A Secure TSM-based Mobile Coupon Payment Protocol on the Mobile Network Operator Level

Shaghayegh Fereidounfar
Electrical and Computer Engineering Department
Villanova University
Email: Sfereido@villanova.edu

Danai Chasaki
Electrical and Computer Engineering Department
Villanova University
Email: Danai.chasaki@villanova.edu

***Abstract*—The increased rate of use of mobile phones for electronic payments has created challenges such as ensuring the security and maximum performance of the payment protocols. Lack of proper infrastructure between mobile operators and banks also causes distrust. This paper attempts to partially solve these issues through mobile coupon payments. By putting the operator in the role of the trust service manager and incorporating the sim-card into authentication, security has been significantly improved without an adverse effect on performance. We design a mobile coupon payment architecture by creating three roles: a) customer, b) retailer, and c) server, and we use the concept of mobile coupon payment by employing shared keys and combining our main three roles with a third-party centric mobile operator as the fourth role. Eventually, the number and timing of user-side calculations are reduced due to the operator's involvement as a trust manager. The security of the proposed method is validated through formal verification.**

***Index Terms*—Mobile payment, payment protocol, trust manager, shared key, mobile coupon, formal verification.**

I. Introduction

THE importance of online payments and the interest of companies in setting up customer clubs, promotion activities and offering flash discounts has continued to grow. Companies are offering a variety of reusable and one-time rebate codes on various platforms such as mobile coupons, e-discounts and loyalty points. At the same time, the mobile phone penetration rate (usually expressed as the ratio of SIM cards to the total population) has been steadily increasing globally. Mobile penetration is projected to continue to grow to 70% by 2025 [1]. As a result, the number of users using mobile payments in stores will also be increasing, highlighting the importance of enhancing efficiency and maintaining financial security. Improving security during the mobile payment process can help grow manufacturer's profits and encourage customers to buy more. It is possible for a rookie hacker to abuse current mobile payment protocols. A simple example is a malicious application that takes screenshots of the payment page during the payment process, compromising the integrity of the process. In order to alleviate security concerns, we use the concept of electronic coupons (or mobile coupons/mobile vouchers) instead of bank payment mechanisms. The direct connection of the coupon with the mobile operator will make the coupons secure and increase the security of mobile payment.

Using coupons to attract new customers, promote products or increase sales is one of the common approaches in business campaigns. Coupons are widely used in both traditional business and e-commerce. Coupons may be published electronically, in the streets, in your mail box, in product packages, in newspapers or even magazines. Coupons available online effectively reduce the cost of printed coupons. The mobile coupon is divided into two groups according to the method of communication and its application. The first type, despite being published online, still needs to be downloaded and printed for use. But in the second type, customers need to digitize traditional coupons before using coupons in virtual stores [2]. According to [3], many companies have begun issuing coupons through mobile devices and wireless connectivity enabling users to store, retrieve and use coupons without external assistance.

In this paper we are designing and implementing a secure mobile coupon protocol to improve the users' mobile shopping experience. Customers can collect loyalty points for using mobile coupons. The coupon we designed not only provides a mobile coupon mechanism but also a customer loyalty points mechanism. In addition, retailers are also able to design their own personalized shop concessions. Some of the most important uses of loyalty points and mobile coupons are: getting credit from the retailer and applying for a loan with a certain number of points, exchanging m-coupons to cash after reaching the specified limit, c) discounts on next purchase, d) buy and sell coupons between customers with specific rules to secure credit, e) public and charitable donations to unexpected events.

Many scholarly articles have discussed mobile payment security within mobile payment protocols. Generally, previous research in the field of security has suggested two approaches: a) the first one is based on secure mobile payment protocols and b) the second one is based on a trust service manager (third-party model approach). The protocol in article [4] aims at unifying many e-commerce proposals into a single framework. Article [5] proposes a secure mobile wallet using digital signature and pseudo-identity techniques. Secure outsourcing has been used to reduce the cost of unreliable server authentication. Article [6] proposes a secure mobile coupon mechanism for NFC smartphones that analyzes the performance and security of coupons. Proper settlement ensures the integrity

979-8-3315-2096-0/25/$31.00 ©2025 IEEE

of the coupon. In this model, it has been tried to put some discounts by placing a discount coupon and zeroing in on them. Article [7] illustrates examples of mobile payment protocols suitable for scenarios with communication constraints where payment security is a new challenge for multi-payment protocol design in a transaction. [8] introduces a new process for encrypted payment and the implementation of its user interface on Android. The plan has been economically and temporally viable. It is highly secure and utilizes artificial intelligence techniques. The protocol in article [9] uses oval curved encryption for more secure and efficient applications of electronic payment systems.

