sensors

MDPI

Article

Mobile Payment Protocol with Deniably Authenticated Property

Yunzhuo Liu [1], Wen Huang [2], Ming Zhuo [1], Shijie Zhou [1,*] and Mengshi Li [2]

[1] School of Information and Software Engineering, University of Electronic Science and Technology of China, Chengdu 610054, China
[2] Colleague of Computer Science, Sichuan University, Chengdu 610017, China
* Correspondence: sjzhou@uestc.edu.cn

Abstract: Mobile payment services have been widely applied in our daily life, where users can conduct transactions in a convenient way. However, critical privacy concerns have arisen. Specifically, a risk of participating in a transaction is the disclosure of personal privacy. This might occur if, for example, the user pays for some special medicine, such as AIDS medicine or contraceptives. In this paper, we propose a mobile payment protocol that is suitable for mobile devices only with limited computing resources. In particular, the user in a transaction can confirm the identity of others in the same transaction while the user cannot show convincing evidence to prove that others also take part in the same transactions. We implement the proposed protocol and test its computation overhead. The experiment results corroborate that the proposed protocol is suitable for mobile devices with limited computing resources.

Keywords: deniable authentication; mobile payment; deniably authenticated encryption; privacy preserving; confidentiality

check for updates

Citation: Liu, Y.; Huang, W.; Zhuo, M.; Zhou, S.; Li, M. Mobile Payment Protocol with Deniably Authenticated Property. *Sensors* **2023**, *23*, 3927. https://doi.org/10.3390/s23083927

Academic Editor: Tamer M. Nadeem

Received: 17 March 2023
Revised: 4 April 2023
Accepted: 10 April 2023
Published: 12 April 2023

1. Introduction

In recent years, mobile communication technology has developed rapidly, especially in rural areas. In the past, due to the lack of infrastructure for communication, people in rural areas did not have easy access to the Internet. With the application of mobile communication devices such as 5G facilities, people in rural areas are becoming increasingly connected to the world. For example, more and more people are shopping via mobile communication networks.

With the popularity of mobile devices and the increasing dependence of people's lives on mobile devices, mobile payment has become widely used around the world. Many organizations, for example, banks, software companies, and mobile operators, have already made many efforts to promote mobile payment services. Google Wallet, MasterPass, Android Pay, and Apple Pay are well-known services in the mobile payment market. One study named Transparency Market Research [1] shows that the market of global mobile payment will reach USD 1602.4 billion.

Although the widespread use of mobile payment services has brought great convenience to people, daunting privacy challenges related to the disclosure of transaction information have arisen, and limited computing resources of mobile devices have also become a problem in the application of mobile payment services. In particular, the private information involved in a transaction brings the risk of disclosure of personal privacy. For example, the US-based mobile social payment platform named Venmo generates payment notes for each Venmo transaction, and these payment notes are visible to all other Venmo users. Rajat et al. found that "41 M notes (10.5%) leak some sensitive information such as health condition, political orientation and drug/alcohol consumption involving 8.5 M (37.8%) users" [2]. In some cases, the risk of participating in a transaction is the disclosure of personal privacy. This might occur if, for example, the user pays for some special medicine, such as AIDS medicine or contraceptives.

To safeguard personal privacy, entities in some application scenarios might not want others to know that they have taken part in a transaction. In a word, it is critical in the circumstances described above that parties in a transaction cannot convince others to believe that a particular person takes part in the transaction.

A deniably authenticated encryption scheme can ensure that it is possible to confirm the identity of entities in the transaction, while also ensuring that there is no chance to affirmatively show the identity of entities in the transaction to other parties [3]. In addition, mobile payment services run on mobile devices with limited computing resources, so there is a need to a design a lightweight mobile payment protocol. Thus, we utilize the deniably authenticated encryption method to construct a mobile payment protocol that can protect the privacy of the client involved in a transaction.

1.1. Contribution

We design a mobile payment protocol that is suitable for application in a mobile payment scenario, where privacy preservation is critical. Firstly, the designed protocol is lightweight. In particular, we adopt an optimized deniably authenticated encryption scheme that can be applied on mobile devices with limited computing resources. Secondly, the proposed protocol can satisfy multiple design goals at the same time. Specifically, the designed protocol has the properties of confidentiality, integrity, deniable authentication, traceability, and non-repudiation.

1.2. Related Work

The discussion about the related works involves two aspects: deniable authentication and mobile payment.

Authentication ensures that the identity of the communicating entity is in line with its own claims. It is a security feature that is necessary in many application scenarios, for instance, smart meters [4], the Metaverse [5], and the location authentication of mobile devices [6,7].

In some application scenarios, an authentication scheme with deniability is needed. Deniable authentication guarantees two particular features. First, it is possible to confirm the identity of the entities in the transaction. Second, there is no chance of showing the identity of entities in the transaction to other entities affirmatively.

On the basis of their original work relating to zero-knowledge proof, Dwork et al. developed a deniable authentication protocol [8]. However, their protocol was limited by a timing constraint called the (α, β) assumption on the response time of processes.

