Design Principles for Third-party Initiation in Real-time Payment Systems

JJ Geewax
Google
jjg@google.com

ABSTRACT

Over the past several years, more and more countries around the world are seeing the value of Real-Time Payment (RTP) systems as a core piece of infrastructure to enable peer-to-peer payments between citizens as well as commercial payments to merchants (e.g., via Quick Response codes (QRs) [12]). The roll-out of Unified Payments Interface (UPI) in India has shown that an important piece of functionality and driver of adoption of that system was the ability for third-parties to initiate payments on behalf of users [15]. This paper aims to define the guidelines and best practices for introducing support for third-party participants on RTP systems, in particular for third-parties aiming to act as payment initiators.

These guidelines have been split into three categories: security, privacy, and user-experience. The conclusions are based on a thorough examination of the downstream consequences of alternatives, real-world experience of integrating with several different systems, and work done to build a reference implementation of Third-party Payment Initiation (3PPI) in an existing open-source RTP system called Mojaloop [7]. Drawing on this work, the final section outlines implementation guidelines for building support for 3PPI in a RTP system.

1 INTRODUCTION

As more and more Real-Time Payment (RTP) systems are coming online around the world (e.g., the FedNow system in the United States [20] or Unified Payments Interface (UPI) in India [9]), one feature of interest is the general ability for third-parties to participate on these systems. And while there are many benefits that third-party participants can offer, one of the most straight-forward and valuable features is Third-party Payment Initiation (3PPI).

1.1 Third-party payment initiation

3PPI is the ability for a third-party participant (Payment Initiation Service Provider (PISP) as defined by Revised Payment Services Directive (PSD2) [18]) to initiate a payment on behalf of an end user, while relying on another participant (e.g., a Financial Service Provider (FSP)) to act as the source of funds being transferred. This ultimately allows an end user to interact exclusively with a PISP (via a website, mobile application, or SMS short code) in order to send a payment, even though the PISP holds no funds or liquidity of its own, and is incapable of transferring funds directly via the RTP system. Instead, the PISP uses the RTP system to request that the Payer's FSP transfer the funds on behalf of the end user, almost as though it were a normal RTP peer-to-peer payment.

Recent metrics have shown that the granting access to third-party participants on RTP systems is correlated with an increase in payments overall, whether that access was specifically for the purpose of payment initiation or some other functionality such as providing account information or analysis. In the case of UPI in India, introduction of third-party participants was correlated with exponential growth between August 2016 and November 2020 [10]. This may have been due at least in part to the fact that FSPs tend to provide the core functionality such as transferring funds and providing account information, whereas third-parties, such as financial technology companies ("fintechs"), are likely to provide more novel functionality built on top of these core features provided by FSPs. It may also be due to the fact that a wider selection of available participants leads to more avenues to lure users to using the RTP system as a whole.

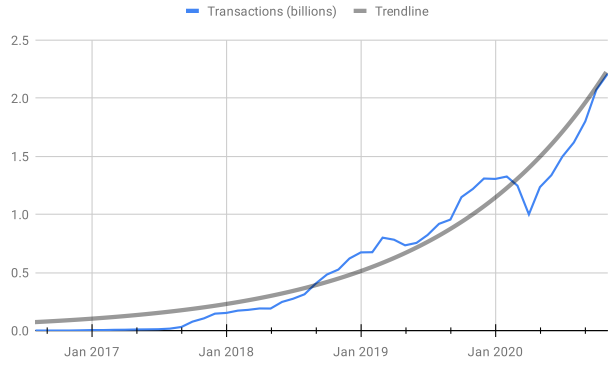

Figure 1: Transactions per month on UPI in India.

While the rapid growth of adoption of RTP systems around the world is leading to more efficient transfers and more financial inclusion, it also has led to more fragmentation. Thanks to an increasing number of technology providers developing software for RTP systems and many countries opting to develop their own RTP systems independently (e.g., Brazil's Pix system [4] or Australia's New Payments Platform (NPP) system [2]), the number of RTP system implementations is vast and varied. And while multiple different implementations of RTP systems is not inherently a bad thing, variations in the design for how third-parties should participate in RTP systems can lead to technological difficulty and extra work for those interested in integrating as a third party participant in any capacity. And since most RTP systems have yet to introduce support for third-party participants, there is still opportunity to avoid the same type of fragmentation for how third parties should interact with various RTP systems, particularly in the case of payment initiation by defining a set of design principles for 3PPI functionality.

1.2 Participants

For any third-party initiated payment, there are always three different participants:

(1) A **Payer**, generally a person with a bank account or mobile wallet account,
(2) A **PISP**, a participant that holds no funds and does not manage a ledger, but interacts directly with the Payer, and
(3) A **FSP**, a mobile wallet, bank, or other financial institution that the Payer maintains an account with. The FSP also manages their ledger and is capable of sending funds using the RTP system.

For any payment handled by the RTP system there is always a recipient of funds, which we'll refer to as a Payee. While relevant to the high-level understanding of the process, the Payee is a passive participant in this process and will therefore only be brought up as needed.

1.3 Phases of third-party initiation

We can break the standard implementation of 3PPI into two pieces:

(1) **Linking**, where we establish shared trust between all three parties (Payer, PISP, and FSP), and
(2) **Transfer**, where we rely on that trust to make a payment from a Payer to a Payee.

In the following sections, we'll cover the primary goals of each phase with the overall objective of processing a third-party initiated payment from a Payer to a Payee.

1.3.1 Linking. The Linking phase has one primary goal: establish trust between all three parties based on a patchwork of existing trust relationships to be relied upon later during the Transfer phase (Section 1.3.2). To start, the Payer trusts their FSP based on a preexisting relationship. The Payer has presumably opened an account with their bank or signed up for a mobile wallet with their telecommunications provider and has a mechanism by which they can authenticate themselves with the provider. Further, the Payer trusts their PISP based on a similarly preexisting relationship. The Payer has presumably signed up for the PISP's services by installing their mobile application, website, or SMS short code application.

With that said, it's important to note that there is no mutual trust between the FSP and the PISP. While it may be the case that the PISP and FSP have both been vetted by the RTP system as participants on the network, this is not always the case and so we must treat this trusted relationship as one that has yet to be established.

