Final Report

Max Dawson

Evan Hassan

Marcel Jonas

CSI-4480

**Due date: April, 16, 2019**

**Introduction/Background Information:**

When a user encrypts confidential data, it is crucial he/she be able to choose a specific access control to determine who will be able to decrypt this data. Enter Ciphertext Policy Attribute Based Encryption. CP-ABE is a system for realizing complex access control on encrypted data. Specifically, a user’s private key with a set of attributes and encrypted ciphertext will specify an access policy over attributes. A user will be able to decrypt if and only if his/her attributes satisfy the ciphertext’s policy. The encryption method was developed in 2006 by John Bethencourt, Amit Sahai, and Brett Waters in a response to a need for a more expressive type of access control. As sensitive data is increasingly stored across many servers, particularly cloud storage, it becomes increasingly difficult to guarantee confidentiality. CP-ABE encryption method along with other variations to date, is a useful tool for achieving access control policy.

**Project goal/Problem solved:** Investigate the Ciphertext Policy Attribute-Based Encryption (CP-ABE) cryptosystem, complete encryption / decryption of ciphertext, and evaluate the performance impact on a Kali Linux system with large numbers of unique attributes.

**Project Schedule:**

|  |  |
| --- | --- |
| **Task** | **Deadline** |
| Research & Analysis | 2-13-19 (2 weeks) |
| Cryptographic tools & testing | 2-27-19 (2 weeks) |
| Midterm Report & midterm presentation preparation | 3-6-19 (1 week) |
| Implementation | 3-20-19 (2 weeks) |
| Final presentation preparation | 4-3-19 (2 weeks) |
| Final report & Peer reviews | 4-16-19 (approximately 2 weeks) |

**Research task:**

Crypto Systems

Little introduction before these two topics below are discussed. In today's world it is hard to maintain data security and has become a hot topic in today's world. One of the ways necessary for data security is to use encryption and that can be many times of system. In our research we were tasked to study two topics and how they contribute to the task of securing data.

Homomorphic encryption(HE) scheme is a cryptosystem with the additional property that there exists an efficient algorithm to compute an encryption of the sum or the product, of two messages given the public key and the encryptions of the messages but not the messages themselves (Zamani…). The way we can visualize this is X1 is a plain text message and X2 is a plaintext message, we desire to put those message together with an encryption key on and then decrypt them with both messages being together. Such as, E(X1)+E(X2) = E(X1+X2) and will become X1 + X2 when decrypted.

Full Homomorphic Cryptosystems(FHE) which supports both addition and multiplication Such a system also preserves the useful algebraic properties(Zamani…). Which can be very useful to certain degree for data security, but is very taxing on the system due to its large demand of processing power. There are several efficient PHE methods but research on efficient FHE schemes is still ongoing. In the near future it is to believe that the rise of power of computing FHE systems will become efficient enough that the tax they currently hold will be manageable.

Partially Homomorphic Cryptosystems(PHE) are the more efficient of the two systems at the moment and are useful for encrypting plain text there are several types of PHE encryption, like RSA, ElGamal, and Paillier(Zamani…). In these libraries they can only support certain types of mathematical computations a good example of this would be RSA vs. Paillier libraries. RSA library can only support multiplication while Paillier can support addition, multiplication by a constant. Which makes them conditional and not multiversital.

Functional Encryption (FE) enables fine control of sensitive data by allowing users to only compute certain functions for which they have a key(Agrawal...). Vast majority of work that is done around FE is focused on deterministic functions, although some may have auditing functions as well. FE is interesting in the sense of control over the data being accessed in the file and could have some practical uses. One such way we could use FE is a form of blurring parts of picture that contain a patient in a hospital to protect that person's privacy, and in this FE has the best use. The other topics discussed before had an all or nothing approach to encryption and decryption of files. It is a good approach, but it lacks the customizations of access that would be very useful for broad aspects of privacy matters.

