Abstract and Plan of work

Title of the project

Light-Weight and Privacy-Preserving Authentication Protocol for Mobile Payments in the Context of IoT.

Group Members: Names and Roll Numbers

Name	Roll Number	Email
Ashutosh Chauhan	S20180010017	ashutosh.c18@iiits.in
Saumya Doogar	S20180010156	saumya.d18@iiits.in

Abstract

In the current world IoT is getting widespread over a wide range of scenarios. From Smart Home to smart cards etc, IoT is becoming the new norm. One of the major characteristics IoT is being lightweight in terms of size, power consumption, performance, storage, etc. The protocol mentioned here provides security while still making sure that the performance and storage are within the capabilities of a IoT system. The protocol provides privacy and authentication for mobile payments in context of IoT. There are a various number of use cases from payments from smart devices like smart electric meters, smart cars, smart watches, monitoring systems, etc. The protocol uses at unidirectional certificateless proxy re-signature scheme, which is of independent interest. Based on this signature scheme, the protocol achieves anonymity, unforgeability and low performance overhead. In this protocol the computational overhead is placed on the Pay Platform. To increase the efficiency of the protocol, a batch-verification mechanism is provided for the Pay Platform and Merchant Server. The security of the protocol is based on the CDH (Computational Diffie-Hellman) Problem.

Related Works

1. **Certificateless Proxy Re-Signatures**

   Proxy re-signature 1998 by **Blaze et al** is an extension of digital signature, with a semi-trusted proxy to transform delegatee's signature into delegator's signature on the same message by using the re-signing key, but the proxy is incapable on signing on its own. In the original proxy re-signature scheme, the public keys of the delegator and delegatee was required to be certified by the digital certificate prior to the verification of signature itself. To mitigate the heavy costs, current solution uses identity-bases proxy re-signature has been introduced, from user's identity. One disadvantage of identity-based proxy re-signature is called "key-escrow", where private key is generated by a fully trusted private key generator. To solve both the certificates management and key-escrow problem, we use a unidirectional proxy re-signature system.

2. **Mobile Payment Protocols**

   We referred two protocol Sureshkumar et al[1] and Yang and Lin[2] but they both had a few drawbacks. Protocol [1] cannot provide non-repudiation and Protocol [2] has very high cost for generation of certificates. Then Qin et al[3] and Yeh[4] proposed a new protocol which provided anonymity, unforgeability, and certificate-less property. Liao et al. [5] found that the verification of Qin et al.[3]'s protocol is insecure that users could collude with the untrusted cloud server to cheat Merchant Server. Then they improved Qin et al.'s protocol to realize secure verification. However, both Qin et al.'s and Yang et al.' protocols will produce multiple pseudo identities to hide the real user identity, so a lot of storage spaces are consumed on the resource-limited users. Most recently, Yeh[4] proposed a transaction protocol based on certificateless cryptographic primitives. In **Yeh**'s protocol, an efficient certificateless signature which does not need any certificate to ensure the legality of public key and private key pairs is adopted to achieve secure transaction. In a nutshell, **Yeh**'s protocol has made great progress in the mobile payment protocol that we can complete the transaction protocol at anytime and anywhere efficiently in smartphones.

BILINEAR MAPS

Here we use G_1 and G_2 to denote two cyclic additive groups with order q. And P is a generator of G_1. If $e : G_1 \times G_1 \rightarrow G_2$ is a bilinear map, it should satisfy the following conditions:

1) Bilinearity, that is, for $\forall x, y \in Z_q$, the equation $e(xP, yP) = e(P, P)^{xy}$ should be hold;

2) Non-degeneracy, that is, $e(P, P) \neq 1$.