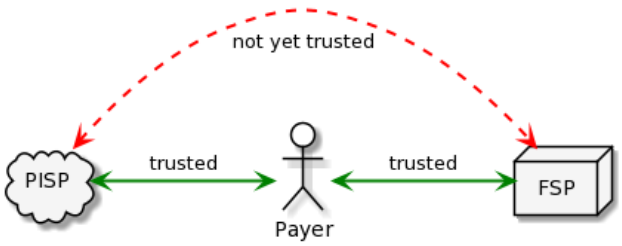

Figure 2: Parties and their lines of trust.

To establish trust between all three parties, we must build upon the individual units of trust and rely on either the Payer or the RTP system to act as a conduit of some secret information to establish the final leg of trust between the PISP and the FSP. This means that after the Payer authenticates with the FSP, there are two options.

One option, shown in Figure 3, is that the Payer can inform the FSP that they intend to link their account so that the PISP can initiate payments drawn on that account (L_1). The FSP can collect some sort of credential (e.g., a public key) and provide that to the RTP system (L_2), which can then be passed onto the PISP, informing the PISP that they are now capable of initiating payments from a specific account with the FSP (L_3). Finally, the PISP can inform the Payer of the established link (L_4). In this case, the Payer is using the trust between themselves and the FSP to inform the FSP of their trust with a specific PISP.

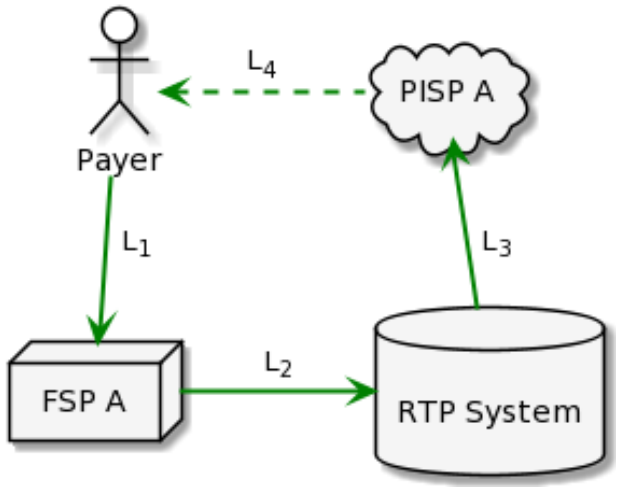

Figure 3: Linking process initiated by the FSP.

The other option, shown in Figure 4 is that after authenticating (L_{1a}) the FSP can provide the Payer with some sort of secret that can be used as a pairing token (L_{1b}). The Payer then has the option of providing this key to the PISP (L_2), and in doing so, the PISP can request access to initiate payments with the FSP acting as the source of funds. It does so by providing the pairing key via the RTP (L_3) to the FSP (L_4) as a way of proving that the PISP is, indeed, trusted by the Payer. In this case, the FSP is relying on the trust between the Payer and the PISP as a way of determining whether to trust the PISP directly.

Regardless of the mechanism, once this mutual, multi-party trust is established the PISP should be able to relay a message to the FSP on behalf of the user and the FSP should be capable of verifying the authenticity of that message. In this case, the message will ultimately be a request to send funds to a Payee, so that once the FSP has verified the relayed message, the FSP can transfer the funds to the Payer as though the Payee had communicated directly with the FSP in the first place.

1.3.2 Transfer. Similarly, the Transfer phase has a single goal as well: transfer funds from a Payer to a Payee. However, rather than

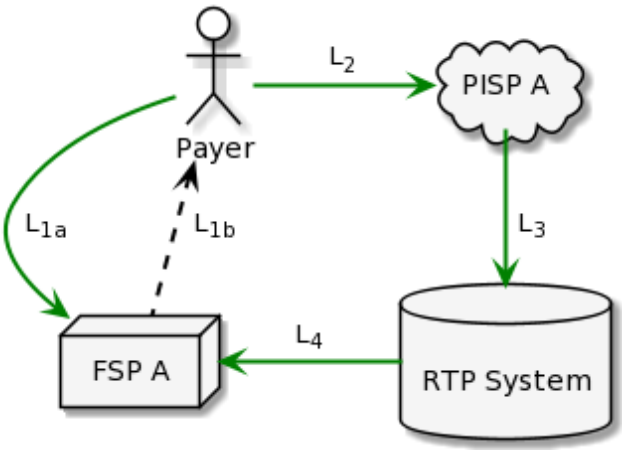

Figure 4: Linking process using a shared pairing token.

sending funds directly (since the PISP is incapable of doing so), the PISP must instead request that the Payer's FSP send the funds on behalf of the Payer. To understand how this works, it's important to first understand how a typical transfer on an RTP system works, separate from one that was initiated by a third-party.

The standard flow of a payment through a RTP network typically consists of four stages, shown in Figure 5. The payment starts with a Payer talking to their FSP to request a payment be sent to a specific Payee. Upon satisfactory authentication and authorization for the payment between the Payer and the FSP (P_1), the FSP then sends the payment request to the RTP system (P_2) to the Payee's FSP via the RTP. The RTP system passes this request onto the Payee's FSP (P_3) and the funds are ultimately delivered to the Payee (P_4).

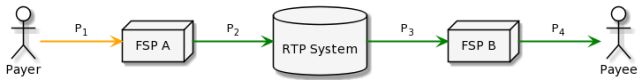

Figure 5: Standard payment flow on RTP systems.

A third-party initiated payment, shown in Figure 6, behaves in much the same way, however the process for requesting the payment in the first place is slightly different. This type of payment starts with the Payer asking their PISP to send funds to a specific Payee ($Init_1$). The PISP conveys this intent to send funds from the Payee onto the RTP system ($Init_2$) to be passed along to the Payer's FSP ($Init_3$). At this point, the payment looks identical to that shown in Figure 5, with steps P_2 through P_4 executing in exactly the same manner. In effect, step P_1 from Figure 5 is replaced by steps $Init_1$ through $Init_3$ in Figure 6.

2 DESIGN PRINCIPLES

Now that we've laid out the groundwork for how third-party payment initiation works we can get into the detailed principles to consider when designing a 3PPI implementation. These principles are high-level and implementation independent, focusing on the end result rather than the manner in which the principle is upheld. For example, rather than stating a specific security feature that must

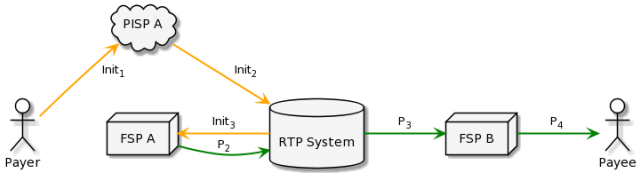

Figure 6: Flow of third-party initiated payments on a RTP system.

be implemented to avoid an attack vector, the principle remains focused on the security aspect in question and why it's important to protect against that specific attack vector.