Although Trusted Execution Environments (TEE) and other methodologies in Table I present viable alternatives, they rely on specialized hardware and incur substantial resource overheads, limiting their feasibility in resource-constrained environments. The proposed design, on the other hand, is lightweight and scalable, utilizing the existing SIM card infrastructure to achieve comparable levels of security without additional cost. This makes it a more practical and accessible solution for a broader range of applications, especially where performance and simplicity are key considerations.

II. Security Architecture

Trust can be established in two ways. The first one is by using a blockchain, which involves a network of half-plus credentials, and the second is by using a trust-management third-party approach. Table I suggests that the second approach - using a trust manager to secure mobile coupons - has more advantages.

A. Suggested method

In this paper, the trust manager server is the SIM card. We treat the SIM card as a secure module and store the keys inside it. The selection of SIM cards as a trust management unit is justified by their widespread usage, robust security features, and interoperability with the majority of devices, which reduces the likelihood of significant compatibility issues. The proposed protocol capitalizes on these inherent advantages, providing a high degree of reliability across different platforms. By the shared key mechanism the retailer and customer receive their shared keys from the operator and do not produce the keys themselves. The duties of the trust manager are: Key production, Key distribution, Resolving Disputes. We should keep in mind that the processor does not perform significant processing; it has very little traffic and overhead because it is only responsible for authentication. In fact, even key validation is not normally the duty of the trust manager and solely solves the problem during litigation.

B. Challenges

The most important challenges in mobile payment are: a) There is no trust between mobile operators and banking payment systems, b) Mobile side security is poor due to the presence of malware and key registrars, c) Electronic payment actors don't agree on cooperating, and d) Lack of standardized and user-friendly infrastructure. To alleviate these concerns, in our work, we start with the mobile coupon payment system in reference [2], and add our innovative design. We combined that system with the concept of the trust manager described earlier and we implemented the whole design. The proposed protocol involves four main actors/roles: Vendors, Mobile coupon service provider (MCSP), Retailers, MCSP members. These roles take part in four distinct phases of the protocol: Agreement and Announcement Phase, Loyalty Receipt and Mobile Voucher Phase, Coupon Use Phase, Settlement Phase.

The basic design is shown in in Figure 1. It includes a system of loyalty points, a method to convert points into mobile coupons and ways to use those coupons. Generating loyalty points is done by a hierarchical combination of two one-way hash chains. The upper horizontal chain is the original primary chain. Suppose we have n customers and no more than 20 loyalty points are issued per purchase. The first element of the matrix is the initial value of the product w0, with the initial hash function applied to the number of clients. Once the main horizontal subsystem is complete, we move to the vertical subsystem which has a secondary hash function to be applied applied hierarchically to the vertices. Finally, one of the elements in each column of this matrix is assigned to each client after a successful transaction and must be sent to the client.