Aumann and Rabin devised a deniable authentication method based on the factoring issue, a tough topic in computing complexity theory [9]. Deng et al. developed a new deniable authentication strategy based on a tough problem in computation complexity theory known as discrete logarithm and factoring problems [10]. The methods of [9,10] assumed that there was a trustworthy third entity who was able to provide a public directory service. On the basis of [11], Fan et al. [12] developed a deniable authentication protocol to improve the protocols of [9,10]. Public key certificates on their protocol prevented person-in-the-middle (PIM) attacks, and their protocol verified the source of messages using digital signatures. However, Yoon et al. [13] proved that [12] is vulnerable regarding authentication. An attacker was able to disguise as the receiver and confirm the identity of the sender.

Based on the ElGamal signature scheme [14], Shao [15] built a deniable authentication scheme over which involved entities did not have to interact with each other. In 2005, Lu developed a deniable authentication method based on a tough problem in computation complexity theory known as the factoring problem, in which involved entities did not have to communicate with each other [16]. The security of their protocol was shown in the random oracle model. Wang [17] created a deniable authentication protocol based on the ElGamal cryptosystem [18] with the inverse of the ElGamal implementing deniable authentication. According to Shao [19], the protocol of [17] has a weakness. An attacker can use this flaw to launch the PIM attack. The attacker can impersonate a legitimate communication entity. Yoon et al. [20] suggested in 2010 that Shao [19]'s enhanced protocol is not safe against a receiver impersonation attack.

Based on ElGamal cryptography, Yoon et al. proposed an improved deniable authentication protocol to overcome this weakness [20]. However, Li and Takagi argued in [21] that [20] did not have the deniable authentication property. The receiver can reveal the sender's identify to other entities. Based on the ElGamal signature scheme [14], a deniable authentication procedure was presented by Lee et al. [22]. Even if the signature mechanism was compromised, the protocol can operate as a signature scheme to provide security.

Based on the DDH assumption, Wang et al. developed a deniable authentication procedure in which participants did not need to engage with one another [23]. Li et al. built a new deniable authentication protocol that used an identity-based cryptology system [24]. Liao et al. developed a system that used a secure certificateless signing mechanism and delegated signature verification to a cloud server [25]. Li et al. presented a technique that can accomplish deniable authentication, confidentiality, and integrity [26]. After Li's scheme, many other deniably authenticated schemes were proposed, such as [27–30].

Researchers from home and abroad have given the definition of mobile payment, which is as follows: a mobile payment is a form of payment transaction processing over mobile communication techniques in which the payer initializes, authenticates, and completes payment by mobile devices [31]. Thus far, there have been some different proposed mobile payment protocols, such as [32,33]. These protocols are based on symmetric cryptography. These protocols are efficient and suitable for mobile devices. However, these protocols have various weaknesses. Other protocols using public key cryptography are not appropriate for mobile devices because they are designed for fixed networks, for instance. In addition, privacy preservation is also an important requirement of mobile payment protocols. Therefore, in this paper, we optimize a deniably authenticated encryption scheme, making it suitable for mobile devices, and then we apply it to mobile payment protocols.

The concept of deniably authenticated encryption was created by Li et al. [26]. It has been applied to many fields. Based on identity cryptography, Chunhua et al. improved the communication costs and ciphertext size of deniably authenticated encryption schemes [34]. Managing certificates costs a lot [35–37], and their scheme can avoid public-key-certificate-based infrastructure. Kasper et al. named the concept of strong and weak deniable authentication and then proposed two efficient encryption schemes that provide deniable authentication [38]. To protect the privacy of location data, Guanhua et al. proposed a deniable authenticated encryption scheme based on certificateless cryptosystems [39]. Their scheme did not depend on public key infrastructure and avoided the problem of key escrow present in identity-based cryptosystems. Chunhua et al. proposed a deniable authentication encryption scheme based on an identity-based environment to avoid public-key-certificate-based public key infrastructure, and then they applied their scheme into the scenario of e-voting. Ahene et al. also constructed an e-voting system based on a deniably authenticated encryption scheme [28]. Kar et al. applied deniably authenticated encryption into the field of e-mail [29]. Wen et al. protected the privacy of entities in communications through a deniably authenticated encryption scheme [40]. Based on bilinear pairings, Chunhua et al. constructed a deniably authenticated encryption for e-mail applications [41]. Chunhua et al. proposed the concept of heterogeneous deniable authenticated encryption, which enabled a sender in a public key infrastructure environment to transmit a message to a receiver in an identity-based environment [42]. Based on blockchain and deniably authenticated encryption technology, Zhang proposed a deniably authenticated searchable encryption scheme to guarantee the privacy and workability of medical image data [30]. Jin et al. used a deniably authenticated encryption scheme to implement location-based services [43]. Based on a two-user ring signature, Shengke et al. constructed an efficient deniable authentication scheme to protect location privacy [44]. Yanmei et al. formalized the syntax and security notions of public-key-authenticated deniable encryption, and then proposed two concrete constructions under a fully deniable framework [45].

1.3. Organization

In the next section, we introduce preliminaries of deniably authenticated encryption. In Section 3, we elaborate on the system model and design goals. In Section 4, the proposed protocol is presented in detail. In Section 5, the security analysis of the proposed protocol is given. In Section 6, we theoretically analyze the computational complexity, and then we implement the proposed protocol and test running time of our implementation. Finally, we provide our conclusion.

2. Preliminaries

2.1. Deniably Authenticated Encryption

The concept of deniably authenticated encryption was proposed by Li et al. for a secure e-mail service [26]. It achieves confidentiality, integrity, and deniable authentication in a logical single step. Then, Wen et al. optimized the deniably authenticated encryption scheme [40]. The deniably authenticated encryption scheme consists of four algorithms, including the setup, keygen, DA-encrypt, and DA-decrypt algorithms.