The principles are broken down into several sections:

(1) **Security** focuses on authorization and authentication to avoid unauthorized transfers,
(2) **Privacy** covers protections for user information stored by FSPs and PISPs, and
(3) **User experience** focuses on the interfaces and flows that users see when linking accounts and making payments (generally, on a mobile device).

2.1 Security

One of the primary concerns about permitting third-party participation on RTPs is that they introduce additional risk related to security ranging from user-provided information leaks to fraudulent transaction initiation. The following principles address what appear to be the most critical potential vulnerabilities. Note that there is a baseline assumption that all participants on RTPs have a sufficient level of security built in as a prerequisite of approval by a regulator to participate on the network, and the system operator has a responsibility to ensure that all participants are complying with the rules and regulations of the network itself. The goal of these principles is to assume that a PISP may become compromised and ensure that the damage is limited as much as possible.

2.1.1 Application Programming Interfaces (APIs) must be defined and serviced by the RTP system. The examples so far have centered around PISPs interacting with a central RTP system as a direct participant much like any other FSP. While this is a perfectly fine model, there may be concern about PISPs connecting directly to the RTP system, particularly when considering questions of which party will take on the risk and liability for fraudulent transactions. This concern is equally shared by the RTP system and the PISPs themselves as both parties are loath to take on any more risk than absolutely necessary. In these cases, some RTP systems have either required or provided the option for PISPs to connect to the RTP system using an existing FSP as a gateway. Since the FSP must already accept the risk in these scenarios, that FSP can partner with a given PISP and negotiate an acceptable arrangement for all parties involved. For example, Australia's NPP system supports both models (direct and indirect).

An example of using FSPs as gateways is shown in Figure 7. In this example, the payment is initiated using FSP Z as the gateway participant ($Init_{2a}$), however it very well could be the case that the Payer may rely on FSP A (their own FSP) or FSP B (the Payee's

FSP) to initiate the payment. Regardless of which FSP is used for initiation, the subsequent steps ($Init_3$ and P_2 through P_4) would remain unchanged.

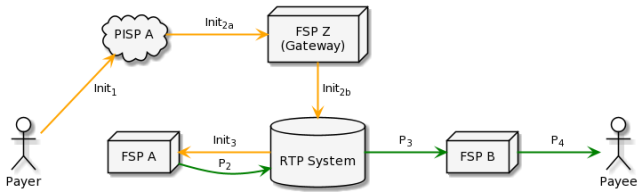

Figure 7: Flow of third-party initiated payments via FSP as gateway.

In this case, since the PISP does not speak directly to the RTP system in this case, it can be tempting for each FSP to define their own API for payment initiation and rely on the FSP to speak the *common* API downstream when communicating with the RTP system itself. The obvious benefit of this is the ability for the FSP to compete with other FSPs on the functionality and ease-of-use of the APIs used for authentication, authorization, and payment initiation. Unfortunately, the drawbacks are significant.

First, this would require multiple integrations in the case that a PISP wants to integrate with multiple FSPs. For many PISPs, integrating with multiple FSPs is a strict requirement in order to achieve the targeted availability requirements for customers using the PISP. And this has shown to be important as FSPs certainly cannot maintain perfect availability as was seen with the collapse of Yes Bank in India in March of 2020 [5].

Additionally, if each FSP defines its own API, the PISP would be effectively locked in to using a single gateway FSP due to technical concerns. If FSPs intend to compete based on their payment initiation APIs, it's very unlikely that the only competitive advantages will be focused on API usability. Instead, it's far more common that the FSP would introduce additional functionality that would be available exclusively with the FSP in question. While new functionality is certainly a benefit in general, the lack of uniformity means that third-parties can be "locked-in" to use a single FSP or risk customers losing access to certain functionality if anything ever happens with the FSP providing this functionality.

The bottom line in this case is that the APIs for 3PPI are not and should not be considered to be an area for competition and innovation anymore than electric companies should be able to experiment with additional voltages or frequencies for power delivery to customers' homes. In this case, drawbacks of a lack of consistency in the APIs far outweigh the benefits of new functionality. Based on this, while FSPs may act as gateways for PISPs to interact with a central RTP system, the technical abilities of the FSP should be limited to acting as a proxy for requests and responses between the PISP and RTP and nothing further.

2.1.2 User consent must be required for each transfer. In many systems available today, the system itself acts as a custodian for the end-user and is itself responsible for the final approval when executing a transfer on a RTPs. This means that, ultimately, a FSP can act independently of the end-user when making a payment, with no consent required for the payment. Since PISPs aren't in control of the ledger and are not fully-empowered participants on the RTP, they must not be able to act in the same manner.

Instead, PISPs must be required to obtain end-user consent for each individual transfer before it is considered valid by the FSP. This consent-driven requirement ensures that even if a PISP is compromised by a malicious party, any stolen data or hijacked infrastructure cannot be used to initiate payments on behalf of users registered with the PISP. Figure 8 below shows user consent being required in order to initiate a payment. At first, the transfer is considered unverified (orange) in Steps 1 through 3. Once the FSP verifies the transaction details, the transaction is considered authorized (green) and is submitted to the RTP.

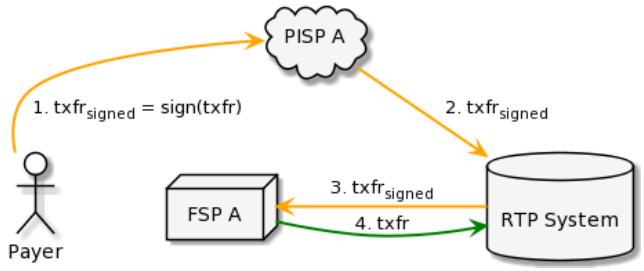

Figure 8: Consent required from the user and verified by FSP before submitting to the RTP.

If this principle is not followed, it's possible for a PISP to initiate a payment that may not have been requested by the end-user. Figure 9 below shows how Step 1 (in orange, indicating that it has not yet been verified) may or may not have been initiated by the actual user. However the FSP would submit the transaction to the RTP without verification (Step 4), potentially making an unauthorized payment.