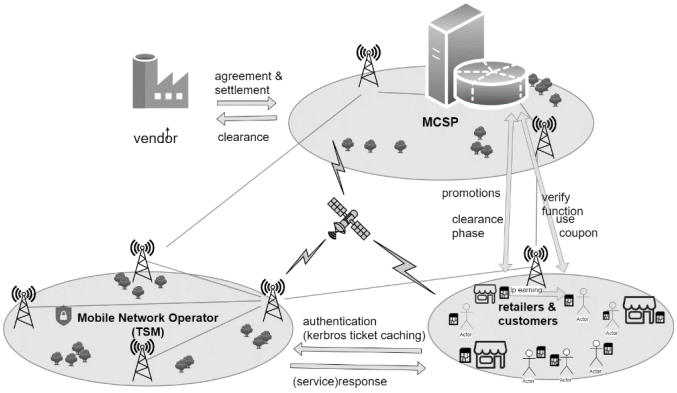

Fig. 1. Our proposed scheme

According to the architecture in Figure 1, in the first phase, vendors aim to increase sales by concluding a contract with a mobile coupon service provider and giving retailers the discount details. The retailer interacts with the customer by informing them about how to obtain loyalty points and discount offers. In fact, the retailer is indebted to the vendor by selling discounted goods to the customer and will eventually settle the discount with the vendor at a specified future time. In the second phase of the protocol, a number of customer loyalty points are awarded after each transaction from the retailer site is completed. The customer can convert it to a mobile coupon by offering a certain number of loyalty points to their server. In the third phase, the customer uses the mobile coupons and the retailer is obligated to accept the customer's redeeming choice after verifying the validity of the coupons. Finally, in the fourth

Method name	Main idea	Cons	Pros
	- Repeat FPGA coding and third-party monitoring updates	- Flexible selection of security functions and features	-Cost
			-Time
The combination of trust manager and HSM hardware [10]	- Build a shared key between the CPU and HSM while maintaining security	-Saving performance	
		- Quick update	- Complexity of hardware design and construction
	- Key generation in HSM with hardware implementation	- runtime support	- High boot time
	-Support mutual authentication with SSL certificate	-Many tenancy	- PKI and certificates are not scalable
		- No need for gate infrastructure	
Authentication and multi-tenancy with the security manager [11]	-Using the Openidconnect OAuth2 protocol to authenticate the gateway user		- No security keys
		Multilateral authentication	
	-Reading and many Api calls	-Select passwords policy	- reverse engineering for malicious user
	-Tervice to the user when authenticating to reduce user overhead computation	High performance and security	
		- High authentication speed	-Limited user time for authentication
Cloud inspection for lightweight, third-party protocol users [12]	- Protecting the privacy of users against third parties by helping to lighten the user's data	- Low overhead	- High cost of validation and validation
		- No need for user data decoding	
		- No need for Android changes	
	- With two user-level sandboxes, third-party access to sensitive api's is controlled.	- Simple practical deployment	
TEE to manage third-party Android application level [13]		-high security	-time limitation
	- Sensitive information is redirected through a proxy and a virtual machine with custom policies.	- No need to root the phone	- Performance overhead
		- Failure to disclose information from tsm	
		- Providing location-based services	- Unexpected stops
Secure third-party service management for mobile clouds [14]	- Reduce cloud computing for mobile users with the added role of trust manager	- Immediate action	-Delay
		- Cloud storage services	- Impose a high cost on drivers
Data protection in the cloud with a trusted third party [15]	- Comparison of two modes, including one key manager and several key managers in the role of trust manager	- Reduce user-side computing	- No security of outsourcing of information to third parties

TABLE I
COMPARISON OF STATE-OF-THE-ART WORKS

phase, the clearing between the retailer and the vendor takes place and the discount promotion program ends.

III. Model design

A. Design Details

1) Selecting the promotion program and gaining loyalty points: First, the user enters the desired promotion program by viewing the list of discount promotions and selecting one of them. The way to get a loyalty point is for the retailer to first connect to the database. Matrix X is examined with two indices a, b. Then the corresponding element in the string is received through the web service as a user loyalty score of a to number b in the user application. This string is actually the discount code (loyalty points).
2) Convert points to m-coupons: Customers can view their loyalty points list. They can also request coupons at any time by a request to the server. In fact, a http request containing a string of customer loyalty codes is sent to the server. Then for control, a request is sent from the server to the retailer and the code is checked for accuracy. In response, the coupon button will be activated for the customer if the code provided is a match.
3) Using a m-coupon: In the client's application, pressing the use m-coupon button causes the promotion discount on the cart.