Setup. The Setup algorithm takes the security parameter λ as input and generates system parameters.

KeyGen. The KeyGen algorithm produces keys for the sender and receiver. In particular, it generates a public key y_s and a private key x_s for the sender, and it generates a public key y_r and a private key x_r for the receiver.

DA-Encrypt. The sender produces ciphertext σ of message m by performing DA-Encrypt algorithm. Specifically, the DA-Encrypt takes system parameters generated by Setup algorithm, message m, sender's private key x_s, sender's public key y_s together with receiver's public key as inputs, and then produces ciphertext σ of message m.

DA-Decrypt. The receiver can extract plaintext m from a ciphertext σ through the DA-Decrypt algorithm. More specifically, it takes the system parameters, ciphertext σ, receiver's private key x_s, receiver's public key y_s, and the sender's public key as inputs, and then outputs the plaintext m of a ciphertext σ if the ciphertext σ is a valid ciphertext. DA-Decrypt outputs nothing if ciphertext σ is an invalid ciphertext

2.2. Security Concepts

Definition 1. *A deniable authenticated encryption scheme is $(\epsilon_{dea}, t, q_e, q_d)$-IND-CCA secure if there is no adversary A who has an advantage of at least ϵ_{dea} in the IND-CCA game under the condition that time is probabilistic t-polynomial time, that deniably authenticated encryption queries are, at most, q_e, and that deniably authenticated decryption queries are, at most, q_d.*

Definition 2. *A deniable authenticated encryption scheme is $(\epsilon_{dea}, t, q_e, q_d)$-DA-CMA secure if there is no adversary A who has an advantage of at least ϵ_{dea} in the IND-CCA game under the condition that time is probabilistic t-polynomial time, that deniably authenticated encryption queries are, at most, q_e, and that deniably authenticated decryption queries are, at most, q_d.*

Definition 1 as well as Definition 2 are from [40], and the detailed information of the IND-CCA game is elaborated in [40].

3. System Model

The model of payment systems is demonstrated in Figure 1. A client, a merchant, and a payment service provider (PSP) make up this model. The client, Alice, is the entity who wants to pay for the merchant. The merchant, Bob, is the entity who receives money from someone else, such as Alice. PSP Bank is a financial infrastructure that provides payment services while also storing the account information of the entities involved. We assume that the bank is a trusted entity and it never does anything evil.

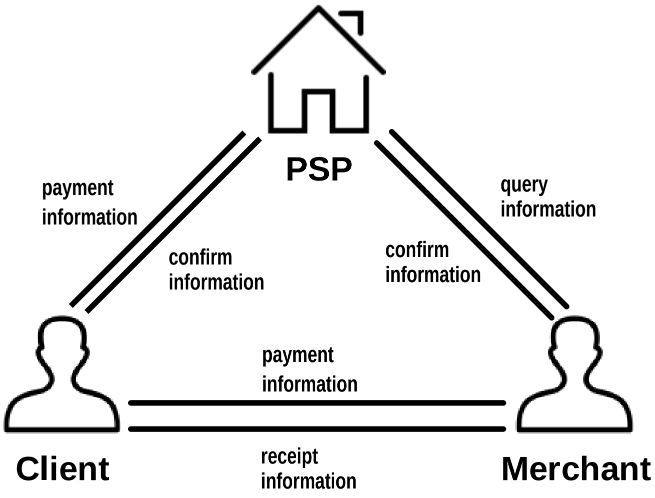

Figure 1. System model.

3.1. Threat Model

In our protocol, we assume that the attack can control the client or the merchant. When the attacker controls the client, his goal is to deny his participant in a transaction so that he can obtain products without non-payment. When the attacker controls the merchant, his goal is to convince others to believe that the client takes part in a transaction. The exposure of purchase records may leak the privacy information of the client, especially the records related to special products, such as special medicines. The merchant may make unjustified profits by selling the client's purchase records when he can convince others to believe that his purchase records are true.

3.2. Design Goal

According to the requirements of the application scenario, the secure mobile payment protocol should satisfy properties of confidentiality, deniable authentication, integrity, traceability, non-repudiation, and small overhead.

Confidentiality: assures that only the intended recipient is aware of the message's contents.

Integrity: ensures that messages sent during the interaction process are not tampered with by unlawful entities.

Deniable authentication: ensures that the entity receiving the message can confirm the identity of the entity sending it. Furthermore, the entity receiving the message is unable to reveal the identity of the entity sending the message to the third entity.

Traceability: ensures that the client cannot deny that she made the payment herself. Meanwhile, the merchant cannot deny the received payment. Otherwise, the unique identity stored in PSP can be used to track them.

Non-repudiation: once a merchant confirms the information message, he or she cannot deny the communication's validity or origin. Additionally, once a client acknowledges the information message, he or she is unable to retract her confirmed payment message.

Lightweight overhead: because the mobile payment protocol is used on mobile devices with limited computing resources, the mobile payment protocol needs to be lightweight.