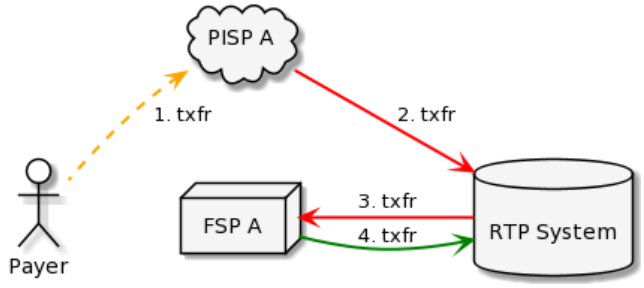

Figure 9: PISP is able to initiate payments without user consent.

It's also important to note that while it is imperative that each and every initiated transaction be properly authorized, this says nothing about how that authorization is to be obtained. However, as noted in Section 2.3.2, this should not require a web redirect for each transaction and instead should rely on credentials exchanged during the linking process.

2.1.3 Consent should be stored by the RTP and shared with interested participants. For the purposes of auditing and analysis of transactions, the FSP and the RTP both already have a history of all transfer details, however with third-party initiated transactions audits will need the ability to quickly validate transactions to ensure that they were properly authorized by the end-user. To ensure this is always possible, the RTP should be informed of and store a copy of the link established between the PISP and FSP, which may or may not be the authoritative source of this information.

This must include the authorization and authentication credentials (e.g., a public key for the end-user) as well as the source account identifiers and the permissions that were granted by the Payer to the PISP (akin to OAuth2 scopes).

If RTPs happen to be the authoritative source for this information, the other relevant parties should be kept informed of the information as well. This means that when this information is first created it should be broadcast to all interested participants (PISP, FSP, and RTP) and these same participants should be further updated when any information about this link changes (e.g., if it is revoked by the Payer). Without this, it's possible that participants' understanding of a relationship is not reflective of the truth and could lead to issues with repudiation of transactions in the future.

2.1.4 Trust must be revocable. During the linking phase, a Payer establishes a trusted relationship across all three involved parties (including themselves): the PISP, the FSP, and the Payer. As a result, this mutual consent allows a PISP to initiate payments on behalf of the Payer with the FSP acting as the true source of funds. While it's certainly not as common as linking accounts, it must be possible and straightforward for Payers to "unlink" or revoke the consent previously granted. It is also critical that this act of unlinking or revoking consent be possible by the Payer acting via the PISP itself or the FSP that acts as the source of funds for the link.

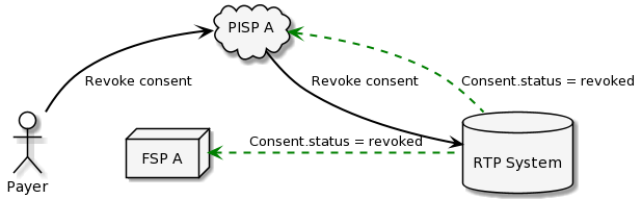

Figure 10: Consent revoked by user request directly to the PISP.

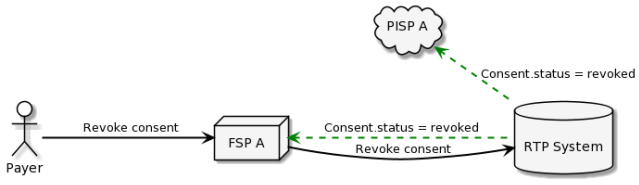

Figure 11: Consent revoked by user request directly to the FSP.

This process must not remove the details of the prior relationship between the related parties, but instead should mark the relationship as "revoked" so that it can no longer be used to initiate payments. This is a critical detail as it must be possible to verify past payments during an audit as having been initiated under the authorization granted by a previously revoked consent. In other words, if we delete the record of the consent entirely, there will be no way to know for sure whether past payments were properly authorized at the time of payment.

2.1.5 Payers must not be able to repudiate transfers. One key concern about any transaction is proving that it was actually requested by the true Payer. In the case of a typical FSP, the authentication is done up-front to establish the user's identity, and only then is the payment allowed to proceed. In this case, the authentication acts as a gateway into a walled garden, such that payments can be executed from within that walled garden based on the assumption that proper verification was done at the entry point. As a result, there is never a question as to whether or not an authorized user was responsible for a payment.

While this works for an FSP, a payment initiated by a third-party is a bit more complicated. For PISP-initiated transfers, there is always the possibility that a FSP will execute a transfer only to have the Payer state later on that they did not authorize that particular transfer (either in its entirety or in some partial capacity such as a different recipient or different amount). This is possibly only in cases where the FSP is not responsible for directly authenticating the user at the time of each transfer being initiated (as stipulated in Section 2.3.1).

Due to the lack of direct (FSP-verified) authentication for each transaction, we must combine authentication and authorization together as the mechanism for proving the authenticity of a given transaction. That authentication-authorization combination must include both something that securely identifies the Payer as well as something that is inherent to the transaction itself (e.g., a digital signature of the transaction details). For example, the reliance of the Fast IDentity Online (FIDO) [14] standard on a digital signature of a nonce of random bytes is sufficient for the authentication portion, but not authorization.

2.2 Privacy

When introducing any new participant into RTPs there is a risk of private information being exposed to unauthorized parties. This section covers some of the considerations necessary to strike a fair balance between functionality and user privacy.

2.2.1 Limited account data should be shareable. In many RTP systems, there is a confirmation step along the way to ensure that funds being sent by a Payer will be directed to the correct Payee. In some RTP systems (like Mojaloop [7] or Singapore's PayNow [3]), this discovery process takes the form of a FSP returning some personal information about the Payee such as a publicly visible "nickname" associated with the Payee's account. The rationale behind exposing this information is to avoid mistaken transfers to the wrong Payee due to typos in the phone number provided for the intended Payee or other user-initiated errors.

While many RTP systems rely on a one-to-one relationship between an identifier (e.g., a MSISDN [19]) and a destination account for a Payee, it's possible that in the future these identifiers may be tied to multiple destination accounts. In this scenario, one of the parties involved in the payment (either the Payer or the Payee) will need to specify which account should be used as the destination for the incoming funds for this particular payment. And if the Payer is the one making that decision on behalf of the Payee, the Payer or the Payer's FSP will need more information (e.g., a unique account identifier as well as the currency in which the account is denominated) in order to properly route the funds to the correct destination account. All of this leads to the conclusion that there is some information that must be considered "publicly visible" for the purposes of routing payments.