B. Adding the Role of Operator and Trusted Service Manager (TSM) to the Scheme

Part One: Securing the relationship between the retailer and mobile coupon service provider (MCSP) with a TSM broker This is done in two phases:

- Phase One: Authentication
  1) The retailer first signs the public key based on the master key with its own private key and then encrypts it with the operator's public key and sends it to the operator as a primary request.
  2) On the other hand, the operator receives the retailer's request, decrypts it with his private key and

extracts the retailer's public key. It then creates a mutual symmetric key and places it in a secure packet, signing with its private key and sending it to the retailer for a SIM card.

3) The retailer decrypts the packet with its private key, retrieves its mutual symmetric key with the operator and stores it in the SIM card. This is illustrated in Figure 2.

- Phase Two: Building Trust
  1) The retailer sends the request for the production of the MCSP (mobile coupon service provider) symmetric key to the operator. He signs his identification information with his private key and encrypts it with his own symmetric key (retailer authentication and authorization process).
  2) The operator receives the request from the retailer and encrypts the new packet containing the shared key for the retailer and MCSP.
  3) The retailer and MCSP will each decode the package received by the operator with their own private key and obtain their shared key. This is illustrated in Figure 3.

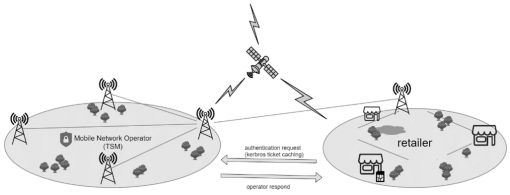

Fig. 2. Retailer authentication

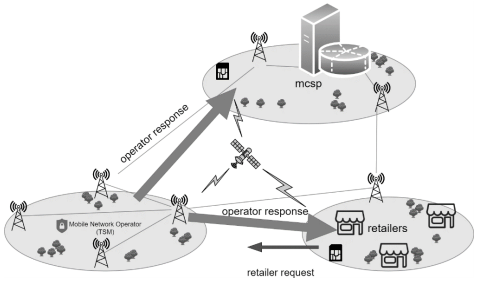

Fig. 3. Trust between the retailer and the MCSP

Part Two: Securing the relationship between member and MCSP with a TSM broker: The steps in the previous section will also be repeated for customers this time as shown in Figures 4 and 5. We have also implemented this section to simulate the operator by a web service and to simulate the sim card by the android application.

IV. Implementation

A. Framework

The MCSP, which is our primary server, runs several ReSTfull web services written in Python using Django's web

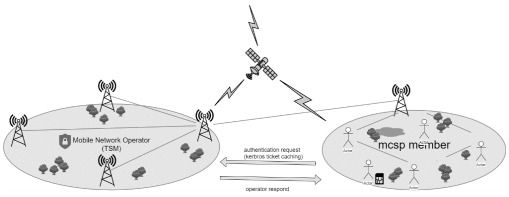

Fig. 4. Member authentication

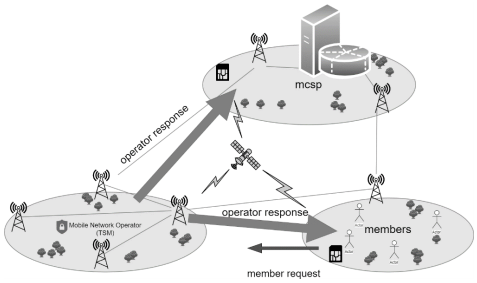

Fig. 5. Trust between the member and the MCSP

frameworks. Two Android apps were written in the Android Studio app, one for the role of retailer and the other as a MCSP member. The two apps were installed on two different mobile phones. The compatible architecture for web-services is REST; it can support the scale of the Internet. The contents of the HTTP request are sent in the sender and receiver details header and the data format is JSON. The body part contains the name/values which send the contents of matrix X (rows of loyalty points and columns of the member id) as an encrypted code that is then passed to the verification function. The client receives the number corresponding to the matrix layer and sends one of the responses to HTTP in response to the server.

A major advantage of our approach is that it can be easily generalized. This ensures compatibility and adaptability across diverse hardware and configurations.

B. Matrix Implementation

The Matrix X is designed on the web-service side of the server, in Python language. It is designed in Java on retailer side. The output of all elements is the same for both codes in both languages. When using a coupon, the verification function is used to match the claim with the server. The number of clients and the performance of the second and third hash functions can vary and scale well in our proposed scheme. Practically, any cryptographic function or even a random number generator can be used to generate matrix elements.

