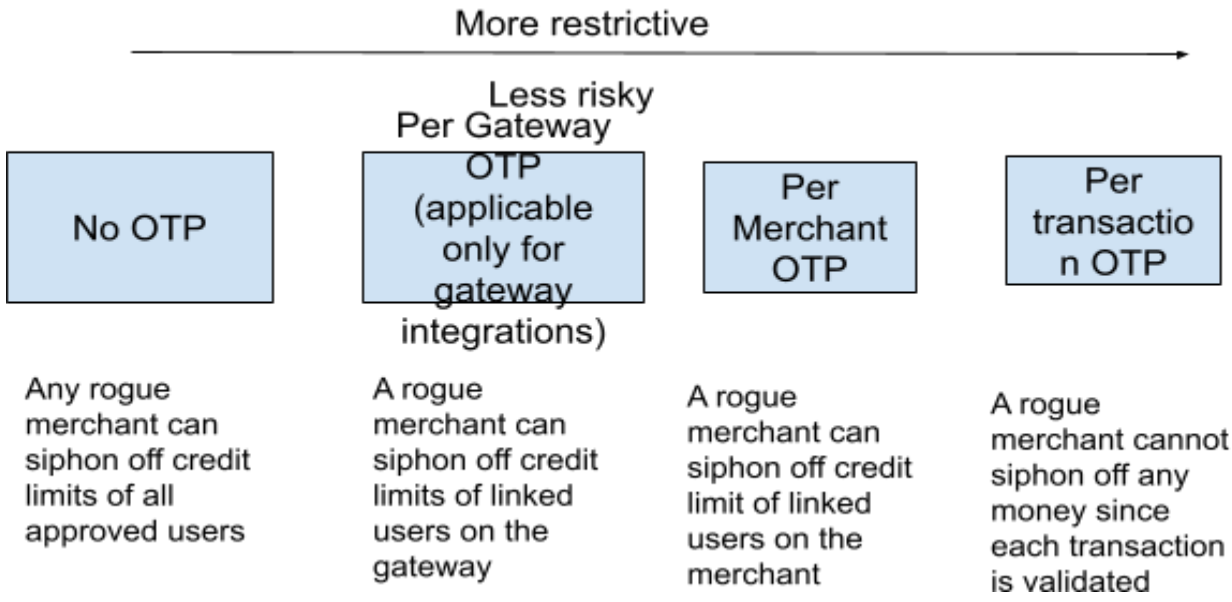
Why do we need OTP verification?

Author: Vivek Pandey
Collaborators: Gauri DehankarSiddhartha BaruaSheekha VermaRobin
Feb 2023

Executive Summary:

There are 4 types of OTP verifications, with various tradeoffs:



Intro:

In my initial days at Simpl, Vijay helped me understand TS, but I could not understand what OTP verification achieves. Even pointed questions did not lead to concrete answers. I left it with a mental picture of "Oh, all merchants do OTP. I know their reason. We do too, there must be some reason" and moved on. So I never understood that. Now there is talk that SoD will benefit if there is no OTP verification, and an obvious point is that well, if we can get rid of OTP, let's get rid of it even in PL and PL will also benefit!

So, I revisit the question: what purpose does OTP serve? I don't fully understand that. I will write down my thoughts here, and I hope that as informed people review it, we can be fully clear about the benefits of OTP verification.

OTP verification on merchants

First, a simpler question: Why do merchants (say zomato, flipkart, amazon) do OTP verification? For merchants, a phone number serves as the identity of the user. So, they need to verify the phone number to show the user with previously ordered items, pending deliveries etc. If they were not to do OTP and just believe the user inputted phone number, I could enter your phone number and see your previous orders.

So, on merchants, the phone number is the login id user, and OTP verification is the password.

How OTP verification is done in PL in Simpl

In v3 integration, OTP verification yields a token ("zero click token"). Simpl stores this token in its servers. Merchant needs to store this token on their side and present it in eligibility and charge calls.

In v4 integration, OTP verification only leads to Simpl setting a flag in its server that the user has been verified. We use the terminology "linked": user has been linked to the merchant: OTP verification is on a per merchant basis (once for each merchant), per gateway basis (once for each gateway) or a per transaction basis (once for each transaction, this is for guest check out)

Some observations as things stand currently

1. In the good case (genuine non fraud users, genuine merchant, no merchant bugs), OTP verification is an impediment to user experience and OTP delay leads to user drop off.

2. Suppose that I am a fraudulent merchant on v4 integration. If there is no OTP verification, I can siphon money from Simpl like the following. Send eligibility calls from different phone numbers (keep trying different phone numbers). When an approval call on a phone number p succeeds, I will get a token. I will use the token to make a purchase on my website. Then I will claim money from Simpl. Owner of phone number p will get a bill from Simpl and will wonder what purchase he did.

This type of fraud has happened even in our current system. However, because of OTP verification, the fraudster needs the owner of phone number p to collude with him for initial linking, and this creates friction. In the fraud case, the fraudulent merchant created a video

asking users to do Rs X transaction on the app, paying by Simpl and they would get Rs X/2 from the merchant.

In this regard, note that with gateway linking, linking on one of the merchants in a gateway links the user to all the merchants on the gateway. However, even in our gateway integrations, OTP linking is on per merchant basis.

3. There is a variant of (2) where due to a merchant bug, eligibility call gets made for a phone number p1 when the user is logged in with a merchant on a phone number p2. In this hypothetical scenario, the same phenomenon as (1) happens, except that it is by bug rather than by malicious intent.

4. There is a legal/perception angle. We send bill reminders / collection calls to a phone number p if phone number p has done a transaction using Simpl. How do we know if the owner of phone number p has actually agreed to pay, and it is not some fraudulent merchant who has done the transaction on behalf of p?

In absence of OTP verification, we are outsourcing the verification to the merchant. But merchants can be fraudulent, and the final responsibility to ensure that we don't start collection calls to a user who never heard of Simpl/never agreed to use Simpl, lies with us.

With one time OTP verification, we ensure that users who never agreed to use Simpl (they just don't want to use BNPL) will never be troubled by our collection efforts. Within the universe of Simpl users fraud can still happen (since OTP verification is not done every time), but persons who will be troubled by Simpl will be Simpl users and thus our case is more defendable.

5. The fraud mentioned in point (2) can be greatly facilitated by a collusion with a Simpl employee. In point (2) the fraudulent merchant needed a set of approved users. That can be provided easily by an internal employee.

Will per transaction OTP make us more secure?

There are four types of OTP linkings (in the order of least restrictive to most restrictive)

1. No OTP linking (no implemented anywhere, but a theoretical possibility)
2. On a per gateway basis
3. On a per merchant basis
4. On a per transaction basis

Option (1) is bad because of several loopholes that are pointed out previously.

Option (2) is also bad since a fraudulent merchant on the gateway can siphon credit limit of the the users who have been linked to some other merchant on the gateway.

That leads us to the question: Are there some loopholes in option (3) too? (This is not the least restrictive option -- that distinction goes to option (4)). In our brainstorming we could not identify loopholes that are realistic. Essential point is that in option (3) we are at the mercy of merchants: if the merchant decides to go rogue and siphon credit limit of linked users, we cannot stop that. (E.g. zomato can send us eligibility + charge calls for those users who have earlier done transaction with simpl even if they are currently not doing a transaction)

Thus we arrive at the diagram which is present at the beginning of this article.