When it comes to third-party payment initiation, linking accounts between a Payer, PISP, and FSP is a scenario in which the PISP may likewise need the ability to fetch some metadata about the accounts available for linking by the Payer. While this could be done exclusively after direct authentication by the FSP, it may lead to a jarring user experience (see Section 2.3) relying on many web-view redirects or requiring the user to leave the PISP mobile application that they're attempting to link with their FSP . As a result, it should be acceptable to expose certain information to an otherwise unauthenticated user given a FSP-specific identifier (e.g., a username for a bank or a phone number for a mobile wallet provider). The issue at this point becomes deciding what information is considered acceptable to share to unauthenticated users purporting to be the Payer.

This information should be limited to the bare necessities for initiating the linking process, such as the available accounts (including a unique identifier and a description useful to the supposed Payer) and the currencies supported by the account. Things outside this scope include the account number used in traditional financial systems (see Section 2.2.2), the balances of the accounts, or the legal names on the accounts (though nicknames for the accounts should be considered acceptable if the account owners have granted consent to share these publicly).

It's also important to note here that RTP systems may decide not to allow any information to be exposed to unauthenticated users, placing the value of user privacy above the potential usability issues arising from a different user experience. Several mechanisms to accomplish this (including linking exclusively via One-time Password (OTP)) are discussed in Section 3.

2.2.2 Account identifiers must be useless outside of the linking flow. As noted in Section 2.2.1, RTP systems will almost always be required to share some limited account information with unauthenticated end-users. Most often this information is nothing more than a nickname of a Payee with a potential Payer during the Transfer phase of a payment, however the Linking phase may also require some additional account information. In particular, when a potential Payer begins the Linking phase with an FSP that supports the ability for the same user to maintain multiple accounts, the PISP must provide a way to choose which account the Payer would like to use as the source of funds for future payments. This multi-account support is not all that common with most mobile wallets, but is far more typical with banks (where users may have both chequing and savings accounts) as well as with any FSP that supports multiple currencies (e.g., a mobile wallet that supports both a local currency and US dollars), and as a result is important to consider when designing a payment initiation API.

As noted in Section 1.3.1, as part of establishing trust between the three parties (PISP, FSP, and Payer), the PISP must also determine which account with the FSP is to be used as the source of funds for payments. Generally, this happens early in the process of linking, such that when the time comes to authenticate the Payer has already indicated the source account. As shown in Figure 12, the Payer begins by asking the PISP to fetch the list of accounts ($Acct_1$). The PISP forwards this request to the RTP ($Acct_2$), which is then passed along to the FSP ($Acct_3$). The response flows back to the Payer in steps $Acct_4$ through $Acct_6$, which ultimately leads to the Payer choosing which account should be used as the source of funds.

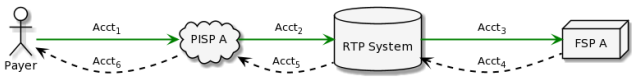

Figure 12: Fetching accounts available for linking given a FSP-specific identifier.

Based on this process, the result returned must include the unique identifiers (and potentially descriptions, currencies supported, etc) of the available accounts. And while it might be convenient to rely on the standard unique account identifiers (e.g., International Bank Account Number (IBAN) [11] or other standard account identifiers), this is strongly discouraged. Instead, the unique identifiers returned in this process should be one-time tokens that are completely useless outside the scope of the Linking phase.

The primary concern is that traditional unique account identifiers are not restricted to push-only payments. As seen in the Level One Principles from the Bill & Melinda Gates Foundation, "Push address credentials, if stolen, can't be used to fraudulently "pull" money out of a consumer's account." [8] And while it would certainly be unfortunate for any PISP to have these account identifiers stolen, it is not impossible that this might happen. As a result, the best practice is to avoid providing any identifiers to PISPs that could be misused and instead rely on tokens scoped to a single PISP for the purposes of identifying a funding source account during the Linking phase (and subsequently during the Transfer phase when initiating payments).

Interestingly, this means that if the same user attempts to pair the same account with two separate PISPs, the identifiers for that same source account will be completely different. This is because the FSP would generate a different token for each account (one for each PISP).

2.3 User experience

Ultimately, any design to facilitate third-party payment initiation will be used by an actual Payer to send payments to others using RTP systems. This means that the experience of these users when sending money is worth exploring to ensure that the designs themselves do not result in a difficult or frustrating experience instead of one that is simple and easy to use. The following principles focus on

the experience that users should expect when initiation payments via a PISP.

2.3.1 Linking should have minimal redirects. Lee, *et al.* [13] has shown that both simplicity and consistency have a significant effect on user's evaluation of the usability of any mobile web portal system, stating, "Simplicity shows a greater effect on usability and credibility than does consistency although consistency also shows a significant effect." It should be no surprise then that anything that interrupts the familiarity and simplicity when linking a FSP (e.g., mobile wallet, bank account, etc) with a PISP is likely to lead to higher rates of abandonment, where the user begins the Linking phase but stops before successfully establishing the mutual trust between the three parties.

Due to the inconsistency in the user interface design, one of the most jarring interruptions to any interaction on a mobile application is a redirect to a different provider's website, in either an embedded web view inside the mobile application or in the device's default web browser application. To minimize the rate of abandonment, the process itself should avoid this as much as possible and attempt to only require as few redirects and different interfaces as are absolutely necessary.

This may mean allowing some portions of the process to be performed while still unauthenticated (e.g., listing the accounts available for linking after providing only a username, as noted in Section 2.2.1), however the decision to do so is always a trade-off between privacy, security, and user-experience. This means that while there is no specific limit on the number of redirects permitted during the linking phase, each RTP system or FSP should aim to minimize this number and the decisions about the trade-offs listed earlier should be made carefully and intentionally.

2.3.2 Transfers must not require web redirects. Analysis of several million transactions using 3-D Secure (3DS) [16], which uses redirects and one-time passwords as an authentication mechanism for online credit card transactions, shows that over 20% of payment transactions are abandoned during the checkout process. While some of this drop-off is almost certainly due to the 3DS system preventing fraudulent transactions as intended, it's unlikely that all abandoned transactions are fraudulent. Instead, it's likely that the cause of some drop-off is due to inconvenience, frustration, or confusion with the user-experience of an extra hurdle after the last stage in the online checkout process.