**Motivation/Importance:**

Security concerns are a primary focus for information systems, given the fact that cloud computing has fundamentally changed the spectrum of computers, storage, and services. Additionally, the emergence of social media platforms have generated more concerns about confidentiality of information. CP-ABE is a countermeasure that provides security and access control. It allows encryption and decryption of data based upon attributes or credentials that describe users. The data owner chooses eligible users to access information. Particularly CP-ABE can be used to encrypt the data outsourced to cloud storage. The key is to establish access control for the encrypted data. CP-ABE and variations thereof are a viable solution for access control in cloud computing and storage. Thus explains our motivation behind our selection of the CP-ABE cryptosystem.

**Implementation:**

After, selecting and investigating the cpabe toolkit, we began working toward implementation. Implementation of our testing started with creating scripts to assist in testing. Several sample scripts were generated to determine what was viable. Once we had created a script to automate the task of key generation, encryption and decryption we had to come up with a way of testing large numbers of unique attributes. We wanted to test up to ten thousand and this required us to create a script that would generate large numbers of attributes. We had multiples scripts that would create, add and remove these attributes from our automation script. Once all of our scripts were created we went into the testing phase. The testing phase consisted of testing 1,10, 50 ,100 ,1000 ,5000 ,10000 attributes to see the performance impact on the system and the time it took to generate a key, encrypt and decrypt a text file. The findings were documented and we concluded our project with these findings being accurate with what we predicted would happen to the system’s performance.

**Challenges: GMP line 67 code missing ;**

In the implementation stage it took some fine tuning to get the cpabe-0.11 and libbswabe-0.9 working properly and together. One of the things that had to be done is to install a PBC library which is ciphertext-policy attribute-based encryption scheme which was built on the GMP library which had a line code error which was missing a (;) , so that was needed to be installed first then the PBC library then install the libbswade library first because that is the first part of the library which had to be allowed certain permissions ($ cd gmp-6.1.2/mpn and chmod +x m4-ccas) and then finally installing the cpabe library. As you can see it took quite a lot of configuring and searching forums to find the exact solutions to the errors the libraries had.

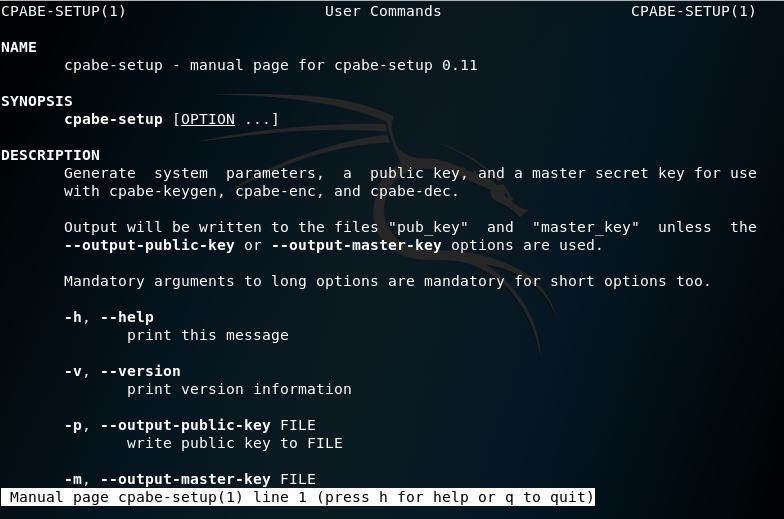
\*Note: We have tried doing other libraries and they require ATLAS library install and that is currently very difficult due to it requiring CPU throttling off.