4. Proposed Protocol

The proposed protocol consists of three phases, including the setup phase, keygen phase, and payment transaction phase. In the setup phase, PSP obtains security parameters λ and outputs system parameters. In the keygen phase, the client obtains the system parameters and outputs the account and the code of the account. Next, the client sends his or her ID and account to PSP. Then, the merchant performs all the same activities as the client. In the payment transaction phase, the merchant and the client interact with each other to transfer the payment. The interaction scenario among the above entities is sketched in Figure 2. Unlike Figure 1 (which elaborates on the components of the whole system), Figure 2 elaborates on how one component interacts with other components in detail.

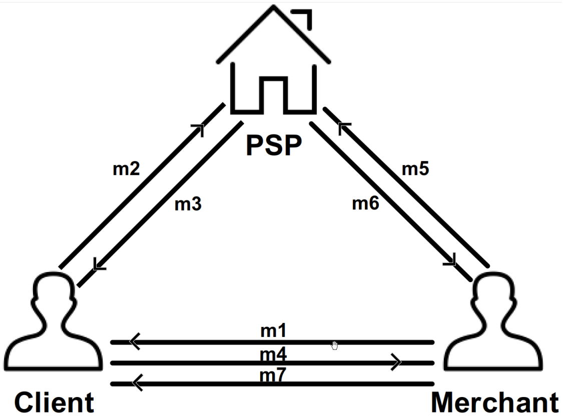

Figure 2. Proposed Protocol.

Next, we give detailed information about the message in the interaction scenario in Figure 2.

$$m_1 = (T_{id}||Amount_t||ID_b||Timestamping)$$

is the payment information, including the ID of the transaction, the amount of the transaction, the ID of the merchant, and the time stamp.

$$m_2 = (T_{id}||Amount_t||ID_b||Timestamping||ID_a||Transfer_a)$$

is the payment message sent from the client to PSP.

$$m_3 = \left(T_{id}||Amount_t||ID_b||Timestamping||ID_a||Transfer_p\right)$$

is the payment confirmation message sent from PSP to the client.

$$m_4 = (T_{id}||Amount_t||ID_b||Timestamping||ID_a||Confirm_a)$$

is the confirmatory payment message sent from the client to the merchant.

$$m_5 = (T_{id}||Amount_t||ID_b||Timestamping||ID_a||Confirm_a)$$

is the query message sent from the merchant or client to PSP.

$$m_6 = yes \ or \ m_6 = no$$

is the message of response to the query message sent from PSP to the merchant or the client.

$$m_7 = (T_{id}||Amount_t||ID_b||Timestammping||ID_a||Confirm_b)$$

is the confirmatory receipt message sent from the merchant to the client.

Implementation of Mobile Payment Protocol

Next, we give a detailed implement of the proposed security mobile payment protocol. As mentioned above, the secure mobile payment protocol consists of three phases.

Setup Phase: Initially, PSP obtains the security parameter λ, runs the setup algorithms of the deniable authenticated encryption scheme [40], and outputs the system parameters $\{n, p, q, g, H_2, H_1\}$. Here, H_2 and H_1 are two hash functions that are randomly selected from the system.

Keygen Phase: PSP runs the keygen algorithm of the deniable authenticated encryption scheme. The client Alice randomly chooses their private key x_a from Z_q^*, computes $y_a = g^{x_a} \bmod p$, and takes (x_a, y_a) as its private/public key pair. The client Alice sends its unique ID_a and y_a to PSP, and then PSP stores them in secure storage. Similarly, the merchant Bob randomly chooses their private key x_b from Z_q^*, computes $y_b = g^{x_b} \bmod p$, and takes (x_b, y_b) as its private/public key pair. The merchant Bob sends its unique ID_b and y_b to PSP, and then PSP stores them in secure storage. PSP randomly chooses its private key x_p from Z_q^*, computes $y_p = g^{x_p} modp$, and takes (x_p,y_p) as its private/public key pair.

Payment Transaction Phase:

①Bob constructs his payment request on his website. The payment request consists of the transaction identity, T_{id}; the amount to be paid, $Amount_t$; his unique identity, ID_b; and the transaction time stamp, $Timestamping$.

$$m_1 = (T_{id} \| Amount_t \| ID_b \| Timestamping)$$

Here, ‖ represents the message concatenation.

With the given message m, the key pair (x_b, y_b) from Bob, and the public key y_a from Alice, Bob encrypts m into a ciphertext, σ_1, using the deniably authenticated encryption algorithm. Bob sends σ_1 to Alice.

② Alice obtains the ciphertext σ_1. With the ciphertext σ_1, the public key y_b from Bob, and the key pair (y_a, x_a) from Alice, Alice decrypts σ_1 using the deniably authenticated decryption algorithm into m_1. If proven valid, Alice constructs her transfer confirmation message

$$m_2 = (T_{id} \| Amount_t \| ID_b \| Timestamping \| ID_a \| Transfer_a)$$

Alice encrypts m_2 to σ_2 using the deniably authenticated encryption algorithm and then sends σ_2 to PSP.

③ PSP obtains the ciphertext σ_2. With the PSP's key pair (y_P and x_P) and Alice's public key y_a, PSP decrypts σ_2 into m_2 using the deniably authenticated decryption algorithm. If σ_2 is valid, PSP stores the content decrypted from the received ciphertext σ_2 in the local database, and PSP constructs its transfer confirmation message m_3.

$$m_3 = (T_{id} \| Amount_t \| ID_b \| Timestamping \| ID_a \| Transfer_p)$$

PSP encrypts m_3 into σ_3 using the deniably authenticated encryption algorithm, then sends σ_3 to Alice.

