Merkle tree used in the algorithm. Upon analyzing the data and plotting it on a line graph, we observed that the signature verification time increased as the parameter set became more complex. Specifically, the 10/16 parameter set had the shortest verification time, followed by the 16/16 parameter set, and finally the 20/16 parameter set, which had the longest verification time. This trend can be attributed to the computational overhead and the complexity of the cryptographic operations involved in verifying signatures with larger parameter sets.

The benchmarking results highlight the trade-off between the security level provided by the XMSS algorithm and the computational performance required for signature verification. While larger parameter sets offer stronger security against quantum attacks, they also require more computational resources and time for verification. On the other hand, smaller parameter sets provide faster verification times but may have lower security levels. Therefore, choosing an appropriate parameter set depends on the specific application's security requirements and performance constraints. It is important to note that the benchmarking results obtained from our test on the PYNQ Z1 board provide insights into the signature verification performance of the XMSS algorithm in a specific hardware setup. The actual performance may vary depending on factors such as the hardware configuration, implementation optimizations, and the size and complexity of the signatures being verified.

To conclude, our research project focused on implementing post-quantum security using the XMSS signature scheme for embedded IoT devices. Through our project plan, emphasizing hardware implementation and incorporating a color detection algorithm for drone verification, we aimed to protect embedded IoT devices used in critical applications against the threat of quantum computing. Our benchmarking results demonstrated the superiority of XMSS over traditional cryptographic algorithms in terms of key generation time and public key size, making it an efficient choice for securing embedded IoT devices. Additionally, by addressing various hardware and software challenges, we improved the reliability and performance of our system. Our project contributes to the field of post-quantum cryptography by showcasing the practical implementation of XMSS and provides valuable insights for developing secure cryptographic solutions in the quantum computing era. Further research is necessary to explore the scalability and adaptability of XMSS in different environments, but our findings lay a foundation for future advancements in post-quantum cryptography and the protection of critical IoT applications against emerging security threats.

Work Cited

[1] [P. W. Shor, ”Algorithms for quantum computation: discrete logarithms and factoring,” Proceedings 35th Annual Symposium on Foundations of Computer Science, 1994, pp. 124-134, doi: 10.1109/SFCS.1994.365700.]

[2] [T. Moore, ”Quantum Computing and Shor’s Algorithm,” Jun. 2016. [Online]. Available: https://sites.math.washington.edu/∼morrow/336 16/2016papers/tristan.pdf.]

[3] [M. Mosca, ”Cybersecurity in an era with quantum computers: will we be ready?,” IACR Cryptology ePrint Archive, Report 2015/1075, 2015. [Online]. Available: https://eprint.iacr.org/2015/1075.]

[4] [Y. Cao et al., ”An Efficient Full Hardware Implementation of Extended Merkle Signature Scheme,” in IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 69, no. 2, pp. 682-693, Feb. 2022, doi: 10.1109/TCSI.2021.3115786.]

[5] [N. Li, ”Research on Diffie-Hellman key exchange protocol,” in Proceedings of the 2010 2nd International Conference on Computer Engineering and Technology, 2010, pp. V4-634-V4-637, doi: 10.1109/ICCET.2010.5485276.]

[6] [Y. Song, X. Hu, W. Wang, J. Tian, and Z. Wang, ”High-Speed and Scalable FPGA Implementation of the Key Generation for the Leighton-Micali Signature Protocol,” in Proceedings of the 2021 IEEE International Symposium on Circuits and Systems (ISCAS), 2021, pp. 1-5, doi: 10.1109/ISCAS51556.2021.9401177.]

[7] [A. Ruggeri, A. Galletta, L. Carnevale, and M. Villari, ”An Energy Efficiency Analysis of the Blockchain-Based extended Triple Diffie-Hellman Protocol for IoT,” in Proceedings of the 2022 IEEE Symposium on Computers and Communications (ISCC), 2022, pp. 1-6, doi: 10.1109/ISCC55528.2022.9912773.]

[8] [M. B¨orsig, S. Nitzsche, M. Eisele, R. Gr¨oll, J. Becker, and I. Baumgart, ”Fuzzing Framework for ESP32 Microcontrollers,” in Proceedings of the 2020 IEEE International Workshop on Information Forensics and Security (WIFS), 2020, pp. 1-6, doi: 10.1109/WIFS49906.2020.9360889.]

[9] [A. Srebro, ”RobotKit 4WD BTS7960 IR SRF05 FollowMe,” GitHub, 2018. [Online]. Available: https://github.com/srebroa/Arduino/blob/master/ArduinoMega2560/RobotKit 4WD BTS7960 IR SRF05 FollowMe/RobotKit 4WD BTS7960 IR SRF05 FollowMe.ino.]

[10] [M. Kumar and P. Pattnaik, ”Post Quantum Cryptography(PQC) - An overview: (Invited Paper),” in Proceedings of the 2020 IEEE High Performance Extreme Computing Conference (HPEC), 2020, pp. 1-9, doi: 10.1109/HPEC43674.2020.9286147.]