As a result, building on the design principle from Section 2.3.1, it's just as important to minimize obstacles when initiating payments as it is when linking accounts with a PISP. In this case, however, it's important to remember that the Linking phase is performed relatively sparingly (typically once per account per PISP) whereas the Transfer phase is executed many times over the lifetime of the relationship between the Payer and the PISP. This means that any redirects that are part of the Transfer phase are seen far more than those from the Linking phase, and therefore have an out-sized influence on the usability and convenience of using a PISP in the first place. Based on this, the guidelines remain the same in spirit as Section 2.3.1, but are simply more strict. Instead of aiming to minimize the number of redirects, the Transfer phase should involve no web redirection whatsoever.

To support this, FSPs should rely on alternative authentication and authorization methods (such as digital signatures based on credentials exchanged in the Linking phase) that avoid web redirect interruptions to the Transfer flow while still providing an acceptable level of certainty that the transfer is, indeed, authentic and requested by the Payer. Ideally, due to liability concerns, these authentication and authorization methods should be standardized by the RTP system, rather than left to each FSP and PISP to decide independently. Also note that this principle really only applies to mobile applications with modern smartphones, as feature phones would handle this via alternative means such as a pre-exchanged Personal Identification Number (PIN) or an OTP sent to the device to authorize the transfer and authenticate the Payer.

There is also a large hidden implication of this principle: if redirects are not required, then transfers could potentially be initiated offline and transported to the RTP system via an alternative connection. For example, by requiring no online redirects to initiate a payment, the door is open to the possibility that a user might digitally sign a payment initiation request, which could then be represented as a Quick Response code (QR) [12] to be scanned and submitted to the RTP by the Payee rather than the Payer. While this is unlikely to be a feature anytime soon, it's worthwhile to consider as a potential future possibility.

3 IMPLEMENTATION GUIDELINES

In this section, we'll explore how a RTP system might implement a payment initiation protocol that adheres to the guiding principles in Section 2.

3.1 Establishing consent

The first, and most important, step toward enabling a third-party initiated payment is to establish trust between the three parties involved (the Payer, the FSP, and the PISP). There are several options available to enable this, with some not quite ready for production use due to various constraints from standards bodies (though modifications are in-progress). Let's start by looking at an option that is ready for live use.

3.1.1 Delegated trust. Delegated trust is the process by which the three parties establish multilateral trust by virtue of two bilateral trust arrangements. The algorithm is as follows (an overview of which is shown in Figure 13):

(1) The Payer proves their identity to the PISP, establishing trust through traditional means.
(2) The Payer proves their identity to the FSP, establishing trust through traditional means.
(3) The FSP provides the Payer with a secret token.
(4) The Payer provides that secret token to the PISP.
(5) The PISP provides that token back to the FSP to establish trust between the PISP and the FSP, scoped to the specific Payer.

Since this secret token is intended for a single use (like an OTP), the next step is to establish a new authentication method that can be used to authenticate multiple times in the future. For this, we can rely on the FIDO2 specification, which itself relies on public-key cryptography for the heavy lifting.

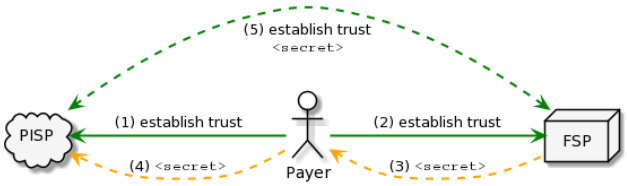

Figure 13: Establishing delegated trust.

The goal is simply to have the Payer generate a public-private keypair and provide the public key to the FSP, which in this case will be the Relying Party (RP). However, there is a problem. In the FIDO protocol, keys that are generated for one RP cannot be used to sign messages for other RPs. In this case, this means that if the FSP asks the Payer to generate a keypair using system-provided FIDO libraries, the PISP will not have access to digitally sign messages using those keys; instead, only the FSP will have that ability.

The result of this scenario leads to multiple options:

(1) Whenever a signature is required (see Section 3.2), the Payer would need to be redirected to the FSP in order to sign with the private key collected,
(2) The PISP will need to generate the keypair and inform the FSP of the public key, or
(3) A centrally run service is responsible for collecting the keypair and the Payer would be directed to a corresponding stand-alone application to digitally sign a transaction during the Transfer phase.

Based on the guidelines from Section 2.3.2, the current recommended best practice is the second option, where the FSP relies on the trust previously established in order to accept the public key provided by the PISP rather than collecting the public key directly from the Payer. This means the process of establishing a shared keypair between the three parties is the following:

(1) The PISP requests that the Payer generate a new keypair.
(2) The Payer generates a keypair and provides the public key to the PISP.
(3) The PISP provides that public key to the FSP (along with the secret to establish trust as shown previously)

This process is shown below in Figure 14, however the process of using the keys for digital signatures is discussed in more detail in Section 3.2.

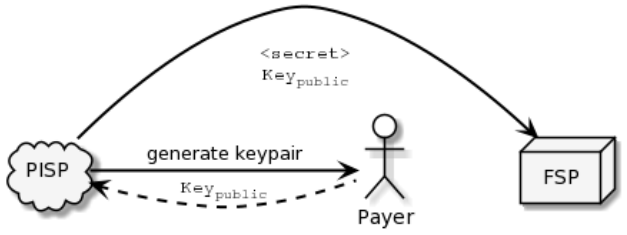

Figure 14: Generating and sharing a keypair for future authentication and authorization.

This leads to a very obvious potential man-in-the middle attack. In this algorithm, there is nothing stopping the PISP from generating their own keypair and providing that to the PISP. This would permit the PISP in the future to initiate payments for the Payer without the Payer's explicit consent, though appearing to have this consent based on the digital signature provided. This man-in-the-middle attack is shown below in Figure 10.

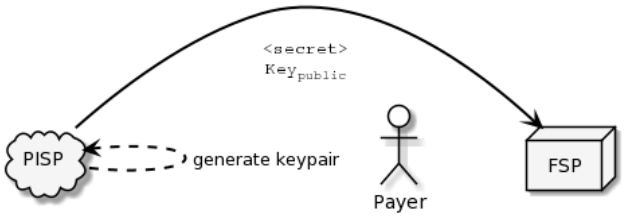

Figure 15: Keypair generated without direct user involvement.

As you can see, in this example the PISP never asks the Payer to generate a keypair. Instead, it generates its own keypair and provides the resulting public key to the FSP. As a result, barring any additional verification (e.g., Android's Key Attestation [6]) the PISP would be capable of initiating transactions without communicating with the Payer at all. To address this, we have two alternatives, discussed in Sections 3.1.2 and 3.1.3.