④ Alice obtains the ciphertext σ_3. With the PSP's public key y_P and Alice's key pair (y_a, x_a), Alice decrypts σ_3 into m_3 using the deniably authenticated decryption algorithm. If σ_3 is valid, Alice constructs her confirmation message m_4,

$$m_4 = (T_{id} || Amount_t || ID_b \| Timestamping \| ID_a \| Confirm_a)$$

Alice encrypts m_4 into σ_4 using the deniably authenticated encryption algorithm, then sends σ_4 to Bob.

⑤ Bob obtains the ciphertext σ_4. With Alice's public key y_a and Bob's key pair (x_b, y_b), Bob decrypts σ_4 into m_4 using the deniably authenticated decryption algorithm. If σ_4 is valid, Bob constructs his query message m_5

$$m_5 = (T_{id} || Amount_t \| ID_b \| Timestamping \| ID_a || Confirm_a)$$

Bob encrypts m_5 into σ_5 using the deniably authenticated encryption algorithm and then sends σ_5 to PSP.

⑥ PSP obtains the ciphertext σ_5. With PSP's key pair (y_P, x_P) and Bob's public key y_b, PSP decrypts σ_5 using the deniably authenticated decryption algorithm into m_5. If σ_5 is valid, we must next check whether there is a message m_5 in the database. If m_5 is found,

PSP constructs a response message, where $m_6 = yes$ or $m_6 = no$. PSP encrypts m_6 into σ_6 using deniably authenticated encryption, then sends σ_6 to Bob.

⑦ Bob obtains the ciphertext σ_6. With Bob's key pair (x_b, y_b) and PSP's public key y_P, Bob decrypts σ_6 into m_6 using the deniably authenticated decryption algorithm. If σ_6 is valid, Bob constructs his confirmation message m_7,

$$m_7 = (T_{id} \| Amount_t \| ID_b \| Timestamping \| ID_a \| Confirm_b)$$

Bob encrypts m_7 to σ_7 using the deniably authenticated encryption algorithm, then sends σ_7 to Alice.

5. Security Analysis

Considering our design goals, we outline how these goals can be achieved using our proposed protocol.

The merchant cannot convince others that the client took part in a particular transaction because the merchant has the ability to forge transaction information.

Theorem 1. *The merchant can forge the legal ciphertext of the client without the private key of the client.*

Proof. Assume that symbol sk_a represents the private key of the client and symbol sk_a represents the private key of the merchant. There is a message m, and the merchant forges their ciphertext as follows:

1. Choose a random number x from Z_q^*;
2. Compute $w = y_b^x \ mod\ p$ and $k = H_1(w)$;
3. Compute $c = m \oplus k$;
4. Compute $e = H_2(m||y_b||y_a||w)$. Here, | | represents the message concatenation;
5. Compute $v = sk_a e + x \ mod\ q$;
6. Compute $z = y_b^v \ mod\ p$;
7. The forged ciphertext is $\sigma = (c, e, z)$;.

In the above procedure, y_b is the public key of the merchant, y_a is the public key of the client; and H_1 and H_2 are two hash functions. □

Due to the merchant's ability to forge the legal ciphertext of the client, the merchant cannot make others believe his claim that some ciphertexts of transaction messages are generated by the client. That is, the client has the ability to deny participation in one transaction, in which it does not matter whether the client took part.

Confidentiality, integrity, and deniable authentication are achieved because the content of the message is encrypted by the deniable authenticated encryption algorithm. Detailed information can be found in Theorem 1 and Theorem 2 in [40].

Traceability: the transaction can be tracked by PSP if necessary because the confirmatory message between the client and PSP includes the ID of the client. For example, the court can ask to trace the transaction. Notably, PSP is assumed to be trusted. Thus, we can believe in PSP if it shows evidence that someone participated in a transaction. PSP ensures that it will not show evidence of whether someone participated in a transaction except when the requester has access rights, such as the police, court staff, and so on.

Non-repudiation: the confirmatory message between the client and PSP includes the ID of the client, and we trust that PSP does not do anything evil. Therefore, the client is not able to repudiate the correctness and the origin of the message once he or she confirms the message. Similarly, the merchant cannot deny his or her confirmatory message if he or she did send the confirmatory message.

6. Efficiency Comparison and Verification

For the simplicity of description, we use the below symbols. The symbol ✓ denotes that the proposed protocol has corresponding characteristics. The length of the message is represented by $|\chi|$. The times for modular inverse, modular multiplication, modular exponentiation, and hash function operations are represented by T_i, T_m, T_e, and T_h, respectively. The theoretical analysis of the computational complexity of the proposed protocol is shown in Table 1.

Table 1. Performance analysis.

Computational Cost		**Ciphertext Size**	**Security**		**Formal Proof**	**No-Interactive**
Sender	**Receiver**		**NID-CCA**	**DA-CMA**		
$2T_h+$ $2T_e+T_m$	$2T_h+$ $2T_e+T_m$	$\lvert m\rvert+$ $\lvert q\rvert+\lvert p\rvert$	√	√	√	√

Next, we took the running time as the standard to quantify computational complexity. The running time is dependent on the equipment on which experiments are performed, so we first introduced our experiment environment. We implemented the proposed protocol on an Ubuntu 12.04 virtual machine, which was installed on an Intel Core i5 2.6GHz computer with 8G RAM, using the MIRACL library [26]. When it comes to implementing huge-number cryptography, the MIRACL library is commonly regarded as a useful building kit for creating cryptography systems. Thus, we chose MIRACL to implement the proposed protocol.

