Money Is All I Transfer

By
Udbhav Singh Sikarwar, Akshat Gupta

1 Introduction

We have developed a solution that enables money transfers without revealing any transaction details, even to intermediaries like NPCI (National Payments Corporation of India). This solution is fully **integrable** with the current digital payment infrastructure, **without compromising the convenience** users expect. The only information disclosed is to the payer, who is notified of the deduction from their account, and to the payee, who is informed when the money is credited to their account. Additionally, we have incorporated features to ensure the protocol's robustness, preventing any individual party from committing fraud or engaging in money laundering.

This paper is divided into five sections: **Introduction**, **Background**, **Protocol**, **User Perspective**, and **Deliverables**. The **Background** section clarifies essential terms before diving into the details, the **User Perspective** explores how users will interact with the system, and the **Protocol** section provides a detailed explanation of the system's operations. Finally, **Potential Modifications** address any changes that might be necessary during implementation, such as adjustments if financial institutions do not fully adopt the protocol.

A reasonable assumption we made is that the payer and payee are physically present near each other during the transaction. This proximity is only necessary to confirm that the transaction has actually occurred, as neither party is notified of the other's transaction status—each is only informed about their own account activity. This assumption is justified, as if the payer and payee were far apart and needed to discuss the transaction, they would likely already have each other's contact information, implying prior communication.

2 BackGround

2.1 Types of Servers

- **Honest Server:** An honest server follows the protocol exactly and does not deviate from the defined steps. It does not attempt to learn any additional information beyond what it is supposed to know.
- **Semi-Honest Server (Honest-But-Curious):** A semi-honest server, or honest-but-curious, follows the protocol correctly but tries to learn as much as possible from the data it processes. It does not alter the input or output but may analyze the data to gain unauthorized information.
- **Malicious Server:** A malicious server deviates from the protocol and can send false data, refuse to follow steps, or collude with other parties. It aims to learn secret information or disrupt the protocol.

  Our Protocol if followed completely should be robust against a **malicious server**

2.2 ECC (Elliptic Curve Cryptography)

It is a public key protocol which provides **Asymmetric encryption** i.e with separate encryption and decryption keys which will be used throughout the protocol for encryption. This was chosen over standard RSA protocol due to its higher efficiency, smaller key sizes and its robust nature.

2.3 Additive Sharing

This is a method used for secret sharing amongst multiple parties without disclosing the contents of

the message to the parties itself. A message x can be split into n additive shares ($n > 1$) with relative ease where each share reveals nothing to its recipient but their combination reveals the message.

$$x = x_1 + x_2 + \cdots + x_n$$

3 Protocol

In any secure cash transaction that ensures transaction integrity, the payer counts the cash before handing it over to the payee, who then reconfirms the amount by counting it again. This transaction process ensures that only the payer and the payee know the amount. To keep NPCI from knowing the full transaction details the instant debit and credit of money from payer and payee banks respectively is kept at a certain time delay (anywhere in the range of 100-500 ms) in accordance with the IST and is thus cumulatively given to NPCI to keep each transaction amount hidden. The protocol further proceeds as follows:

Note: Steps 1, 2, and 3 do not need to be repeated every time.

1. **Generate QR**: A request is put in by the Payee to the Payee PSP for a QR to initiate the transaction.

2. **Generation of QR**: The Payee UPI ID is and split into 2 additive shares to disable direct communication between Payer and Payee PSP. In addition a key pair K_1, K_2 for encryption of the amount is generated.

3. **Payee PSP Response a:** an additive share is sent along with a message ID to NPCI for future validation.
   Payee PSP Response b: a message ID along with a QR sent which contains the encryption key K_1 and the second share of encrypted UPI ID.
   The **message ID is sent throughout** to keep track of the transaction.

4. **QR Scan:** Payer gets the information stored in the QR.

5. **Credentials:** The payer enters his amount and chooses the his prefered bank account.

6. **Pay request:** Payer clicks on the pay option after verifying details, essentially counting his cash.