C. Designing Applications

After designing the matrix, we implemented two Android member and retailer apps. The front-end and back-end sections as well as the server-side functions are written in Java; the async required functions are part of the Java code as well. Using the HTTP protocol, we create a request and response structure between the server and the application. The volley

library in Android and the HTTP protocol have been used for this purpose.

The retailer application login page includes a mobile number, a password and an optional email. The retailer sends a $w0$ random value request to the server once per each promotion. The server sends the random value in response for the retailer to produce the matrix X itself. This value is constant until the end of the promotion. Whenever a customer makes a successful transaction, he or she receives a number of loyalty points, and the retailer sends the matrix entry corresponding to the memberid row and the number of loyalty points column to the customer. This matrix entry, which is actually the same as encryption code, is sent to the customer via SMS or NFC. A list of supported web-services is described here: a) authentication function - separate for customer and retailer, b) uploading various promotion activities from the server, c) showing these promotions in the retail app, d) validation check for loyalty points code, e) allocating a certain matrix entry to each member, f) receiving a random initial value from server side to retailer and g) equating the specified number of loyalty points to coupon.

D. Token for authentication

There are several ways to authenticate a user. In our method, there is a third party (trust manager) who provides a token request and checks for authentication. We allocate a 64-bit long string that is random and unpredictable to each user. A user who wants to request a mobile coupon server will only request a hashed string instead of raw information, and on the server side this string will be searched in the database. If it is exists in the database the authentication is complete. The token line cannot be guessed, but it could be intercepted. The concept of a shared key was used to address this problem. The key is symmetric between the receiver and the sender. Each part of the key can be accessed by the key manager of the same sender or receiver. So if interception occurs, the intruder only gets part of the key and cannot abuse it.

E. SIM card Simulation with Android Application

Our simulation requires the application to interact with the operator without user interference in the mobile background. The goal is to have two Android member apps while the sim simulator interacts with each other in the background. This is possible in Android through the methods of activity pages, broadcast and service usage. In the sim simulator application we wrote a service called a sim card service that sends a web service to the operator and receives the response (Figure 6). To test the functionality of our sim simulator application, we send a request to the server to generate and distribute the key, and then we test the results with the postman software. The implementation stages of the request are in accordance with the proposed two-phase protocol. If there is a change in the shared key, trust will not occur. An error has been generated and the shared key of the SIM card and the coupon server will not be sent to them. A message based on this system will also be received if the request is successfully sent to the server. This web-based implementation of the operator simulator service app could be connected to any other app within the simulator Android app, which makes our design versatile.

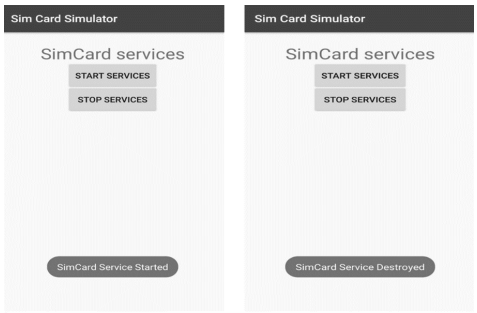

Fig. 6. Application Overview of the SIM Simulator Android application: This application contains the standard aidl library sim card simulation implementation that we can connect to any other Android application that requires an operator connection with the Python Web Service.

V. Formal verification based on AVISPA and SPAN

A. Avispa

In this section, the Automated Validation of Internet Security Protocols and Applications (AVISPA) [16] is employed to validate the correctness and security of our proposed scheme. The AVISPA tool provides comprehensive and reliable verification of the protocol's robustness against a wide range of potential threats, such as replay or man-in-the-middle attacks, establishing a solid foundation for security evaluation. Its ability to simulate various attack scenarios ensures that the protocol's defenses are validated effectively in initial stages.