In the first experiment, we tested the running time(computational complexity) of the encryption operation. In particular, p is 1024 bits and q is 512 bits in the proposed protocol. We employed eight different encryption key lengths, starting with 64 bits and gradually increasing this number by 64 bits. At an eight-type encryption key length, Figure 3 shows the running time (sum time to execute 1000 iterations of deniably authenticated encryption).

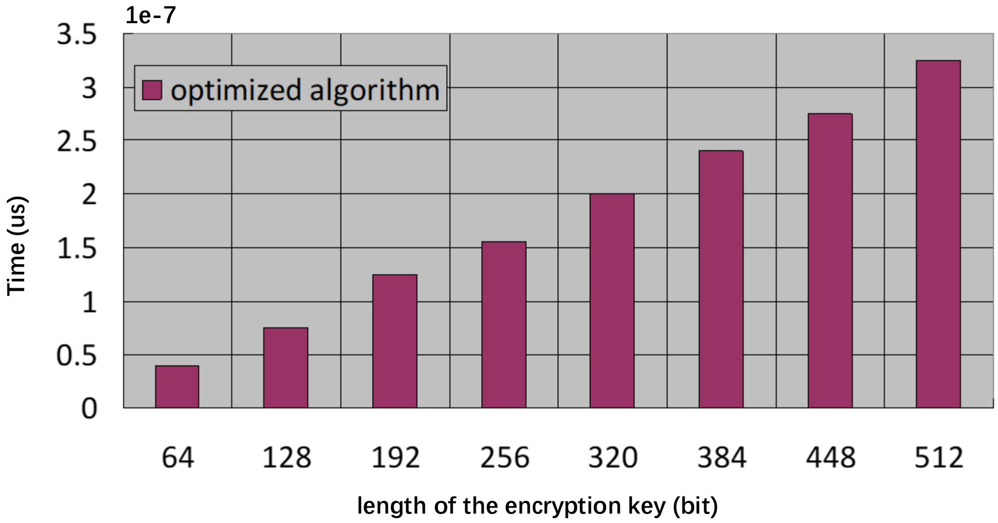

Figure 3. Encryption Time Comparison.

In the second experiment, we tested the running time of the decryption operation. Specifically, we chose 1024 as the value of p, and we chose 512 as the value of q. We employed eight different decryption key lengths, starting with 64 bits and gradually increasing this number by 64 bits. The experiment results are demonstrated in Figure 4. According to the experimental results, the proposed protocol is lightweight; that is, the proposed protocol can be applied into the application scenario of mobile payments.

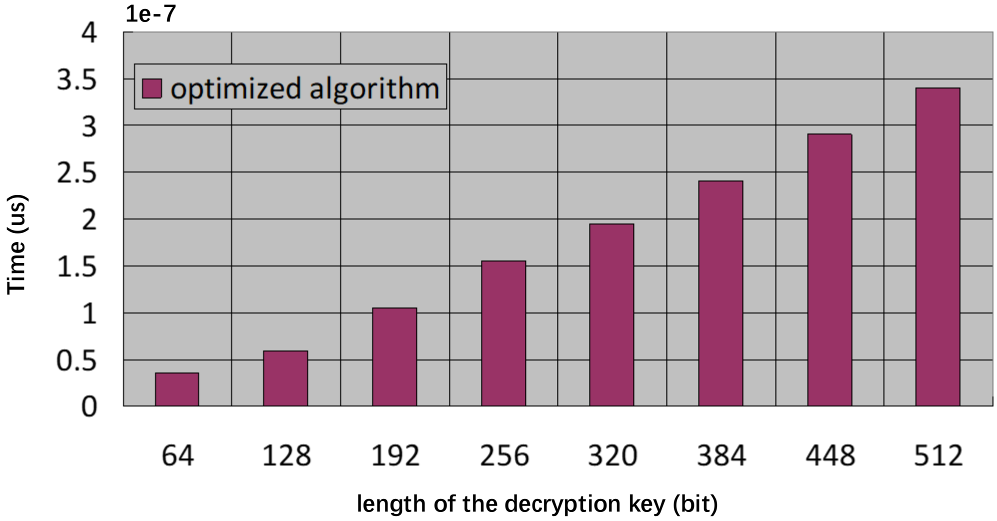

Figure 4. Decryption Time Comparison.

7. Conclusions

In order to preserve the privacy of entities of particular transactions, we designed a mobile payment protocol that can be applied into mobile devices, such as a POS machine, with limited computing resources. The entity in a transaction can confirm the identity of other entities involved in the transaction, and the entity in a transaction cannot show convincing evidence to prove that other entities are involved in the transaction. We implement the proposed protocol by an optimized deniably authenticated encryption, making it suitable for mobile devices.