7. **Payer PSP Debit Response a:** Bank name with additive shares of Payer IFSC code, Account information,UPI ID along with the amount encrypted with K_1 is sent.
   Payer PSP Debit Response b: Additive shares of Payer IFSC code, Account information (all information needed to withdraw money) along with the amount is sent.
   This step ensures NPCI would not know the payer or transaction details but only verify them.

8. **NPCI validation:** NPCI computes and gets the encrypted UPI ID and validates the encrypted amount with the Payee PSP.

9. **Payee PSP validation a:** Payee PSP decrypts the amount and UPI ID and sends a confirmation link to the Payee.
   Payee validation b: Payee confirms the payment after verifying the amount is correct, essentially counting the cash again.

10. **NPCI Debit Request:** After next timestamp NPCI sends Each Payer Bank all its pay requests which contains the remaining additive shares of IFSC code and account number which is matched by the unique message ID.

11. **Debit and store:** The bank debits the amount mentioned in step 7b from each account and cumulatively stores it.

12. **Payer Debit Status:** The bank provides failed and successful transaction for each message ID.

13. **Payer Bank Debit Response:** The sum is transferred to NPCI where it is held for verification.

14. **Status to Payer PSP:** Status details are sent to their respected PSPs via message ID.

15. **Debit status to Payer:** Failed or money debited successfully.

16. **Sum for NPCI Payer Bank Validation:** Sum is provided by Payer PSP for each bank to NPCI for validation of total successful transaction amount.