AVISPA provides a suite of applications for the analysis of large-scale Internet security-sensitive protocols. It utilizes the High-Level Protocol Specification Language (HLPSL) to describe the security protocols and specify its intended security properties. AVISPA integrates four different back-ends, including On-the-fly Model-Checker (OFMC), Constraint-Logic-based Attack Searcher (CL-AtSe), SAT-based Model-Checker (SATMC), and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP) to implement a variety of state-of-the-art automatic analysis techniques. AVISPA adopts the standard intruder model of Dolev and Yao (DY model), in which the intruder, in addition to having all the capabilities of an honest agent, such that all messages sent by the agents will go to the intruder, may intercept, analyze, and/or modify messages, and send the new ones impersonating other agents [17].

In the following part, we will formally specify our proposed scheme. Our proposed scheme described in HLPSL. Although, the current version of the HLPSL only supports the general secrecy and authentication goals, it is sufficient to specify a large types of problems. In our scheme, we verify two strong authentication and one secrecy goals. Finally, the Security Protocol Animator for AVISPA (SPAN) [18] with the OFMC is used for falsification and verification, which provides a graphical user interface for the protocol designer to easily interact

	Original data	Compressed data	Our data
The amount of data	7KB	4KB	2.25KB
Transmission time	3S	1.3 $\sim$ 2s	o.482S

TABLE II
DATA SIZE - TRANSMISSION TIME

	Execution time of [6]	Our execution time
The one-way hash function	0.0 0 015 s/t	0.0 00 015 s/t
The signature verification	0.08 s/t	0.0 0 022 s/t
Connecting to TSM	It does not have	0.0 63 s/t

TABLE III
RUNTIME COMPARISON

	retailer	member	TSM	MCSP
Data blinding	-	-	-	-
Authenticator generation			O(n)	
Verify operation				O(c)
Challenge generation			O(c)	
Hash function	O(1)			
Make matrix	O(n^2)	O(1)		

TABLE IV
COMPUTATIONAL COMPLEXITY

with AVISPA capabilities. Figure 7 shows a sample output of the model checking results of our scheme. Our proposed scheme holds the four roles namely merchant, member, MCSP and third-party. It can resist malicious attacks such as replay attacks, man in the middle attacks, and secrecy attacks under the test of SPAN using the OFMC back-end.

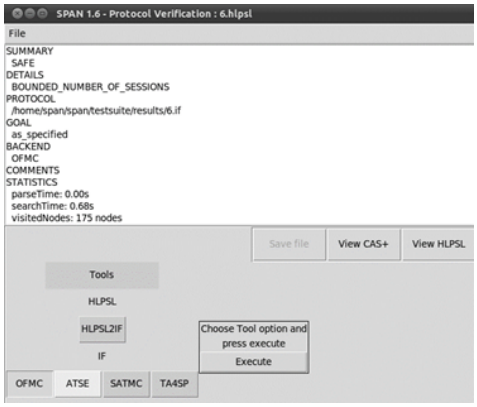

Fig. 7. Results reported for our scheme by the OFMC back-end

VI. SECURITY ANALYSIS

In previous works, the user would request a token from MCSP, and the server would issue the user with a username and a one time password that would allow the intrusion. However, in our proposed protocol, the member requests a shared key from the operator. A token is also exchanged between the member and the operator. So the operator makes sure that the member owns the authentication SIM card and can not impersonate anyone else. In [2] at the bottom row of the matrix the columns were clustered together and used as checkpoints. But in our proposed protocol this part is not implemented and instead the system of authentication is implemented at the mobile network operator level. Both public and private keys were also defined in [2], but we removed the simple public-key encryption system and implemented the token system as an API in our protocol. Although the token is exchanged between two members, the operator is also in the token cycle. Thus, a secure coupon payment system is designed using the SIM card platform.

Phase	Active roles	The one-way hash function (average)	The signature generation
Lp earning	Retailer	1	1
	MCSP member	0	0
m-coupon redemption	MCSP member	0	1
	MCSP	1	0
m-coupon using	MCSP member	0	1
	Retailer	0	0

TABLE V
NUMBER OF COMPUTATIONS

	Number of http requests	Number of encryptions
First phase	2	3
Second phase	1	6
Total	3	9

TABLE VI
NUMBER OF ENCRYPTION OPERATIONS

A. Runtime Comparison

The keys that the operator produced in this study had a maximum of 64 characters. Since each character is equivalent to 32 bits and requires nine times the key exchange, then 18432 bits of our data volume is equivalent to 4608 bytes and therefore 2.25 KB (Table II). The total runtime is obtained from the total response times of http requests and is shown on Table III.