Author Contributions: Conceptualization, Y.L.; Methodology, W.H.; Software, W.H.; Validation, M.Z.; Investigation, Y.L. and M.Z.; Resources, S.Z.; Writing—original draft, Y.L. and M.Z.; Writing—review & editing, M.L.; Supervision, S.Z.; Project administration, S.Z.; Funding acquisition, S.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Anonymous. Biopesticides Market—Global Industry Analysis, Size, Share, Growth and Forecast 2015–2023. Available online: https://www.prnewswire.com/news-releases/biopesticides-market---global-industry-analysis-size-share-growth-and-forecast-2015---2023-300224816.html (accessed on 9 April 2023).
2. Tandon, R.; Charnsethikul, P.; Arora, I.; Murthy, D.; Mirkovic, J. I know what you did on Venmo: Discovering privacy leaks in mobile social payments. *Proc. Priv. Enhancing Technol.* **2022**, *3*, 200–221. [CrossRef]
3. Kar, J. Provably secure certificateless deniable authenticated encryption scheme. *J. Inf. Secur. Appl.* **2020**, *54*, 102581. [CrossRef]
4. Hossain, M.B.; Natgunanathan, I.; Xiang, Y.; Zhang, Y. Cost-Friendly Differential Privacy of Smart Meters Using Energy Storage and Harvesting Devices. *IEEE Trans. Serv. Comput.* **2022**, *15*, 2648–2657. [CrossRef]
5. Zhao, R.; Zhang, Y.; Zhu, Y.; Lan, R.; Hua, Z. Metaverse: Security and Privacy Concerns. *arXiv* **2022**, arXiv:2203.03854. [CrossRef]
6. Dong, X.; Zhang, W.; Zhang, Y.; You, Z.; Gao, S.; Shen, Y.; Wang, C. Optimizing Task Location Privacy in Mobile Crowdsensing Systems. *IEEE Trans. Ind. Inform.* **2022**, *18*, 2762–2772. [CrossRef]
7. Zhou, M.; Zheng, Y.; Wang, S.; Hua, Z.; Huang, H.; Gao, Y.; Jia, X. PPTA: A location privacy-preserving and flexible task assignment service for spatial crowdsourcing. *Comput. Netw.* **2023**, *224*, 109600. [CrossRef]
8. Dwork, C.; Naor, M.; Sahai, A. Concurrent zero-knowledge. *J. ACM* **2004**, *51*, 851–898. [CrossRef]

9. Aumann, Y.; Rabin, M.O. Authentication, enhanced security and error correcting codes. In Proceedings of the 18th Annual International Cryptology Conference, Santa Barbara, CA, USA, 23–27 August 1998; Springer: Berlin/Heidelberg, Germany, 1998; pp. 299–303.
10. Deng, X.; Lee, C.; Zhu, H. Deniable authentication protocols. *IEE Proc.-Comput. Digit. Tech.* **2001**, *148*, 101–104. [CrossRef]
11. Diffie, W.; Hellman, M.E. New directions in cryptography. In *Secure Communications and Asymmetric Cryptosystems*; Routledge: England, UK, 2019; pp. 143–180.
12. Fan, L.; Xu, C.; Li, J. Deniable authentication protocol based on Deffie-Hellman algorithm. *Electron. Lett.* **2002**, *38*, 705–706. [CrossRef]
13. Yoon, E.J.; Ryu, E.K.; Yoo, K.Y. Improvement of Fan et al.'s deniable authentication protocol based on Diffie–Hellman algorithm. *Appl. Math. Comput.* **2005**, *167*, 274–280. [CrossRef]
14. Sow, D.; Camara, M.G. Provable security of the generalized elgamal signature scheme. *J. Math. Res.* **2019**, *11*, 1–77. [CrossRef]
15. Shao, Z. Efficient deniable authentication protocol based on generalized ElGamal signature scheme. *Comput. Stand. Interfaces* **2004**, *26*, 449–454. [CrossRef]
16. Lu, R.; Cao, Z. Non-interactive deniable authentication protocol based on factoring. *Comput. Stand. Interfaces* **2005**, *27*, 401–405. [CrossRef]
17. Wang, Y.; Li, J.; Tie, L. A simple protocol for deniable authentication based on ElGamal cryptography. *Netw. Int. J.* **2005**, *45*, 193–194. [CrossRef]
18. ElGamal, T. A public key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inf. Theory* **1985**, *31*, 469–472. [CrossRef]
19. Fun, T.S.; Beng, L.Y.; Likoh, J.; Roslan, R. A lightweight and private mobile payment protocol by using mobile network operator. In Proceedings of the 2008 International Conference on Computer and Communication Engineering, Kuala Lumpur, Malaysia, 13–15 May 2008; IEEE: Piscataway, NJ, USA, 2008; pp. 162–166.
20. Yoon, E.J.; Yoo, K.Y.; Yeo, S.S.; Lee, C. Robust deniable authentication protocol. *Wirel. Pers. Commun.* **2010**, *55*, 81–90. [CrossRef]
21. Li, F.; Takagi, T. Cryptanalysis and improvement of robust deniable authentication protocol. *Wirel. Pers. Commun.* **2013**, *69*, 1391–1398. [CrossRef]
22. Lee, W.B.; Wu, C.C.; Tsaur, W.J. A novel deniable authentication protocol using generalized ElGamal signature scheme. *Inf. Sci.* **2007**, *177*, 1376–1381. [CrossRef]
23. Wang, B.; Song, Z. A non-interactive deniable authentication scheme based on designated verifier proofs. *Inf. Sci.* **2009**, *179*, 858–865. [CrossRef]
24. Li, F.; Xiong, P.; Jin, C. Identity-based deniable authentication for ad hoc networks. *Computing* **2014**, *96*, 843–853. [CrossRef]
25. Liao, Y.; He, Y.; Li, F.; Zhou, S. Analysis of a mobile payment protocol with outsourced verification in cloud server and the improvement. *Comput. Stand. Interfaces* **2018**, *56*, 101–106. [CrossRef]
26. Li, F.; Zhong, D.; Takagi, T. Efficient deniably authenticated encryption and its application to e-mail. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 2477–2486. [CrossRef]
27. Zhang, Y.; Wen, L.; Zhang, Y.; Wang, C. Designated server certificateless deniably authenticated encryption with keyword search. *IEEE Access* **2019**, *7*, 146542–146551. [CrossRef]
28. Ahene, E.; Jin, C.; Li, F. Certificateless deniably authenticated encryption and its application to e-voting system. *Telecommun. Syst.* **2019**, *70*, 417–434. [CrossRef]
29. Kar, J.; Naik, K.; Abdelkader, T. An efficient and lightweight deniably authenticated encryption scheme for e-mail security. *IEEE Access* **2019**, *7*, 184207–184220. [CrossRef]
30. Zhang, Y.l.; Wen, L.; Zhang, Y.j.; Wang, C.f. Deniably authenticated searchable encryption scheme based on Blockchain for medical image data sharing. *Multimed. Tools Appl.* **2020**, *79*, 27075–27090. [CrossRef]
31. Bojjagani, S.; Sastry, V.; Chen, C.M.; Kumari, S.; Khan, M.K. Systematic survey of mobile payments, protocols, and security infrastructure. *J. Ambient Intell. Humaniz. Comput.* **2023**, *14*, 609–654. [CrossRef]
32. Alidadi Shamsabadi, F.; Bakhtiari Chehelcheshmeh, S. A cloud-based mobile payment system using identity-based signature providing key revocation. *J. Supercomput.* **2022**, *78*, 2503–2527. [CrossRef]
33. Alshammari, M.; Nashwan, S. Fully Authentication Services Scheme for NFC Mobile Payment Systems. *Intell. Autom. Soft Comput.* **2022**, *32*, 401–428. [CrossRef]
34. Jin, C.; Zhao, J. Efficient and Short Identity-Based Deniable Authenticated Encryption. In Proceedings of the Third International Conference, ICCCS 2017, Nanjing, China, 16–18 June 2017; Revised Selected Papers, Part II; Sun, X., Chao, H.C., You, X., Bertino, E., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 244–255.
35. Zhang, Y.; Xu, C.; Lin, X.; Shen, X. Blockchain-Based Public Integrity Verification for Cloud Storage against Procrastinating Auditors. *IEEE Trans. Cloud Comput.* **2021**, *9*, 923–937. [CrossRef]
36. Zhang, Y.; Xu, C.; Cheng, N.; Shen, X. Secure Password-Protected Encryption Key for Deduplicated Cloud Storage Systems. *IEEE Trans. Dependable Secur. Comput.* **2022**, *19*, 2789–2806. [CrossRef]
37. Li, S.; Zhang, Y.; Xu, C.; Cheng, N.; Liu, Z.; Du, Y.; Shen, X. HealthFort: A Cloud-Based Ehealth System with Conditional Forward Transparency and Secure Provenance Via Blockchain. *IEEE Trans. Mob. Comput.* **2022**, 1–18. [CrossRef]

