Analyzing the Energy Cost of Cryptographic Ciphers on an Android Smartphone

REU fellow: Michael Bolot[1]
Faculty Mentor: Tao Zhang[2], Ziqian Dong[2]
Affiliation: [1] University of Dallas, [2]New York Institute of Technology
Emails: mbolot@udallas.edu, {tzhang, ziqian.dong}@nyit.edu

ABSTRACT

This paper compares the power efficiency of AES, Blowfish, RC4, RC6, Curve 25519, and a custom implementation of XXTEA on a Nexus 5 smartphone through the use of PowerTutor. The results demonstrate that Lightweight Cryptographic algorithms are not categorically better than their conventional counterparts for power efficiency on an android device, and that algorithm security and key generation cost should be the primary concerns when encrypting smaller files. In addition, the results demonstrate that asymmetric algorithms remain categorically less power efficient than symmetric algorithms.

BACKGROUND

Problem

- Smartphones are becoming used more frequently for important tasks, from banking to monitoring vital signs as in WBAN networks[1]
- Many of these tasks require encryption to maintain the confidentiality and integrity of user's information
- It is vital that encryption require as little power as possible to maintain functionality of the phone

Potential Solutions

Prior Work

- A prior NYIT team compared the power cost of AES, Blowfish, and RC4
- However, these results may be inaccurate [2]
- The selection of algorithms is narrow
- No Lightweight or Asymmetric Algorithms
- Doesn't include decrypt or key generation costs

Lightweight Cryptography

- Lightweight Algorithms are designed for power efficiency and computational simplicity
- They aren't typically considered suitable for smartphones [3]
- XXTEA [4] was chosen for simplicity of implementation [5] to reduce error due to implementation

Elliptical Curve Cryptography

- Elliptical Curve Algorithms are more efficient than commonly used asymmetric algorithms [6] by a significant margin [7]
- Differ from conventional cryptography in foundational elements, which involve calculations done over elliptical curves (shown in figures 1 and 2 below)

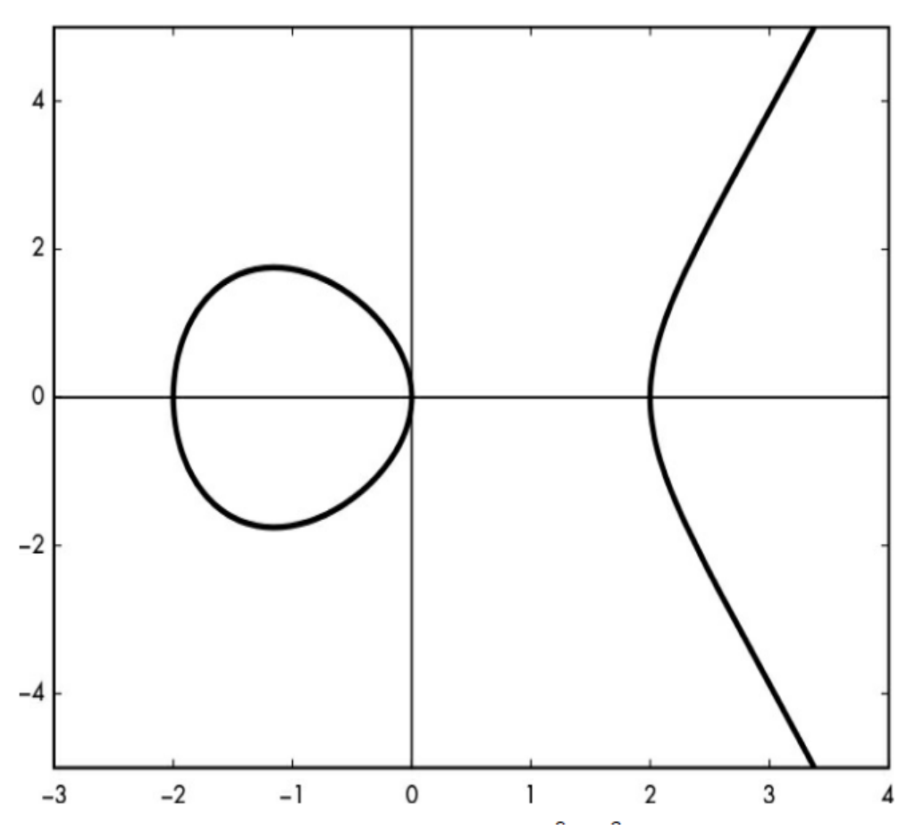


Figure 1: An elliptic curve with the equation $y^2 = x^3 - 4x$, shown over the real numbers [6]