B. Performance Analysis

With our proposed architecture we do not need to blind sensitive user data during transmission. Data loading and data auditing are inherently protected against malicious intruders because both the server and the retailer create their own matrix elements and not exchange data during the http request and response. So there is no need for extra privacy methods like code obfuscating or data blinding. As a result, our architecture is very efficient and achieves computational complexity of $O(n)$, where n is the total number of data blocks and c is the number of challenged blocks to compare. The computational complexity of different entities in the process is summarized in Table IV.

C. Number of Coupon Generation Calculations

The number of computations varies depending on the type of function that performs the cryptographic operation. For example, SHA1 generates a 160-bit (20-byte) hash, typically presented as a hexadecimal number of 40 digits. SHA1 is widely used in the existing SHA hash function suite. However, the SHA1 algorithm may not be secure enough for continuous use, so we experimented with other algorithms as well. Tables V and VI show the number of computations and the number of encryption operations of our proposed protocol.

D. TSM Http Requests Response Time

The size of the requests are 450, 240, and 243 bytes, respectively. Response times were 7, 220 and 255 milliseconds, respectively. Based on the header information, the data length of the json request was estimated to be 274, 65 and 68 bytes respectively. As an example, the matrix construction time with the SHA1 algorithm is given in Figure 8. It should be noted that the SHA1 algorithm is used in Google's SSO implementation. We also modified the SHA1 algorithm to other well-known algorithms such as SHA256, SHA512 and MD5. The results for the execution time of matrix X are shown in Figure 8 as well.

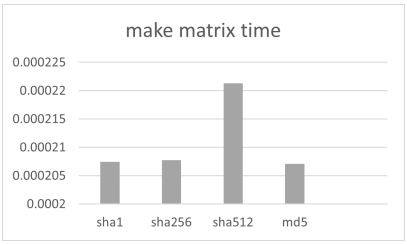

Fig. 8. Matrix computation time calculation code using various hash functions

CONCLUSION

In this paper, we discuss mobile payment challenges and propose a secure approach based on a trust manager architecture. The first three challenges mentioned in Section II-B have been addressed. The fourth challenge is part of future work.

More specifically, our work ensures: 1) Confidentiality - since data is not sent at the request phase the confidentiality of the information is guaranteed, 2) Accountability - Due to the simulation of the authentication process and the fact that each member has signed a message with his /her private key at each step, it is not possible for the server or retailer or users to deny it, 3) Fraud avoidance - Due to the one-way hash used in the process of sending the coupon, it is not possible to reverse the process. So the prior and next rows of the matrix cannot be accessed from each other. In addition, each coupon is specific to a specific customer and its counterfeiting will be useless in any way to anyone but the SIM card holder, 4) Non-Repudiation - Due to the use of cryptography and digital signatures it will not be possible to deny customer loyalty points. Since the matrix X is a combination of two hierarchical one-way hash chains, it cannot be reversed. The customer may gain more unrealistic loyalty points. But it cannot obtain the value of the next layer from its value in the matrix because the hash chain is an one-way function. Similarly, other matrix layers are not derived from other privilege codes, 5) Double Spending Detection - Double-spending is the result of successfully spending some loyalty points more than once. The double-spending attack can happen offline when the merchant cannot verify the transaction message with the TSM. In our approach, the conversion of points to coupons takes place online and is registered by the server. Thus, double spending loyalty points is detectable,

ACKNOWLEDGEMENT

This work is part of the course project in the graduate class ECE 8476 - Cryptography and Network Security at Villanova University.

REFERENCES

[1] GSMA, "The state of mobile internet connectivity," https://www.gsma.com/r/wp-content/uploads/2023/10/The-State-of-Mobile-Internet-Connectivity-Report-2023.pdf, 2023.