**Technical reference dictionary for cpabe:**

# Generates a public key and a master secret key.

$ cpabe-setup

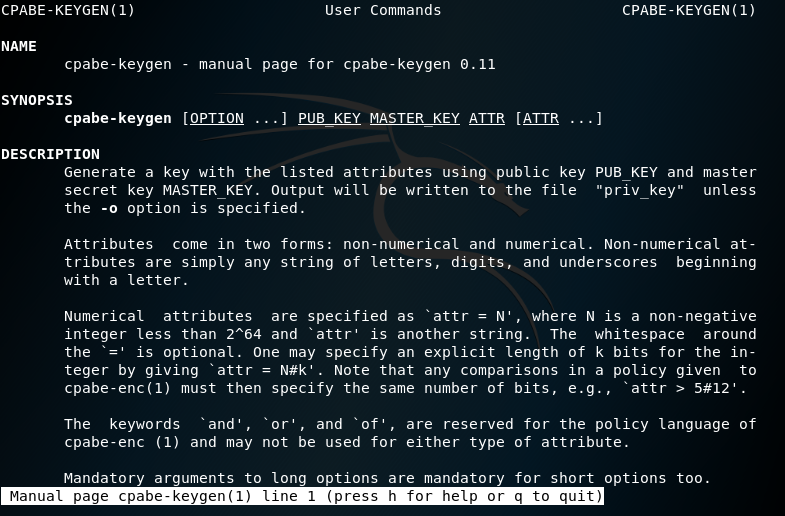
Screenshot of cpabe-setup manual page:



# Generates a private key with a given set of attributes.

$ cpabe-keygen

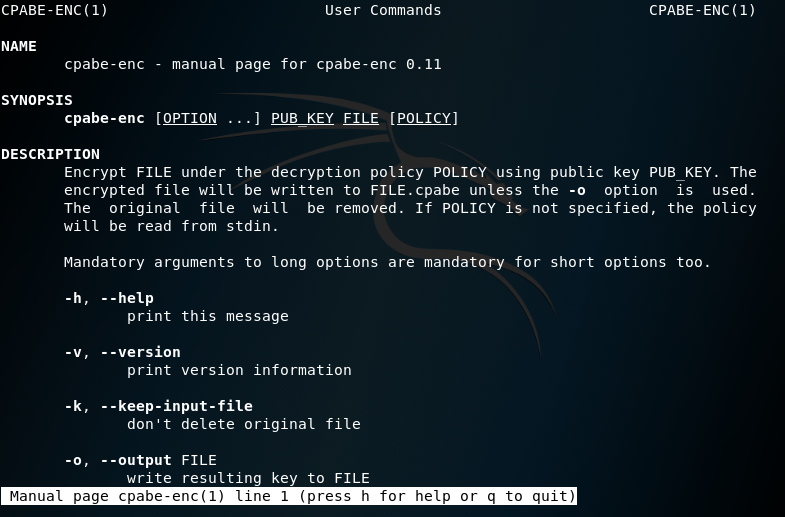
Screenshot of cpabe-keygen manual page:



# Encrypts a file according to a policy, which is an expression in terms of attributes.

$ cpabe-enc

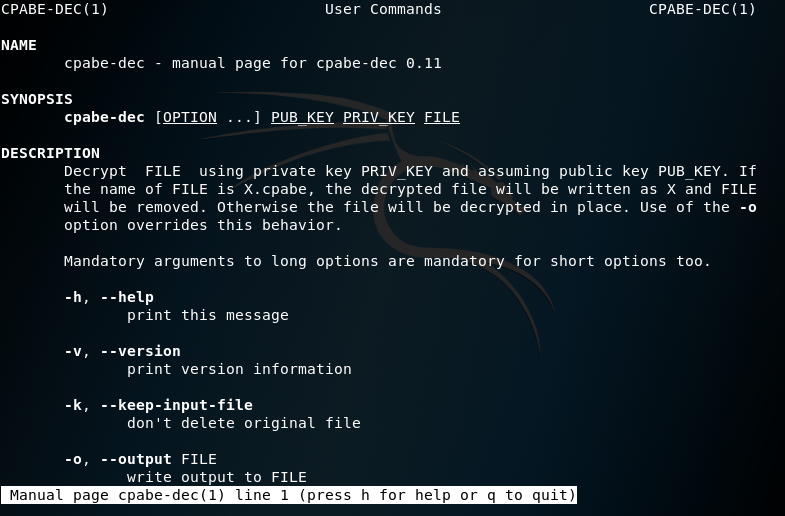
Screenshot of cpabe-enc manual page:



# Decrypts a file using a private key.

$ cpabe-dec

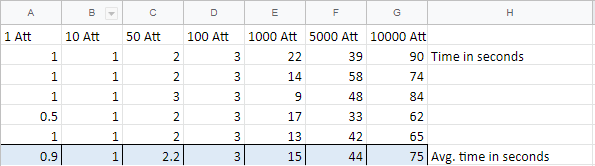
Screenshot of cpabe-dec manual page:



**Outcome of testing:**

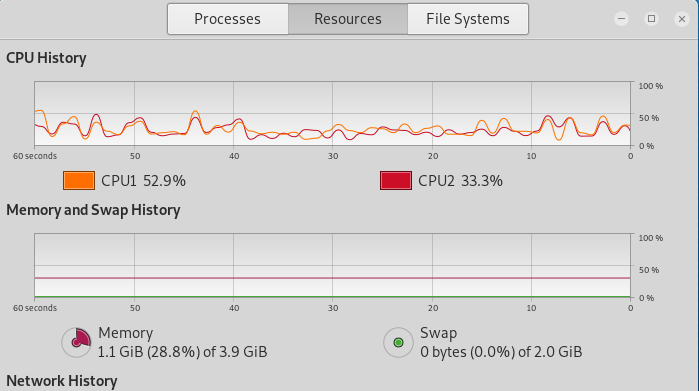
During testing we had the systems CPU usage monitored during each test for the different number of attributes. We tracked the execution time for each test run of the automation script that would generate the key with attributes, encrypt and decrypt a text file.

Documented time outcome



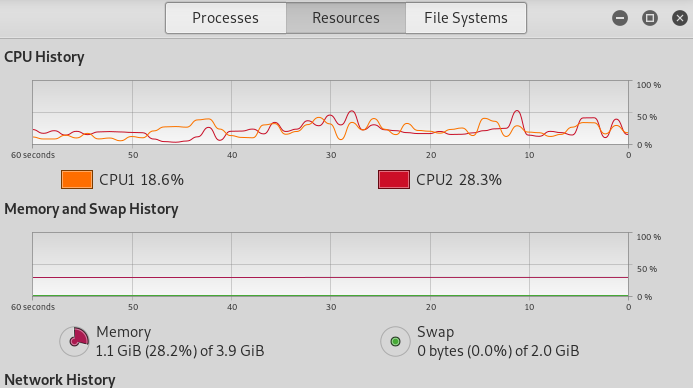
Noticeable variation in time it took for each test. Due to the variation in times for each test, an average of 5 runs was taken for each number of attributes.

1 Attribute test



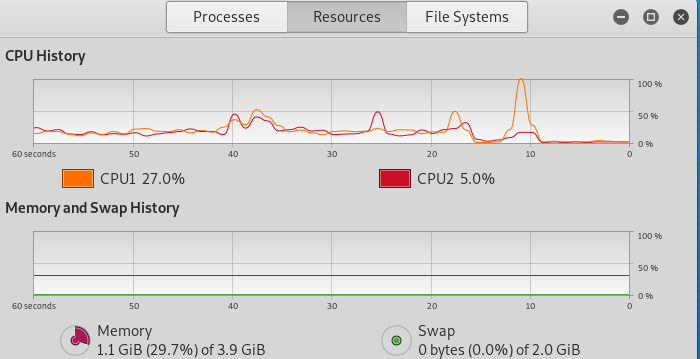
Test of 1 attribute shows no noticeable change in CPU usage.

