Network Provisioning Encryption: Securing Digital Money

A Technical Report submitted to the Department of Computer Science

Presented to the Faculty of the School of Engineering and Applied Science

University of Virginia • Charlottesville, Virginia

In Partial Fulfillment of the Requirements for the Degree

Bachelor of Science, School of Engineering

Ali Ibrahim

Spring, 2022.

On my honor as a University Student, I have neither given nor received unauthorized aid on this assignment as defined by the Honor Guidelines for Thesis-Related Assignments

Daniel Graham, Department of Computer Science

Network Provisioning Encryption: Securing Digital Money

CS4991 Capstone Report, 2022

Ali Ibrahim
Computer Science
The University of Virginia
School of Engineering and Applied Science
Charlottesville, Virginia USA
ai9sk@gmail.com

Abstract

The problem my team of software engineering interns at a major bank had to solve was making encryption of payment card information more efficient and centralized into a single codebase. This was significant as it would allow for users to add their credit and debit cards to mobile wallets more quickly as well as allow future software engineers to more easily update the codebase should encryption requirements change. To solve this problem, we used an AWS serverless solution to host an API that would accomplish the encryption. To design the solution, we first settled on an architecture to host our solution within the cloud (AWS). Then we created a web API in python and finally deployed into our cloud architecture. The API correctly encrypted information for one mobile wallet without using as many resources as the previous implementation. Furthermore, our API would be able to host all the different encryption requirements in one centralized codebase. Future work requires implementation of different encryption schemes for different mobile wallets (for example, Apple Pay and GPay) as well as testing of these new endpoints.

1. Introduction

As mobile payment options become more and more popular, the use of physical currency in the form of cash and coin declines. As this trend continues, societies may inevitably reach a point at which physical money will no longer be used or required and its production will cease. If this point is reached, a futuristic society in which all monetary transactions are done digitally will usher in: a cashless society.

In a society where all transactions are handled digitally, security and efficiency are of the utmost importance. Just a few seconds of delay in handling transactions can result in severe monetary costs. Even worse consequences might occur if the mobile wallets we use to carry out these transactions are not secured because hackers might be able to steal payment information. Network Provisioning Encryption aims to solve both of these problems through making the process of adding your cards to your wallet more efficient while conforming to encryption standards.

2. Related Works

Some of the earlier efforts to secure non-cash transactions were in the form of the EMV chip. The EMV chip, a chip located on many modern credit and debit cards, is a huge step above using the magnetic stripe on the back of the card because the information on the chip is encrypted, making it much harder for hackers to steal the actual payment card information. This is in contrast to the magnetic stripe located on the back of the card in which the information is not encrypted (Payments, 2022). This was important to my project as this was one of the

earliest methods of using encryption for transactions.

Another modern method of payment that is on the rise and considered to be secure is payment by cryptocurrency. Cryptocurrencies typically use asymmetric encryption methods to secure transactions. However, most establishments do not yet accept cryptocurrencies (Seth, 2021). Cryptocurrencies are also important to my project as they are an extremely secure way to pay and rely on asymmetric encryption that most mobile wallet technologies make use of.

Now, mobile wallets such as Apple Pay, Google Pay, and Samsung Pay, as well as many others, are on the rise. Transactions done through these mobile wallets are heavily encrypted and often tokenized and are thus very secure (Apple Pay security and privacy overview, 2021). Furthermore, adoption by establishments and consumers is steadily rising for mobile wallets.

3. Process Design

There were three major phases in the development of our project, preparing the could infrastructure, actually programming the API endpoints, and finally deploying to QA for testing.

3.1 Preparing Cloud Infrastructure

Before my team could begin work on programming the various endpoints, we had to decide how to host the project on the cloud. This decision was important as we had to make sure our setup was optimal for balancing loads as well as having some redundancy in the event of failure in one region. We deployed route 53s, load balancers, and lambda functions to achieve these goals as these are highly scalable AWS resources.