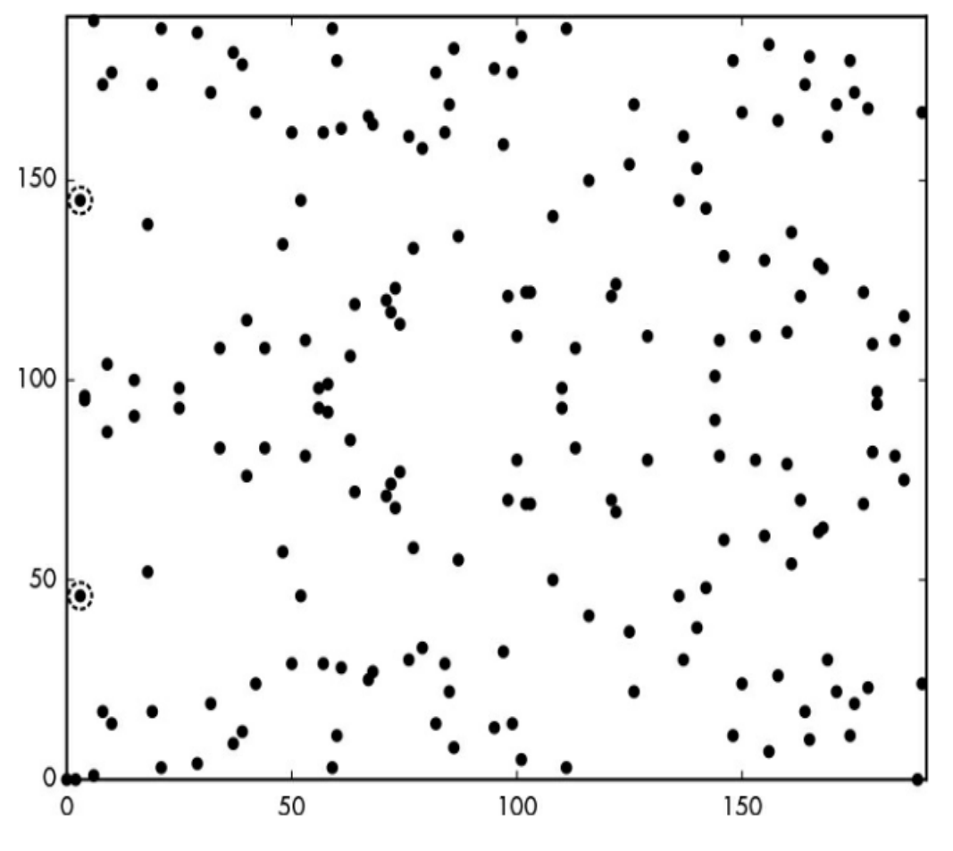
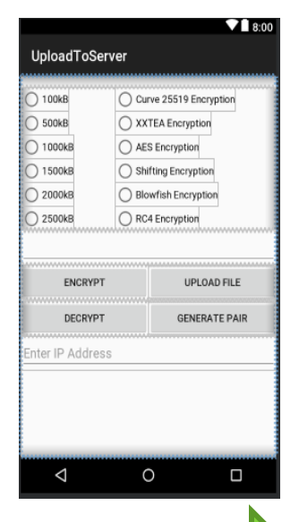


Figure 2: The elliptic curve with the equation $y^2 = x^3 - 4x$, shown over $\boldsymbol{Z}_{191}$, the set of integers modulo 191 [6]

Methodology

Data Generation

- The UploadToServer app (pictured below), which was originally designed by [1], was modified and used for encryption, decryption, and key generation
- Files of size 100kB, 500kB, and 1500kB were encrypted and decrypted
- Key size was 128 bytes for all symmetric algorithms
- SpongyCastle default was used for Curve 25519
- Algorithms available were: AES, Blowfish, RC4, RC6, Curve 25519, and XXTEA
- Power Usage Data was collected by PowerTutor in batches of 5 results per file size and algorithm combination
- Encryption, Decryption, and Key Generation results collected separately
- 25 total records per file size and algorithm combination (and 25 per algorithm for key size)



Data Collection and Display

- PowerTutor outputs the collected results to a log file (one log file to one record)
- These log files are collected onto an external system and parsed by python
- The python scripts extract the power usage data for 30 seconds (for encryption or decryption records) or 15 seconds (for key generation records) of the application running
- These results are then placed into excel for display