38. Rasmussen, K.; Gasti, P. Weak and Strong Deniable Authenticated Encryption: On their Relationship and Applications. In Proceedings of the 2018 16th Annual Conference on Privacy, Security and Trust (PST), Belfast, Ireland, 28–30 August 2018; pp. 1–10. [CrossRef]
39. Chen, G.; Zhao, J.; Jin, Y.; Zhu, Q.; Jin, C.; Shan, J.; Zong, H. Certificateless Deniable Authenticated Encryption for Location-Based Privacy Protection. *IEEE Access* **2019**, *7*, 101704–101717. [CrossRef]
40. Huang, W.; Liao, Y.; Zhou, S.; Chen, H. An Efficient Deniable Authenticated Encryption Scheme for Privacy Protection. *IEEE Access* **2019**, *7*, 43453–43461. [CrossRef]
41. Jin, C.; Chen, G.; Yu, C.; Zhao, J. Deniable authenticated encryption for e-mail applications. *Int. J. Comput. Appl.* **2020**, *42*, 429–438. [CrossRef]
42. Jin, C.; Kan, G.; Chen, G.; Yu, C.; Xu, C. Heterogeneous Deniable Authenticated Encryption Protocol. In Proceedings of the Third International Conference, FCS 2020, Tianjin, China, 15–17 November 2020; Xu, G., Liang, K., Su, C., Eds.; Springer: Singapore, 2020; pp. 331–346.
43. Jin, C.; Kan, G.; Chen, G.; Yu, C.; Jin, Y.; Xu, C. Heterogeneous deniable authenticated encryption for location-based services. *PLoS ONE* **2021**, *16*, e0244978. [CrossRef]
44. Zeng, S.; Zhang, H.; Hao, F.; Li, H. Deniable-Based Privacy-Preserving Authentication Against Location Leakage in Edge Computing. *IEEE Syst. J.* **2022**, *16*, 1729–1738. [CrossRef]
45. Cao, Y.; Wei, J.; Zhang, F.; Xiang, Y.; Chen, X. Efficient public-key authenticated deniable encryption schemes. *Comput. Stand. Interfaces* **2022**, *82*, 103620. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.