3.2 Programming API Endpoints

Once the team deployed all of the cloud resources, we had to begin programming the API's. Although AWS lambda supports a variety of languages, the team chose to proceed in python due to its vast library, support within the company, and familiarity among the team members. There were 6 API endpoints the team was responsible for deploying.

The reason for different endpoints because the encryption requirements differed between the mobile wallets as well as the type of card: credit or debit as well as Visa or MasterCard. In order to program the endpoints, we had to become familiar with the various encryption requirements. To do this, we consulted documentation provided by the mobile wallets and card networks as well as previous implementations and began programming, focusing on one endpoint at a time.

3.3 Deploying to QA and Testing API

One familiar with the encryption requirements, the team programmed and ultimately deployed one endpoint but concluded work on two. As work concluded on an endpoint, the API had to was deployed to a quality assurance (QA) environment for testing. Each function had multiple unit tests. Furthermore, we tested the deployed API to make sure it could handle realistic loads as well as malformed requests.

4. Results

Although only two endpoints were completed, documentation on encryption requirements were compiled by the team for all 6 endpoints, which should make future efforts on the other endpoints much quicker to complete. Furthermore, since all of the endpoints are similar, many of the existing functions written by the team will be leveraged to finish work on the other endpoints.

Although at the time no endpoints were in production, one was production-ready. Once this endpoint is in production, the API will theoretically be more efficient than the previous implementation as well as require less maintenance as it was designed to automatically scale up and down as the load on the server changes. Additionally, the API will be easier to maintain by future team members since all of the endpoints were centralized into one codebase with all necessary documentation attached.

5. Conclusion

Although there were already functioning APIs that existed within the company that could achieve what our API could achieve, NPE was incredibly useful as it increased efficiency and was a serverless application that made maintenance much easier. Additionally, the API included functions that would be useful for a range of projects, such as functions for retrieving certain payment card information as well as calls to other services. Since these functions had to be created before any endpoints could be configured, the existence of these functions will make adding the rest of the endpoints much easier.

Furthermore, NPE was the first serverless effort on the team, therefore the documentation and deployment files were already being used for other serverless efforts by software engineers within the team.

6. Future Work

Although a lot of progress was made, the API is still far from finished. One endpoint is production-ready but still needs to be deployed into production. Two other endpoints have been completed but still need to be tested before they can be considered production-ready. The remaining three endpoints need to be coded, tested, and deployed.

Once all of the endpoints are production-ready, the entire project needs to be pushed into production. The last step would be to inform the downstream dependencies to migrate from using the old service to using NPE.

7. UVA Evaluation

My courses at UVA prepared me for a career in CS by instilling useful problem-solving skills as well as the ability to teach myself using the resources available to me. I did not necessarily use much of what I learned in class outside of basic software development techniques, but having the ability to problem-solve was incredibly important. Software development is always changing. What you learn today might not be used at all in a few years; therefore, having the ability to quickly learn and teach yourself new tech stacks is far more important than learning a couple of languages and frameworks. I believe that UVA's CS curriculum has given me that ability to quickly learn and adapt.

References

Payments, E., 2022. Why Are EMV Cards More Secure Than Traditional Debit And CC | EMSpayments. [online] EMSpayments. Available at: <https://emspayments.com/why-are-emv-cards-more-secure-than-traditional-cc/#:~:text=EMV%20chip%20cards%20were%20originally,each%20time%20it%20is%20accessed.&text=Traditional%20debit%20and%20credit%20cards%20have%20magnetic%20stripes%20that%20store%20data%20statically.> [Accessed 24 February 2022].

Seth, S., 2021. Explaining the Crypto in Cryptocurrency. [online] Investopedia. Available at: <https://www.investopedia.com/tech/explaining-crypto-cryptocurrency/> [Accessed 24 February 2022].

Apple Support. 2021. Apple Pay security and privacy overview. [online] Available at: <https://support.apple.com/en-us/HT203027> [Accessed 24 February 2022].