RESULTS

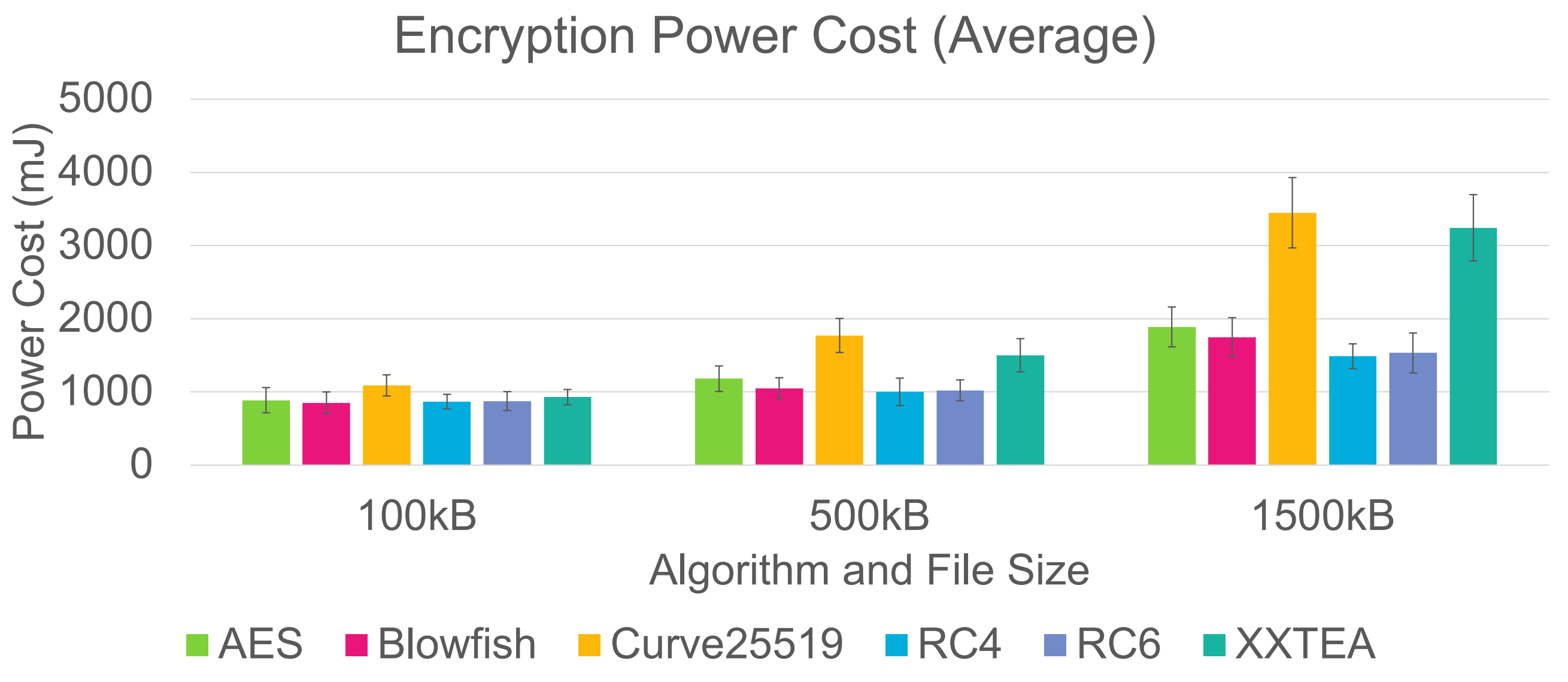


Figure 3: Encryption Results