10 Attribute test



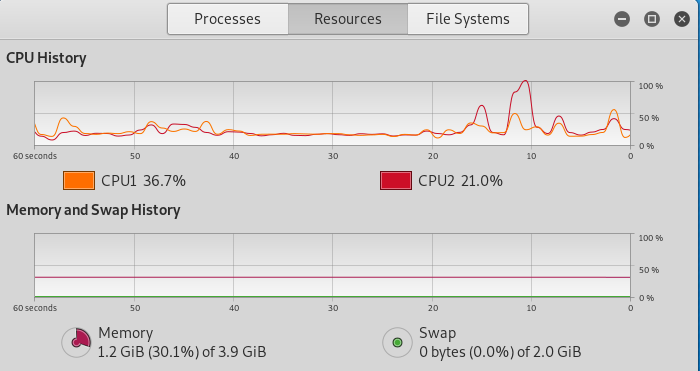
Test of 10 attributes shows no noticeable change in CPU usage.

50 Attribute test



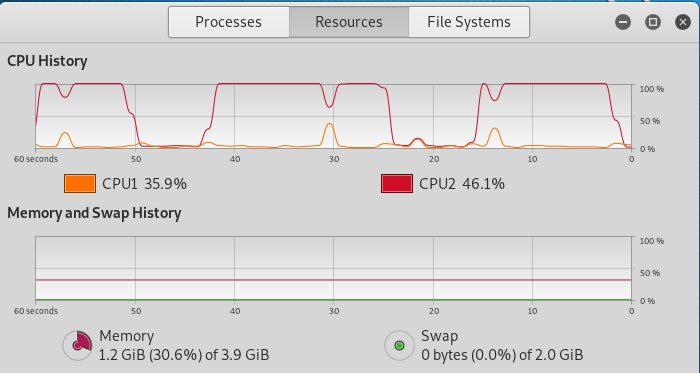
Test of 50 attributes shows noticeable change in CPU usage for short duration of time.

100 Attribute test



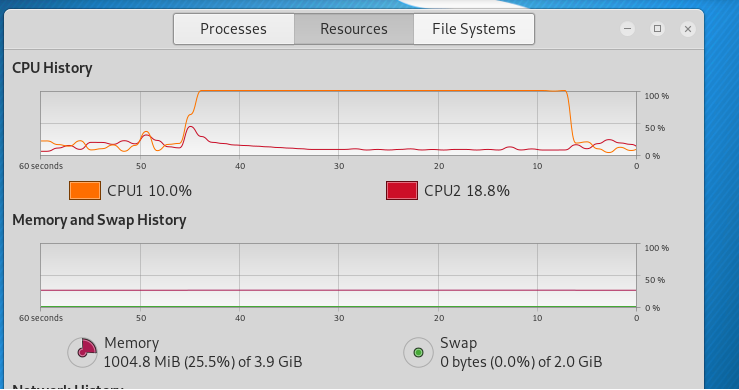
Test of 100 attributes shows noticeable change in CPU usage for short duration of time.

1000 Attribute test



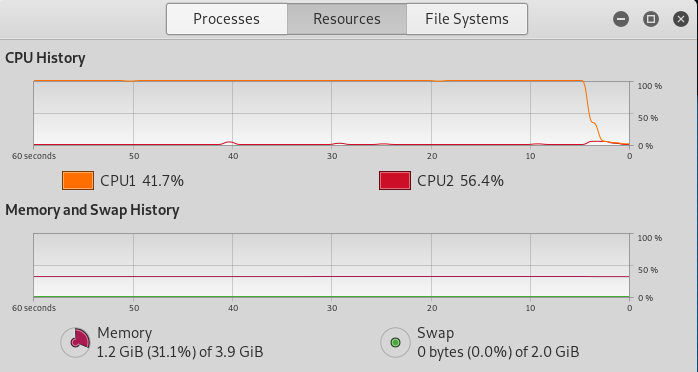
Test of 1000 attributes shows noticeable change in CPU usage for long duration of time. Image shows multiple tests of 1000 attributes.