3.1.2 Cross-RP FIDO registration. One alternative to preventing the man-in-the-middle attack shown in Section 3.1.1 involves modifying the FIDO specification in order to allow the party that initiates the registration process to specify additional relying parties that should be able to use the generated keypair. In other words, the FSP would be able to register the FIDO credential and specify that the generated keys should be able to be used by the PISP. In this case, the FSP has first-hand guarantees that the credential was provided directly by the Payer, and therefore does not need to rely on delegated trust at all. This process is as follows, and shown below in Figure 11:

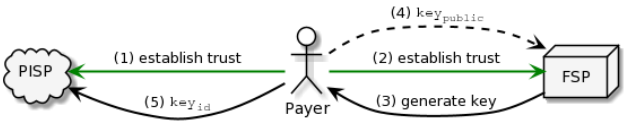

Figure 16: Establishing direct trust via cross-RP FIDO registration.

Unfortunately, this process only works if the FIDO libraries on various devices, provided by the mobile operating system (e.g., Android or iOS) will support this multi-relying party functionality. This is not currently the case, however there are proposals in progress, with the ultimate intention to submit these to the W3C and the FIDO Alliance for evaluation and a decision on whether this functionality will be accepted into the specification.

3.1.3 Push registration. This process could also be done in a "push" manner, where the PISP has no idea about the registration process but is instead informed via a background communication channel that they are now able to initiate transfers for a given FSP by using the indicated set of FIDO credentials. Regardless of whether this is a background notification or an active redirect to the FSP as part of the PISP registration process, the high-level conceptual flow is almost identical to Figure 16, shown below in Figure 17.

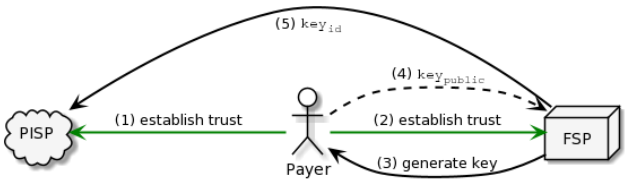

Figure 17: Establishing direct trust via FSP-pushed registration.

Note that the only difference here is in the final step where the FSP informs the PISP directly, via a background communication channel of how to initiate payments using the indicated FIDO credential.

3.2 Authorizing transfers

The most important aspect of initiating a transfer is to be certain that the transfer itself is authorized directly by the Payer. The best way for the FSP to ensure this would be to have first-hand knowledge of the authorization, however using traditional channels this would result in a redirect to the FSP for direct authorization, which leads to a categorically worse user experience. Instead, we can rely on the protocol dictated by the FIDO specification with a specially chosen challenge to allow for authorization rather than simple authentication.

In the traditional FIDO login flow, a server sends a random challenge to the client, which is then digitally signed with a pre-exchanged keypair, and evaluated against the public key held by the server. This process is fine for authentication because it proves that the client has access to the private key. In its current form, however, it is not an acceptable form of authorization because the challenge being signed is random and therefore meaningless. If a Payer were to repudiate a transaction in the future, our digital signature of some random bytes would not be acceptable proof of authorization, and would not settle any disputes.

To address this, we can make a simple change to the standard FIDO login flow: ensure the challenge is meaningful. Instead of digitally signing some random bytes, we can digitally sign a cryptographic hash (e.g., SHA256) of the transaction details. By doing this, the FIDO signature is now both a form of authentication (proving possession of the private key established during registration) as well as authorization (digitally signing details of a specific transaction). This process, shown below in Figure 18, is as follows:

(1) The Payer requests PISP initiate a transfer.
(2) The PISP requests the Payer digitally sign the details of that transfer.
(3) The Payer provides the resulting signature of this (meaningful) information.
(4) The PISP sends this digitally signed payment request to the FSP for payment
(5) The FSP verifies the signature to ensure it matches the credentials provided at registration.

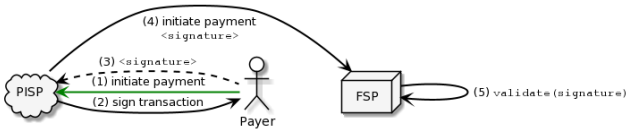

Figure 18: Initiating payment with digital signatures.

4 CONCLUSIONS AND FUTURE WORK

Based on our most recent experience, these design principles, if followed, are those most likely to lead to RTP systems with the simplest and effective support for third-party payment initiation and PISP participants. Further, these principles are likely to encourage more usage of PISPs on RTP systems thanks to an intuitive and consistent interface, free from unnecessary redirects or other hurdles that have been linked to payment abandonment.

While the principles themselves are unlikely to change much over time, there is functionality and changes to specifications in progress that may influence some implementation recommendations. For example, if the FIDO Alliance [1] is able to approve adding support for cross-RP credential registration, the recommendation for how keys are exchanged will change to encouraging this new functionality rather than the current recommendation that relies on delegation and indirect exchange of credentials between the Payer and the FSP.

Further, more research is needed in areas of usability specifically focused on payment initiation in environments that involve security hurdles (e.g., 3DS). While we have an indication that some abandonment is due to fraud being prevented versus drop-off due to user confusion or frustration, it's unclear what the division of causes truly is.

Finally, this paper focuses exclusively on centralized RTP systems and omits entirely the alternative distributed standardization designs, such as UK's Open Banking API standards [17]. Certainly more research is necessary to evaluate each of these objectively, with a comparison of the drawbacks and benefits of each and a conclusion of which model leads to the best results for a national payment system.