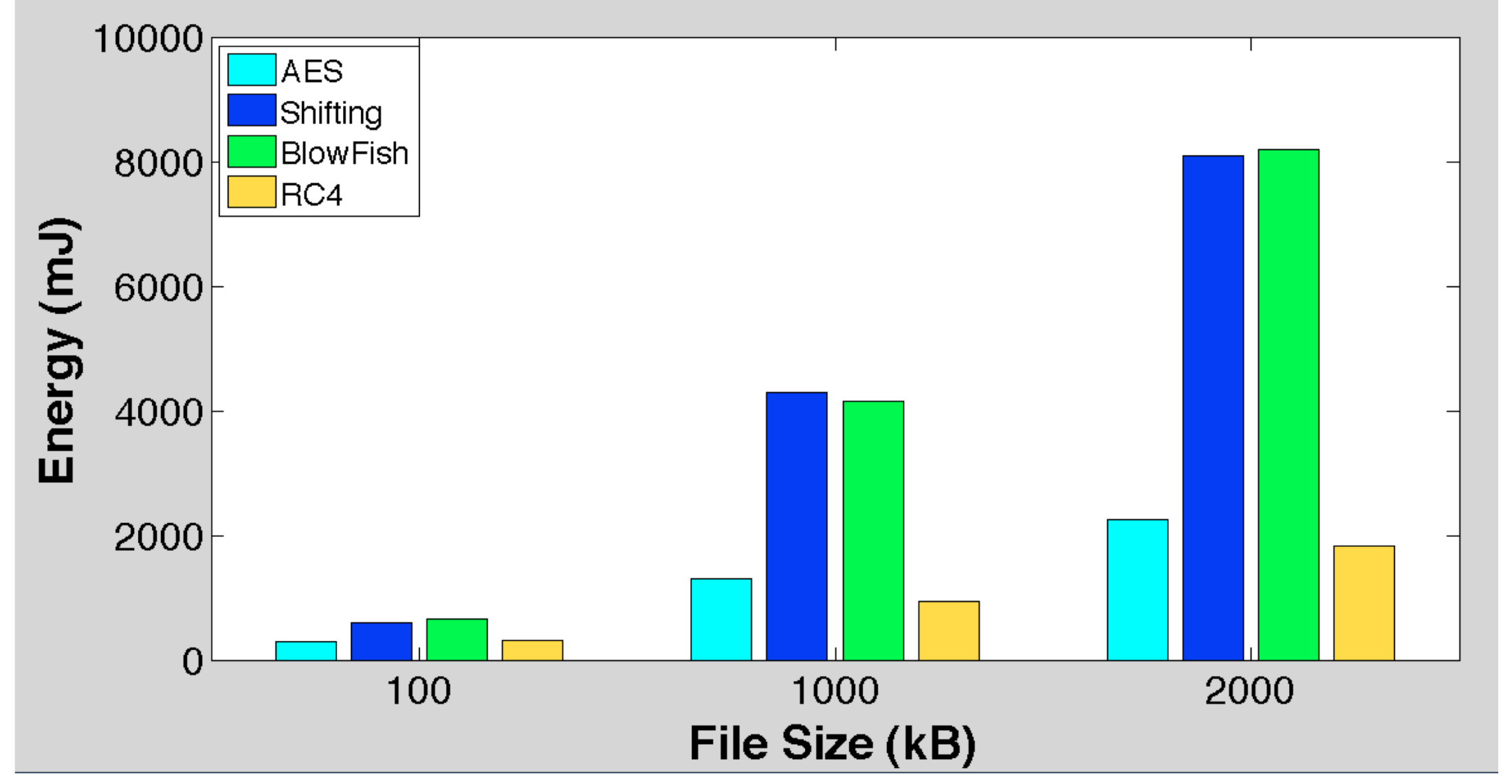


Figure 4: Prior Work Encryption Results [1]