[2] J.-L. Ferrer-Gomila, M. F. Hinarejos, and L. Huguet-Rotger, "A survey on electronic coupons," *Comput. Secur.*, vol. 77, no. C, p. 106–127, aug 2018. [Online]. Available: https://doi.org/10.1016/j.cose.2018.03.007

[3] S. Balasubramanian, R. Peterson, and S. Jarvenpaa, "Exploring the implications of m-commerce for markets and marketing," *Journal of the Academy of Marketing Science*, vol. 30, pp. 348–361, 10 2002.

[4] W. Li, Q. Wen, Q. Su, and Z. Jin, "An efficient and secure mobile payment protocol for restricted connectivity scenarios in vehicular ad hoc network," *Comput. Commun.*, vol. 35, no. 2, p. 188–195, jan 2012. [Online]. Available: https://doi.org/10.1016/j.comcom.2011.09.003

[5] Z. Qin, J. Sun, A. Wahaballa, W. Zheng, h. Xiong, and Z. Qin, "A secure and privacy-preserving mobile wallet with outsourced verification in cloud computing," *Computer Standards Interfaces*, vol. 54, 11 2016.

[6] Y.-Y. Chen, M.-L. Tsai, and F.-J. Chang, "The design of secure mobile coupon mechanism with the implementation for nfc smartphones," *Computers Electrical Engineering*, vol. 59, pp. 204–217, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790616301069

[7] S. Kungpisdan, S. Bala, and P. Le, "A secure account-based mobile payment protocol," vol. 1, 05 2004, pp. 35 – 39 Vol.1.

[8] C. Butpheng, K.-H. Yeh, and H. Xiong, "Security and privacy in iot-cloud-based e-health systems—a comprehensive review," *Symmetry*, vol. 12, no. 7, 2020. [Online]. Available: https://www.mdpi.com/2073-8994/12/7/1191

[9] S. Chaudhry, M. Farash, S. Naqvi, and M. S. Ramzan, "A secure and efficient authenticated encryption for electronic payment systems using elliptic curve cryptography," *Electronic Commerce Research*, vol. 16, 06 2015.

[10] A. Coughlin, G. Cusack, J. Wampler, E. Keller, and E. Wustrow, "Breaking the trust dependence on third party processes for reconfigurable secure hardware," in *Proceedings of the 2019 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, ser. FPGA '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 282–291. [Online]. Available: https://doi.org/10.1145/3289602.3293895

[11] S. Nakandala, H. Gunasinghe, S. Marru, and M. Pierce, "Apache airavata security manager: Authentication and authorization implementations for a multi-tenant escience framework," 10 2016, pp. 287–292.

[12] W. Shen, J. Yu, H. Xia, H. Zhang, X. Lu, and R. Hao, "Light-weight and privacy-preserving secure cloud auditing scheme for group users via the third party medium," *Journal of Network and Computer Applications*, vol. 82, pp. 56–64, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804517300218

[13] Y. Zhou, K. Patel, L. Wu, Z. Wang, and X. Jiang, "Hybrid user-level sandboxing of third-party android apps," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*, ser. ASIA CCS '15. New York, NY, USA: Association for Computing Machinery, 2015, p. 19–30. [Online]. Available: https://doi.org/10.1145/2714576.2714598

[14] M. Aloqaily, B. Kantarci, and H. T. Mouftah, "Trusted third party for service management in vehicular clouds," *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 928–933, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:22951527

[15] P. Zhang, H. Chi, J. Wang, and Y. Shang, "Data security protocol with blind factor in cloud environment," *Information*, vol. 12, p. 340, 08 2021.

[16] A. Project, "Automated validation of internet security protocols and applications v1.1 user manual," https://www.avispa-project.org/package/user-manual.pdf, 2006.

[17] J. Cao, M. Ma, H. Li, Y. Fu, B. Niu, and F. Li, "Trajectory prediction-based handover authentication mechanism for mobile relays in lte-a high-speed rail networks," in *2017 IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.

[18] Y. Glouche, T. Genet, O. Heen, and O. Courtay, "A security protocol animator tool for avispa," 01 2006.