5000 Attribute test



Test of 5000 attributes shows noticeable change in CPU usage for long duration of time. The CPU usage is used for a significantly longer duration of time compared with previous tests.

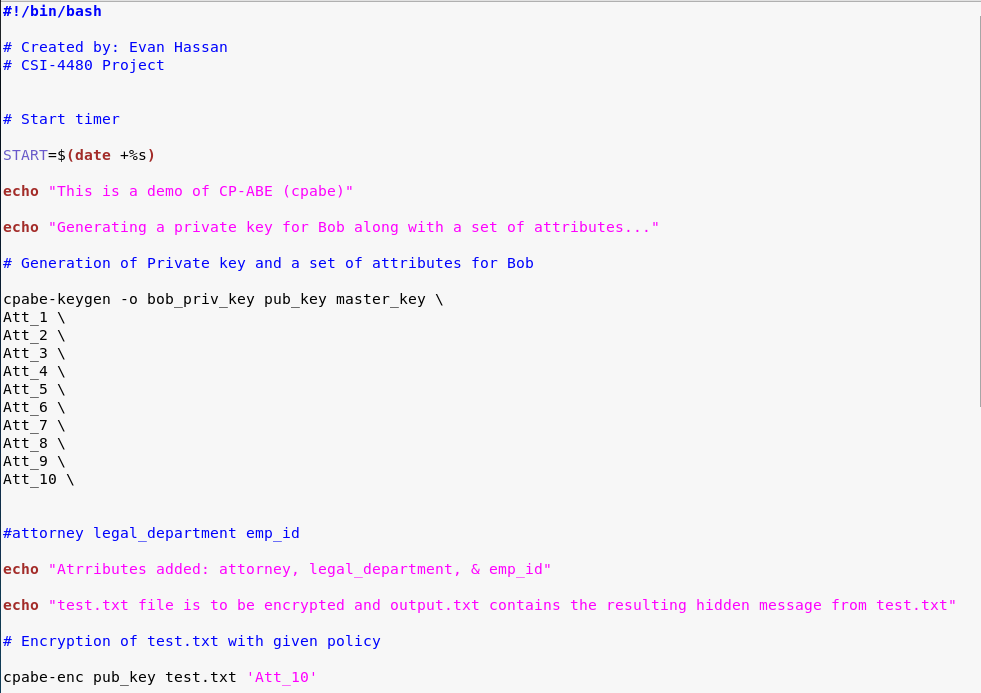
10000 Attribute test

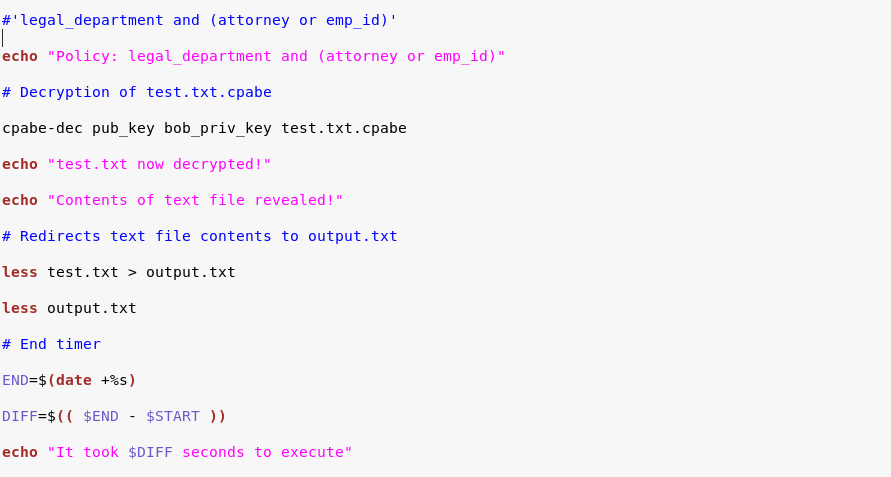


Test of 10000 attributes shows the most significant amount of CPU usage for large duration of time compared to previous tests.