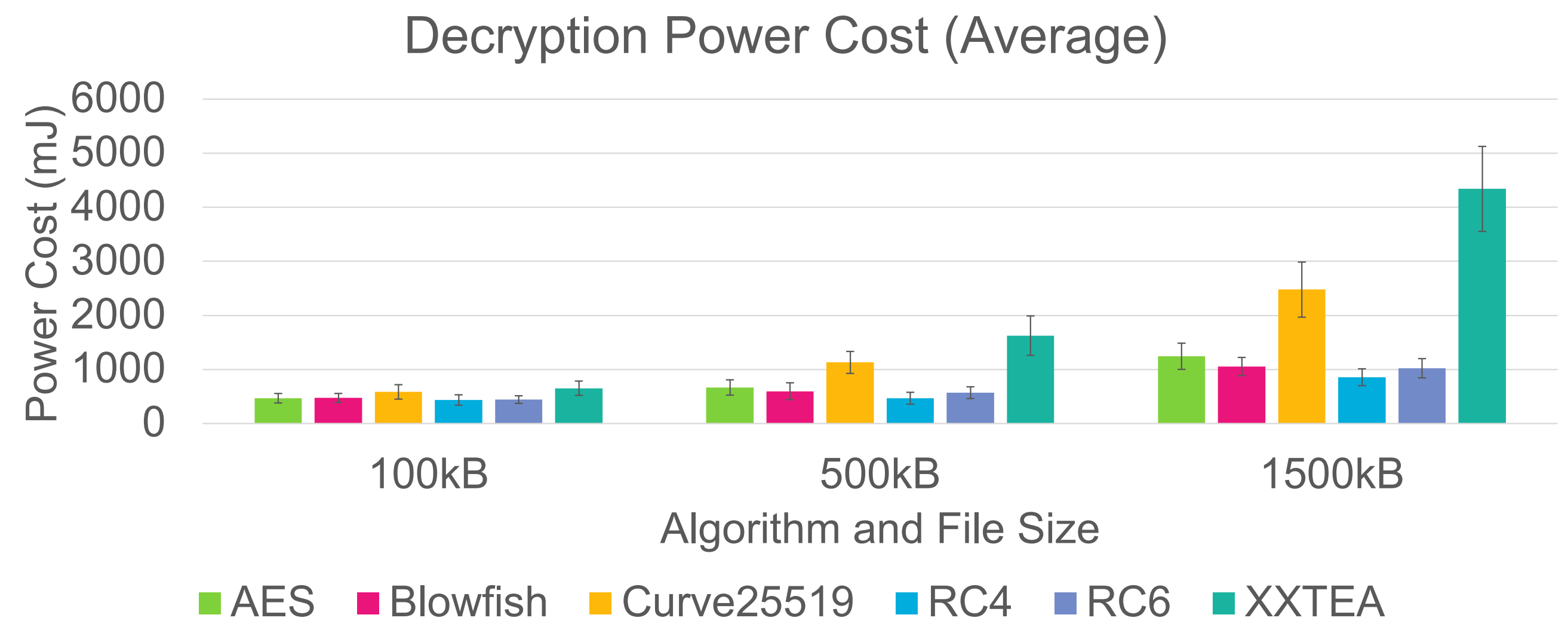


Figure 5: Decryption Results

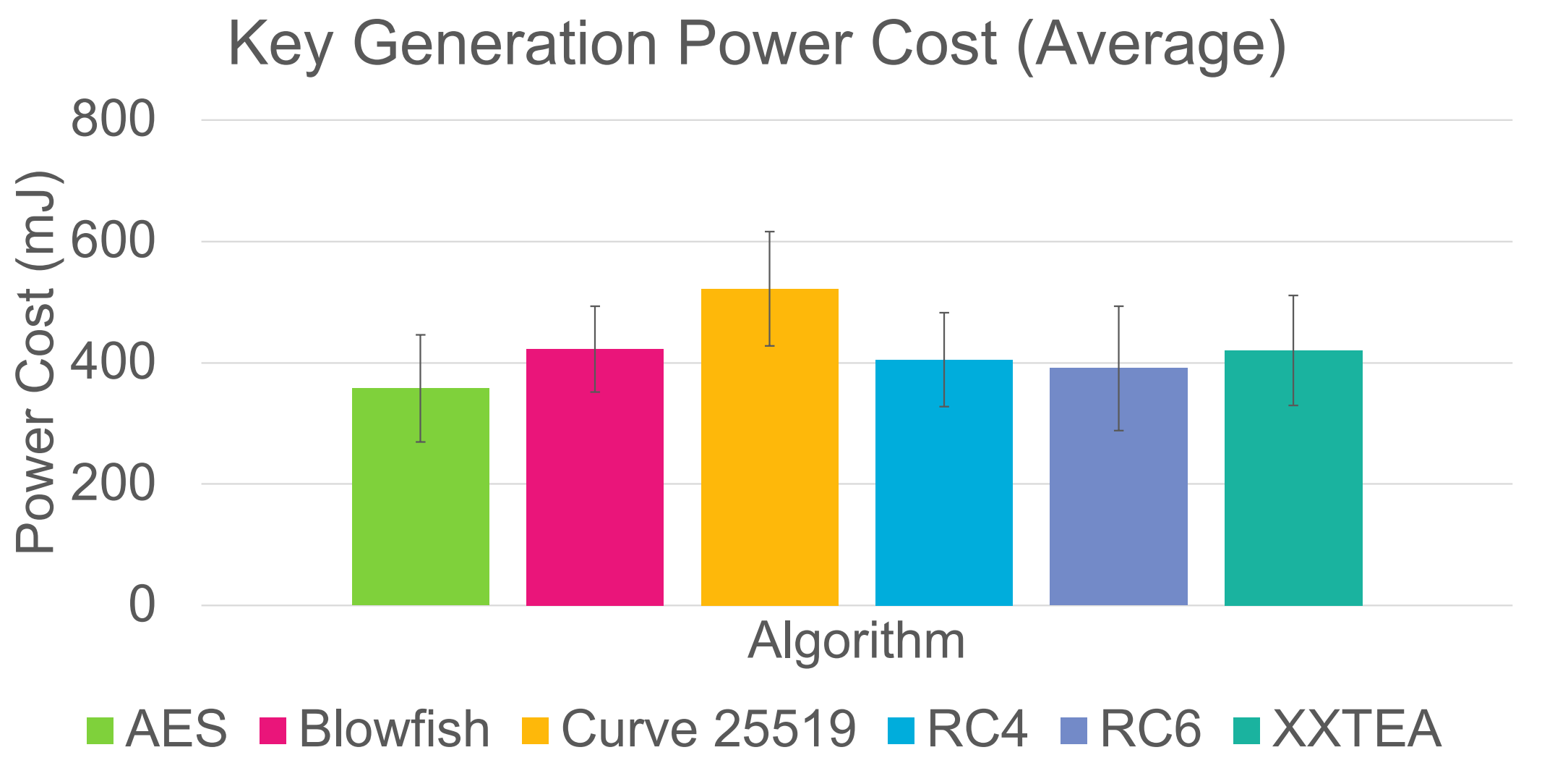


Figure 6: Key Generation Results

RESULTS ANALYSIS

Encryption

- As seen in figure 3, the stream ciphers (RC4/RC6) performed the best
- Curve 25519 performed the worst, followed by XXTEA
- The power cost differences between algorithms is much less at 100kB than at 500kB and 1500kB
- Results were much more consistent with [3] than the results seen in figure 4

Decryption

- The Decryption results in figure 5 were largely consistent with the encryption results in figure 3
- XXTEA performed the worst
- This was possibly due to issues with the implementation (specifically the padding)
- The stream ciphers remained the most power efficient
- Curve 25519 was still very inefficient compared to the other algorithms

Key Generation

- As figure 6 shows, AES performed the best in Key Generation
- These results should be taken with skepticism, as PowerTutor records results in mJ and key generation costs as seen in [3] can be in μJ
- Curve 25519 was again the most inefficient

CONCLUSIONS

- Lightweight algorithms are not categorically better for power efficiency on Android devices
- Given the small differences in power cost of algorithms when encrypting lower file sizes, the security of algorithms and key generation cost should be the paramount concerns when dealing with small file sizes
- Stream ciphers were the most efficient at encryption and decryption, so lightweight stream ciphers might be the most power efficient lightweight algorithm
- RC4 was more efficient (when accounting for Key Generation cost, which was lower with RC6) but the power cost difference between RC4 and RC6 was negligible
- While some inefficiency of XXTEA may be caused by implementation details, it is still far less efficient than the similar algorithms (AES, Blowfish) that were also tested