ACRONYMS

3DS 3-D Secure. 7, 9
3PPI Third-party Payment Initiation. 1–4

API Application Programming Interface. 3, 4, 6

FIDO Fast IDentity Online. 5, 8–10
FSP Financial Service Provider. 1–10

IBAN International Bank Account Number. 6

NPP New Payments Platform. 1

OTP One-time Password. 6, 7

PIN Personal Identification Number. 7
PISP Payment Initiation Service Provider. 1–10
PSD2 Revised Payment Services Directive. 1

QR Quick Response code. 1, 7

RP Relying Party. 8, 9
RTP Real-Time Payment. 1–7, 9, 10

UPI Unified Payments Interface. 1

GLOSSARY

3-D Secure A protocol designed to be an additional security layer for online credit and debit card transactions that relies on a three-domain model to tie financial authorization with online authorization.. 7, 9, 10

Third-party Payment Initiation The ability for a third-party participant (e.g., a PISP) to initiate a payment on behalf of an end user, while relying on another participant (e.g., FSP) acting as the source of funds.. 1, 9, 10

Application Programming Interface A computing interface that defines multiple interactions between multiple software intermediaries.. 3, 9, 10

Fast IDentity Online An authentication standard relying on secure public-key cryptography.. 5, 9, 10

Financial Service Provider An institution (e.g., bank, telecommunications provider, etc) capable of providing some financial services (e.g., mobile banking) to an end-user.. 1, 9, 10

International Bank Account Number An internationally agreed system of identifying bank accounts across national borders to facilitate the communication and processing of cross border transactions with a reduced risk of transcription errors.. 6, 10

New Payments Platform An industry-wide payments platform and national infrastructure for fast, flexible, data rich payments in Australia. 1, 10

One-time Password A password that is valid for only one login session or transaction, on a computer system or other digital device.. 6, 10

Payment Initiation Service Provider A service provider participant on a RTP network capable of initiation payment drawn on another FSP. 1, 10

Revised Payment Services Directive An EU Directive to regulate payment services and payment service providers throughout the European Union and the European Economic Area.. 1, 10

Quick Response code A type of matrix barcode (or two-dimensional barcode) first designed in 1994 for the automotive industry in Japan.. 1, 7, 10

Relying Party A web site or other entity that uses a FIDO protocol to directly authenticate users (i.e., performs peer-entity authentication).. 8, 10

Real-Time Payment A system run by a central authority capable of transferring funds from one party to another.. 1, 10

Unified Payments Interface India's national RTP system.. 1, 10

authentication The process of determining a given party's true identity to some acceptable level of certainty.. 3–9

authorization The act of granting explicit permission to perform a given action.. 3–5, 7–9

MSISDN A number uniquely identifying a subscription in a Global System for Mobile communications or a Universal Mobile Telecommunications System mobile network. Most commonly, a telephone number.. 6

Payee An end-user participant in a transaction that is the recipient of funds from another participant.. 2, 3, 5–7

Payer An end-user participant in a transaction that is sending funds to another participant.. 1–3, 5–9

Pix Brazilian Instant Payment Scheme. 1

REFERENCES

[1] FIDO Alliance. *FIDO Alliance Overview*. URL: https://fidoalliance.org/overview/ (visited on 12/06/2020).

[2] Reserve Bank of Australia. *Launch of the New Payments Platform*. URL: https://www.rba.gov.au/media-releases/2018/mr-18-02.html (visited on 12/05/2020).

[3] Association of Banks in Singapore. *PayNow Singapore*. URL: https://www.abs.org.sg/consumer-banking/pay-now (visited on 08/12/2020).

[4] Banco Central do Brasil. *Pix starts its full operation with 734 institutions on November 16, 2020*. URL: https://www.bcb.gov.br/en/pressdetail/2361/nota (visited on 12/05/2020).

[5] Archana Chaudhary and Bibhudatta Pradhan. *View: The urgency Yes Bank's collapse should have triggered, but didn't*. URL: https://economictimes.indiatimes.com/news/economy/policy/view-the-urgency-yes-banks-collapse-should-have-triggered-but-didnt/articleshow/74606376.cms (visited on 12/05/2020).

[6] Android Developer Documentation. *Verifying hardware-backed key pairs with Key Attestation*. URL: https://developer.android.com/training/articles/security-key-attestation (visited on 12/04/2020).

[7] Mojaloop Foundation. *Mojaloop Technical Overview*. URL: https://mojaloop.io/how-it-works/technical-overview/ (visited on 08/10/2020).

[8] The Level One Project (The Bill & Melinda Gates Foundation). *Level One Project Guide 2019*. URL: https://www.leveloneproject.org/wp-content/uploads/2020/07/L1P_Guide_2019_Final.pdf (visited on 12/05/2020).

[9] National Payments Corporation of India. *NPCI presents Unified Payments Interface(UPI) system*. URL: https://www.npci.org.in/PDF/npci/press-releases/2016/UPI_Launch_Press_Release_April_11_2016.pdf (visited on 12/04/2020).

[10] National Payments Corporation of India. *UPI Product Statistics*. URL: https://www.npci.org.in/what-we-do/upi/product-statistics (visited on 11/30/2020).

[11] *Financial services – International bank account number (IBAN) – Part 1: Structure of the IBAN*. Standard. Geneva, CH: International Organization for Standardization, Mar. 2007.

[12] *Information technology — Automatic identification and data capture techniques — QR Code bar code symbology specification*. Standard. Geneva, CH: International Organization for Standardization, Feb. 2015.

[13] Jongtae Lee et al. "Factors affecting the perceived usability of the mobile web portal services: comparing simplicity with consistency". In: *Information Technology and Management* 14.1 (2013), pp. 43–57.

[14] Rolf Lindermann et al. *FIDO UAF Protocol Specification v1.0*. URL: https://fidoalliance.org/specs/uaf-v1.0-id-20141122/fido-uaf-protocol-v1.0-id-20141122.html (visited on 06/25/2020).

[15] Google LLC. *Real-Time Payments Systems & Third Party Access*. URL: https://static.googleusercontent.com/media/pay.google.com/en//about/business/static/data/GPay_RTP_2019.pdf (visited on 07/01/2020).

[16] Ravelin Ltd. *Global online payment regulation*. URL: https://pages.ravelin.com/hubfs/All%20shareable%20content%20/Global%20payment%20regulation%20infographic%20PDF%20200919.pdf (visited on 08/14/2020).

[17] Ltd. Open Banking. *What is Open Banking?* URL: https://www.openbanking.org.uk/customers/what-is-open-banking/ (visited on 12/05/2020).

[18] "Payment services (PSD2) Directive (EU) 2015/2366 of the European Parliament". In: *Official Journal of the European Union* 58.L337 (2015). URL: https://eur-lex.europa.eu/legal-content/EN/TXT/PDF/?uri=OJ:L:2015:337:FULL&from=EN.

[19] International Telecommunications Union. *E.164: The international public telecommunication numbering plan*. URL: https://www.itu.int/rec/T-REC-E.164/ (visited on 08/14/2020).

[20] Federal Reserve Bank of the United States of America. *Federal Reserve Actions to Support Interbank Settlement of Faster Payments*. URL: https://www.federalreserve.gov/newsevents/pressreleases/files/other20190805a1.pdf (visited on 12/04/2020).