FRAMEWORK OF CERTIFICATELESS PROXY RE-SIGNATURE

The unidirectional certificateless proxy re-signature scheme consists of the following eight algorithms:

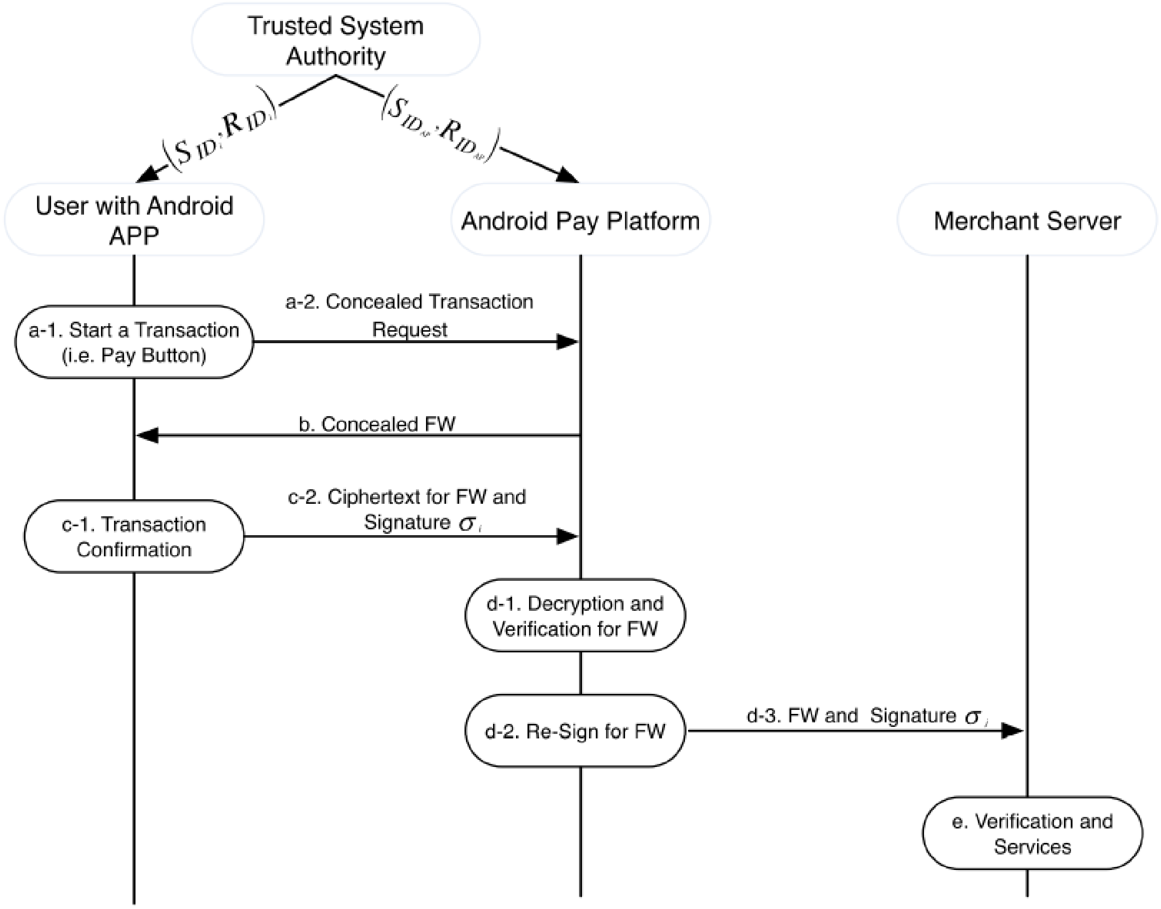
- Setup: On input the security parameter k, the algorithm generates the master secret key msk, the master public key $PK_{pub}$and the system parameters params.
- Partial-Private-Key-Extract: On input the system parameters params and an identity ID of the user, the algorithm generates the user's partial private key D_{ID}.
- Set-Secret-Value: On input the system parameters params and an identity ID of the user, the algorithm generates the user's secret value x_{ID}.
- Set-Public-Key: On input the system parameters params and the user's secret value x_{ID}, the algorithm generates the user's public key P_{ID}.
- ReKey: On input the system parameters params, the delegatee's identity ID_i and public key P_i , as well as the delegator's secret key (D_j, x_j) associated with the identity ID_j and public key P_j , the algorithm generates the re-signature key $rk_{i,j}$.
- Sign: On input the system parameters params, a message m the user's secret key (D_{ID}, x_{ID}) associated with the identity ID and public key P_{ID}, the algorithm generates two kinds of signatures σ on message m.
- ReSign: On input the re-signature key $rk_{i,j}$, the delegatee's public key P_i and a signature σ_i on message m with the identity ID_i , the algorithm generates the re-signature σj on message m with the identity ID_j .
- Verify: On input the system parameters params and the user's public key P_{ID}, the algorithm checks the validity of signature σ on message m under the identity ID. If σ is valid, the algorithm outputs 1; ⊥, otherwise

Plan of Implementation

SYSTEM MODEL OF OUR TRANSACTION PROTOCOL

The considered system consists of four types of entities: the trusted system authority (TSA), the user app, the merchant server, and the Pay Platform [9].