FUTURE WORK

- Expand testing to include other lightweight algorithms (such as sparx, speck, or SEA), specifically lightweight stream ciphers (such as Grain, Trivium, or Mickey)[3]
- Expand testing to include other standard algorithms, including those in Android's standard library
- Develop a more precise and reliable tool to measure power use of each individual application

REFERENCES

[1] C. DelBello, K. Raihan, and T. Zhang, "Reducing energy consumption of mobile phones during data transmission and encryption for wireless body area network applications," Security and Communication Networks,8.

[2] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, "Analyzing the energy consumption of security protocols," Proceedings of the 2003 Internal Symposium on Low Power Electronics and Design, no. 03, 2003.

[3] K. A. Mckay, L. Bassham, M. S. Turan, and N. Mouha, "Report on lightweight cryptography,"National Institute of Standards in Technology, 2017.

[4] D. J. Wheeler and R. M. Needham, "Correction to xtea," October 1998.

[5] ——, "Tea: A tiny encryption algorithm," October 1998.

[6] J. Aumasson, Serious Cryptography. No Starch Press, 2018.

[7] R. Alvarez, C. Caballero-Gil, J. Santonja, and A. Zamora, "Algorithms for lightweight key exchange," Sensors (Basel), Jun 2017.

ACKNOWLEDGEMENT

This research was supported by the National Science Foundation (NSF) Research Experiences for Undergraduates (REU) program. We would like to thank all of the mentors and research fellows at the New York Institute of Technology who have provided their helpful insight and expertise that greatly assisted with our research.

The project is funded by National Science Foundation Grant No. CNS-1559652 and New York Institute of Technology.