**Tools Created By Group:**

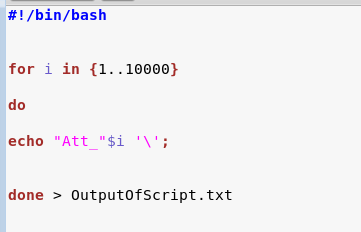
test.sh (test automation script)





Script created to run the CP-ABE automatically when the script runs rather than doing it manually.

WriteScript.sh (attribute creation script)



Script used to generate large numbers of attributes and write the attributes to a text file.

OutputOfScript.txt (container file that holds created attributes)

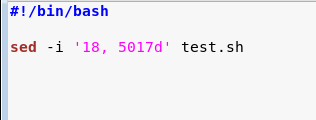


AddAttributesToFile.sh



Sends attributes contained in txt file to test.sh automation script file

RemoveAttFromFile.sh



Removes attributes from automation file when used

**Link to Final video demonstration:** [**https://drive.google.com/a/oakland.edu/file/d/1d71oyjGB18wPBIk1abzy7sLk1EUD3Blc/view?usp=sharing**](https://drive.google.com/a/oakland.edu/file/d/1d71oyjGB18wPBIk1abzy7sLk1EUD3Blc/view?usp=sharing)

**Conclusion:**

Our overall goal in testing was to determine the performance impact on a Kali Linux system using multiple attributes with CP-ABE. We created scripts to help facilitate the testing of this goal. We tested our goal and found no significant deviations from what we assumed would happen. Our implementation and testing was successful and we were satisfied with the outcome.

**Summary of contributions:**

Max

* Research of CP-ABE topics
* Creation of attribute generation and implementation scripts
* Testing of the CP-ABE with different amounts of attributes
* Documenting testing outcomes

Evan:

* Research of CP-ABE & cpabe toolkit
* Developed initial automation script template (test.sh)
* Testing and fine-tuning of usable script (test.sh)
* Created plus outlined documents (e.g. reports and presentations)

Marcel -

* Research of the CP-ABE topics
* CP-ABE crypto library setup and configuration
* Assistance with implementation of scripts

References:

1. Lai, Junzuo & Deng, Robert & Yang, Yanjiang & Weng, Jian. (2014). Adaptable Ciphertext-Policy Attribute-Based Encryption. 10.1007/978-3-319-04873-4\_12.
2. Bethencourt, John & Sahai, Amit & Waters, Brent. (2007). Ciphertext-Policy Attribute-Based Encryption. Proceedings - IEEE Symposium on Security and Privacy. 321-334. 10.1109/SP.2007.11.
3. “Ciphertext-Policy Attribute-Based Encryption.” *Advanced Crypto Software Collection*, 1 Dec. 2006, <http://hms.isi.jhu.edu/acsc/cpabe/>
4. Buchanan, Bill. “Ciphertext-Policy Attribute-Based Encryption.” *YouTube*, uploaded by Bill Buchanan OBE, 3 January 2019, <https://www.youtube.com/watch?v=pd42NCGiryQ>.
5. Parit, Sneha C. and Rachh, Rashmi “Ciphertext Policy Attribute Based Encryption” *International Research Journal of Engineering and Technology (IRJET),* vol. 4, no. 4, 2017, https://www.academia.edu/33454777/Ciphertext\_Policy\_Attribute\_Based\_Encryption. Accessed 20 March 2019.
6. Zamani, M., & Movahedi, M. (n.d.). 8 Homomorphic Encryption. Retrieved April 4, 2019, from <http://mahdiz.com/crypto/basics/page-8.html#prev>
7. Agrawal, S., & WU, D. J. (n.d.). Functional Encryption: Deterministic to Randomized Functions from Simple Assumptions. Retrieved April 4, 2019, from https://www.cs.virginia.edu/dwu4/talks/UCLASeminar0916.pdf