1. **Trusted System Authority**: **TSA** is a trusted third party organization that provides registration services for User’s App and Pay Platform. TSA also distributes system params and partial private keys for registered users to ensure the whole scheme successfully works.
2. **User’s App**: Any software that requires a payment function is called User’s App, such as Apple pay, etc. This application needs to be registered with the TSA to obtain the corresponding system params and partial private key. It also generates its own user secret value and public key. Then User’s App completes the signature using its full private key, which consists of partial private key.
3. **Pay Platform**: Pay Platform is an application offered by a trusted party, of course, it also needs to register with the TSA to obtain system params and private key. Simultaneously, in order to protect the user’s information of the transaction, Pay Platform will provide re-sign service, that is, the Pay Platform transforms signature of User’s App into signature of Pay Platform.
4. **Merchant Server**: Merchant Server is utilized by a merchant, it verifies the correctness of the transaction information to check the product is given to the right user.


Trusted System Authority
(S_{ID_i}, R_{ID_i})
$(S_{ID_{AP}}, R_{ID_{AP}})$
User with Android APP
Android Pay Platform
Merchant Server
a-1. Start a Transaction (i.e. Pay Button)
a-2. Concealed Transaction Request
b. Concealed FW
c-1. Transaction Confirmation
c-2. Ciphertext for FW and Signature σ_i
d-1. Decryption and Verification for FW
d-2. Re-Sign for FW
d-3. FW and Signature σ_j
e. Verification and Services

Performance Analysis

We will time the average time for multiple transaction in batches of 10, 50, 100, 250 to get a accurate min, max, average, median time taken per transaction.

We will check storage requirements based on key sizes.

We might also verify memory analysis if possible.

We do not have access to low power system configurations so we will try to use our personal machines for the same. But will try to extrapolate the estimated performance analysis for the same.

Experimental Setup: System configurations, Libraries used

Configuration for User's App (Original)

- **CPU**: PXA270 processor 624MHz
- **RAM**: 1GB memory

Configuration for Payment Platform (Original)

- **CPU**: Intel i3-380M processor 2.53GHz
- **RAM**: 8GB memory

Hash Function: SHA-3

G_1, Z_q → 64 Byte

G_2 → 128 Byte

ECC → $y^2 = x^3 + x$

Paper's Usage

- VC++ 6.0
- PBC library

Our Usage

- GNU G++
- PBC library

Summary of the results

We will provide the summary reporting the performance and security benefits of the platform and the feasibility to us in IoT solutions.

References

[1]: V. Sureshkumar, A. Ramalingam, N. Rajamanickam, and R. Amin, “A lightweight two-gateway based payment protocol ensuring account□ability and unlinkable anonymity with dynamic identity,” Comput. Elect. Eng., vol. 57, pp. 223–240, Jan. 2017.

[2]: J-H. Yang and P.-Y. Lin, “A mobile payment mechanism with anonymity for cloud computing,” J. Syst. Softw., vol. 116, pp. 69–74, Jun. 2016.

[3]: Z. Qin, J. Sun, A. Wahaballa, W. Zheng, H. Xiong, and H. Qin, "A secure and privacy-preserving mobile wallet with outsourced verification in cloud computing," Comput. Standards Interfaces, vol. 54, pp. 55–60, Nov. 2017.

[4]: K.-H. Yeh, "A secure transaction scheme with certificateless cryptographic primitives for IoT-based mobile payments," IEEE Syst. J., doi: 10.1109/ JSYST.2017.2668389

[5] Y. Liao, Y. He, F. Li, and S. Zhou, "Analysis of a mobile payment protocol with outsourced verification in cloud server and the improve□ment," Comput. Standards Interfaces, vol. 56, pp. 101–106, Feb. 2018, doi: 10.1016/j.csi.2017.09.008.