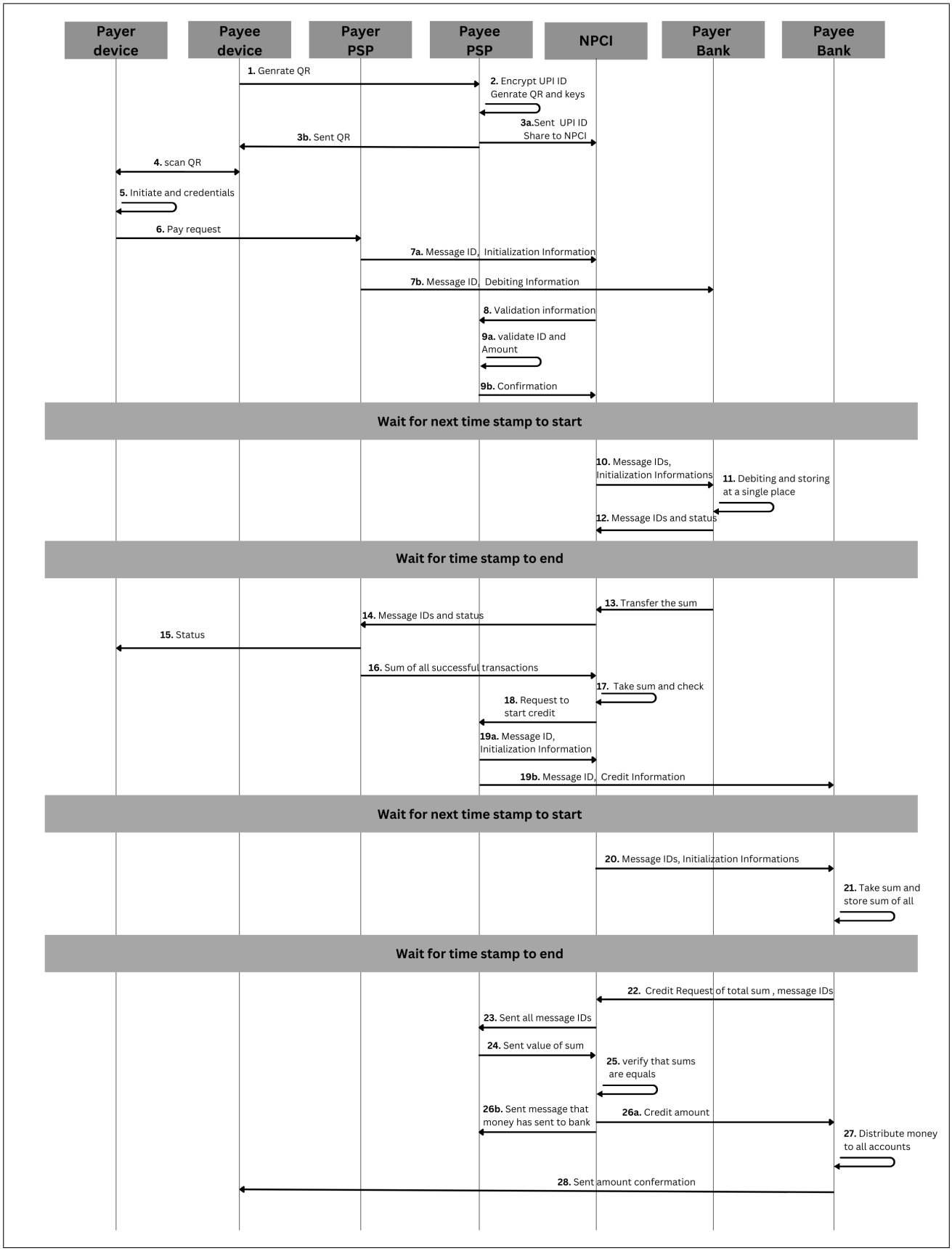

Fig.4.1. The process flow diagram. **Note:** The diagram is provided to assist in understanding the protocol. However, relying solely on the diagram may be misleading. For instance, at step 7b, the debiting information does not include any sensitive details, such as account numbers or personal information.

17. **NPCI Payer Bank Validation:** Take sum of values sent by different PSP for specific banks and check value.
18. **Request to Payee PSP:** Marks the end of Debit Sequence and initiates Credit Sequence.
19. **Payee PSP Credit Response a:** Bank name with

additive shares of Payee IFSC code and Account information is sent.
Payer PSP Debit Response b: Additive shares of Payee IFSC code and Account information along with the amount is sent.

20. **NPCI Credit Request:** After next timestamp NPCI sends Each Payee Bank all its credit requests which contains the remaining additive shares of IFSC code and account number (details needed to credit money) which is matched by the unique message ID.
21. **Payee Bank Sum:** Payee Bank then accumulates the amount to be credited from each successful debit to get a total sum.
22. **Payee Bank Request:** Payee Bank sends total sum to NPCI for verification.
23. **NPCI Payee PSP Request:** NPCI asks for sum of credit via message IDs.
24. **NPCI Payee PSP Response:** Payee PSP gives bank wise sum for verification with Payee Bank request.
25. **NPCI Sum Validation:** Validate that that sum is equal to the sum of all values sent by all different PSPs.
26. **NPCI Credit Response a:** Required sum is transferred from NPCI to Payee Bank for redistribution.
NPCI Credit Response b: Status Update to Payee PSP.
27. **Payee Bank Redistribution:** Redistribution according to each amount for each account.
28. **Payee Credit Status:** Final credit message from Payee Bank to Payee finishing the protocol.

4 User Perspective

The **Payer perspective will not change** as they would still have to scan and pay in the same way they currently do. **The Payee perspective will change** in this protocol due to amount verification (analogous to recounting cash), which is the accept payment message specifying the amount which is currently given by a person.

5 Deliverables

- **User Privacy:** The Details of each user is known only by 3 entities which are User Device, User PSP and User Bank.
- **Transaction Integrity:** The Payer Bank can check if Payer has sufficient balance on **step 11**. If not then an error message of insufficient balance would be sent.
- **Fraud Prevention:** Checks and verification points are put throughout the protocol to ensure that there is no chance of fraud from the Payer and Payee side due to **no Payer to Payee communication**. Bank Fraud is also an unlikely event as the combined sum of credit and debit is monitored by NPCI. If one were to manipulate by, for example increasing their own credit amount and decreasing the other then **due to difference in amount verified through the PSP and amount credited**, the User is alerted to the fraud attempt. **Note:** We have not implemented **zero-knowledge proof** because it still carries a probability, however small, of fraud. When handling such large sums of money, even a minimal risk is unacceptable. Our protocol completely eliminates this possibility, ensuring total security.